

# Unit 1 - The Internet

What can be represented with a single bit and how do we get a single bit of information from one place to another? This unit explores the technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices. Topics include: the digital representation of information - numbers, text, images, and communication protocols.

Many of the lessons that follow have worksheets and student guides associated with activities. Those worksheets are listed in the relevant lesson plan, or you can check out all unit 1 student-facing activity guides [here](#).

## Chapter 1: Representing and Transmitting Information

### Big Questions

- Why do computers use binary to represent digital information?
- How does data physically get from one computer to another?
- Are the ways data is represented and transmitted with computers laws of nature or laws of man?

### Enduring Understandings

- 2.1 A variety of abstractions built upon binary sequences can be used to represent all digital data.
- 3.3 There are trade offs when representing information as digital data.
- 6.2 Characteristics of the Internet influence the systems built on it.
- 7.2 Computing enables innovation in nearly every field.

### Week 1

#### Lesson 1: Personal Innovations

##### Unplugged

Welcome to Computer Science Principles! Groups make a “rapid” prototype of an innovative idea and share it. Students watch a brief video about computing innovations.



#### Lesson 2: Sending Binary Messages

##### Unplugged | Concept Invention

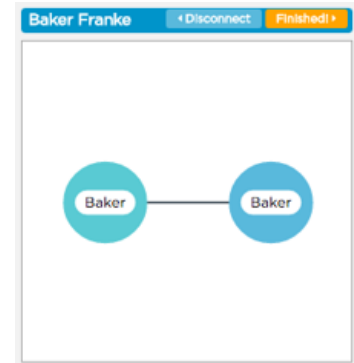
Students collaborate in an iterative design process to make a "bit sending device" using classroom supplies and everyday objects. They develop their own systems for encoding and sending simple binary messages over some physical distance.



## Lesson 3: Sending Binary Messages with the Internet Simulator

### Internet Simulator | Group Problem Solving | Concept Invention

Students use the Internet simulator for the first time in this lesson. It is configured to represent two parties connected on a single shared wire that only hold one of two possible states. Students invent a binary call-response protocol that can overcome the coordination, timing and synchronization problems that arise when forced to use such a truly binary system.



### Optional Lesson: Sending Bits in the Real World

#### Optional

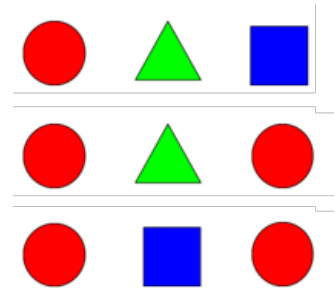
In this lesson, students will be introduced to how bits are transmitted over the most common mediums (copper wire, fiber-optic cable, and radio waves) used to connect devices on the Internet. They then chose a device that transmits bits and research that device and the system it uses. Students create a poster presenting their findings, and the lesson concludes with a gallery walk of the posters.

## Week 2

### Lesson 4: Number Systems

#### Unplugged | Concept Invention

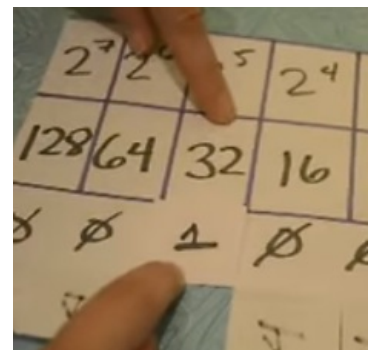
Students will explore the properties of number systems by inventing their own number system using only three shapes: a circle, triangle and a square. This lesson is a precursor to looking at several other number systems important to computing, especially binary and hexadecimal.



### Lesson 5: Binary Numbers

#### Unplugged | Reinforcement

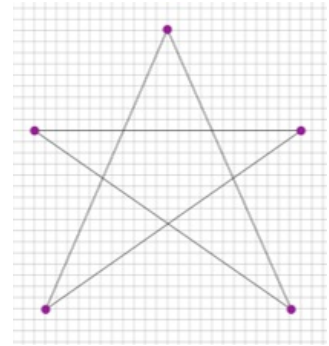
Students become more familiar with the binary number system by making a "flippy do" using the binary odometer widget and practicing doing binary-to-decimal number conversions.



## Lesson 6: Sending Numbers

### Internet Simulator | Group Problem Solving

Students invent a binary protocol for sending a line drawing represented as list of grid coordinates (numbers). Students test and hone their protocols using a new version of the Internet Simulator which is now configured to automatically send and receive streams of bits.



### Optional Lesson: Encoding Numbers in the Real World

#### Optional

In this lesson, students explore some fascinating stories from the news and history (and the future) about number encodings in computers. These stories should serve to illuminate how the kinds of decisions students have been making about number encodings are the same kinds of things that real scientists in the world have to worry about, sometimes with disastrous consequences. While this lesson has the possibility of running long, it is meant only as a short excursion into real-world application and should be limited to one class period.

## Week 3

## Lesson 7: Encoding and Sending Formatted Text

### Group Problem Solving | Internet Simulator

Students invent a protocol that uses ASCII (the standard plain text encoding scheme) to encode formatted text such as fonts, colors, sizes, etc. Students test their protocol using the Internet Simulator, now configured to send and interpret ASCII text, to see if a partner can properly render (draw on paper) the formatting encoded in a message.

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F

# Chapter Commentary

## Unit 1 Chapter 1 - What's the story?

This chapter is a story about bits - what they can represent and, importantly in the age of the Internet, **how** they physically get from one place to another. Through the lessons in this chapter we look at different types of primitive information (mostly numbers and text) and repeatedly ask a pair of questions: 1) how can I represent that with bits? And 2) how do I get those bits from **here to there**.

Ultimately we want to give students a sense of the layers of abstraction involved in representing and transmitting information. Sending a single bit becomes sending a stream of bits; a stream of bits can represent number; a number can represent a letter of the alphabet, and so on. In order to communicate using only bits, two parties must agree on **protocols** for encoding and decoding information.

A strong foundation for learning computer science starts with developing a certain level of comfort with abstraction. Computer science is unlike other scientific disciplines in that many of its rules and laws have been defined by people, sometimes arbitrarily, rather than nature. The closest thing to a natural law we can rely on are things related to binary representation, and the associated logic and mathematics that come along with it.

## Our Approach to the Content

We want students to go on a bit of a journey in which they repeatedly solve problems related to representing information. A broad term for our approach in this chapter is "**concept invention**" in which we ask students to invent their own solution to a problem before revealing the conventional solution. For example, we ask students to use sequences of geometric shapes - a circle, a triangle, and a square - to invent a number system as precursor to learning about binary.

One advantage of concept invention is that it tends to foster a more **equitable** classroom environment. We try to structure the activities so that they do not advantage prior knowledge of the concept. The key pattern to this approach is to introduce vocabulary and definitions **after** students have had some shared experience with the activity and its associated concepts. A pilot teacher of the course coined the phrase: **Activity Before Concept, Concept Before Vocabulary or “ABC / CBV”**.

A seeming disadvantage of concept invention is that students may initially be resistant to learning this way if they are fixated on getting the “right” answer. In an exploratory or discovery-based activity, of course, there is no one right answer. Early on you, yourself, might worry about “chaos” or the fact that students are learning things “wrong.” It is worth being patient and sticking with it because much of computer science (especially programming) is about coming up with your own solutions to problems using the tools you have at hand. There is no one right way to write a program either. Furthermore, this more open approach will have a positive effect on your classroom environment. After a certain amount of trust has been established that you are not setting students up to fail, but rather encouraging them to create and invent, you’ll see a shift in attitudes and **overall student engagement**.

We make heavy use of the **Internet Simulator** in these lessons to enforce rules about binary representation or the activity designed to use it. The configuration of the Internet Simulator changes slightly for each lesson that uses it, but it is always about point-to-point communication: you sending a message to a partner using only bits. The simulator is intended to be a tool for experimentation and play. We encourage you to let students investigate any new features rather than explaining it to them at the outset. Similar to the approach for concept invention, our philosophy is that you can always explain how the tool works after students have had some experience (or even frustration) poking around. In fact, they’ll be much more receptive to the explanation after having a hands-on experience.

## Chapter 2: Inventing the Internet

### Big Questions

- Who and what is “in charge” of the Internet and how it functions?
- How is information transmitted from one computer to the other when they are not directly connected?
- How can the Internet keep growing? How does that work?

### Enduring Understandings

- 2.1 A variety of abstractions built upon binary sequences can be used to represent all digital data.
- 6.1 The Internet is a network of autonomous systems.
- 6.2 Characteristics of the Internet influence the systems built on it.
- 7.3 Computing has a global affect -- both beneficial and harmful -- on people and society.

### Week 3

#### Lesson 8: The Internet Is for Everyone

##### Unplugged | Research

Students read portions of a memo written by Vint Cerf entitled “The Internet is for Everyone”, a call-to-arms that highlights the benefits of a free and open Internet and identifies threats to this system.

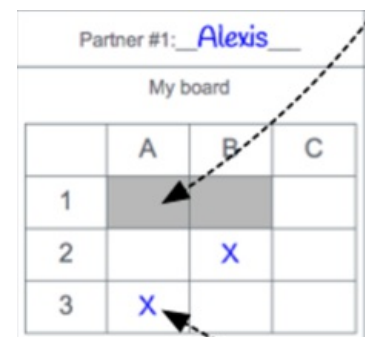


### Week 4

#### Lesson 9: The Need for Addressing

##### Group Problem Solving | Internet Simulator | Unplugged

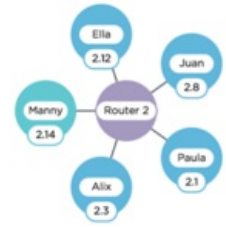
Students invent a protocol (similar to the real Internet Protocol IP) to encode the necessary elements for playing a simplified, but multi-person, game of Battleship. Students first play unplugged, then using a new version of the Internet Simulator configured to allow a user to broadcast messages to a group.



## Lesson 10: Routers and Redundancy

### Internet Simulator | Whole Class Discovery

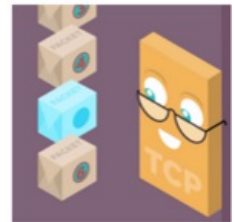
Students use a new version of the Internet Simulator configured with simulated IP addresses and routers to explore the benefits (and potential security concerns) associated with routing traffic across the Internet. Students should see that messages go through many different routers, may not always take the same path to reach a destination, and all the traffic is publicly viewable!



## Lesson 11: Packets and Making a Reliable Internet

### Internet Simulator | Group Problem Solving

Through an unplugged activity students are introduced to packets, and issues with packets being delayed and dropped. Students invent a protocol to reliably send a message over an unreliable network using the Internet Simulator now configured to be "unreliable" by delaying and randomly dropping packets sent between routers.



### Optional Lesson: Algorithms Detour - Minimum Spanning Tree

#### Optional

In this and the subsequent lesson, we consider some of the strategies used to construct networks and find paths for data in them. While this has a connection to ideas about the Internet, the focus of these lessons is on algorithms, formal techniques, and processes for solving problems. Students will explore and solve the Minimum Spanning Tree (MST) problem, first, in an unplugged fashion on paper. The real challenge is not in solving a particular instance of the minimum spanning tree, but to develop an algorithm, a clear series of steps, that if followed properly, will solve any instance of the problem. There is a possible misconception to look out for: the MST has a definite, verifiable optimal solution, as opposed to the Text Compression problem (from Unit 1), which does not.

### Optional Lesson: Algorithms Detour - Shortest Path

#### Optional

In this lesson students will explore the Single Source Shortest Path problem, by solving the problem with pencil and paper first, then by following a famous algorithm that solves the shortest path problem known as Dijkstra's Algorithm. Even though this is an algorithms detour, there is a strong connection in this lesson to routing algorithms used on the Internet. This lesson also introduces ideas about how we analyze algorithms: looking for correctness, efficiency and running time. As foreshadowing: in the next lesson students will act out another distributed shortest path algorithm used by routers to learn about the Internet dynamically.

### Optional Lesson: How Routers Learn

#### Optional

This lesson is the last of the algorithm series. Building off of the previous lesson about shortest path algorithms, the activity in this lesson shows how routers learn about the rest of the Internet in order to route traffic so it takes the shortest path. In the previous lessons, students use the Internet Simulator to send packets to other students through simulated routers. The path that the packet follows, and how the router knows where to send it, however, has been largely untouched. Today, students simulate the process of a router joining a network and generating a router table that would allow them to send packets to anyone else in their network as efficiently as possible. They then reflect on the process by comparing the similarities between the SSSP problem and the process the used today, and how it facilitates the structure of the Internet.

## Lesson 12: The Need for DNS

**Unplugged | Group Problem Solving | Internet Simulator | Research**

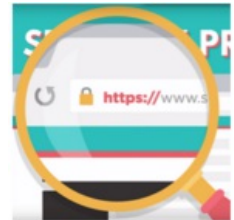
Through an unplugged activity students see the difficulties in trying to maintain a universal name-to-IP address mapping. The Domain Name System (DNS) is introduced and students can experiment with a simplified version using the Internet Simulator.



## Lesson 13: HTTP and Abstraction on the Internet

**External Tools | Exploration**

Students learn about and investigate HTTP by looking at HTTP traffic generated within their own browser. Students visit a variety of websites and use the browser's built-in tools to view all the traffic.



## Lesson 14: Practice PT - The Internet and Society

**Practice PT | Research | Presentation**

In this Practice Performance Task students research and prepare a flash talk about the controversial issues around either Net Neutrality or Internet Censorship. The talk and artifacts produced should exhibit students' knowledge of the Internet.



# Chapter Commentary

## Unit 1 Chapter 2 - What's the story?

The story of this chapter is in many ways the story of the Internet itself: how it came into existence, and how its protocols were designed to build on each other and allow for its rapid expansion and scaling. Instead of inventing protocols to send information directly from one peer to another, we now consider the protocols necessary for any one computer to communicate with any other computer, assuming they are all connected via multiple interconnected networks - i.e. over the Internet.

The lessons in this chapter build on each other just like the real internet protocols do. For example, in a networked world, for any one computer to communicate with any other you need a protocol for computer addresses (**IP addresses**). Once you have addresses, you can consider how data should be routed over multiple networks to find the address it's supposed to. Once you have that, you can consider breaking large messages into pieces (**TCP packets**) for more reliable transfer. You also need a way to translate text addresses (like code.org) into IP addresses (**DNS**). Finally, you need a way to communicate high-level text information from one place to another (**HTTP**) that relies on all of these other pieces working as designed.

Again, we want to give students a sense of the **layers of abstraction** involved the creation of these protocols: each protocol solves one problem and solves it well, allowing a more complex problem to be solved on top of it. This philosophy the "open Internet" has allowed an enormously complex global system like the Internet to not only be built in the first place, but also allows everyone to participate in growing, scaling, and adapting to new problems over time.

## Our Approach to the Content

Similar to the last chapter, our general approach to learning is through **concept invention**. Each lesson in the chapter poses one of the fundamental problems that had to be solved for Internet communication to work. Because the concepts are often abstract and hard to see, we typically have a robust **unplugged** activity that pairs with each Internet protocol. The activity is designed to illuminate the problem in a

physical, experiential way, so that students have a **common experience** to draw on when working to develop a solution.

The solution to a problem is always the invention of a new protocol, and we usually enforce the rules of that protocol by using the **Internet Simulator**, which is configured differently for each lesson. Each configuration incorporates the solution to the last major problem so students don't have to solve it again, while at the same time it exhibits some new behavior or problem that must be solved. For example, after students have invented an addressing protocol, all subsequent versions of the simulator have a simple IP address built in that is handled automatically by the system, but perhaps it will impose a limit on the size of any message, thereby forcing students to invent a protocol to handle "packeting" of information.

Even though designing protocols and testing them on the Internet Simulator is not programming, per se, it has many of the same skills. It requires students to be precise, to manage a growing complexity of systems, to creatively and iteratively solve problems with imposed constraints, and to "debug" those solutions using the tools at hand. Also there is no one right answer most of the time.

We believe that many of the lessons help to improve students' collaborative problem solving behaviors, and bring out other characteristics of computer science work that we want to encourage. We believe these non-programming, computational problems are fun, engaging, and novel. It is unlikely that many students, even those with some background knowledge in computer science, will have considered problems about devising communication protocols. Our theory, again, is that this helps to foster an **equitable** classroom environment that does not unduly advantage students' prior knowledge. We want every student to have an equal stake in the proceedings, to discuss problems, and work together to test out solutions.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.



# Lesson 1: Personal Innovations

## Unplugged

## Overview

Welcome to Computer Science Principles! The first lesson is about getting students excited about the course and connecting their own personal interests to computer science. Students are asked to share something they know a lot about and teach it to a small group. Groups make a “rapid” prototype of an innovative idea and share it. Students watch a brief video about computing innovations. The lesson ends with students logging into the Code.org CSP course web site, and answering a brief prompt about what “computer science” means to them.

## Purpose

This activity plants the initial seed for students to think about the ways in which they might be able to solve some problems relevant to their lives with technological innovations. The AP CS Principles framework describes 6 Computational Thinking Practices that need to be evident in the course. This first lesson is more about beginning to engage with those practices - beginning to think, act, and behave like computer scientists. In particular, the practices in which students should be engaged in this lesson are:

- **P1: Connecting Computing:** Describe connections between people and computing.
- **P5: Communicating:** Describe computation and the impact of technology and computation.
- **P6: Collaborating:** Foster a constructive, collaborative climate by facilitating the contributions of a team member; exchange knowledge and feedback with a partner or team member.

## Agenda

### Tech Setup

#### Before class

#### At the beginning of class

### CSP Pre-Course Survey

**Important! Have your students take the CSP Pre-Course Survey!**

### Getting Started (15 mins)

#### Teacher Message

**What could you teach somebody?**

**Group introduction and share out**

### Activity (25 mins)

**Identify impacts and prototype an innovation**

**Brainstorm Technological Innovation**

**Rapid Prototype one idea**

**Share Prototypes**

**Welcome students to the course**

### Wrap-up (5 - 10 min)

**Show Video - “Computer Science is Changing Everything”**

### Assessment Questions

### Extended Learning

## Objectives

### Students will be able to:

- Communicate with classmates about computing innovations in their lives.
- Describe positive and negative effects of computing innovations.

## Preparation

- Procure post-its or paper and tape
- Procure poster paper for sharing innovations
- Queue up CS is Changing Everything video
- Setup section in Code Studio for this course
- Have student sign-up link ready to share

## Links

### For the Teacher

- [Code Studio Teacher Dashboard Video Tutorial](#)
- [How to Administer a Locked Assessment \(PDF | DOCX\)](#)

### For the Students

- [Personal Innovations - Activity Guide \(PDF | DOCX\)](#)
- [Computer Science is Changing Everything - Video \(download\)](#)
- [Personal Innovations - Rubric \(PDF | DOCX\)](#)
- [Unit 1 on Code Studio](#)

## Vocabulary

- **Innovation** - A novel or improved idea, device, product, etc, or the development thereof.

# Teaching Guide

## Tech Setup

Throughout the year, in order for you to be able to see student work, and for students to be able to connect to each other on tools like the Internet Simulator, all students need to be in the same section on Code Studio. To see assessments and answer keys that Code.org provides, you also need to be a "verified teacher".

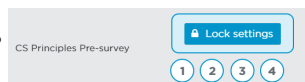
### Before class

#### Ensure you are registered on Code Studio as a "verified" teacher account

Anyone can create a teacher account on Code Studio, which means that we need an extra layer of authorization to allow CS Principles teachers to see assessments, answer keys, and any other collateral that students should not be able to trivially get access to. If you attended a Code.org TeacherCon during the summer, you should already have this access.

To check if you have access:

1. Navigate to the **Unit 1 course overview page**
2. Do you see the CS Principles Pre-course Survey as the first "stage" of the course?



3. If not, please fill out **this form**. Note that it can take a day or so to become a verified teacher account, so please do this step early!

If you are **not** a verified teacher account, you can still create a section for your class, but you will not be able to administer the pre-course survey on the first day.

#### Create a class section on Code Studio.

To create a section:

1. Navigate to the **Teacher Home Page**
2. Click 'Student Accounts and Progress'
3. Click 'New section'
4. In the space provided, give your section a name (e.g. CSP Period 1), select Login Type: 'email', the grade of your students, Course: 'Unit 1: The Internet' and click 'Save'
5. Once the section is created, click the name of the section
6. On the new page, you will see a unique Join URL that you will distribute to your students in class.

For a video walkthrough of these steps and more on viewing student work in Code Studio, watch the **Code Studio Teacher Dashboard Video Tutorial**

### At the beginning of class

1. Have students create a Code Studio account at <https://studio.code.org> if they don't already have one
2. Share the section Join URL with students and tell them to navigate to it to join your section
  - You can confirm that a student successfully joined your section by having the section progress page on the Teacher Home Page open and hitting refresh as students join. Students should see a small green bar at the top of their page that says 'You've successfully joined ...'
3. From [studio.code.org](https://studio.code.org) have students locate the Unit 1: The Internet tile and click 'View course'

Once students are looking at the Unit 1 overview page, they will be ready to take the CSP Pre-Course survey

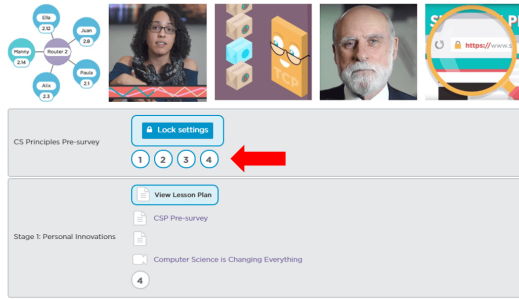
## CSP Pre-Course Survey

### Important! Have your students take the CSP Pre-Course Survey!

Students can find a link to the survey in Code Studio as the first item on the Unit 1 overview page. To ensure that students only take the survey at the appropriate time, it is "locked" and unviewable by them until you "unlock" the survey. The **How to Administer a Locked Assessment** document provides details on how to do that when you are ready. Note that the instructions for administering an assessment and a survey are the same.

## CSP Unit 1 - The Internet

This unit explores the technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices. The unit then explores the structure and design of the internet and the implications of those design decisions.



### How much time does it take?

The survey does take some time - it is roughly 30 questions. You might consider administering it on an admin day at school, or as an early homework.

Because it is a pre-course survey it is important that students take it as early in the course as possible, before they have had much (or any) exposure to the class so that we may accurately gauge changes in attitudes and beliefs caused by the course.

Thank you!

-- The Code.org Team

## Getting Started (15 mins)

### Teacher Message

#### Remarks

- Welcome to Computer Science Principles! This year we're going to learn a lot about computer science, but before we begin, we have to learn about everyone in this room.
- One thing that makes this class unique is that you have to invent solutions to problems and create things all the time, both alone and with others. Everyone has a unique and creative perspective they bring to the table.
- We'll talk about the syllabus and rest of the year at the end of class.
- But let's start by seeing what everyone brings with them, and how creative we can be right now!

Take out a journal or a piece of paper.

### What could you teach somebody?

#### Prompt:

**"What's something that you know a lot about? Something that you could teach somebody?"**

- This doesn't have to be a subject in school - it very well might not be.
- As a person, as an individual who is living and breathing in this world, there is something that you probably know a lot about - maybe you feel like you know more about than most people. What is that thing?

#### Give students 1 minute to write that thing down on a post-it and stick it on the wall

- Note: putting name on the post-it is optional -- if you don't have post-its, use note cards and tape, or scraps of paper. Anything that allows for these topics/areas of interest to be seen in one place.

**Survey the post-its on the wall to see the diversity of responses, maybe invite students to do this as well.**

### Group introduction and share out

- Take post-its and make groups of 4 that represent a diverse set of interests.
  - For example, grab: "video games", "basketball", "cooking", "growing vegetables"
- Once in groups, give each student 2 minutes to:

### Why do the pre-course survey?

- A major goal of CS Principles is to broaden participation in computer science.
- It is **crucial** therefore to have insight into students' attitudes and beliefs about computer science **before** the course so that we can measure the amount of change that occurred **after** the course is over.
- Please note that this survey is **anonymous for students**
- Completing it also helps us understand important improvements we can make to the curriculum to improve the teacher and student experience.

**Please help by having your students contribute to this vital dataset. Their voices make the difference!**

### Teaching Tip

What you're trying to do here is get students to state something that they are interested in, but also know a lot about - something they might have insights into.

A big part of students' enthusiasm for sharing will come from your enthusiasm and genuine interest in getting to know them.

Students might need prodding: there is something that makes them interesting and unique. Something they like to do, have interest in, read about, have some expertise in, a hidden talent.

- introduce themselves
- explain the thing they know a lot about
- teach the group something about it, or tell the group something interesting about it
- **You may want to set a timer and tell them when to switch.**

## Activity (25 mins)

Students should remain in groups established during the getting started activity.

### Identify impacts and prototype an innovation

#### Remarks

People seem to say that technology is all around us, that it affects everything we do. Is that true? Technological innovation is about recognizing a problem that needs to be solved, or recognizing something needs improving and then building a tool to solve it.

**As a class we're going to see how innovative we can be. We're going to do something called "rapid prototyping."**

"Prototype" is a fancy word that means a preliminary sketch of an idea or model for something new. It's the original drawing from which something real might be built or created.

### Brainstorm Technological Innovation

Go around the group, and for each individual's area of interest:

1. Identify some way that technology is used with, or affects that thing
2. Make a suggestion for either:
  - a way that technology might be improved to make it better, faster, easier to use
  - a creative or innovative new technology might help solve some problem within that area, or at least make better?

Everyone in the group should make suggestions for any of the areas of interest at your table.

### Rapid Prototype one idea

As a group you have just brainstormed about the technology ideas at the table.

Now, come together and get excited about **one** of them.

As a group, nominate the idea you've discussed that you think would be the **most interesting** to everyone else in the class.

**Start to sketch out that idea on a poster. Make a visual representation of your ideas.**

- Remember this is a rapid prototype. Just something to quickly convey the idea.
- Give students a decent amount of time to work and sketch together.

#### Teaching Tip

Keep things quick. If a group is worried about not being innovative enough, remind them that very small ideas can have big consequences. People once thought it was ridiculous that you would want to send a short text message to another person over a phone.

Alternatively, a group may have a great idea that they want to spend more time on. They can do that later. For now, just remind them it's a rapid prototype.

### Share Prototypes

Do a "Gallery Walk" or a whip around so that each student can see all of the other students' work.

- Put prototype posters on the wall
- Give students time to survey the various posters
- Time permitting, ask an individual from each group to explain what the the thing is or what their innovation is.
- If time is short, ask for one or two volunteers, or hand pick a poster or two for a student to explain. You can leave the posters up for people to look at tomorrow.

### Welcome students to the course

Take this opportunity to explain the importance of bringing individual interests and perspectives to this course.

Students are not only encouraged to find areas of personal interest in this course, it is actually mandated as part of the Performance Tasks. From day one, students should be thinking about how to apply the principles they learn to their own lives, and hopefully they will be excited to do so.

#### Remarks

- We are just starting this class, but you all bring passions and knowledge about things you care about. And whatever those things are it is likely (if not inevitable) that they involve computing technology in some way.
- This class is about the principles of computer science. The principles that underlie much of the technology around us.
- The reason this class exists is because not enough people in our school/town/city/country/world know or understand things about computer science. Not enough people know the basics that you need to know to be able to see where you can make a difference. Not enough diverse views, interests, opinions exist among the people who do more than use technology - they create new technology
- This class is about taking a step in that direction.
- Your job as a student in the class is to be on the lookout, to be alert, for where and how computer science affects or impacts the things you care about, the things you know about. Whatever it is you want to do in your life, it's pretty likely that technology affects it in some way, or that some technological innovation is called for.
- In fact, for the AP assessment you must create an app that reflects a personal interest or problem you're trying to solve. You must also explore a technological innovation and write about it.
- Don't worry, we'll practice these things, and through the course you'll learn about the principles that underlie all this stuff.
- The way the class works, we often ask you to invent your own solution to problems. Even if it's a problem that's been solved before, thinking like a computer scientist is a different mindset to be in.
- We study a lot of things in this class not only because it's foundational knowledge, but because of the **way it makes you think**, they way it asks you to solve problems, on your own or in collaboration with others.
- Inventing things, and having insights about how things work, and how they might work better is what this class is about. Everyone has the ability to contribute. Everyone has unique insights into something. Whatever it is you care about, whatever it is that makes you different, is the value you bring to the class.
- Everyone is unique, but we're all in this together.

#### Teaching Tip

If this is your first time teaching this course, you may not know exactly how to present an entire year of activities and exploration on the first day.

The "Remarks" given at left are some possible ways to contextualize today's activity and build excitement about the rest of the year.

Feel free to do what best matches your own style, and use the ideas to craft your own presentation of the course.

## Wrap-up (5 - 10 min)

### Show Video - "Computer Science is Changing Everything"

- Show: **Computer Science is Changing Everything - Video**
- The video is available for students in Code Studio as well.

## Assessment Questions

- Select an innovation. Describe the positive and negative impacts it has had on the world. (A satisfactory answer reflects awareness and analysis of the effects on populations beyond the students' immediately locale.)
- Speculate on how students 25 years from now will answer, "What computing innovation has had a significant impact on your life?" (A satisfactory answer includes a non-trivial consideration of how technology will likely change from, and perhaps build upon, what is currently available)

## Extended Learning

- Ask students to think about how the adults in their lives might answer the question, "What computing innovation has had the most significant personal impact on your life?"
- To reinforce visual learning, suggest that students create time-lines showing the years when the various innovations they hear about from other adults were invented or became available to consumers.
- Blown to Bits <http://www.bitsbook.com/> - read Chapter 4, Needles in the Haystack, pages 141-142 (Placements, Clicks, and Auctions), then answer the following question about innovation:
  - Discuss the positive and negative results of Overture's three search engine innovations. How did those innovations turn out today?.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CI** - Community, Global, and Ethical Impacts

## Computer Science Principles

- ▶ **7.1** - Computing enhances communication, interaction, and cognition.
- ▶ **7.2** - Computing enables innovation in nearly every field.
- ▶ **7.3** - Computing has a global affect -- both beneficial and harmful -- on people and society.
- ▶ **7.4** - Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 2: Sending Binary Messages

Unplugged | Concept Invention

## Overview

In this lesson students work in groups using classroom supplies and everyday objects to develop their own systems for encoding and sending simple binary messages, messages that only have two possible values. Students will think about what can be usefully conveyed in such a simple message and build a “device” to communicate the message over some physical distance. Then students are asked to consider how to use their binary messaging devices to send a more complex message - a message with more than two possibilities, say four, or eight, or even thousands of different messages. Students will collaborate in an iterative design process in the “maker ethos” of rapidly building and improving their “device” for sending messages.

## Purpose

One of the least understood things about the internet is how it works as a physical system. Bits of information physically travel from one place to another. On the “real” internet bits are transmitted using a variety of different mediums, and most of the time a single bit physically travels over a variety of systems to get where it’s going. Students should see the issues in the physical transmission of bits as a separate problem from interpreting and encoding the messages sent on top of those systems.

While building and modifying their “binary message devices” students should recognize that it’s easier to invent a system of communication that uses combinations of binary signals with a simple device, rather than making a new, or increasingly complex device for each new problem. This lays the foundations for:

- encoding complex information in computers using combinations of bits
- developing clear communication protocols

## Agenda

### Getting Started (5 mins)

**Introduction**  
**Think, Pair, Share**

### Activity

**Prepare for Binary Message Device Activity**  
**Introduce Activity: Binary Message Device Challenge**  
**Challenge 1: Send a Binary Message**  
**Challenge 2: Four possible messages**  
**Challenge 3: Eight possible messages!**  
**Challenge 4: N-possible messages**

### Wrap-up

### Assessment

### Extended Learning

## Objectives

**Students will be able to:**

- Create a device for sending a single bit of information - state A or state B - over a distance.
- Analyze the possibilities and limitations that arise when sending binary messages.
- Explain or demonstrate how to use a binary message sending device to send messages that have more than two states.

## Preparation

**Optional** Print activity guide (see **teaching tip**)

One box of “maker supplies” for each student group. Example supplies include:

Small flashlights

Simple noisemakers such as bells, triangles, or whistles

Cups and string

Slinky

Legos

Colored paper

## Links

### For the Teacher

- [U1L02 - Teaching Tips & Tricks Video](#) - Video

### For the Students

- [Binary Message Devices](#) - Activity Guide ([PDF](#) | [DOCX](#))
- [Binary Message Devices](#) - Rubric ([PDF](#) | [DOCX](#))
- [Unit 1 on Code Studio](#)

## Vocabulary

- **Binary** - A way of representing information using only two options.
- **Bit** - A contraction of “Binary Digit”. A bit is the single unit of information in a computer, typically represented as a 0 or 1.

# Teaching Guide

## Getting Started (5 mins)

### Introduction

Computer science is commonly thought to be the study of computers themselves - the physical machines we have on our desks and carry around in our pockets. Another way that computer scientists think about would be to say that **computer science is the study of information and information processes**.

Today we're going to think about what "information" and "information processes" means. What is it that you think you might be studying?

### Think, Pair, Share

Let's get started with a thinking prompt...

💡 **Prompt:** What is your personal definition of "information?" Take a minute to write it down.

- Give students enough time to write their ideas or just think silently.
- Then let them confer with a neighbor

💬 **Discuss:** Have students share ideas about what "information" means to them. Explore how they're similar or different, or the definition changes depending on the context in which the word is used.

### 🎤 Remarks

- There are many ways we can think about the word "information," but one possible definition we'll explore today is that information is the answer to a question.
- Perhaps the simplest question one can ask is what we'll call a **binary question**, or a question to which there are only two possible answers.
- For example, the question "Do we have a quiz today?" is a binary question, as there are only two possible responses, "yes" and "no." Typically we think of binary questions as ones that can be answered "yes/no," "true/false," etc.
- Many either/or type questions have binary responses as well, such as in "Which do you prefer: Coke or Pepsi?" or "Country music or hip hop?"
- The information in this response we'll call a **binary message**, or a message that can only have one of two possible values.

### 💡 Teaching Tip

It might be surprisingly tricky for students to define "information". If necessary you might **modify the prompt** and ask students to just do word association:

- **What other words come to mind when you hear the word: "information"?**

### 💬 Discussion Goal

Landing on a correct definition of information is less important than getting students thinking about the question, and realizing that "information" is a rich, multi-faceted word.

Your discussion will set up the next several lessons, which are going to explore how information is sent between devices, and how it's understood once it's been sent.

### 🎓 Content Corner

The word "binary" is not actually a term from computing. "Binary" is used to describe anything which involves or relates to two possible things.

## Activity

### Prepare for Binary Message Device Activity

Put students in partners for the activity

Today we're going to work more with binary questions and messages. First you and your partner need to come up with a binary question.

💡 **Prompt:** Imagine that you and your friend have not been able to communicate for the entire summer, and you have a chance to ask her one binary question that she will answer. What binary question do you want to ask?

- Give partners a minute to chat
- They can choose to ask a yes/no question but students may have more fun choosing an either/or type question.
- You might invite students to share their ideas with the class. Record some of the questions on the board or other display.

### 💡 Teaching Tip

The activity asks students to go beyond representing only binary questions, so you might want to coerce them into choosing questions that aren't simple true/false, yes/no type questions. A question like: "Coke or Pepsi?" could easily be expanded to include other options.

You might think about it like a multiple choice test question, where for right now you only have two possible answers.



Assess whether they are indeed binary questions and correct any errors or misconceptions.

**Transition:** Now that you've come up with a binary question, let's talk about how you need to answer it. Answering a binary question is easy when we speak to each other, but it becomes more difficult when we are separated.

Today we will focus on how a binary message can be sent over a distance. **You will build the device that sends them.**

## Introduce Activity: Binary Message Device Challenge

### 🔦 Distribute Activity Guide: **Binary Message Devices - Activity Guide**

(From the activity guide)

**Introduction:** You are going to build a device out of classroom supplies to send information to a classmate on the other side of the room. There are some basic rules and constraints:

- **Stay on your side.** You may not walk to the other side of the room.
- **No language.** That means no writing or talking to communicate.
- **No projectiles!**

### Challenge 1: Send a Binary Message

For the first challenge you will have **5 minutes** to construct a device out these supplies to send a simple binary message - one of the two possible answers to your binary question - to your partner on the other side of the room.

You should try to make it fail-proof. Consider obstacles that might be thrown in your way. Would your device still work if:

- There was something in between you and your partner?
- You couldn't see your partner?
- You were in a loud room?
- Your partner wasn't paying attention?

**GO!**

Give students 5 minutes to put together their devices.

After 5 minutes....

#### **Quick Share / Demonstration**

- Ask teams to demonstrate how to use their device
- Go around the room to see the different kinds of things people came up with.

### Challenge 2: Four possible messages

Not all questions have only two possible answers. Your new challenge is to invent a way to use your device to send an answer to a question that has **4 possible answers!** Think about these things:

- Should you modify your device?
- Should you use it in a different way?
- Should you make a new device entirely?

You've got 5 minutes!

**GO!\***

After 5 minutes....

#### **Quick Share / Demonstration / Discussion**

- Ask teams to demonstrate how to use their device
- Go around the room to see the different kinds of things people came up with.

### 💡 Teaching Tip

Distributing the activity guide is **optional** as you may prefer to control the action yourself.

- You may treat the activity guide as your script for conducting the activity rather than printing and distributing.
- It is perfectly acceptable to have students simply jot notes on a blank sheet of paper or in their journals.
- If you wish to assess what students write down, the rubrics will still apply.

### 🎓 Content Corner

The word for a single binary state is bit. In the curriculum you will see references to "bits" starting in the next lesson where the word is introduced. It could be introduced here if you like, it is up to you.

While typically "bit" refers to a 0 or 1 for this lesson if you use the word "bit" you should use it to refer to states A and B to be consistent with the activity.

### 💡 Teaching Tips

- You should start referring to these two states as **state A and state B**.
- Encourage students to test and iteratively improve their designs as they reach what appears to be "the end." Repeatedly encourage them to make their design better and better.
- Ask students to demonstrate their device for you and try to present them with challenges they might not have considered, like their line-of-sight is blocked.
- Encourage students to try to make their method of signaling as simple as possible.
- Students will often create devices that have a third "neutral" state or do-nothing state. Challenge students on whether their device really only has two states and encourage them to improve their devices accordingly.

- Example of student response:

Your Binary Question:

*Do we have a quiz or a test today?*

Message	How to send with your device
A: <i>quiz</i>	<i>one tug on the string</i>
B: <i>test</i>	<i>Two quick tugs on string</i>

- **Questions:** did most people modify their device to add states? Did anyone not change the device but just use it differently?



### Challenge 3: Eight possible messages!

What if you wanted to ask an even more complex question with **8 possible answers**?

Just as before update your device and test it out. Record how to use your device.

You've got 5 minutes!

**GO!\***

After 5 minutes....

#### Quick Share / Demonstration / Discussion

- Ask teams to demonstrate how to use their device
- Go around the room to see the different kinds of things people came up with.
- **Questions:** did most people modify their device to add states? Did anyone not change the device but just use it differently?

**Transition:** How long can we keep doing this?

### Challenge 4: N-possible messages

Could we keep increasing the number of messages forever?

Could our devices be used for questions with 16, 32, or 1,000,000 possible responses? Some things to think about...

- Our alphabet only has 26-letters, yet we can spell many words
- Our number system only has 10 digits yet we can represent many numbers
- Think back to your simple two-state device. Could you simply use it differently, rather than modifying it?

**Discuss with your partner:** How could you use your device to respond to much more complex questions (for example one with 1,000 possible responses). Come up with a system for using your device and describe it in such a way that another group could pick up your device and use it to send messages this way.

Give students a few minutes to discuss and write.

**Transition into the wrap up.**

## Wrap-up

**Discussion:**

1. Could you use another group's device to send your set of messages? Why or why not? What would you need to know from the other groups?
2. Based on what you've seen in today's activity what do you think are the limitations on the kinds of information we can send with binary messages?

## Assessment

**Rubric/Questions:** Distribute the **Binary Message Devices - Rubric** rubric for today's activity, which asks students to assess their performance and respond to a series of questions. Students can record their answers individually or as a team. Note: Students can also respond to these questions in **Unit 1 - Sending Binary Messages - Code Studio** on Code Studio.

1. Multiple Choice: A binary question is defined as:

#### Activity Goal

The purpose of the whole activity is to build toward an understanding that, from an engineering perspective, the simplest way to physically send an infinite number of messages over some distance is to make a binary message device, and to send **unique sequences of binary states**. This is how the internet at physical level actually works.

After each little challenge the purpose of the demonstration and questions is to:

- Highlight different tactics for addressing the problem
- Foreshadow the next challenge

There are 3 basic ways that students will probably focus on to make a device send 4 different messages:

1. Modify the device so it can assume 4 different states
2. Similarly, they might just **use** the device differently in such a way that they can do 4 different things with it.
3. **Use the same representations of states A and B from your original device and come up with 4 unique sequences, for example A-A, A-B, B-A, B-B**

Most students will likely do the first two options - modify the device or use 4 different ways - but some might do the 3rd option at this point. Certainly, the **preferred solution** is number 3 and that's what we're working toward with the activity.

If a group figures out to send a binary sequence at any point **make sure to highlight that solution** and move the class in that direction.

If at some point students start to sense the pattern and it becomes stale, you can dispense with the discussion if you don't think it will be valuable and just transition to the next challenge or move quickly to the 4th and final challenge.

Note: For 4 messages the optimal solution is to use sequences of 2 binary messages (AA, AB, BA, BB). For 8 messages you can use sequences of 3 binary states (just add an A or a B to each of the previous 4 messages). For 16 messages, you can use a sequence of 4 binary messages and so on.

#### Teaching Tip

Resist the urge to teach binary-to-decimal conversions here. We'll be learning about binary over the next several lessons.

The point of this activity is NOT to teach the binary number system, but rather to understand that using physical devices that have binary states, we can send sequences of states to represent any number of things.

As long as students come up with some kind of system for sending sequences of binary states - even if they don't map the classic binary number system - that is perfectly fine for now.

- A piece of information that is sent in pairs
  - Two questions which share the same answer
  - A message which can be in two possible states
  - A question which can be answered in only one of two possible ways
2. Provide an example of a question that could not be answered with a binary message. Explain why this is the case, making reference to the definition of a binary message.
  3. Modify your question so that it could be answered with a binary message. Explain why it can now be answered with a binary message.
  4. Can you send a message in binary to someone you've never before communicated with? If yes, how? If no, what does the person receiving a message need to know in order to successfully decode the message?
  5. Why did you choose your particular message encoding strategy? List at least two reasons that justify your decision. List two issues or problems that could be improved in your encoding strategy.

### Discussion Goal

The goal here is just to clarify things that came during the activity.

**Yes, the devices are interchangeable because they all have ways to represent binary states. To use another group's devices you would need to know: 1. How to use the device to signal each state: A or B and 2) Some kind of guide or instructions explaining how many states per message, and what each unique sequence means. For example: "each message is a sequence of 5 binary. AAAAA (Alaska) AAAAB (Alabama)..etc"**

- **Physical Limitations: signal loss, noise, unclear binary states, obstructions, obstacles, etc.**
- **Informational Limitations: the length of a message limits the number of things you can represent as binary states. Anything that doesn't map easily or obviously to discrete states would be hard or impossible to represent.**

## Extended Learning

### Prompts:

- Switch devices with another group. Make sure you get their instructions for indicating state A and state B. Using their device, send your message to your partner. Why is this possible?
- What binary messaging systems are currently used in this room, school, or at home? How effective are they?
  - Possible answers: Many kinds of alerts or notification systems are binary in nature: smoke detector, metal detector, some traffic signals, brake lights on cars, cell phone ringtones, etc.
  - Similarly, vibrations can signal a variety of conditions based on length or number.
- What are the most important aspects of coordination when trying to send a single binary message from one place to another?
- What can't be represented in a binary message?
- Could you use another group's device (and the associated instructions for how to indicate state A and state B) to send your message? Why or why not?

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CL** - Collaboration
- ▶ **CT** - Computational Thinking

### Computer Science Principles

- ▶ **2.1** - A variety of abstractions built upon binary sequences can be used to represent all digital data.
- ▶ **3.3** - There are trade offs when representing information as digital data.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 3: Sending Binary Messages with the Internet Simulator

Internet Simulator | Group Problem Solving | Concept Invention

## Overview

Students are introduced to the Internet Simulator, a tool they will return to many times in the first two units of the course. Today, the Internet Simulator will be used to simulate a single shared wire, connecting two people. The wire can only be in one of two possible states (state A or state B) and either partner may set or read the state of the wire at any time, but this is the only way in which students may communicate. Students must invent a binary call-response protocol using this system. Coordination, speed and timing are problems that need to be solved. At the conclusion of the lesson, students compete to demonstrate the speed and accuracy of their protocols, and calculate the bit rate of their message exchange.

## Purpose

The major purpose of this lesson is to engage students in a rather challenging problem of engineering a physical network for digital communications. If you must communicate in binary by setting the state of some object (such as a wire) to one of its two possible states, merely defining "State A" and "State B" is insufficient because there is no way to distinguish between a single bit "A" and a string of "AAA" for example. Some element of time must be incorporated into a communication protocol to make it functional for exchanging bits. This time-per-bit leads naturally to calculating a bitrate for a given device, or a measure of how quickly a system transmits digital data.

This is the first lesson in which we use the Internet Simulator. The Internet Simulator is a robust learning tool developed by Code.org specifically for Computer Science Principles, that has many different variants that you will see in the first two units of the course. It was designed to give students a hands-on experience experimenting with (and solving) different kinds of problems associated with networked computers and the Internet. A goal of using the internet simulator was put best by a student near the conclusion of Unit 1 who said "Whoa, I feel like I just invented the Internet!"

## Agenda

### Getting Started (10 mins)

#### Binary Signal Test

### Activity (35 mins)

#### Quick Vocabulary

#### Introduce the Internet Simulator

#### Experiment: Coordination and Single-Bit Protocols The 2-bit Message Exchange Challenge!

### Wrap-up (10 mins)

#### Video - The Internet: Wires Cables and WiFi

### Assessment


### Extended Learning

## Objectives

### Students will be able to:

- Explain how synchronization and coordination enable the transmission of binary messages.
- Develop a protocol for exchanging binary messages in two directions.
- Calculate the bit rate for a binary message exchange.
- Provide a definition of "bit" and relate it to the binary messages they have seen so far.

## Preparation

 Code Studio section setup (required for using Internet Simulator)

## Links

### For the Teacher

- [U1L03 - Teaching Tips & Tricks Video](#) - Video ([download](#))

### For the Students

- [Coordination and Binary Messages](#) - Activity Guide ([PDF](#) | [DOCX](#))
- [Flashlight Binary Signal Test](#) - Presentation
- [Internet Simulator - Part 1](#) - Video ([download](#))
- [The Internet: Wires, Cables & WiFi](#) - Video ([download](#))
- [Worksheet - Video Guide for "Wires, Cables & WiFi" \(optional\)](#) - Video Worksheet ([PDF](#) | [DOCX](#))
- [Unit 1 on Code Studio](#)

## Vocabulary

- **Bandwidth** - Transmission capacity measure by bit rate
- **Bit** - A contraction of "Binary Digit". A bit is the single unit of information in a computer, typically represented as a 0 or 1.
- **Bit rate** - (sometimes written bitrate) the number of bits that are conveyed or processed per unit of time. e.g. 8 bits/sec.
- **Latency** - Time it takes for a bit to travel from its sender to its receiver.
- **Protocol** - A set of rules governing the exchange or transmission of data between devices.

# Teaching Guide

## Getting Started (10 mins)

### Remarks

- Yesterday you all made your own binary message devices.
- We learned that we could compose any number of messages by sending a sequence of states.
- In order to interpret the message we needed to know
  1. which signal meant A and which meant B and
  2. some kind of mapping between sequences of signals and a possible message.
- What we were really doing was beginning to develop a **communication protocol**
- Today you're going to develop a protocol to solve a problem.

### Content Corner

The word "**protocol**" has a variety of meanings outside of computing. It typically refers to a formal procedure or set of rules. In computing a protocol usually implies a "communication protocol" that governs how devices should transmit and interpret data. There are many similarities between **protocols** and **algorithms** that can be made later in the course.

## Binary Signal Test

The following demonstration can be done in two ways:

1. Teacher use a flashlight
2. Use this presentation **Flashlight Binary Signal Test - Presentation** which has animations that simulate the flashlight.

### Instructions:

**"Imagine that you and your friend have made a binary signaling protocol using a flashlight. The light on is state A, off is state B."**

#### Test 1

**"Your friend sends you this message. What is being signaled here? Write down what you think the message is."**

- Pulse the flash light on and off at ~1 second intervals 4 times
- Give students a few seconds write

- Students can confer with neighbor or as whole class as to what the message was.

#### Test 2

**"Uh oh! Your friend realizes she actually made a mistake encoding the message from before and decides to re-send the message. Decode this new version of the message and write it down."**

- Pulse the the flashlight on and off quickly, then a few pulses at ~1 second intervals
- Give students a few seconds to write
- Again, discuss with neighbor

### Teaching Tip

Many students likely decoded the first message as B A B A B initially. The second message might have more variants, but BB A B A BB AA BB is one possibility.

As soon as they received the second message they may have updated their original answer to account for the implied timing they saw in the second message. Depending on how fast you think you could interpret signal changes the first message could easily be **any number** of consecutive Bs and As.

### Discussion

Lead a discussion that explores the assumptions made when decoding these messages, and more importantly, explores what information they would need in order to decode it.

- Did this new message make you think about your answer to the first question? Do you want to change your answer to the first question?
- What assumptions did you make in interpreting these messages?
- Is this protocol specific enough to allow useful communication of a binary message? If not what information would need to be added to it?

### Discussion Goal

This brief warm-up activity and discussion is meant to draw out two main points:

1. A flashlight turning on and off **truly** has only two states. Many of the devices students made, even if they used flashlights, probably had at least a third "do nothing" state that was used to signal a break between As and Bs.
2. When there are truly only two states, **time** and synchronization become huge factors and must be accounted for in a **protocol**.

## Activity (35 mins)

## Quick Vocabulary

We need to get some terminology down so that we can speak about our problems and solutions more efficiently.

- **Protocol** - For our purposes today a "protocol" is simply a set of rules about sending, receiving and interpreting binary messages.
- **Bit** We will call each element of a binary message a **bit**. "Bit" is short for **binary digit**. So for example if you have a binary message A B B A , we would say that is a 4-**bit** message.

Today you and your partner will be developing a **protocol** for exchanging 2-bit messages using an Internet Simulator.

## Introduce the Internet Simulator

We encourage you to use a **"discovery-based introduction"** to this and all other tools in the Code.org curriculum. (See teaching tip at right).

### Remarks

- Today we will be using the Internet Simulator to explore some of the challenges of sending bits on the Internet.
- This tool simulates a single wire connecting two people who cannot otherwise see or speak to each other.
- The Internet Simulator helps to enforce "the binary rule" for sending messages -- it will present challenges that they will need to problem solve around.

**Demonstrate** how to connect to a partner in the widget **but explain nothing else**.

Give students the following 5-minute discovery challenge:

- **"Your job is to explore this tool with a partner - click all the buttons, type in the text areas what you can."**
- **"You cannot break it so don't worry."**
- **"There is a bit of a mystery in what the tool does...and doesn't do. Can you figure it out?"**
- **"You and your partner have 5 minutes to poke around and see what you can find."**

After 5 minutes bring the group back together and poll the group to see what they found.

It is likely that they will have uncovered most of the properties of the simulator, but if they haven't you can explain now.

Clarify any misconceptions about the tool at this point and then move on with the challenge.

- The major challenge is that **the wire is SHARED** between the two people.
- The wire can hold **only a single state** - A or B - and it can be set by **either** person at **any time**.
- At **any time either person** can **read** the wire to see which state it's in
- **misconception:** that either partner is "sending" or "receiving" anything. Partners can set and see the current state of the wire.

The **challenge** of today's activity is to figure out a way to coordinate actions with your partner to make this tool into a functioning two-way bit-sending device.

### Video Demonstration of the Widget:

If necessary, you can show or have students watch the tutorial video: **Internet Simulator - Part 1 - Video** which covers how to: \* join or connect with another classmate \* set the state of the wire \* read the wire.

You can show it to the whole class or students can watch it on the Code Studio stage for this lesson.

## Sending Bits with the Internet Simulator

### Experiment: Coordination and Single-Bit Protocols

Students experiment with sending a single bit and develop protocols that allow them to send more complex messages with more bits.

- Distribute a copy of **Coordination and Binary Messages - Activity Guide**

### Use a discovery-based method

#### Discovery-based introduction of tools in nutshell:

- Get students into/connected to the tool
- Give students working in partners a fixed amount of time (5 minutes or so) to poke around and see if they can figure out what it does and doesn't do -- typically this might be presented as a mystery worth investigating
- Ask the group to report what they found
- Teacher fill in any gaps or explanations of how the tool works **afterwards**

The Internet Simulator - and all other tools created by Code.org - are meant as learning tools. You cannot break them so you are encouraged to let students play and investigate to figure out how the tools work.

These discovery-based methods of introducing tools have been tested in professional development and have worked well for teachers who use this curriculum. This method is effective for a few reasons, but overall students find this approach more engaging and fun, and they tend to be more receptive to, and motivated to hear, explanations of how the tool works **after** trying to "solve the mystery" themselves.

#### First time with Internet Simulator?

If you are using the Internet Simulator for the first time you should watch the **U1L03 - Teaching Tips & Tricks Video - Video** that shows what the classroom environment should look like and how students should be interacting. You should also watch the student video: **Internet Simulator - Part 1 - Video**





- Place students in pairs and send them to the Internet Simulator.
- It's recommended that partners who want to send data to each other sit close together so they can talk, even though they need to be on different computers.

## The 2-bit Message Exchange Challenge!

Students will practice relaying a 2-bit sequence with their partners. The **goal** is to exchange 2-bit messages (partner sends a 2-bit message, other partner sends a 2-bit message back) as quickly and accurately as possible. In other words, students are trying to get the fastest bit rate possible.

- Each initial sender should be able to confirm that the message was received correctly and resent because the message that they have received is the same as the original.
- As students iterate and perfect their protocol, they should try to extend their protocol to send more bits or send them more quickly, while maintaining accuracy. When each group has a working protocol and has time to practice (~10 minutes), have an all-class showdown to see who can do it the fastest.

### Rules for the Challenge:

- Student pairs can decide who sends first.
- Teacher reveals sequence of bits to be sent. Partner A of each group may enter this sequence of bits into Internet Simulator but may not begin sending them.
- Teacher will say "GO" and partners will exchange messages.
- During the challenge students may not communicate with one another. (**Note:** feel free to put partners on opposite sides of a room or in separate rooms altogether.)
- Students yell "STOP" once they have completed their exchange.
- Teacher will verify that each partner received what the other sent by checking outgoing and incoming messages.

### Teaching Tip

Students will likely need some time to connect with their partners and experiment with the interface of the Internet Simulator. Once they feel comfortable with the basics of the tool, they should proceed as a class through the challenges found in **Coordination and Binary Messages - Activity Guide**.

#### Tips for Teaching with the Internet Simulator

- Understanding the concept of the "shared wire" can be very challenging at first. The bit-sending devices students created in the previous lessons and the tool they are using today function differently, and students will likely need help developing the appropriate mental model.
- Students are setting the state of a wire but have no guarantee their partners will receive that message.
- In fact it is very likely students will set the state of the wire and then read their own message.
- Only through carefully coordinating when they set and get the state of the wire can students use this tool to communicate.
- Highlight this difference to students and feel free to emphasize that coordination / timing is the goal of the activity.
- A timer has been included in the widget to help coordinate timing. NOTE: the timer doesn't get or set the wire - it's just a pulse that students can follow to coordinate getting and setting.
- Groups should be encouraged to reconvene to add to or modify their protocol in order to be able to coordinate this kind of call-and-response message.

## Wrap-up (10 mins)

### Video - The Internet: Wires Cables and WiFi

- Show **The Internet: Wires, Cables & WiFi - Video**
- **NOTE:** The video is also available to students through Code Studio - it's the level after the Internet Simulator level.

#### Discussion:

Relate what's shown in the video to what students had to do in the previous two lessons:

- first making a bit-sending device
- second solving the coordination problem of sending bits using a wire that can only have 1 of 2 possible states.
- These are both problems that real engineers had to solve as well when building the internet for the first time.

If students have not already done so, they should calculate the fastest bit rate they were able to achieve. Lead a quick classroom discussion about the following topics, using these prompts:

- How has today's activity added to or altered your definition of a bit?
- A major focus of today's activity was timing and coordination. What things did you need to coordinate or agree about ahead of time in order for your protocol to work?
- What is the best bit rate you were able to achieve? What would it take to go faster?
- How fast do you think computers transmit bits?



## Assessment

### Content Corner

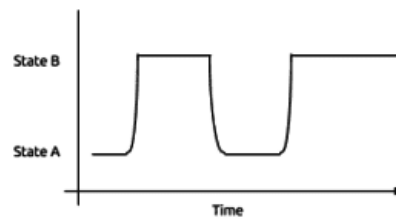
Students would be hard-pressed to achieve a bit rate much faster than 1 bit/sec in this activity. You might think that, say, 1,000 bits/sec would be fast for a computer, but even typical household modems in the early 90s had bit rates higher than about 10,000 bits/sec.

These days transmission rates are well in excess of millions of bits/sec (or Mbps - Megabits per second). Typical mobile phone speeds range from 5-10 Megabits per second,

wikipedia: modem

**Questions:** Note: Students can also respond to these questions in Code Studio.

- A binary message consisting of four bits was sent to you by a friend. The message was supposed to be ABAB. Unfortunately, your friend set the bit on the wire once every 2 seconds, but you read the wire once every second. Assuming that the first bit was sent and read at the same time, what message did you receive instead?
- A binary message was recorded as a wave, as shown in the image below. Can you decode the message? Explain what information you would need in order to successfully decode the message into A's and B's.



## Extended Learning

- Don't allow partners to decide who goes first ahead of time. Set up the challenge so that one partner could send a message to the other at any time and must verify that the partner is there and listening first.
- Continue to increase the number of bits that students must be able to transmit in the challenge and showdown.
- Require students to relay a message, but the response must be sent backwards. This ensures partner B must receive the entire message before responding. (e.g. if partner A sends: ABAB partner B should respond: BABA.)
- **Blown to Bits ([www.bitsbook.com](http://www.bitsbook.com)), Chapter 1 - Say:** Read about the following koans (or truths) of bits related to the Internet in Chapter 1, pages 4-13.
  - Koan 1: It's All Just Bits
  - Koan 2: Perfection Is Normal
  - Koan 6: Nothing Goes Away
  - Koan 7: Bits Move Faster Than Thought
  - Pick one of the above koans and address the following questions:
    - Argue if you agree that it is a "truth" and if it will always be a "truth."
    - How does this koan intersect with your life as a student?

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CL** - Collaboration
- ▶ **CT** - Computational Thinking

### Computer Science Principles

- ▶ **2.1** - A variety of abstractions built upon binary sequences can be used to represent all digital data.
- ▶ **2.3** - Models and simulations use abstraction to generate new understanding and knowledge.
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.3** - There are trade offs when representing information as digital data.
- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.



# Lesson 4: Number Systems

Unplugged | Concept Invention

## Overview

Students will explore the properties of number systems by effectively inventing a base-3 number system using circles, triangles and squares as the symbols instead of arabic numerals. Students are asked to create rules that explain how each arrangement of symbols can be generated or predicated as an orderly, logical series. The objective is to understand that you can represent **any** number with any agreed-upon set of symbols that appear in an agreed-upon order. This is as true for circles, triangle and squares as it is for the digits 0-9, or the number systems we commonly see in computer science (binary and hexadecimal).

## Purpose

In computer science it's common to move between different representations of numbers. Typically we see numbers represented in decimal (base-10), binary (base-2), and hexadecimal (base-16). The symbols of the decimal (base-10) number system - 0 1 2 3 4 5 6 7 8 9 - are so familiar that it can be challenging to mentally separate the written symbols from the abstract values they represent. As a result, using the digits 0 and 1 can be a distraction when learning binary initially, so we don't in this lesson.

We want to expose the fact that numbers themselves (quantities) are laws of nature, but the symbols we use to represent numbers are an arbitrary, man-made abstraction. Sometimes students memorize conversions from one number system to another without really understanding why. By effectively inventing their own base-3 number system in this lesson the goal is for students to see that all number systems have similar properties and function the same way. As long as you have 1) a set of distinct symbols 2) an agreement about how those symbols should be ordered, then you can represent any number with them.

## Agenda

### Getting Started (5 mins)

How many ways can you represent "7"?

### Activity (30 mins)

**Circle-Triangle-Square Activity - Create a number system using symbols**

### Wrap-up

**Present Student Circle-Triangle-Square Number Systems**  
**Discuss the Rules Created for the Number Systems**  
**Connect to Number Systems and Binary Numbers**

### Assessment

### Extended Learning

Extend to a 4 digit numbering system

## Objectives

Students will be able to:

- Reason about permutations and symbols as arbitrary abstract concepts that can be used to represent numbers.
- Invent their own "number system" with symbols and rules for getting from one permutation to the next.

## Preparation

Print and cut out (or have students cut out) the shapes from the **Teacher Resource - Number Systems: Circle-Triangle-Square - Resource** or from colored paper.

Each group of students needs at least 3 of each shape (a full set would be 10 of each shape).

## Links

### For the Teacher

- Teacher Resource - Number Systems: Circle-Triangle-Square** - Resource ([PDF](#) | [DOCX](#))
- U1L06 - Teaching Tips & Tricks Video** - Video ([download](#))

### For the Students

- Activity Guide - Number Systems: Circle-Triangle-Square** - Worksheet ([PDF](#) | [DOCX](#))
- Unit 1 on Code Studio**

# Teaching Guide

## Getting Started (5 mins)

### How many ways can you represent "7"?

Prompt:

"How many different ways can you represent the quantity "7"? Take one minute to write your ideas down before sharing with your neighbors."

- Give students about one minute for silent work and then use a share-out activity to share ideas from the class. We recommend just writing them somewhere the whole class can see them.

#### Discussion Goal

This discussion aims to prepare students to distinguish between the familiar symbols we use to represent numbers and the quantities we are actually representing.

For example, if there are seven apples on the table we can represent that fact by writing "seven", "7", "VII", "\*\*\*\*\*", seven tallies, seven apple drawings, and so on.

#### Discuss

- Do a whip-around and have students suggest the different symbols used to represent numbers
- Once you've got a decent list up that everyone can see ask a few directed questions:

#### Suggested Prompts:

- "If we kept going how many ways of representing "7" do you think we could come up with?"
  - There exist an infinite number of representations.
- "Why do you think we use the symbols we do use to represent numbers? Who decided that?"
  - To learn more about numbering systems read **Roman and 'Arabic' Numerals**.
- "If we were going to design a new system for representing numbers, what features would this system need to have?"
  - The answer to this is the subject of today's lesson.

#### Transitional Remarks

In the previous lessons you all invented ways to represent a set of messages with bits. Today we will focus on representing numbers. By the end of class, you will have invented your own number system.

## Activity (30 mins)

### Circle-Triangle-Square Activity - Create a number system using symbols

Preparation:

- Form teams of 2 or 3 students each.
- Distribute **Activity Guide - Number Systems: Circle-Triangle-Square - Worksheet**
- Each group should be provided paper shapes (at least seven of each). Alternately, provide students materials for making their own, or use tiddlywinks, poker chips, or any other little doo-dads you can find, as long as there are three different types of objects.

#### Activity: Circle-Triangle-Square

**Instructions:** (from the activity guide: **Activity Guide - Number Systems: Circle-Triangle-Square - Worksheet**)

Given 3 places to work with, make as many unique patterns as you can using only circles, triangles and squares.

The diagram on the right shows a few examples of some 3-place patterns. **NOTE: Order matters -- so, for example: Circle-Triangle-Square is a different pattern than Square-Circle-Triangle, even though both have one of each possible shape.**

#### Challenge 1 - Find all of the 3-place patterns

- **Record all** of the unique 3-place patterns you can find in the template started below.
- **How many are there?** Number each one to keep track. (Note there may be more or fewer total patterns than spaces provided)

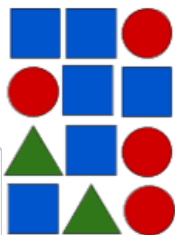
#### Content Corner

The purpose of using these three symbols (as opposed to digits or letters of the alphabet) is to ensure that the activity becomes a more true problem-solving exercise or puzzle. The shapes are enough in most cases to jolt students out of the context of math class and truly invent a number system of their own without realizing it at first.

Even though students might not come up with systems that we would think of as "common", creativity should be encouraged. It's possible to invent all kinds of rules to get from one pattern to the next.

The point is that number systems are man-made sets of symbols and rules.

- **Suggestion: try to find the permutations in some kind of organized or systematic way, rather than just randomly.**



\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

...

### Challenge 2 - make a system for generating all the patterns

Now that you've listed out all of the 3-place patterns of circles, triangles, and squares, let's put them in a systematic order. You can use any system you like, as long as you create and follow a clear set of rules for getting from one line to the next.

- **Jot down the rules of your system below.**
- **Suggestion: to test your rules, have someone follow them to see if they can recreate your organized list above.**

### Work Time

Give students a fair amount of time to get into groups and start to arrange the shapes. They should be trying to discover patterns and rules in an effort to find all the possible unique configurations.

Students will need to do three things:

- Use the cut-out shapes to explore and generate all the possible patterns.
- Organize the set of patterns in an ordered system of their own design.
- Write down the rules of their ordering system; a good set of rules will allow someone else to predict or generate each subsequent permutation is in the list.

### Teaching Tip

There are two major "beats" to this activity:

1. Discover all the 3-place patterns<sup>1</sup>
2. Figure out a way to order them so that the sequence is predictable.

Students should discover a total of 27 unique patterns.

When students record all of the patterns that they come up with on paper and number them it foreshadows assigning a numeric value to a distinct set of symbols.

You may need to emphasize that the goal of the activity is not merely to list all 27 permutations, but to develop a set of rules that could be followed to generate all of them.

Some students might quickly recognize that there are 27 distinct groupings. However, ordering them is frequently a challenge for students because outside the context of math class they might not immediately apply what they already know about number systems, especially place values.

Nevertheless, creativity should be encouraged. It's possible to invent all kinds of rules to get from one pattern to the next.

Good questions to direct students towards thinking in this way include:

- Could you always tell me which permutation comes next?
- Could a classmate easily follow your rules to generate the same order?
- Would your rules still work if I only asked you to make all the permutations of length 2? What if I asked you instead to make all the permutations of length 4 or 5?

<sup>1</sup> We're using the phrase "unique 3-place patterns" rather than the common mathematical word **permutation**. You may use **permutation** if you like, but it is not necessary vocabulary for CSP.

## Wrap-up

### Present Student Circle-Triangle-Square Number Systems

Use a share out activity to allow students to share their systems with their classmates. Either in groups or as a class, students should read through their classmates' rules, assess whether they are clear, and test them out to see how they are similar to or differ from their own rules and the patterns they generate.

### Remarks

#### You just made a number system!

If you have good rule for generating all the patterns, and for getting from one pattern to the next, and you have numbered each pattern so you have a symbol-to-number mapping, you have the beginnings of a number system!

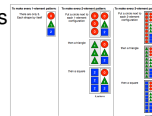
**Recall:** how many different ways you could write the number 7? Well, you now have another way using a system you just made up.

### Teaching Tip

Take a look at the next lesson on binary numbers to gauge how deeply you need to go into number systems for this wrap up. You might be able to blend the end of this lesson with the beginning of the next more seamlessly.

### Content Corner

**The diagram on the right** (click to expand) shows a technique for generating all the unique patterns with 3 shapes. The strategy mimics how we typically count in most number systems. Then, instead of shapes, if you just say circle = 0, triangle = 1, and square = 2 you can see how you might represent any pattern with 3 digits.



## Discuss the Rules Created for the Number Systems

### Prompts:

- Were some sets of rules easier to use than others? If so what do you think led to this difference?
- Do you think there are any limits to the number of the symbols we could use to represent numbers?

## Connect to Number Systems and Binary Numbers

At the end of the lesson a connection to number systems in general can be made and binary numbers in particular.

For example, after demonstrating a rule for the circle, triangle, square number system you could ask:

- **"What if we only had two symbols: a circle and square? Could we still make a number system?"**
  - The answer is yes, of course, and this is how binary numbers work (we'll see more in the next lesson)

You might then ask the related question:

- **"What if we had 10 symbols: a circle, a triangle, a square, a star, and so on...Could we still make a number system?"**
- This question should drive the point home - if you had 10 different shapes you could just make them operate like the digits 0-9.

If it would be useful, you might show the **Binary Odometer - Code Studio** widget that appears in the next lesson.

- The **Binary Odometer - Code Studio** shows a rolling count for 5 different number systems at once.

### NOTE:

By the end of this lesson students do NOT need to know the binary number system or be able to convert between decimal and binary. We will address **binary numbers in the next lesson** much more specifically, including the idea of **place value**. Only a general understanding of the concept of number systems must be established.

## Assessment

**Peer-assessment:** For a student peer-assessment, give students note cards or blank sheets of paper and direct them to write the first few permutations of their system on it. They will then trade their papers with another group, and see if the other group is able to predict the next two permutations in the system.

### Prompts:

- If you just had a circle and a square, how many 3-shape permutations could you make?
- Reflection: In 50 words or less, describe the concept of a number system. Why are rules required for a number system to be useful?

## Extended Learning

### Extend to a 4 digit numbering system

- Have one or more additional shapes cut out to provide to students. Ask students to extend their number systems to account for this additional shape.
- Ask students to extend their number systems to include 4 shapes or more.
- Try to identify which number a random permutation represents without counting all of the permutations that appear before it. Can you develop any rules?
- Peter Denning explains how "representations of information are at the heart of computing" in this article: **Computation: A new way of science**. Suggested activity: assign students to read and summarize the content. Follow with a class discussion.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CT** - Computational Thinking

### Computer Science Principles

- ▶ **2.1** - A variety of abstractions built upon binary sequences can be used to represent all digital data.
- ▶ **2.3** - Models and simulations use abstraction to generate new understanding and knowledge.

### Discussion Goal

The goal of this final discussion is to establish the general properties of all number systems. You can raise the idea that the systems students developed might be just as legitimate as the ones they use every day - just not commonly accepted. The only requirements for developing a number system are:

1. You must have a set of unique symbols
2. You must agree on a fundamental ordering of those symbols.

For example: circle comes before triangle, triangle before square. (similarly: 0 comes before 1, 1 before 2, and so on.)

If you have that, then you can count, and represent any number.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 5: Binary Numbers

## Unplugged | Reinforcement

### Overview

In this lesson, students will gain more familiarity with binary numbers. The lesson will transition away from the number systems that students created in the the circle-triangle-square activity, and begin to focus on representing numeric values using the binary number system. Though students have communicated with binary before, developing a **number system** is a little different. Previously, students mapped patterns of binary values to a small set of fixed messages. A number system is infinite, and also has rules for counting - or how to get from one value to the next.

### Purpose

Number systems help us express and reason about quantities. Early number systems were merely a system of tallies that allowed humans to record and perform simple arithmetic with values. The number system we use today uses the concept of place value to allow us to express any value we wish by combining only 10 symbols (0, 1, 2 ...). We therefore call it a “base 10” number system. When developing a number system for a computer, we only have two symbols available to us, corresponding with the two states of a single bit. However, the power of place value allows our binary or “base 2” number system to express any value we wish.

When using this binary representation of numbers, certain values (1, 2, 4, 8, 16, etc.) are seen repeatedly. When written in binary, these values are 1, 10, 100, 1000, 10000, and so on, and so are the incremental place values in this binary number system.

### Agenda

#### Getting Started (15 mins)

**Recap: Number Systems: Circle-Triangle-Square activity and properties of number systems**

**Presentation: From Circle-Triangle-Square to Binary**

#### Activity (30 mins)

**Construct a Flippy Do (10 mins)**

**Show the Binary Odometer (5 mins)**

**Complete the Binary Practice Activity Guide (15 mins)**

#### Wrap-up

**Discuss Students' Completed Activity Guides**

#### Assessment

#### Extended Learning

### Objectives

#### Students will be able to:

- Describe how to use bits to create a functioning number system
- Understand the relationship between the powers of 2 and the number of bits needed to express a number of a certain magnitude. e.g. How many bits do I need to represent the number “15”, or “32”, or “1492”?
- Determine, for a given number of bits, both the number of possible numbers that can be represented and also the range of those numbers

### Preparation

✂ Scissors (many pairs)

🖨 Printed copies of the **Flippy Do Template - Worksheet**

### Links

#### For the Teacher

- **Presentation Slides - Circle Triangle Square to Binary** - Slides
- **How to Make a Flippy Do** - Teacher Guide (**PDF** | **DOCX**)
- **U1L07 - Teaching Tips & Tricks Video** - Video (download)
- **Circle Triangle Square to Binary** - Video
- **KEY - Binary Practice** - Answer Key
- **Binary Game AppLab App** - App

#### For the Students

- **Binary Odometer** - Code Studio
- **Flippy Do Template** - Worksheet
- **Binary Practice** - Activity Guide (**PDF** | **DOCX**)
- **Unit 1 on Code Studio**

# Teaching Guide

## Getting Started (15 mins)

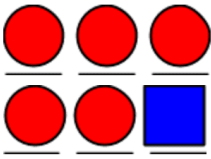
### Recap: Number Systems: Circle-Triangle-Square activity and properties of number systems

#### Recall

In the previous lesson you created 27 different 3-place patterns out of circles, triangles and squares, and tried to define a system of rules to generate all of the patterns.

#### Prompt:

- 🎓 "What if you only had a circle and square? With only a circle and square, how many 3-place patterns are there? Let's start some for you below, you make the rest. How many are there?"



#### Compare & Discuss:

Ask students to compare their results with their classmates' and clear up any discrepancies together or with the help of the class. Then lead a discussion around the following prompts:

- Why might we want to create a number system that includes only two symbols?
  - Then we can use bits as the building blocks of our number system.
- How large of numbers do you think this system can represent? How could we go higher?
  - Adding more bits (increasing the size of our permutations) allows us to count as high as we wish.

### Presentation: From Circle-Triangle-Square to Binary

To help with the transition from circle-triangle-square to binary you may use the following presentation - which we use at professional development to summarize the lesson.

The screencast video is intended to give the teacher a sense of how the presentation can go. You may show this video to students, though you are encouraged to try to do it yourself.

- Slides: [Presentation Slides - Circle Triangle Square to Binary - Slides](#)
- Video: [Circle Triangle Square to Binary - Video](#)
- Widget (used in video): [Binary Odometer - Code Studio](#)

#### Remarks

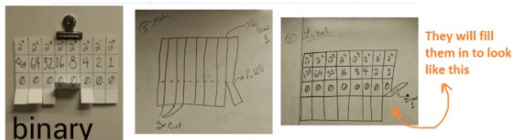
The number system we have just been introduced to is called the **binary number system** and can be constructed entirely from bits. This is the number system implemented in almost every computer. While it may look different from our familiar number system, as we'll see in today's lesson, it can be used in the same way and shares many properties.

## Activity (30 mins)

### Construct a Flippy Do (10 mins)

#### Remarks

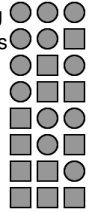
For today's activity you will be creating your own "Flippy Do."



#### Discussion Goal

The goal here is to begin the transition to binary by making a bridge from a number system using shapes again but this time using only 2 shapes (a circle and square) which is effectively a binary system.

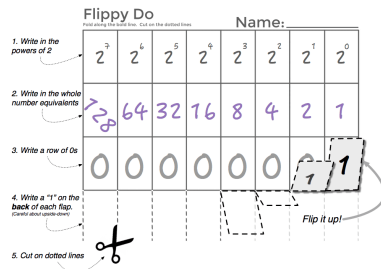
It should not take long to discover all of the 3-place patterns using only 2 shapes - **there are 8 in total** (see image at right)



This is a tool that will allow you to quickly and easily translate between decimal numbers and binary numbers, or in other words, between the numbers we use every day and bits.

## ✍ Make the Flippy Do

- **Distribute: Flippy Do Template - Worksheet** - one per student.
- **Display: How to Make a Flippy Do - Teacher Guide** (or just click the diagram image below to enlarge it)



**Complete Flippy Do:** Lead students through completing their Flippy Do's.

## Show the Binary Odometer (5 mins)

We'll keep filling in our Flippy Do's but you should also know about a helpful widget -- It was used in the opening demonstration -- called the "Binary Odometer"

- Go to the **Binary Odometer - Code Studio** (linked in Code Studio)
- This tool allows you to watch how counting is related across multiple number systems, binary, decimal and other number bases as well.
- Students should explore in pairs.
- Give students a few minutes to get acquainted with the Binary Odometer.

Tasks/Prompts to consider:

- "What's the largest number you can make in binary with the binary odometer?"
  - BIN: 1111111111 --> DEC: 1023
- "What happens when the odometer run out of numbers?"
  - Overflow! The binary odometer rolls back over to all zeros but the other numbers keep going up.

## Complete the Binary Practice Activity Guide (15 mins)

Students should:

- Complete the **Binary Practice - Activity Guide** individually (They can work with a partner but each should complete their own sheet )
- Use their Flippy Do and Odometer as resources

If students finish early, you may direct them to the **Binary Game AppLab App - App**.

## Wrap-up

### Discuss Students' Completed Activity Guides

Open up the floor to respond to any lingering questions that may have developed over the course of today's activity.

You can refer to the **KEY - Binary Practice - Answer Key** for answers.

Alternative: students may spend this time completing the two reflection questions on Code Studio.

## Assessment

### 💡 Teaching Tips

It may be necessary to demonstrate how values can be calculated by flipping up a "1" for each value required to arrive at the sum of values equal to the decimal number.

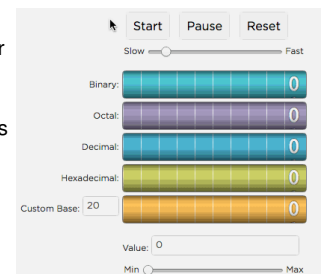
If students are having a difficult time understanding the rules of the system, remind them of the concept of place value and relate to base 10.

### 💡 Teaching Tip

The binary odometer is simply meant to be an interactive tool for looking at numbers in various bases. You do not need to linger on it when introducing it. Students can make use of it when completing the activity guide.

One thing to note about the widget is that you can rearrange the various odometers to place ones you want next to each other.

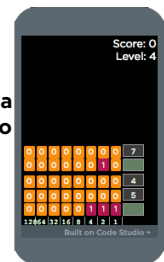
Just click and drag to rearrange.



### 💡 Teaching Tip

#### Binary Game AppLab App - App

Note this app was put together as a stand-in for the more popular Cisco Binary Game - Applet, the web-based version of which was discontinued in Sept. 2016. You can find it as a free iPhone app or if you register for the Cisco web site.





## ASSESSMENT

### Suggestions

- The Activity Guide can be assessed, or generate similar questions asking students to translate back and forth between decimal and binary. Encourage students to try to do so without the assistance of their Flippy Do once they become more comfortable.
- **Prompt** (written): Explain why this joke is funny: "There are 10 kinds of people in the world, those who understand binary and those who don't."

## Extended Learning

- How much memory does your music player, phone, or computer have? Do you notice anything related to binary?
- Practice translating between bases using the Cisco Binary Game.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CT** - Computational Thinking

### Computer Science Principles

- ▶ **2.1** - A variety of abstractions built upon binary sequences can be used to represent all digital data.
- ▶ **2.3** - Models and simulations use abstraction to generate new understanding and knowledge.
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 6: Sending Numbers

Internet Simulator | Group Problem Solving

## Overview

In this lesson students will return to the Internet Simulator in order to send a simple line drawing to a classmate. Students will be presented a grid on which they will draw an image (connecting 3-7 dots with straight lines). They must develop a protocol which will allow them to send any image they might create on their grids, paying particular attention to how many bits are used to represent each binary number. Students will therefore have additional practice encoding and decoding binary numbers and develop further intuitions about the properties of binary numbers in a hands-on way. The lesson concludes by testing protocols using a teacher-supplied test-image to transmit.

## Purpose

The main purpose of this lesson is to reveal that just because we now know how binary numbers work, there are still challenges of information representation and communication. We use the Internet Simulator to enforce some of the constraints imposed by real computers and systems. In particular we are interested in developing a more robust protocol for sending a list of numbers over the internet.

Students will need to make choices about how to structure their messages in terms of number of bits to use for each number and even how to communicate the beginning and end of the list. While the binary number system can be used to represent any value we wish, in practice the range of values we are able to represent is limited by the number of bits we use. Thus, protocols for exchanging binary information must specify in advance how many bits will form a single number among other things. Without knowing this information the receivers of a message have no way of determining how to break up an incoming stream of bits into individual numbers; it will just appear to be a random string of 0s and 1s.

## Agenda

### Getting Started (5 mins)

Review binary numbers.

Introduce the idea of a communication protocol.

### Activity (35 mins)

Introduce the Internet Simulator

Develop a Number Sending Protocol.

### Wrap-up (10 mins)

Does it work?

### Assessment

### Extended Learning

## Objectives

Students will be able to:

- Calculate the range of values that can be represented using binary numbers of a given size in bits.
- Invent a simple communication protocol for sending a list of numbers that represent coordinates on a cartesian grid.
- Use the Internet Simulator to send the list of points that make up the drawing to a friend.

## Preparation

(Optional) Rulers

(Optional) Extra copies of the **Sending Numbers for Graphing - Worksheet**

## Links

For the Students

- **Internet Simulator - Part 2** - Video ([download](#))
- **Sending Numbers** - Activity Guide ([PDF](#) | [DOCX](#))
- **Sending Numbers for Graphing** - Worksheet ([PDF](#) | [DOCX](#))
- **Unit 1 on Code Studio**

## Vocabulary

- **Protocol** - A set of rules governing the exchange or transmission of data between devices.

# Teaching Guide

## Getting Started (5 mins)

### Review binary numbers.

Review properties of binary numbers covered in the previous lesson, especially the relationship between the powers of 2 and the number of bits you need to encode a particular number.

Here are a few sample questions you can throw out:

- **"How many more numbers can be represented with 4 bits as opposed to 3?"**
  - Twice as many. Every time a bit is added the number of values we can represent doubles
- **"What is the highest value I can count to using 3 bits? What about with 4?"**
  - With 3 bits, 7. With 4 bits, 15. In general the answer is  $2^{(\text{number of bits})} - 1$
- **"Justify the following claim: Regardless of the number of bits in our binary number system, the first value we represent is 0."**
  - This is actually the case in any number system that uses place value. The smallest value we can represent is composed of all 0s and is exactly 0.

#### Discussion Goal

Students will likely need more time to review the binary number system if they have not seen it prior to this class. Provide students an opportunity to review key topics from yesterday, as they will be applying them in today's lesson.

#### Teaching Tips

Binary review should take about 5 minutes. Do not overly emphasize binary review here if you spent enough time with binary in the previous lessons. As long as students have a reasonable facility with the decimal equivalents of 5- or 6-bit numbers, that's enough to do the main activity.

### Introduce the idea of a communication protocol.

#### Remarks

In computer science, and when talking about transmitting information between computing devices, a protocol is a set of rules that tell us how to encode, communicate and exchange information.

Today, you and a partner will invent a communication protocol that allows you to send a list of numbers to represent a drawing.

DON'T WORRY - we have a **new version of the Internet Simulator** widget that abstracts away the coordination problems you had to deal with previously.

## Activity (35 mins)

### Introduce the Internet Simulator

#### Use a discovery-based method

(This tip first appeared in Unit 1 Lesson 3 introducing the "A/B" version of the internet simulator. The same principles apply here.)

#### Discovery-based introduction of tools in nutshell:

- Get students into/connected to the tool
- Give students working in partners a fixed amount of time (5 minutes or so) to poke around and see if they can figure out what it does and doesn't do -- typically this might be presented as a mystery worth investigating
- Ask the group to report what they found
- Teacher fill in any gaps or explanations of how the tool works **afterwards**

#### Familiarize yourself with the Internet Simulator?

If you are using this version of the Internet Simulator for the first time you should watch the student video: **Internet Simulator - Part 2 - Video**

### Code Studio levels

Levels

1

(click tabs to see student view)

## Levels

 3 4

(click tabs to see student view)

We encourage you to use a **"discovery-based introduction"** to this and all other tools in the Code.org curriculum. (See teaching tip at right).

 **Remarks**

- Today we will again use the Internet Simulator to explore some challenges of sending numbers on the Internet.
- The tool **has changed** from the last time you saw it! Let's find out what's different.

**Demonstrate/Remind students** how to connect to a partner in the widget **but explain nothing else**.

Give students the following 5-minute discovery challenge:

- **"Your job is to explore this tool with a partner - click all the buttons, type in the text areas what you can. You cannot break it so don't worry."**
- **"What's different now? There is a bit of a mystery in what the tool does...and doesn't do. Can you figure it out?"**
- **"You and your partner have 5 minutes to poke around and see what you can find."**

After 5 minutes bring the group back together and poll the group to see what they found.

Clarify any misconceptions about the tool at this point and then move on with the challenge.

- The major change is in how you send and receive messages:
  - You no longer have to coordinate setting and reading the wire with your partner.
  - Now the simulator is a two-way street, and everything you send will be received by your partner automatically.
  - You send a full sequence of symbols all at once, rather than one symbol at a time.
- Now the binary sends 0's and 1's instead of A's and B's,
- It is possible to see the decimal interpretation of the binary

 **Video: New Version of the Internet Simulator**

If it would be helpful, show the video: **Internet Simulator - Part 2 - Video** which explains and shows how the new version of the Internet simulator works.

 **Teaching Tip**

It is your choice whether to show the video here or not. Students may also watch it on their own once the activity is started if you prefer that approach.

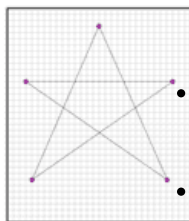
**Develop a Number Sending Protocol.**

**Distribute:**

- Activity Guide: **Sending Numbers - Activity Guide**(which includes both the activity and its rubric)
- Graph templates: **Sending Numbers for Graphing - Worksheet**.

**Challenge: (from the activity guide)**

Groups must develop a protocol or set of rules for communicating a drawing to their partners using only bits.

**Challenge Rules: (from the activity guide)**

- The image will be a line drawing created by connecting points on a grid, like the one seen here.
- The parties can discuss and agree on a protocol ahead of time, but the image exchange must happen without communication between the two parties, other than through using the Internet Simulator.
- You can only send a single message through the Internet Simulator to describe the whole image.

**During the Activity**

- Give students time to develop their protocols and practice encoding and decoding their images into bits.

 **Teaching Tips**

While students are creating their protocol, encourage by asking:

- What order will the points be sent in?
- How does the recipient know when one number ends and the next begins?

Keep the focus on what's required of a protocol. How many bits do you need to represent the numbers you need to represent the drawing, or coordinates? You want to be efficient but also support different kinds (sizes) of images.

To test protocols, direct groups to trade protocols and see if they can send an image using the alternate protocol. Students should give advice to each other to make suggestions for improvement.

- Students should be encouraged to make a simple drawing. A geometric shape, or the first letter of their name is a good option. A shape that can be drawn with about 3-5 points is a good goal. (The 5-pointed star example is a good one, and is shown in the activity guide)
- Students have been provided three different-sized graphs, on which they could scale their initial drawing to be different sizes or draw different images. Either way, it may necessitate the creation of different protocols.

#### Test It Out:

Students should test their protocols with one another under the conditions defined on the sheet. Let them know that there will be a more formal testing of their protocols at the conclusion of the lesson, in which you will provide the image that needs to be transmitted. Are they absolutely sure they've accounted for every possible scenario? Does everyone feel comfortable encoding and decoding a message?

## Wrap-up (10 mins)

### Does it work?

Give students an image as a "test" of their protocol.

- Make simple line drawings on graph paper for students to send to each other.
- Give the drawing to one partner and ask him/her to use the protocol to communicate it to the other partner.
- The other partner must draw what she receives on a blank grid.
- Compare the results to see if it came out right.
  - If it didn't, have students write a sentence or two about what went wrong, if the problem was preventable, and whether they need to change their protocol.
- Use the rubric to assess.

## Assessment

**Questions:** You may assess the outcome of the challenge run at the conclusion of the activity. Assessment questions appear both on the rubric and in Code Studio.

#### Additional Questions:

- Develop a protocol that allows the user to send a calendar date (mm/dd). What is the minimum number of bits necessary?
  - Answer: The minimum number is 9 bits.
- Develop a protocol that allows the user to send a time (use 24hr military time hh:mm:ss). What is the minimum number of bits necessary?
  - Answer: The minimum number is 17 bits.

## Extended Learning

- Develop a protocol designed for a graph that is not square, e.g. 50 \* 200. How does this alter the format of your protocol?
- Develop a protocol that can communicate locations on the surface of the earth. Longitude and latitude might be helpful tools to use.
- The "Scalable Vector Graphics" (SVG) image file format represents images as a text-based protocol. It can be used with HTML or as a standalone file. Check it out on [W3Schools](#) or [Wikipedia](#).

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ CT - Computational Thinking

### Computer Science Principles

- ▶ **2.1** - A variety of abstractions built upon binary sequences can be used to represent all digital data.
- ▶ **2.3** - Models and simulations use abstraction to generate new understanding and knowledge.
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 7: Encoding and Sending Formatted Text

Group Problem Solving | Internet Simulator

## Overview

In this lesson, students are first introduced to the standard number-to-text encoding scheme used in computers and on the Internet known as [ASCII](#) encoding. Students will invent a communication protocol that uses only plain text ASCII characters to encode fancier formatting for text such as fonts, colors, sizes, etc. Students will demonstrate their protocol by using the Internet Simulator to send an encoded message to a partner, who must correctly interpret the formatting and draw the result on a piece of paper.

## Purpose

This lesson gives a glimpse into "coding languages" by having students invent a way to use plain ASCII text to encode other text. At this point we really begin to see how layers upon layers of encodings - all tracing back to binary - work together to encode complex information.

We also want to make a connection to the Internet and protocols. Information traveling across the Internet will often need to contain both the contents of the message, itself, and information that helps to format, route, or interpret this data.

Developing means for differentiating between these two types of information has led to the creation of a number of ubiquitous protocols and languages. HTML (short for HyperText Markup Language) is the language in which the content and formatting of a web page are written. And the protocol HTTP, or HyperText Transfer Protocol, is another ASCII-based protocol that is the foundation of communication on the web - it was designed to send and receive web page data over the internet. In both cases, plain ASCII text is imbued with deeper meaning through the development of well-defined protocols.

## Agenda

### Getting Started (15 mins)

**Think Pair Share - How would you encode text?  
Introduce the ASCII encoding scheme.**

### Activity

**Formatting Text Challenge: Create a protocol for encoding formatted text**

### Wrap-up

**Discuss the results of the activity  
Compare/contrast encoding schemes with HTML  
Discuss layers of encodings**

### Assessment

**Lesson Assessment  
Chapter Assessment**

### Extended Learning

## Objectives

**Students will be able to:**

- Describe the ASCII encoding scheme.
- Design/invent a protocol for sending formatted text using the Internet Simulator.
- Invent a text formatting language.
- Explain the connection between binary and more complex encodings of formatted text

## Preparation

(Optional) Poster Paper

Markers or Crayons

Section prepped to use Internet Simulator in Code Studio.

## Links

**For the Teacher**

- [How to Administer a Locked Assessment \(PDF | DOCX\)](#)

**For the Students**

- [Sending Formatted Text - Activity Guide \(PDF | DOCX\)](#)
- [Rubric - Sending Formatted Text - Rubric \(PDF | DOCX\)](#)
- [Unit 1 on Code Studio](#)

## Vocabulary

- **ASCII** - ASCII - American Standard Code for Information Interchange. ASCII is the universally recognized raw text format that any computer can understand.
- **code** - (v) to write code, or to write instructions for a computer.
- **Protocol** - A set of rules governing the exchange or transmission of data between devices.

# Teaching Guide

## Getting Started (15 mins)

### Remarks

In previous lessons we explored how to encode numbers in binary, and you also developed protocols for sending a list of numbers. Today we're going to take that method one step further and look at how we can encode text with a binary representation. Hopefully you are beginning to realize that if we can figure out a way to represent information as a set of numbers, then we can encode it in bits and store that information in a computer or send it over the Internet.

### Think Pair Share - How would you encode text?

#### Prompt:

**"One of the most powerful uses of the internet is sending text to people. Since the internet can only send bits around we need a way to encode text with bits..."**

**"If it were up to you, how would you encode text in binary? Quickly, jot down an idea for encoding text."**

- Provide students a couple of minutes to write their thoughts.
- Emphasize that they do not need to actually specify every detail of the scheme but merely need to outline its structure.

#### Discuss

Have students compare and contrast encoding schemes with their neighbor first, then open up the discussion to the class. Here are a few prompts:

- Did you and your neighbor come up with the exact same idea? What was different?
- How many bits does your encoding scheme require? For example, how many bits would you need to say "hello"?
- Did you account for anything besides the letters of alphabet (or whole words)?
- Of the encoding schemes mentioned so far, which one is "best"? why?

### Introduce the ASCII encoding scheme.

### Remarks

You just invented your own scheme for encoding text with numbers. It turns out that there is a standard encoding for most of the symbols you can type on an American keyboard.

That encoding is called the **American Standard Code for Information Interchange** or **ASCII** (pronounced: "Ask-ee").

- Project or share an ASCII table for your students. **Here's an example chart.**
- Then present the key components of this encoding scheme.
  - ASCII codes were originally 7 bits long and so there are 128 possible values.
  - 0-31 are "control characters" that are largely defunct and go unused; they were formerly used to control various aspects of machines and printers.
  - 32-126 are printable characters and include the numbers 0-9, all 26 letters (both lowercase and uppercase), and many common punctuation symbols.
  - 127 is the symbol for delete.
  - Over time, 8 bits became a standard "chunk-size" for encoding information. ASCII made the transition to this 8-bit encoding by just adding an extra 0 to the front of the old 7-bit codes.

### Quick Activity: write your name in ASCII codes

Using the ASCII table, translate your first name from letters to numbers using the ASCII table.

### Teaching Tip

- A conventional way to do this is to make a simple mapping of the 26 letters of the alphabet to numbers, something similar to the example on the right. This is not, however, the only solution.
- Students should have the freedom to invent a text-encoding scheme however they like. Some students may wish to give common phrases or words their own sequences of bits rather than letter-to-number mappings.
- If students are moving quickly encourage them to add more functionality. How will their protocols encode punctuation, capital letters, special characters?

0	[space]
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I
...	...
26	Z

### Discussion Goal

Points to draw out in the discussion:

- Most probably invented a scheme that mapped letters of the alphabet and other text characters to numbers.
- There are tradeoffs in deciding how many bits per character you think you need. How much do students' suggestions vary?
- In order to **transmit** text data we're all going to have to agree on an encoding scheme (foreshadow [ASCII](#))

### Teaching Tip

Time permitting you might send students off to do their own "rapid research" on ASCII and report back. You should be able to draw out the same points.

- Write your name as: \*Name!" (capital first letter, exclamation point at the end)
- Give students a minute to do this

## 🎤 Transitional Remark

- Having a standardized protocol like ASCII to encode text enables us to send and receive textual information.
- This is very useful, but there are still instances when we will want even greater expressive power in our digital communications.

## Activity

### Formatting Text Challenge: Create a protocol for encoding formatted text

#### Introduction:

"What if you wanted to send formatted text that included things like the ability to underline, bold, or italicize words....specify a different font size, or color?"

💡 Things like this:

- ALL CAPS
- Highlighting
- Bold
- Italics
- Underline
- Font Color
- Large font

#### 💡 Teaching Tip

You might poll the class for other kinds of formatting you see with text. Some things that might come out: **tables or grids of text** different font faces (e.g. Arial, Times, gill sans, etc.) \* placement of text on the page, like floating text boxes, etc.

## 🎤 Remarks

Today your challenge is to:

- Invent a protocol for sending formatted text
- Use the Internet Simulator to test out your protocol.

You will also **notice that the Internet Simulator has been updated** so that you can now type ASCII text characters to send.

#### Transition to Code Studio

- Place students in groups of 2 or 3
- Distribute a copy of **Sending Formatted Text - Activity Guide** to each group and **Rubric - Sending Formatted Text - Rubric**.
- Students should **connect to their partner in the Internet Simulator**

💡 • Students should notice the new ability to send text.

### 📝 Invent a protocol for sending formatted text with the Internet Simulator

(from the activity guide)

#### 📝 Directions

Work with a partner or in a small team to develop a protocol that allows you to send formatted text.

#### Guidelines

Both the text and the formatting instructions must be derived from the printable ASCII character set (i.e. codes 32-126).

Your protocol must encode at least:

- bold, italics, and underlining
- three different font sizes (large, medium, and small)
- three different font colors (red, black, blue)
- You will demonstrate that your protocol works by sending a message with the **Internet Simulator**
- You will send a message and the recipient must be able to faithfully draw (or produce in some fashion) the formatted text, based only on the data she received. Here's a sample message:

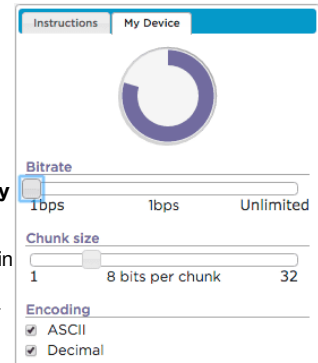
Whoa, this message sure has a lot of **FORMATTING** in it!

#### 💡 Teaching Tip

Before starting the activity you might want to have students just practice sending ASCII text messages to each other to get a feel for the new environment.

While doing this you should point out that **everything is still binary** underneath the hood.

You can use the **My Device** tab in the Internet Simulator to turn different encoding schemes on or off.





## Develop Your Protocol

Use the space below to brainstorm ideas for your protocol. Iteratively improve your protocol by testing it out with simple sample messages.

### Demonstrate:

- Test students' protocols by providing a formatted message to one member of the team and having them send the message to their partner using the Internet Simulator.
- Students may recreate the message by hand or in a text document and compare results to the intended message.
- For stronger proof, ask one member of each group to move to the other side of the room or hallway.

## Wrap-up

### Discuss the results of the activity

**Discuss** Use a group discussion strategy to address these questions:

- Were most groups successful?
- If not, what caused the most trouble?
- Were some components of the challenge easier to address than others?

### Compare/contrast encoding schemes with HTML

- Today's activity is motivated by the real-life challenge of bringing web pages to life.
- Students more familiar with HTML / CSS may recognize many of these ideas, but it can still be instructive to show the class that much of the information required to view a web page isn't the content itself, but information on how it should be formatted.

### Discuss layers of encodings

Do a quick **Think Pair Share** or perhaps assign this question as written work.

#### Prompt:

**"Take a moment to think about the layers of encodings that allowed for formatted text to be transmitted over the Internet."**

**"Imagine someone pointed to piece of formatted text and asked: 'Can you explain to me how this is encoded in binary?' How would you explain it?"**

- Give students a moment to jot ideas down.
- Discuss with neighbor.
- Share explanations with the class.

## Assessment

### Lesson Assessment

#### Rubric:

- Use **Rubric - Sending Formatted Text - Rubric** to assess students' work on the activity.

### Activity Goal

Students must define a protocol that allows for the encoding of text formatting, while only making use of the printable ASCII character set, i.e. 32-126.

A conventional way to do this is to specify a set of "reserved words" or characters, that should be interpreted as formatting instructions. For example, HTML uses angle-brackets <...> with opening and closing tags, for example:

```
<b>this is bold</b> and this isn't  
this is bold and this isn't
```

Students who have seen HTML prior to this class may come up with schemes for the simple problems easily; give them some of the more challenging encodings, such as adding column formatting to their text, placing text at an arbitrary location on the page (like a text box), creating tables, etc.

### Teaching Tip

Give teams time to develop their protocols, either on their Activity Guides or on a separate poster, document, slideshow, etc.

Encourage students to iteratively test their protocols to make sure they have not overlooked any gaps in their protocol.

### Content Corner

What students likely did in the activity was invent a text-based code. Whether they are formatting languages like HTML, or Markdown or programming languages like Java, C, or Python, all of these languages have one thing in common: they use ASCII text to encode other text or information.

Don't be shy about telling students that they just invented a coding language. At this point in the course a code and a protocol are very similar. Even though it's probably something no one else will use, the process students just went through gives a taste of inventing any kind of formal language or protocol that ultimately needs to be interpreted and processed by a computer.

### Optional Demonstration

- Most web browsers allow you to view the source code for a web site (e.g. Chrome allows this in Developer Tools).
- Point out the tagging system used to structure the text of a website. Ask students to consider how similar or different this protocol is to their own.
- W3Schools Introductory HTML:  
[http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

🗣️ **Questions** (found both on the Rubric and in Code Studio):

- How many bits are required to store the number “150” in ASCII?
  - 3 bits
  - 8 bits
  - 16 bits
  - 24 bits
  - 32 bits
- The word “Apple” translated into its ASCII number equivalent is:
  - 097 112 112 108 101
  - 097 108 108 111 119
  - 065 112 112 108 101
  - 065 110 110 105 101
  - 065 108 108 111 119
- What problems arose in your efforts to create a working protocol? How did you think about the problems in order to solve them?
- Describe one instance in which collaboration with a partner influenced the final protocol your team produced.

### Discussion Goal

There are many ways to answer this question. Any response that acknowledges:

- sequences of binary states can be used to represent numbers
- numbers can be assigned to letters of the alphabet to encode text
- with plain text you can make a **code** you can use to apply other meanings (or formats) to text
- You can invent a “formatting language” (like HTML) to represent different ways you want text messages to appear.

## Chapter Assessment

There is a multiple choice assessment for this chapter available in **Unit 1 on Code Studio**. It can be found on the stage right after this lesson and uses the Lockable Stages feature. If you are new to Lockable Stages check out **How to Administer a Locked Assessment**.

### Teaching Tip

Some questions on these assessment may seem ‘out there’ or only indirectly related to the material in the lessons. That is intentional as it is a good simulation of the kinds of questions students might find on the real exam. In many cases, a student can probably use their judgment and intuition based on what they’ve learned to make a pretty good guess at a question.

However as always these resources are just a suggestion and you should use them as best suits your class and their needs. The goal of CSP is to grow participation in computer science, so if offering this as a high stakes test early in the year will go against that goal maybe try going through the assessment in a lower stakes way (allow students to work with a partner, make the assessment worth less points, etc.) to practice for the future.

## Extended Learning

- Additional Formatting Encoding Challenges:
  - Special characters not found in the ASCII encoding (e.g. ñ)
  - Multiple columns of text
  - A text box at any location on the screen
  - A table of information
- Continue the exploration of HTML by determining how you would complete today’s activity in HTML. Further compare your own protocol with HTML.
- Read **Blown to Bits (www.bitsbook.com), Chapter 3**, Ghosts in the Machine, pp. 73-80 (What You See Is Not What the Computer Knows), then answer the following questions:
  - Give an example of your own when just knowing what a computer did wasn’t sufficient - you really needed to know how and why it was doing what it was doing as well.
  - Talk about file metadata and how it “fingerprints” a file. Include a discussion of file metadata benefits and challenges.
- Read **Blown to Bits (www.bitsbook.com), Chapter 3**, Ghosts in the Machine, pp. 80-88 (Representation, Reality, and Illusion), then answer the following questions:
  - How does highlighting in a PDF doc work? What are the computational ideas utilized?
- Continue the exploration of HTML by determining how you would complete today’s activity in HTML. Further compare your own protocol with HTML.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CL** - Collaboration
- ▶ **CT** - Computational Thinking

### Computer Science Principles

- ▶ **2.1** - A variety of abstractions built upon binary sequences can be used to represent all digital data.
- ▶ **2.2** - Multiple levels of abstraction are used to write programs or create other computational artifacts
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.3** - There are trade offs when representing information as digital data.

- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 8: The Internet Is for Everyone

## Unplugged | Research

### Overview

This lesson sets the stage for why we want to learn about how the Internet works. First students share what they currently know about how the Internet works through a KWL activity.

Then students watch a short video that introduces Vint Cerf and the Internet at a high level. Students then skim a memo written to the Internet Engineering Task Force (IETF) by Vint Cerf in 2002 entitled "The Internet is for Everyone," which calls out a series of threats to the prospect that the Internet should be an open, easily and cheaply accessible resource for everyone on the planet.

Finally we foreshadow the practice PT at the end of the unit. Many of the questions and challenges raised by Vint Cerf still apply today, and students will be asked to research and present on one for the Practice PT.

### Purpose

The purpose of this lesson is to set up and motivate students to be receptive to learning about some of the technical aspects of how the Internet functions. We want the message to be clear that a huge part of being able to solve these problems, or even to function as an informed citizen, is to be educated about how the Internet actually works as a system that is built, engineered, and maintained by people.

It is both interesting and important to know that the protocols or rules by which Internet traffic is governed are not owned or controlled by any government or business (at the moment). It's a group of well-meaning citizen-engineers dedicated to keeping the Internet free, open and robust for all.

The Internet Engineering Task Force (IETF) is the group of mostly volunteer citizens that proposes and develops all of the standards and protocols that exist on the Internet. Request for Comments (RFC) documents - like the one we use in the lesson - are how these standards and protocols are defined and published for all to see on the IETF website. They are some of the best-written technical documents in existence.

And with a little background in how bits work, what's necessary for protocols to work when bits are transferred over wires, they are relatively accessible reading. Here's the full set: <https://www.ietf.org/rfc.html>. RFC 000 (the first one) is related to what we ask students to do in the next several lessons.

### Agenda

#### Getting Started

#### Activity 1 - KWL The Internet (25 mins)

##### KWL the Internet

#### Activity - Vint Cerf: The Internet is for Everyone (20 mins)

##### Video: Introducing Vint Cerf

##### Vint Cerf: The Internet is for Everyone

#### Wrap-up (5 mins)

##### Foreshadow the Practice PT: Internet and Society

#### Assessment


#### Extended Learning

### Objectives

#### Students will be able to:

- Connect a personal experience to one challenge related to the idea that "The Internet is for Everyone".
- Cite one example of how computing has a global affect -- both beneficial and harmful -- on people and society.
- Explain that the Internet is a distributed global system that works on shared and open protocols.

### Preparation

 Copies of the activity guide to distribute (or distribute online)

### Links

#### For the Students

- **Primary Source: "The Internet is for Everyone"** - Activity Guide ([PDF](#) | [DOCX](#))
- **The Internet Is For Everyone - Abbreviated Version** - Resource ([PDF](#) | [DOCX](#))
- **Unit 1 on Code Studio**
- **What is the Internet?** - Video ([download](#))
- **KWL Chart** - Handout ([PDF](#) | [DOCX](#))

### Vocabulary

- **IETF** - Internet Engineering Task Force - develops and promotes voluntary Internet standards and protocols, in particular the standards that comprise the Internet protocol suite (TCP/IP).
- **Internet** - A group of computers and servers that are connected to each other.
- **Net Neutrality** - the principle that all Internet traffic should be treated equally by Internet Service Providers.

# Teaching Guide

## Getting Started

### Remarks

- So far in this class you have solved a few problems by creating and using small protocols for transmitting data over a wire to one other person.
- But the Internet is obviously **much bigger** than a single wire connecting two people. It connects billions of people and even more billions of machines.
- In the next several lessons, we're going to look at some of the technical issues involved with having lots of machines trying to communicate at the same time.
- In other words, we're going to take a deep dive into the inner-workings of the Internet.

### Teaching Tips

Don't linger too much with this introduction or the subsequent activity. This lesson is intended to be a quick one-day motivator lesson to have students understand the scale of the questions we're going to consider about the Internet and why learning about how the Internet functions is an important part of addressing those questions intelligently. This lesson should not go longer than one class period.

## Activity 1 - KWL The Internet (25 mins)

### KWL the Internet

#### Remarks

You may already know a few things about how the Internet works. Maybe you feel like you don't know anything, but in that case you might have questions or be curious about how things work.

To get started learning more about the Internet we want to get out into the open what we **know** and what we **want to know** more about.

It's okay if you don't know the **whole** thing. We want to collect the bits and pieces that we do know, and over the course of the next few lessons we'll put it all together.

#### Distribute KWL chart

- You can either have students just make 3 columns in their journals
- or (OPTIONAL) print out this version **KWL Chart - Handout**

**Thinking prompt:** (2 mins)

**"When you enter a web address in a browser and hit enter, what happens? At some point you see the web page in the browser, but what happens in between? What are all the steps?"**

**"Write down the series of things that you think (or have heard) happen right after you hit Enter. What happens first, second, third and so on. "**

**"Don't worry if you don't know all the pieces or how they all fit together. If you don't know a step, or you are fuzzy on some details, or there's a gap, that's okay. Just write down the parts that you know."**

Give students about 2 minutes to write.

**Pair** (5 mins)

Ask students to compare notes with their neighbor.

Generate list of things on post-its?

**Share** (10 mins)

On the board at the front of the room or in a shared document, have students put the things they know.

Cluster ideas by type, in a way that seems to make sense to you. Some categories that might come up:

- Routing messages to the right place

### Activity Goal

If you are familiar with KWL charts you can make this a more formal activity, but the intent is for this to be rapid and relatively informal.

When doing KWL as a group you should be cautious about accidentally making it seem like you're trying to make students feel "dumb" by exposing what they don't know. The Internet is complicated and hard for **anyone** to explain all at once. The point of the exercise is three-fold:

1. Get the bits and pieces that students do know of how the Internet works out into the open because you can refer back to these later.
2. That will hopefully generate a bunch of questions about how the Internet actually works - which we are likely to address in this unit or in unit 4
3. Motivate a desire to answer those questions, both intrinsically, out of genuine curiosity, and extrinsically, because of the Practice PT that concludes the unit.

- Addresses (e.g. IP addresses)
- Packets (breaking message down into small parts)
- Security/encryption concerns
- Translating URL to IP address
- Clients and servers
- Databases
- Internet Service Providers
- Physical mediums (e.g. WiFi v. Fiber optic v. copper wires, etc.)

**As a class review the list and Write down want to knows** (7 mins)

- Review the clusters of things students came up with.
- Identify gaps or mysteries that exist in the process.
- Is it totally clear what the sequence of steps is?
- You can get things started by asking some big questions like:
  - **"Who is in charge of the internet?"**
  - **"Does any country have its own Internet?"**
  - **"Is it possible to control what you see or have access to on the Internet? Who is in control, at what level? At what part of the process do they operate? How does it work?"**

- What are the gaps in what we know?
- What questions do you have at this point?

Give students a few minutes to write down their own questions.

#### 💡 Teaching Tip

To motivate some want-to-knows it's useful to think about a number of societal issues that hinge on knowledge of the internet and how it works.

- How does a web page come back to you and not someone else?
- Is a web page one big message? or multiple messages?
- How does a website remember who you are?
- What happens if a cable gets cut? Does the Internet fix itself?
- Who is in charge of the Internet?
- Who pays for the Internet?
- Where/how might someone spy on you?
- Who controls what you are allowed to see?
- If a government wanted to restrict access to the internet, how would they do it?
- Who can see your Internet activity? how?
- Are there parts of the internet you're not allowed to see?
- Where/how would a hacker steal your identity?
- What exactly is being attacked during a cyber attack?

## Activity - Vint Cerf: The Internet is for Everyone (20 mins)

### Video: Introducing Vint Cerf

Show this brief 2-minute video in which Vint Cerf (one of the fathers of the Internet) asks "What is the Internet Really?" **What is the Internet? - Video**

#### 🎤 *Transitional Remarks*

In the video Vint Cerf says that nobody and everybody is in charge of making the internet work but the reason it all works together because everyone uses the same protocols.

So who develops these protocols? Who makes the final decisions? Who is in charge? The amazing thing is that no single person, government, or corporation is in charge.

Rather, it is a collection of citizens and volunteers interested in defining the standards who formed a volunteer organization called the **Internet Engineering Task Force** to develop and promote voluntary internet standards [IETF](#).

#### 💡 Teaching Tip

Vint Cerf is widely regarded as one of the fathers of the modern internet. He will be a recurring character for the rest of the unit, mostly in videos as a kind of tour guide. We introduce him here asking some big questions.

### Vint Cerf: The Internet is for Everyone

**Distribute either:**

- The full document: **Primary Source: "The Internet is for Everyone" - Activity Guide**
- Abbreviated version with just the 9 challenges **The Internet Is For Everyone - Abbreviated Version - Resource**

**Review the Introduction and Background to the document.**

(This text is in the full version of the handout)

After the terrorist attacks of Sept. 11, 2001, Vint Cerf was concerned about maintaining some of the principles of openness and access upon which the Internet was built and founded.

He wrote a memo entitled **The Internet is for Everyone** as a plea to the the people who write and define standards (the IETF) to make sure the way the Internet is engineered maintained the philosophy that the Internet should be available and useable by everyone in the world, regardless of race, gender, creed, location, or wealth.

In the memo, he lays out the state of the Internet and its usage in 2002, predicts the future, and then presents a series of nine challenges or threats to the idea that the Internet is for everyone. The last several paragraphs begin with the phrase: **"Internet is for everyone - but it won't be if..."**

#### Instructions for students

- **With a partner, skim the document** and look at the 9 "Internet is for everyone - but it won't be if..." challenges laid out at the end.
- **With your partner pick one or two of the challenges** that are the most meaningful to you, or relate to some experience you've had in your life.
- **Be prepared to:**
  - Read the statement you chose: "Internet is for everyone - but it won't be if..." and then explain in your own words what it means.
  - Explain why that particular challenge is meaningful to you or relates to some experience you've had.

#### Whip around share-out:

1. Ask for a volunteer to briefly state:
  - One of the challenges they chose.
  - The example they thought of for that challenge.
2. Ask for another volunteer who chose a different challenge to do the same thing...and so on until all or most of the challenges have been addressed or you are out of time.

## Wrap-up (5 mins)

The "Internet is for Everyone" is actually a philosophy about how people should be connected. That philosophy is expressed in the way the Internet standards and protocols were engineered. In order to understand that philosophy over the next several lessons we'll be learning about the systems of protocols that work together to make the internet function.

### Foreshadow the Practice PT: Internet and Society

Why learn about how the Internet works? As Vint Cerf Says: "You can't escape from contact with the Internet. So why not get to know it?" But you don't have to take Vint Cerf's word. Some of the largest issues facing society hinge on an understanding of how the Internet functions.

**At the end of this unit you will do a Practice PT** about one of these societal issues. You will research one of these modern issues and prepare a flash talk (a 2 minute presentation) that explains the technical aspects of the Internet that underlie that issue. As you go through these lessons keep your ears and eyes open for how things work.

Many of the issues are related to people taking advantage of the open protocols that make the internet function and present us with tricky dilemmas.

For example there are two major issues to think about:

- **Net Neutrality** is a raging legal debate about the principle that Internet service providers should enable access to all content and applications regardless of the source, and without favoring or blocking particular products or websites.
- **Internet Censorship** is the attempt to control or suppress of what can be accessed, published, or viewed on the Internet by certain people. This can be used to protect people (i.e. to not allow access to child pornography) but can also be used to limit free speech.

To have an informed opinion though it helps to understand the technical underpinnings of how the internet works.

Finally, a major issue that our society faces is that far too few people actually understand how the Internet works! We are going to change that over the next few lessons.

## Assessment

**Questions** (also on Code Studio):

### Teaching Tip

In the interest of time, you may want to review the background material on the first page together as a whole class, or just read it to students yourself. NOTE: The **abbreviated** version of the document **does not** contain this introductory text but the full version does.

Students should know the circumstances under which "The Internet is for Everyone" was written and generally what its goals were. They should also know what the **IETF** is.

#### While students work with the document...

- Students will probably need 10 minutes or more to process the document.
- Students should not be doing research but should be trying to make personally meaningful connections to the challenges presented in the document.
- Most of the threats posed in the document are either about access, or freedom and privacy which should be things students can relate to on some level.

### Discussion Goal

The important factor here is to make the class student-centered. They should get to know each other's stories. From the students' examples, there should emerge at least a few powerful examples how much the Internet means to our society, our quality of life, and freedoms. Difficult economic issues of access to the Internet may emerge, as might sensitive issues related to cyberbullying or crime. Be sensitive to these issues if they emerge.

1. The Internet Engineering Task Force (IETF) defines the protocols and standards for how the Internet works. The members of the IETF are:
  - a. An International coalition of government agencies who oversee the Internet in their countries.
  - b. The worldwide leaders of the Tier 1 Internet service providers.
  - c. A loosely organized collection of citizens and engineers who communicate mostly by email.
  - d. Political leaders and heads of state.
  - e. There are no members. IETF is an "organization" in name only.
2. Choose the two best answers to complete the sentence: If I understand how the Internet works, then I will be able to:
  - a. make informed choices to support or oppose decisions my government makes about access to the Internet.
  - b. connect the latest devices to the Internet.
  - c. make informed choices about my privacy on the Internet.
  - d. get the best price for my cell phone plan.
  - e. speed up my downloads of movies I purchase.
3. Explain one challenge raised by Vint Cerf in "The Internet is for Everyone" and give one example of it that you know about.

**More Examples and reasons to learn how the Internet functions:**

- The people who propose the laws, or judge them often don't really know how the Internet works, what makes sense and what doesn't. (See: The Internet is a series of tubes).
- Citizens don't know how to protect themselves from cybercrime, or often live in a cloud of fear and uncertainty not only about the law, but their rights, and about what's actually technically possible.
- Hackers take advantage of unknowing, unsuspecting people.
- Corporations who provide Internet service need to balance profitability with providing access to everyone.
- Countries restrict access to the internet or monitors activity - how is that technically possible? What is it they are actually doing?

## Extended Learning

- Reading RFCs can actually be fun, especially the early ones! Here are all the RFCs: <https://www.ietf.org/rfc.html>
- **Blown to Bits (www.bitsbook.com), Chapter 1**, pp. 4-13. - Read about the following koans (or truths) of bits related to the Internet:
  - Koan 1: It's All Just Bits
  - Koan 2: Perfection Is Normal
  - Koan 3: There Is Want in the Midst of Plenty
  - Koan 6: Nothing Goes Away
  - Koan 7: Bits Move Faster Than Thought
  - Pick one of these koans and address the following questions:
    - Argue if you agree that it is a "truth" and if it will always be a "truth."
    - How does this koan intersect with your life as a student?

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CI** - Community, Global, and Ethical Impacts

### Computer Science Principles

- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.
- ▶ **7.3** - Computing has a global affect -- both beneficial and harmful -- on people and society.
- ▶ **7.4** - Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).



# Lesson 9: The Need for Addressing

Group Problem Solving | Internet Simulator | Unplugged

## Overview

In this lesson, students explore more deeply how communication between multiple computers can work over the Internet. They do this by playing a simplified game of Battleship, in which the first game is played unplugged, in their table groups, and the second game is played using the Internet Simulator, so that multiple students can connect to each other and see each other's messages. Students must devise a messaging protocol that makes it clear who is sending the message and who the intended recipient is.

Students then devise a **binary protocol** for playing this game which will entail developing an addressing system for players, as well as a formal packet structure for transmitting data about the state of the game.

**NOTE:** this is a large lesson that will likely need to span 2 days of class.

## Purpose

This lesson builds up to students learning about the "Internet Protocol" addressing system, commonly known as IP Addresses.

Up to this point in the course, the Internet Simulator has only assumed point-to-point communication. The "protocols" students were inventing mostly had to do with encoding information in binary, rather than also encoding communication information required for successful delivery of the message.

The lesson mimics a real problem that designers of the internet had to solve. Many computers are connected together by networks. So if a computer sends some bits out, those bits pass through many computers. How does a computer know who those bits are for? If a response is necessary, how does it send it back? This is a complicated question, but the first part of the answer is that you need an addressing system, and some kind of fixed structure for messages that everyone agrees on (a protocol) so messages can be interpreted properly. (We'll get to routing later).

The Internet Simulator for this lesson is setup to "broadcast" every message to every person in the group. Since everyone receives any bits that were sent by anyone else, a method for identifying the intended sender and receiver of the message is needed. Students will invent a protocol for addressing messages.

Many network systems, such as local ethernet and WiFi, rely on addressing schemes to make sure bits are received by the correct computer based on address and for other computers to simply ignore messages not intended for them.

## Agenda

### Getting Started (20 mins)

**Introduce "Broadcast Battleship"  
Battleship Setup**

### Activity: Silent Broadcast Battleship (25 mins)

**Introduce new version of Internet Simulator (Broadcast)  
Refine and Reflect.  
Play Game. Round 2.**

### Activity 2: Invent a Binary Protocol for Battleship (20 mins)

**Overview**

## Objectives

**Students will be able to:**

- Explain why messages need to contain addressing information (sender/recipient identification).
- Invent an informal addressing protocol for use in the Battleship game.
- Recall that browsing the Internet entails computers sending each other requests and sending back data to satisfy those requests.

## Preparation

Copies of the Battleship game board for each student

Poster paper or methods/materials for students to present

## Links

### For the Teacher

- **U2L02 - Teaching Tips & Tricks Video** - Video ([download](#))
- **Video Guide KEY for "IP Addresses and DNS"** - Answer Key

### For the Students

- **Broadcast Battleship Game Board** - Activity Guide ([PDF](#) | [DOCX](#))
- **Broadcast Battleship Rules** - Activity Guide ([PDF](#) | [DOCX](#))
- **Internet Simulator - Part 3** - Video ([download](#))
- **IP and DNS Video Worksheet (Optional)** - Video Worksheet ([PDF](#) | [DOCX](#))
- **Invent a Binary Protocol for Battleship** - Activity Guide ([PDF](#) | [DOCX](#))
- **Unit 1 on Code Studio**
- **The Internet: IP Addresses and DNS** - Video ([download](#))

## Vocabulary

- **IP Address** - A number assigned to any item that is connected to the Internet.
- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information.
- **Protocol** - A set of rules governing the exchange or transmission of data between devices.

**Discuss & Share**  
**(Optional) Exemplar Solution**

**Wrap-up**

**Real IP Addresses - Background and Video.**  
**Discuss**

**Assessment**

**Extended Learning**

# Teaching Guide

## Getting Started (20 mins)

### Remarks

So far we have only solved Internet problems when you are connected to one other person (so-called "point-to-point" communication). Obviously, the Internet is bigger than that, and today we're going to look at problems that involve multiple people.

### Introduce "Broadcast Battleship"

- We will start by playing a game that simulates some issues that arise when constructing the internet.
- We will play a crazy game of Battleship where instead of playing against one other person you will **play multiple games against multiple other people simultaneously** -- We call this "Broadcast Battleship"
- In our version today, you will play in **groups of 3** (4 is OK)
- To make it easier to track, we've also simplified the playing board to just a **3x3 grid** (instead of the classic 10x10)
- And we'll just play with paper and pencil.

### Battleship Setup

- Arrange students in table groups of 3. (A group of 4 is OK, but not recommended.)
- (Optional) Distribute **Broadcast Battleship Rules - Activity Guide** to each student.
- Distribute **Broadcast Battleship Game Board - Activity Guide** to each student.
- Review the rules for the multi-player version, and answer any logistical questions.

### Play Broadcast Battleship - Round 1 - unplugged

- **Start Game Unplugged.**
- Give students a couple minutes to play the game.
- Students might not finish the game, but students should get a good feel for how the game works and what the mechanics of the game are.
- **Pause the Game After 5 minutes**
- After a couple minutes interrupt the game
- Assure students they'll return to playing in a minute
  - they record their current spot in the game if necessary to pick up where they left off

### Transitional Comment

- It seems that most of you have figured out a way to play battleship with your group.
- You will get to continue to play your game in a few minutes.
- We are going to add a **new challenge: Your team is going to have to play Battleship without talking.**
- You will only be able to **use the Internet Simulator to communicate**

### Teaching Tip

#### How to demonstrate the rules of Battleship

Not every student may have played Battleship before and even if they have, this mini-version of the game is different from the real thing. You should consider how you want students to learn the new rules and play with pencil and paper rather than the typical peg board. In the resources above there is a printable copy of the **Broadcast Battleship Rules - Activity Guide**, but there are alternatives to simply distributing those rules.

**One possibility:** Demo two player version in large format.


- Draw a large version of the battleship grids from the activity guide on two pieces of chart paper or on the blackboard where everyone can see.
- Nominate two students to stand back to back (so each student cannot see what the other is marking) to mark up the grids following your instructions - they will each place one ship that is two boxed in size, and then try to location and sink the other student's ship.
- Once people get the gist of playing this paper and pencil version of battleship, ask "okay, what if we added a 3rd person so that each person was playing against two others simultaneously?"
- Then have the students get into the actual activity and figure out the rest in their group of three.

#### Other possibilities:

- Project the rules on a screen at the front of the room
- Have students read/review a digital copy of the rules
- Have students play the game for "homework" - some free online versions of the game exist.
- Ask a student who knows the rules to explain. Who has played Battleship? How does the game work? What's the object of the game? What are the rules?

## Activity: Silent Broadcast Battleship (25 mins)

### Introduce new version of Internet Simulator (Broadcast)

 Transition to the Internet Simulator found in Code Studio for this lesson

- Students log in to Code Studio, and navigate to the lesson: "The Need for Addressing"

Highlight the **differences in the new version**:

1. You connect to a "Room" with other people, instead of an individual partner.
2. Every message that is sent gets **broadcast** to everyone in the "room", including you!

## Continue the game on the Internet Simulator

- Give students 5-10 minutes to finish their game using the Internet Simulator.
- Enforce the **no-talking rule**
- Students should quickly discover that they need to "address" their messages.
- That is, they need to indicate the sender (themselves) and the intended recipient somewhere in the message, in order for the game to work.
- Different tables (and different player pairs) may invent different ways of doing this. Some examples:
  - "Hi, Tom, it's Mary."
  - "Mary to Tom."
  - "Mary → Tom"
  - "To Tom, From Mary"
  - "Mary is asking Tom right now", etc.

Some methods may be more informal, or buried in content of the message:

- "Tom, B4. This is Mary, by the way."
- "That B4 message was for Tom. It's Mary."
- "I just sent that message. This is Tom. No, the other message."

## Refine and Reflect.

**After the 5-10 minutes** allow groups to talk. They should discuss standardizing their protocol for sending messages. They should consider:

- What protocol have they been using? Did they have a protocol at all?
- How can they standardize their communication?
- How do they make their message as clear as possible?
- How do they make their message as efficient (short/easy to interpret) as possible?

## Play Game. Round 2.

After groups have had a chance to coordinate and refine their protocols, give them a chance to try it out on a fresh game.

- Have students return to the Internet Simulator, start a new game and test out their protocol.
- Remind them of the No Talking rule.
- If necessary, distribute a clean copy of **Broadcast Battleship Game Board - Activity Guide** for use in this round.

**After 5 minutes** or enough time to try out their new protocol, transition to the next activity.

## (Optional) Individual Reflection

- Before moving on to the group whip around, have students complete the reflection question in Code Studio or in their journal:
  - How did your group decide upon a system for sending messages in the Battleship simulator?
  - Were there disagreements or problems?

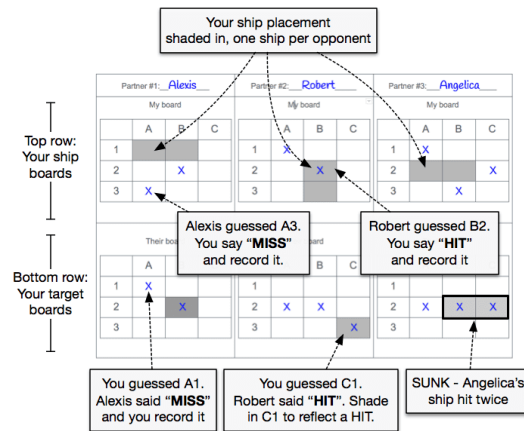
## Activity Goal

### Why Battleship?

Teachers may want to substitute other games for battleship in this activity for a variety of reasons. However, if you do that be aware that battleship was chosen because it has several properties that force concepts of internet protocols to come out. Battleship has a relatively fixed size message length, a need to encode several different pieces of information (coordinates), to and from addressing, and a verification message (hit or miss).

### Example Game Board

Below is an example game board included in **Broadcast Battleship Rules - Activity Guide** (click to enlarge)



## Teaching Tip

The first time students visit Code Studio for this lesson a **tutorial video** will pop up to show how this works.

You may want to do one of the following:

- show this video to the whole class
- have students watch it individually
- have students skip the video and demonstrate the behavior to everyone

## Teaching Tip

- You will probably need to **enforce the NO TALKING rule**
- Without talking, communication protocols will organically come out as students try to play the game in this new mode of the Internet Simulator.
- Assure students that they **will get a chance to talk and collaborate** before trying again.

- How were they resolved?

## Activity 2: Invent a Binary Protocol for Battleship (20 mins)

Distribute and go over **Invent a Binary Protocol for Battleship - Activity Guide**

(From the Activity Guide)

### Overview

Previously you came up with a method for exchanging messages on an open broadcast channel to play multiple games of Battleship at once. Now that you've played Battleship this way, with your group or with a partner, describe an efficient **binary protocol for playing a 3-person game of Battleship that can be played accurately over the Internet Simulator**.

Let "efficient" mean that your protocol uses the smallest reasonable number of bits (0s and 1s) to make messages for Battleship that still contain all of the necessary information for playing the game.

#### Give groups about 10 minutes to work

- Use the teaching tips to help students along if they struggle to start.

### Discuss & Share

After groups have had some time to work bring the class together.

- Ask if any group would like to share their protocol
- After the group has presented open up the discussion to draw out similarities and differences. Here are a few prompts:
  - "Did any other group do something similar? or something completely different?"
  - "What pieces of information were common across all of the protocols? Sender, recipient, cell number, hit/miss info."
  - "If we were to play a different game, what data would stay the same? What would change?"

Use this question to **transition** to the wrap-up:

**What do you think all of this has to do with the Internet?**

### (Optional) Exemplar Solution

You may choose to share this with your students or not. You might just use this exemplar as background knowledge to help you guide students' solutions.

**A simple, but clever encoding protocol is to recognize that the addresses, coordinates, and hit/miss messages can each be encoded with 2 bits (i.e. a list of 5 numbers in the range 0-3), and that you can make a combined message protocol that would allow a player to communicate anything needed for the game.**

**The scheme shown below uses 10 bits total. The last two bits are interesting: they indicate whether this is a shot being fired for the given row and column or whether it's a hit/miss response to the given row and column. Since the respondent who says hit or miss must verify the coordinates of the shot they are responding to, the recipient doesn't have to remember the last shot they called.**

#### Teaching Tip

At this point, playing the game is less important than moving towards developing a clear and efficient protocol.

Students might notice that a concise protocol makes it harder for them as humans to play the game, but remind them that they are the computers in this activity.

Why would it be important to keep the message short and standardized? Make a connection to the protocols from previous lessons: **Sending Numbers** and **Sending Text**.

#### Teaching Tip

If students are having difficult getting started:

- The activity guide contains a number of suggestions and questions to get thinking started -- Start by lead students through these questions
- Use the exemplar solution as a guide to coerce students into thinking about at least some element of that to get un-stuck.

Here are a few other prompts you can use

- How will you standardize the recipient and sender addresses?
- Should you encode people's names or (hint) use a number?
- How would a recipient of your message know where address ends and the other begins?
- What other information do you need to include?

#### Discussion Goal

A good solution uses a **numeric** address (not tied to a person's name) and a **fixed length** message. The number allows you to uniquely identify players (like a phone number) and having a fixed length helps know where each "chunk" of the message starts and ends.

**How many bits do you need for addresses?** -- For 3-person Battleship, you only need 2 bits for an address, since there are only 3 people; each can have a unique code.

**How many bits are needed for the total message?** -- Answers to this will vary and creativity should be encouraged. However, it should be evident from the students' protocols that they recognize the fundamental elements of the game and what might be able to be compressed into only a few bits.

(See the example solution).



## Wrap-up

### Real IP Addresses - Background and Video.

It turns out computers on the Internet are addressed in a similar way to phones for many of the same reasons. The real addresses used on the Internet are called "Internet Protocol Addresses" or IP Addresses for short.

#### Video: The Internet: IP Addresses and DNS - Video

- Have students watch the video (display for all, or have students watch in Code Studio - it's the level after the Internet Simulator)
- Have students complete **IP and DNS Video Worksheet (Optional) - Video Worksheet**
- You may **stop the video at 4:10** so that only the content on IP addresses is covered, as DNS will be covered later. In this case, you should split the video worksheet into two parts.

### Discuss

You should review the material covered in the video.

The questions in the **IP and DNS Video Worksheet (Optional)**

- **Video Worksheet** are a good place to start. The essential ideas and vocabulary students should know are:

- IP Address
- IP Packets
- IPv4 versus IPv6

**Optional Quick activity:** What's the IP address of the computer you're using right now?

There are a number of ways to figure out your IP address. An exploratory way is to do a web search for: "What's my IP address?"

That will lead to finding something, but you should be cautioned that the answer isn't always simple.

For example, if you are in a computer lab and use a site like **What's my IP Address?** it's likely that all of the computers in the classroom will report the same address. This is because the computers in a computer lab are probably using the same shared connection and might look to the outside world like they're all coming from the same place.

You can also look at a computer's settings to see what it thinks it's IP address is.

## Assessment

**Questions** (also on Code Studio):

1. To improve communication in playing Battleship on the Internet Simulator your group invented a(n):
  - a) heuristic

#### Teaching Tip

##### Do some rapid research (Optional)

At this point, you could have students look up the real IP packet structure and it might make some sense to them. They might find it satisfying and empowering to be able to understand a more technical article. The Wikipedia article on IPv4 packets shows how all the bits are laid out:

[https://en.wikipedia.org/wiki/IPv4#Packet\\_structure](https://en.wikipedia.org/wiki/IPv4#Packet_structure)

However, we have not yet covered a major aspect of packets, and the real need for their existence, which is that they are used to breakdown very large messages into smaller chunks to improve reliability of transmission. **We cover packets in a future lesson.**

- b) algorithm
  - c) protocol
  - d) encoding
2. Which of the following information is not necessary in your protocol to play Battleship?
- a) cell number
  - b) from
  - c) to
  - d) time
3. Even with a good protocol, what are some privacy and security challenges while playing Battleship on the Internet Simulator?

## Extended Learning

- **Security and Privacy:** If the messaging scheme your group designed was really used on the Internet, what could go wrong in terms of security and privacy? Think about what a malicious person could do. What if the messages sent weren't about a game but about something more personal like a conversation with a friend, or communicating with a bank?
  - There are two major security problems with the current version:
    - Anyone can fake a return address. Bob could send you a message and claim it's "from Alice," and you'd have no way to confirm who sent it.
    - Everyone can view everyone else's messages! Because all messages are "broadcast" over Internet Simulator, messages have no expectation of privacy.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CL** - Collaboration

### Computer Science Principles

- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.
- ▶ **6.3** - Cybersecurity is an important concern for the Internet and the systems built on it.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 10: Routers and Redundancy

Internet Simulator | Whole Class Discovery

## Overview

In this lesson students explore the benefits (and potential security concerns) associated with routing traffic across the Internet. Building on their introduction to IP addresses in the previous lesson, students use a version of the Internet Simulator that allows messages to be sent only to an intended recipient, as indicated by the IP address. The Internet Simulator also allows students to examine the traffic that goes through all of the (simulated) routers on the network. They will discover that messages go through many different routers, may not always take the same path to reach the final destination, and that the routers (and their owners) can **see all of this traffic!**

## Purpose

The main purpose here is to use the Internet Simulator to dive into examining the network traffic that's generated by students in the class. Seeing traffic like this in real life is very hard if not impossible since only engineers of large networks have access to router data on a large scale, and it's also more complicated to understand. While our network is simulated and provides a simple view, the basics are the same, and quick investigations reveal interesting things. We want students to simply get a taste for both the scale and the implications of what routers do.

CS Background: A router is a computer designed to receive and redirect packets of information based upon the addressing information (e.g. an IP address) contained in the packet. Routers will either deliver a packet to its final destination or forward it to one of several other routers it is connected to. By monitoring current network conditions, a router can determine which of these will allow the packet to reach its destination fastest. There will often be redundant paths between two locations on the Internet, and so if one path is experiencing traffic or otherwise out of service, additional paths will be available. This redundancy makes the Internet more reliable and also helps the Internet to scale, accommodating new users (and routers!) as they are connected to the system.

## Agenda

### Getting Started

#### Activity

**Introduce: New Version of the Internet Simulator - Routers**  
**Activity: Investigate Routed Traffic**

### Wrap-up

#### Assessment

#### Extended Learning

## Objectives

### Students will be able to:

- Describe the redundancy of routing between two points on the Internet.
- Evaluate the benefits and security concerns associated with the use of a routed system of sending packets.
- Send messages using a numeric addressing protocol with the Internet Simulator.

## Preparation

Review functionality of Internet Simulator with address fields as well as router / network tables

## Links

### For the Students

- **Routers and Redundancy** - Activity Guide ([PDF](#) | [DOCX](#))
- **Unit 1 on Code Studio**



# Teaching Guide

## Getting Started

### Thinking Prompt:

Imagine you were going to send a letter to a friend living in another state. List the steps you imagine your letter would have to take through the different parts of the postal system. Don't worry if you're not sure about your answers, just make an educated guess.

### Discuss:

Students can share their steps with a partner or group. Combine ideas from the class to create a single model of the steps involved, perhaps by writing / drawing it on the board. Draw attention to the number of times a person, post office, etc. must make decisions about where to send the letter next, i.e. "route" the letter.

- Example: Mailbox → Mail Carrier → Local Post Office → Regional Collection Center (of origin) → Regional Collection Center (of destination) → Local Post Office → Mail Carrier → Friend's House

### **Transitional Remarks**

When we send messages through a network we don't actually want everyone on the network to receive them. If we include information about who the message is intended for then we can allow portions of the network to focus on sorting and routing messages, so that they can continue on their way to their intended target. In the mail system, mail facilities, post offices, or a mail carrier fills this role.

In a network of computers, certain computers called "routers" do the same thing, directing messages towards the target computer based on the IP addresses included in the message.

## Activity

### Introduce: New Version of the Internet Simulator - Routers

### **Remarks**

Today's activity introduces the newest incarnation of the Internet Simulator

- Messages now have "To" and "From" addresses in a format similar to IP addresses (just an 8-bit address instead of a full 32 bits)
- This allows messages to be sent to a single intended recipient.
- The Internet Simulator also simulates the routing of messages across a network, with messages possibly being routed across multiple routers in unpredictable sequences before finally being delivered.
- This is done to simulate the way traffic travelling across the Internet is constantly rebalanced in response to over- or under-usage of some channels.
- A message will usually make it to its destination, but we can't know for sure how it will get there.



**Go to the Internet Simulator on Code Studio:**

Log into Code Studio and find today's lesson.

### Choose a Router:

Add a router if you need more space. Then join a router with a few of the people sitting closest to you. Ideally, you'll have 3-4 classmates with you on your router.

### Send a quick test message:

Send a simple "hello" to a friend who is connected to the same router.

- Find friend's (small) IP address
- Send a message to that address
- Friend should send a response

### Discussion Goal

Prepare students to think about ways to overcome shortcomings of broadcasting messages (sending them to everyone on a shared line) through the use of routing. The prompt draws parallels between routing in the mail service and routing of traffic on the Internet.

### Teaching Tips

The exact details matter less than identifying multiple points where the letter is sorted and rerouted based on its intended destination. No need to go into great detail here, but the hope is to help students develop a mental model for how a system for routing messages looks and what role a router serves.

## Activity: Investigate Routed Traffic

### Distribute Activity Guide: Routers and Redundancy - Activity Guide.

- Students do not need to be in groups but it may help, especially early in the activity, if they are physically close to students who are connected to the same router as they are.
- Lead students through the activity guide
- Each of the following sections appears in the activity guide
  - There are some notes for each of the sections, with suggestions for what you might do or say for each.
- Lead students through the sections as you see fit.

#### Find a Classmate on Your Router:

Ask students to join a router with some of their classmates, adding more if they need them. There should be at least three students on each router. Point out that messages now include a "To" and "From" IP address in their header. Ask students to use the network visualization to find the IP addresses of some of their classmates.

	To	From	Message
ASCII	13.2	13.8	Hey there!
Binary	1101 0010	1101 1000	01001000 01100101

96/Infinity bits

#### Have a Conversation:

Ask students to conduct a short conversation (e.g. a simple greeting or a question and answer) with two friends on their router. They should verbally confirm that both sides are receiving their messages. Provide students a few minutes to practice. Help students construct their messages. If their To address is not constructed properly the message will be dropped.

#### Read the Traffic:

Once students have conducted their conversations, ask them to pause and demonstrate the ability to view router traffic in the Router tab. Students should use the data generated here to answer the first set of questions on the Activity Guide. Have students click the "Log Browser" button to view the table of router data.

Stats	
Uptime	0:10:14
Queued Packets	0
Memory Use	0b / Unlimited
Throughput	0bps / Unlimited
Routed Packets	1 / 1 (100%)
Routed Data	8B / 8B (100%)

Logs

Log Browser

#### Activity Goal

1. have students become familiar with using an addressing protocol similar to IP
2. highlight features of routed traffic on the actual Internet in an interactive way.

Ideally, students will recognize that:

- some messages may be dropped
- messages may take different paths to the same location
- routers can theoretically read **all** traffic moving across them.

Ask questions and direct investigations that help students encounter some of these surprising things.

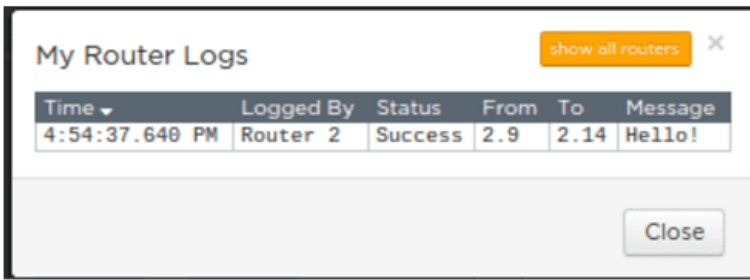
#### Teaching Tips

Big Picture: recognize that this activity is a repeated pattern of steps:

1. Find a friend's IP address.
2. Have a short conversation.
3. Analyze the traffic using the "router logs" to learn something about the structure of the Internet and answer questions.

- First time: this is done with classmates on the same router.
- Second time: students communicate with classmates on a different router.

The format of the lesson lends itself to being completed as a class, rather than individually. The activity instructions here aim to help you structure the activity so that the class can move forward as a unit and explore the functionality of the tool together.



### Transitional Remarks

- Routing messages functions very differently from broadcasting.
- The messages are only sent to the user they are intended for.
- This is just like when you write a letter to a friend, where an address on the outside of the letter tells the mail service how to properly route your letter.

#### The Router Logs

- The view we have in this table of traffic wouldn't be available to a normal user on the network -- this is a simulation after all -- but it WOULD be visible to the router.
- A router needs to read the header of the message to route it correctly \* But the contents of the message are right there too, and if it wanted, **a router could read all the messages** going across it, just like you did.
- This should raises some questions about privacy and security -- we'll talk more about this later.

#### Find a Classmate on a Different Router:

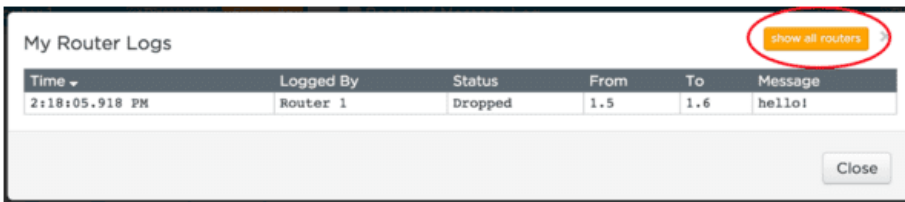
The Internet Simulator can route messages between routers, allowing the entire class to communicate. Ask students to find two classmates on a different router and ask for their IP addresses. (They'll need to actually talk to one another; they currently won't be able to do it on the Internet Simulator.)

#### Have a Conversation:

Again, students should conduct a short conversation with their two partners, confirming verbally that the messages are being received. Troubleshoot errors students may be encountering as before.

#### Reading Network Traffic:

Once students have been able to talk with classmates, direct them again towards the "Log Browser" button. It is possible to read traffic across the entire network, not merely your router by clicking the "Show all routers" button. Ask students to open the network traffic and examine the traffic in order to answer the questions listed there.



#### Explaining Redundancy:

The simulations students used in today's lesson model many features of the actual Internet, most notably its redundant nature (there are many paths between locations). Assist students in identifying these features with the following comments, and note that at the top of the next page you'll find a routing table that shows the different routes taken by the messages.

- |           |         |      |     |            |
|-----------|---------|------|-----|------------|
| Router 5  | Success | 6.12 | 5.5 | cool beans |
| Router 8  | Success | 6.12 | 5.5 | cool beans |
| Router 13 | Success | 6.12 | 5.5 | cool beans |
| Router 6  | Success | 6.12 | 5.5 | cool beans |

#### Teaching Tips

Some of the behavior for the full-class Internet can be difficult to fully comprehend the first time. It may be a good idea to join a router and send a message to another router yourself, and then as a class, trace the path it takes hopping across multiple routers. If you send multiple messages with different contents, then you can see examples of a message taking different paths to the same location.

### Remarks

On the Internet Simulator we see messages appearing in the Router Logs multiple times (like in the image to the right). This is done to simulate a message passing through multiple routers on its way to its destination. A row is created each time it shows up at a new router.

- **Prompt:** If you trace carefully you'll notice that messages between two people don't always visit the same routers along the way. This is not a mistake; it's modeled after the way the actual Internet was designed. Why might the Internet have been designed to be flexible

about how messages get from one person to another? Why go through the trouble of creating multiple paths between users?

- Efficiency: Communication lines can become backed up with traffic. The ability to choose whichever one currently has the least traffic improves the speed of transmission.
- Reliability: Having many redundant paths means if one goes down, there are many others that a message can take.

#### All Router Logs

Time	Logged By	Status	From	To	Message
10:43:48.364 AM	Router 6	Success	5.5	6.12	first time was 6 - 8 - 5, second time was 6 - 13 - 8 - 5
10:43:48.133 AM	Router 8	Success	5.5	6.12	first time was 6 - 8 - 5, second time was 6 - 13 - 8 - 5
10:43:47.913 AM	Router 5	Success	5.5	6.12	first time was 6 - 8 - 5, second time was 6 - 13 - 8 - 5
10:43:39.114 AM	Router 6	Success	13.2	6.9	Hello Lillian, This is Vicky from Router 13.2
10:43:38.846 AM	Router 8	Success	13.2	6.9	Hello Lillian, This is Vicky from Router 13.2
10:43:38.579 AM	Router 7	Success	13.2	6.9	Hello Lillian, This is Vicky from Router 13.2
10:43:38.275 AM	Router 13	Success	13.2	6.9	Hello Lillian, This is Vicky from Router 13.2
10:43:27.347 AM	Router 5	Success	6.12	5.5	cool beans
10:43:27.155 AM	Router 8	Success	6.12	5.5	cool beans
10:43:26.957 AM	Router 13	Success	6.12	5.5	cool beans
10:43:26.740 AM	Router 6	Success	6.12	5.5	cool beans
10:43:24.677 AM	Router 5	Success	6.12	5.5	k
10:43:24.486 AM	Router 12	Success	6.12	5.5	k
10:43:24.273 AM	Router 6	Success	6.12	5.5	k
10:43:21.013 AM	Router 13	Dropped	13.13	14.12	Hey did you get my message? vcasas
10:43:20.324 AM	Router 12	Success	2.1	14.5	he's ignoring me...
10:43:20.002 AM	Router 2	Success	2.1	14.5	he's ignoring me...
10:43:11.974 AM	Router 6	Success	6.10	6.5	Why thank you!
10:43:11.947 AM	Router 2	Success	14.5	2.1	do you have a shorter route to my 14 router
10:43:11.385 AM	Router 8	Success	14.5	2.1	do you have a shorter route to my 14 router
10:43:10.678 AM	Router 6	Success	14.5	2.1	do you have a shorter route to my 14 router
10:43:10.115 AM	Router 14	Success	14.5	2.1	do you have a shorter route to my 14 router

## Wrap-up

**Reflection:** Ask students to answer the reflection questions at the bottom of this document: **Routers and Redundancy - Activity Guide.**

**Discuss:** Allow students to share their answers, either in groups or with the class. The discussion should touch on:

- Benefits of redundancy in a routed system, to provide many paths between users of the network.
- Routed messages aren't seen by all users, but they can still be seen by owners of the router.
- If you control all the routers, you entirely control the way messages travel through the network, and additionally can see all the traffic.

## Assessment

**Questions** (also included on Code Studio):

1. In the Internet Simulator, your router knows to send your message to a different router because
  - a) You set the "other router" bit in the message.
  - b) Everyone is on the same router.
  - c) The first 4 bits of the address specify the router number that your friend is on.
  - d) Messages are routed over different paths.
2. Choose two: If the post office delivered mail exactly like routers deliver messages on the Internet, which 2 of the following statements would be true?
  - a) One mail carrier would be responsible for delivering a letter from sender to receiver.
  - b) The mail carrier would sometimes take different paths to deliver each letter to your home.
  - c) Letters would be written on the outside of envelopes for all to read instead of being put inside envelopes.
  - d) Your mail could not be delivered if a road your mail carrier was planning to take were under construction.
3. Choose two: Which of the following statements are true about routing on the Internet?
  - a) Information travelling between two computers over the Internet will always take the same path.

### Teaching Tips

Here's a cheat sheet for the features of the Internet students saw today and how the Internet Simulator is simulating an actual feature of the Internet:

- **To / From Address:** Like an IP address, included on every message sent over the Internet.
- **Dropped Messages:** Poorly formed messages cannot be delivered and so are dropped, just like a letter with a bad address on it. Tomorrow you'll discuss more technical reasons messages are dropped.
- **Multiple Hops:** A message travelling across the Internet will visit many routers as each tries to forward it along the most efficient path to its destination.
- **Different Paths:** Routers respond to traffic on the Internet in real time. The best path at one moment might be backed up a few seconds later. Routers choose the current best path to get the message through.

**Note:** In the Internet Simulator the "hops" are chosen randomly. This is done to create multiple paths. Actual routers use algorithms to determine the best path to send a message along.

- b) Your router receives all the traffic currently travelling across the Internet and delivers to your computer only the messages intended for you.
  - c) There is one router which receives and redirects all the traffic of the Internet.
  - d) A packet travelling between two computers on the Internet may be rerouted many times along the way.
  - e) A packet contains addressing information to allow routers to decide how best to forward along that packet towards its destination.
4. Describe the relative benefits of routing over a broadcast style of communication. Is routing traffic more secure than broadcasting? Justify your answer.
5. Can you know in advance the path a message will take between you and another computer on the Internet? Justify your answer.

## Extended Learning

- In order to connect to the Internet, your traffic must be travelling through a router. Either investigate your router yourself to learn more about its capabilities, or invite an IT person from the school to talk about the structure of the network within your school building.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CL** - Collaboration

### Computer Science Principles

- ▶ **3.3** - There are trade offs when representing information as digital data.
- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 11: Packets and Making a Reliable Internet

Internet Simulator | Group Problem Solving

## Overview

In this lesson student develop a protocol for reliably sending a message over an unreliable internet. The Internet Simulator has been setup for this lesson to restrict messages to no more than 8 characters each, and messages get dropped messages with some probability on every hop.

Students are given time to experiment with the Internet Simulator and develop their own protocol, possibly testing or demonstrating their protocol to their peers. At the conclusion of the lesson, students watch a short video explaining how these challenges are addressed in the real world with TCP - the Transmission Control Protocol.

## Purpose

This is a problem-solving lesson. On the real Internet packet sizes are limited, and transmission is unreliable. In this lesson we have setup the Internet simulator to restrict packet-size to be very small - 16-bits for the **to** and **from** addresses plus **only 8 ASCII characters**. Also, the **Simulator drops packets** pretty regularly. In a set of 10 messages it's very likely one or two will be dropped. You may need to justify these restrictions to students, acknowledging that, yes, this is a simulation, but these types of constraints are a very real problem on the Internet.

The problem students have to solve is how to use the 8-ASCII-characters-worth of data to include both a piece of message you're trying to send, as well as information about how many messages (packets) there are in the whole message, and which number **this** packet is.

Students should invent or at least appreciate the need for something like the Transmission Control Protocol TCP. TCP was designed to overcome the inherent unreliability of the Internet. A small but non-negligible percentage of packets are lost in transmission because of faults in the infrastructure of the Internet. In order to constrain the extent of these errors, larger messages are divided into many packets which are individually routed to their recipient. The receiving computer will send an acknowledgement confirming the receipt of that packet. If the sending computer does not receive an acknowledgement, it will resend the packet until all packets have been acknowledged.

Since packets may arrive out of order, additional data must be included to indicate the order in which the packets should be arranged. Thus, while individual packets cannot be guaranteed to arrive, eventually an entire message can be accurately reconstructed.

## Agenda

**Getting Started (5-20 mins)**

**Optional Unplugged Activity (15 mins)**

**Activity (35 mins)**

**Introduce: new version of Internet Simulator - packets and unreliability.**

**Generate Traffic**

**Develop a Protocol**

**Test Protocols**

**Wrap-up (10 mins)**

## Objectives

**Students will be able to:**

- Explain why protocols are necessary to overcome the underlying unreliability of the Internet.
- Justify the need for acknowledgements and packet numbering in TCP.
- Develop a protocol for reliable communication on the Internet.

## Preparation

Review functionality of Internet Simulator with transmission failure

Copies of the Activity Guide

## Links

**For the Teacher**

- **U2L05 - Teaching Tips & Tricks Video** - Video ([download](#))
- **Packets Unplugged Activity** - Teacher Guide ([PDF](#) | [DOCX](#))

**For the Students**

- **The Internet: Packets, Routing, and Reliability** - Video ([download](#))
- **Packets and Making a Reliable Internet** - Activity Guide ([PDF](#) | [DOCX](#))
- **Unit 1 on Code Studio**

## Vocabulary

- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information.
- **TCP** - Transmission Control Protocol - provides reliable, ordered, and error-checked delivery of a stream of packets on the internet. TCP is tightly linked with IP and usually seen as TCP/IP in writing.

**Video: The Internet: Packets, Routing, and Reliability**  
**Connect activity to TCP**

**Assessment**

**Extended Learning**



# Teaching Guide

## Getting Started (5-20 mins)

### Prompt:

Your friend sent you a message on the Internet, but you never received it. Based on what you already know about routers and the physical Internet, list what reasons might explain this fact.

### Discuss:

In groups or as a class, students should share their lists. Use this opportunity to review the layers of the Internet already covered and potential flaws that could lead to a message getting dropped. Some example points of discussion:

- Wires are cut
- Interference on a radio channel
- Router malfunctions or cannot keep up with traffic being directed to it

### Discussion Goal

- Build on prior knowledge of the structure of the Internet to reflect on the unreliability of the Internet
- Introduce the use of packets
- Motivate the need for protocols to ensure reliability when sending messages across the Internet.

## Optional Unplugged Activity (15 mins)

### Teacher Guide for unplugged activity: Packets Unplugged Activity - Teacher Guide

**Synopsis** The activity simulates issues of packets traveling on the internet with a paper-passing activity:

- Students pass pieces of paper (post its?) with writing on them to another student in the class
- Students cannot move from their seat - they only pass to someone within reach
- Teacher might provide some "interference" by grabbing post-its in transit and delaying or destroying them.

### Teaching Tip

This is an unplugged activity that was invented by teachers over a few different teacher professional development workshops for this curriculum. We've included it here because it was effective at both highlighting some of the issues around packets as well as very nicely setting up the problem students need to solve with the internet simulator.

## Transitional Remarks

When we communicate on the Internet, we are not just sending short text messages as we did yesterday. We also use the Internet to exchange documents, videos, music, and scientific data, and these files can easily grow to enormous size.

All of this would not be a problem if the Internet were perfectly reliable, but in reality, errors sometimes occur. Wires can be cut, routers can be overwhelmed with traffic, and interference with electric or radio signals can cause messages to become corrupted. The response to this problem is to split large messages into smaller pieces of information called packets.

It turns out that splitting up a message into packets provides many benefits. If a faster route opens up halfway through transmitting a large file, it is easy to reroute later packets in the transmission through that route.

Splitting up a message into smaller chunks doesn't solve all the problems of unreliability on the Internet. Packets can still be dropped or arrive out of order.

Today's challenge is to develop a protocol to reliably send messages even though the network itself is unreliable.

## Activity (35 mins)

### Introduce: new version of Internet Simulator - packets and unreliability.

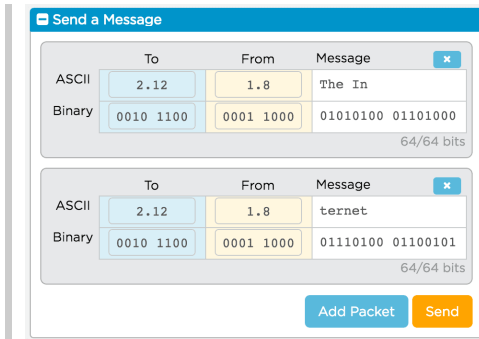
## Remarks

The version of the Internet Simulator we will be using today has been structured to simulate the unreliability of the Internet.

In particular you'll notice a few changes:

- You will **only** be allowed to send packets containing **8 characters** of text!
- Anything larger than 8 characters will be cut off...
- However, you may construct multiple packets prior to sending them, by clicking "Add Packet", and then send them all with one click of the "Send" button.
- Every message has a small chance of being dropped on each "hop" it makes between routers.





### Teaching Tip

You might want to demonstrate these new features of the Internet Simulator while explaining them.

If you have the ability to project your screen you can demonstrate before having students connect by:

- Opening 2 windows and connect to the simulator as yourself - you will see your name appear twice
- Send messages to yourself (using one of the IP addresses)
- Open the router logs to show packets dropped.

### Open Internet Simulator:

Distribute the Activity Guide: **Packets and Making a Reliable Internet - Activity Guide**

- Place students in groups of 2 to 4 members.
- Students (individually or as a group, either is OK) should log into Code Studio and access the Internet Simulator.
- Add at least four routers (six or more is ideal; there's no need to max out the connections on each router - it's fine if only one or two students are connected to a router)
- Have students join a **DIFFERENT** router from their groupmates.

### Generate Traffic

- Students will need some time to get used to the changes introduced to the Internet Simulator.
- Provide them a couple minutes to exchange messages across routers.
- Make sure students try to construct a multi-packet message and send multiple packets at once to someone else
- Make sure they try the same thing for a classmate on ANOTHER router and then view the router logs to examine the result of these transmissions.

This is a clever way to send a drawing with ASCII text and students can come up with ways to break it up. The wider the drawing is, the more challenging the protocol will be to develop since each individual message is limited to only 8 characters.

Students should recognize:

- Packets are dropped with some frequency
- Packets of more than 8 characters are always truncated to just the first 8 characters
- Packets sometimes arrive out of order.

If students do not immediately recognize these points, draw them out in subsequent conversations.

### Develop a Protocol

- Students will spend the rest of the class developing a protocol that will allow them to overcome the unreliability of the network so that a message can be sent and both sender and receiver can be confident the full message was received.

#### Guidelines for Protocol:

- All communication can only be done through the Internet Simulator.
- The full message sent will be at least 80 characters long - broken into at least 10 packets - and might be entirely random (i.e. there's no way to use human intuition to reconstruct the message).
  - The message is not known beforehand.
  - **Note:** The Teaching Tips & Tricks video mentions a 20 character long message. 80 characters (10 packets) is the preferred length, as it results in more packets being dropped.
- The sender and receiver must be confident the full message was successfully transmitted and reconstructed.

#### The real Problem to Solve

### Teaching Tip

#### What to Look For

In student solutions, you should primarily be looking for:

1. Some way to confirm receipt of a packet / message
2. Some way to define the ordering of packets

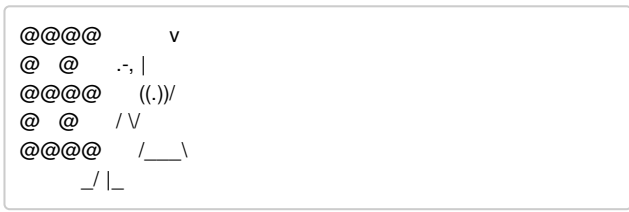
This activity is designed to provide students a realistic encounter with the actual challenges of the Internet. It is OK if their solution does not perfectly match TCP, the actual solution to this problem.

#### The Focus

Should be on the protocol for the recipient to acknowledge or re-request missing packets. That is the real challenging part, and other parts of the protocol emanate from it. See: **"The REAL problem to solve"** below.

#### Suggested message format

To come up with a non-english message you could have students send a message as "ASCII" art. For example, you can treat the message as a grid and draw something with characters. Here are two little examples: a letter "B" and a little character holding a torch.



The real problem to solve is to think about what the recipient of the messages should do to inform the sender of what's missing and needs to be re-sent.

It's tricky because any message the recipient sends back to the sender also stands a chance of being dropped or lost. Your protocol needs to overcome this unreliability for both the sender or receiver.

Provide students time to work on their protocol, iteratively testing their work and recording their final protocol in the space provided in the activity guide.

### Test Protocols

- Provide students an opportunity to either present their protocols in written or visual form, or test them out by sending a randomly generated message of your choosing.
- The goal of this share-out would be to identify the main components of the protocols that groups developed in common.

## Wrap-up (10 mins)

### Video: The Internet: Packets, Routing, and Reliability

We're going to watch a short video that talks a little more about the way this protocol and others help us ensure the reliability of the Internet.

**Video: The Internet: Packets, Routing, and Reliability - Video**

### Connect activity to TCP

Either say these points or draw them out through **discussion**

- The challenges we encountered in today's activity very closely mirror those that exist on the actual Internet.
- The response was the development of a protocol called the Transmission Control Protocol, or more simply, TCP.
- TCP divides larger messages into smaller packets which have ordering information added to their header.
- When a packet arrives at a destination computer, an acknowledgement is sent to the sender, letting them know they don't need to resend that packet.
- Once all the packets have arrived, the ordering information in the headers of the packets allows them to be reordered to create the original message.

## Assessment

**Questions** (also available on Code Studio):

1. Choose two: Pick the two statements about packets and routing on the Internet which are true.
  - a) Packets travelling across the Internet take a standardized amount of time and so can be counted on to arrive in the order they were sent.
  - b) TCP ensures messages can be reliably transmitted across the Internet.
  - c) TCP depends on the infrastructure of the Internet to be reliable enough to ensure no packets are lost in transmission.
  - d) TCP must account for the fact that packets may not arrive at a destination computer in the intended order.
2. Which of the following is NOT true about packets?
  - a) Packets are numbered, so if they arrive out of order, the message can be reassembled.
  - b) A message sent across the Internet can always be contained in a single packet.
  - c) Packets are routed on different paths from sender to receiver.
  - d) The receiver computer must confirm to the sending computer that each packet was received.

### Content Corner

#### A Simple Solution

**Sender:** The simplest solution is to develop something akin to TCP, in which you use part of the message to indicate the the packet number and how many there are total. For example:

- Message 1: 14 Hi
- Message 2: 24 Th
- Message 3: 34 er
- Meesage 4: 44 e!

**Receiver:** should know how many messages there are and can request a re-send of any that are missing. For example if you received messages 1, 2 and 4. You could send a message back, something like one of these:

req: 3 or need 3 or send 3

The way the real TCP works is this: every once in a while TCP sends back a "cumulative acknowledgement" of how many in-order packets it has received. For example, in an 8-packet message if you received packets:

1 2 \_ 4 5 6 \_ 8

Then you would send back a message like:

ack 2 or maybe ack 3

This would indicate to the sender that next packet you need is packet 3. The sender would send 3, and once you got it, your next message would be:

ack 6 or ack 7

Because you received packets 4, 5 and 6 already.

3. In your own words, what were the primary obstacles you needed to overcome in today's challenge? How do these obstacles mirror actual problems encountered on the Internet?

## Extended Learning

- **There and Back Again: A Packet's Journey** and discuss questions that arise about how a packet travels from one person to another. (3:21)
- **Easy Way to Understand Packets.** Note: Headphones are recommended; this one has some dramatic music! (13:00)
- Protocols are developed by the Internet Engineering Task Force (among other groups), who publish Requests for Comment (or RFCs) detailing the components of the protocol. If students wish, they may have a look at the one of the actual RFCs for TCP here <https://www.rfc-editor.org/rfc/rfc793.txt>. RFCs exist for many of the protocols covered in this course.
- **Blown to Bits (www.bitsbook.com), Appendix** - The Internet as System and Spirit, pp. 301-309, "The Internet as a Communication System," then answer the following question:
  - It is interesting to think about the foresight (or luck?) that caused initial Internet designers to choose a packet-switched network as the basis for the Internet. What do you think, foresight or luck?
- **Blown to Bits (www.bitsbook.com), Chapter 3** - Ghosts in the Machine, pp. 91-92, "Data Formats as Public Property," then answer the following question:
  - What do you think the Internet would be like today if different companies' network protocols were still competing for adoption for the Internet, instead of TCP being the de facto standard?

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CL** - Collaboration

### Computer Science Principles

- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 12: The Need for DNS

Unplugged | Group Problem Solving | Internet Simulator | Research

## Overview

The core idea of this lesson occurs in the unplugged activity that kicks off the lesson, in which students try to keep track of IP addresses that had been randomly assigned to each student in the class, while at the same time the teacher occasionally changes students' addresses. This leads to identifying the need for an authoritative system for name-to-address mappings, known as the Domain Name System or DNS.

Students then briefly experiment with a DNS protocol in the Internet Simulator. The activity is similar, in that students will have to grapple with IP addresses changing in real time and use the built in DNS protocol to resolve the issues.

The lesson ends with students doing some rapid research about DNS and some of its vulnerabilities, particularly what are known as Denial of Service Attacks.

## Purpose

The basic purpose of this lesson is to show students in kinesthetic and interactive ways some of the challenges solved, and created by, DNS. At its core, the DNS is "simply" a hierarchical system of computers and databases, that maps IP addresses to domain names. It enables Internet users to connect human-language locations on the Internet with numeric addresses used by IP. While distributed and hierarchical, it can be treated in the abstract as a centralized registry of locations on the Internet, allowing users to quickly find locations they are looking for and register themselves so that others may find them.

## Agenda

### Getting Started (15 mins)

**DNS Unplugged  
Discussion**

### Activity

**DNS in the Internet Simulator  
Video: DNS**

### Activity 2

**Rapid Research: DNS and DDoS attacks**

### Wrap-up

**What is DNS?**

### Assessment

### Extended Learning

## Objectives

### Students will be able to:

- Give a high level description of DNS as a name-to-IP-address mapping system used on the Internet
- Give a few reasons why DNS is useful and necessary
- Describe at least one vulnerability of DNS and how an attack on it works

## Preparation

- MUST print out and have prepared IP Address Labels for students walking into class
- Copies of Worksheet and Activity Guide
- Familiarize yourself with the Getting Started activity and what you need to do

## Links

### For the Teacher

- **U2L09 - Teaching Tips & Tricks Video** - Video ([download](#))
- **IP Address Labels** - Teacher Resource ([PDF](#) | [DOCX](#))

### For the Students

- **The Internet: IP Addresses and DNS** - Video ([download](#))
- **Names and Addresses** - Worksheet ([PDF](#) | [DOCX](#))
- **DNS Partner Questionnaire** - Activity Guide ([PDF](#) | [DOCX](#))
- **Research: DNS in the Real World** - Activity Guide ([PDF](#) | [DOCX](#))
- **Unit 1 on Code Studio**

## Vocabulary

- **DNS** - The service that translates URLs to IP addresses.

# Teaching Guide

## Getting Started (15 mins)

### As students walk into class...

- Hand each student a single IP address (precut by teacher from **IP Address Labels - Teacher Resource**). Students should be instructed to hold onto their IP addresses, not share them, and await instructions.
- Make copies of **Names and Addresses - Worksheet** available -- each student will need one.

### 💡 Teaching Tip

This Getting Started activity is a bit more involved than usual. It will probably take 10-15 minutes to complete and requires a lot of physical movement. The "Activity" portion of the lesson may be a little shorter than usual.

## DNS Unplugged

### Instructions:

- When you walked in, I handed you a slip of paper with an IP address on it.
- You also should have a **Names and Addresses - Worksheet** worksheet.
- For the next 5 minutes, your goal is to complete an accurate list of IP addresses and names for all students in the room. You may only talk to one person at a time, but you may exchange as much information with that person as you want.
- **GO!**

**Teacher Participation:** As students are working, circulate quietly through the room.

- Approach a student and silently take their IP address slip away from them.
- Give that person a new IP address slip (or a re-used IP address).
- Repeat the above two steps as many times as you can, as you circulate the room.

## Discussion

- **Prompt:** Why did I keep taking your IP addresses?
  - This simulates the fact that a computer's IP address does not stay the same. For example, a person's IP address on their phone changes quite frequently as they move around throughout their day and their phone tries to connect to the Internet from different locations.
- **Prompt:** Do you think the system we just simulated is an efficient way of collecting IP addresses? Are there any inefficiencies you observe? How could it be made better?
  - A central list would be better, and the Internet has a system for that.

### 🎓 Content Corner

#### Background Info for the Activity

When computers talk to one another on the web, they don't refer to each other by names; they use an IP address to indicate who they are and who they are sending a message to. Without this system, the packets would be impossible to route across the Internet. The problem with this system is that, while computers are fine with referring to other computers by numbers, humans are really terrible at remembering long strings of seemingly-random numbers. We'd prefer to identify a web page by its name, and besides, it makes remembering our favorite locations on the Internet ("Code.org" vs. 174.129.14.120) much easier!

### 💡 Teaching Tip

Students may object that you're making the task impossible. Part of the point of the exercise is that people's IP addresses change all the time, and it's difficult to keep track if no one, or no system, can be appealed to as an authority for name-to-address mappings.

You don't need to linger with this activity, once you think students see what's happening. Do enough to make the point and move onto the discussion.

### 💬 Discussion Goal

Students should understand that this is a realistic scenario -- IP addresses change all the time. This system of everybody-keep-track-of-your-own-list is highly inefficient. Every student is developing an identical list to one another. A central list would be better, and the Internet has a system for that.

## Activity

### 🖥️ Transition to the Internet Simulator (see Code

#### Studio):

- Direct students to the Internet Simulator and distribute **DNS Partner Questionnaire - Activity Guide** or direct students to the online document.
- Explain the new configuration of the simulator that includes a DNS server.

- o A DNS server now appears attached to every router.
- o We no longer can see anybody's IP address. To get an IP address, we have "ask" the DNS server using a text-based protocol.
- o Demonstrates how to send a request to the DNS for someone's address. Let students try the DNS protocol to get the address of someone who is attached to their router.
- After a minute or two, get students' attention to explain what will happen in the following activity.

### Teaching Tip

This is the first time with the Internet Simulator where students are communicating with a machine using a protocol, rather than with other human beings.

While students were able to fudge their protocols in the past, they will no longer be able to do so; it is essential that they use the exact protocol presented.

## DNS in the Internet Simulator

(Explanation also in activity guide)

When you go to the Internet Simulator now, you will see a "DNS server" attached to the router. In order to communicate with someone else, you must first find their IP address by asking the DNS.

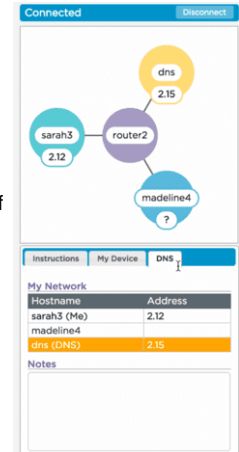
1. To begin, click over to the "DNS" tab to see all the hostnames of people on the router. You will see the address of the DNS (always 15) but will not see an address for anyone else on the router.
2. The DNS server responds to a text protocol that will give you someone's IP address. The protocol is:

```
GET <hostnameOfPerson>
```

For example:

```
GET madeline4
```

1. After the DNS has returned an IP address, you can type that IP address into the "To" field, enter a message, and then press "send."



### Directions:

(from the activity guide)

- You are going to interview/have a conversation with a classmate using only the Internet Simulator. We've created a list of interview questions (on the next page) and you should both jot down each other's responses.
- To find the person, you will have to ask the DNS for her IP address. When you have retrieved the IP address, start the interview.
- **HOWEVER...**As you're working, if your teacher taps you and your partner, you **both MUST disconnect and reconnect from the simulation**. This is to simulate changing IP addresses.
- Even though your IP address will change, your hostname will stay the same, so you'll need to re-join a router and ask the DNS for your partner's new IP address in order to continue having your conversation!

### Teacher Participation:

As in the previous activity, you will "interfere" with the students' activity by walking around the room and directing **BOTH members of a pair to disconnect and reconnect**.

When an IP address changes, there is no visual cue to anyone else on the Internet that the person they are talking to has moved.

Instead, they should send a request to the DNS to make sure that the IP address is correct every time they want to send a person a message.

### Content Corner

Especially in the world of mobile devices it is very common for a device to lose an IP address and need to re-acquire one - entering a tunnel, flying on a plane, even moving from one WiFi hotspot to the next.

It is true that at this point in time it's less likely that a web site with a registered domain name (like Code.org, google.com, or facebook.com) would rapidly change IP addresses, but the great thing about DNS is that even if they did, that change would be transparent to the public and you won't have to worry about it. Even so huge operations like Facebook, Twitter, etc. actually serve their sites on many hundreds (or thousands) of computers, all with different IP addresses - DNS helps manage that too, so you can just type Microsoft.com and it direct you to the right place.

As an analogy, think about a time when a friend or family member changed their phone number. That affects everyone who needs to know that number and causes some annoying problems. If we had a system like DNS for phone numbers, you'd never have to concern yourself with anything but remembering the person's name. Of course, every person on earth would have to have a unique name for this to work, so it's a bit impractical.

## Video: DNS

## Remarks

You may remember (from the IP/DNS video that we saw several lessons ago) that you learned about the Internet system (v DNS) for sharing names and IP addresses. Let's watch that section again!

- Show the DNS portion of this video as a **transition** to the next activity.
- **Video: IP and DNS - start at 4:12**



## Activity 2

### Remarks

- Hopefully we all get the basic idea: the DNS is the large-scale system that translates human-readable web addresses into their numeric IP addresses so that computers can communicate.
- This system however was **not designed to be secure** and that has resulted in some major security incidents over time.
- You're now going to learn about some of them and how they work.

## Rapid Research: DNS and DDoS attacks

### Introduction:

Place students in groups of 4-6 people to complete their readings.

- Distribute: **Research: DNS in the Real World - Activity Guide**, one copy per student.
- Assign each group an article about DNS and DDoS attacks
- The list of articles can be found on the **first student page (bubble 1) on Code Studio for this lesson.**
- Students should complete the first page of the worksheet

**Give students 15-20 minutes** to read their article and make sense of the content.

- Groups should work together to complete their activity guide.
- Encourage groups to work together and make sure that at the end of the reading time, all students in the group feel comfortable with the technical content in the article.

### Jigsaw:

- Do a count-off to form new groups in which each member read a different article.
- Students should exchange information they learned with one another, recording the key points from each article in the space provided on the second page of the activity guide.

### Teaching Tips

Some articles are shorter than others and may not take the full 15 minutes to read. Consider giving these to students with lower reading levels or pairing them together so that a group reads two articles.

## Wrap-up

### What is DNS?

Lead a discussion reviewing the IP and DNS systems.

#### Prompts: What is DNS?

- Why does the Internet use IP addresses?
- Why don't we need to know IP addresses?
- Why do we need a Domain Name System?
- Why don't we all maintain our own DNS?
- Is there one big DNS for the entire Internet?
- How do you think all these DNS servers are maintained?

#### Prompts: Security

- What is one vulnerability of DNS and how is that vulnerability attacked?
- What are the implications of an attack on a DNS server (or servers) - how does this affect your life?

## Assessment

### Questions:



1. A single central register of IP addresses and names (i.e. a DNS-style system) is an efficient means of translating human-readable names to IP addresses. Which of the following is NOT solved by DNS?
  - a) It's inefficient to have everyone on the Internet maintain their own list of IP addresses.
  - b) There are too few IP addresses to meet the current demand.
  - c) When someone new joins the Internet, they need to inform everyone of their IP address.
  - d) When an IP address changes, it is impossible to locate a computer until the owner announces the change.
2. Why do computers need to periodically check the DNS for websites you have already visited?
3. Why don't we need to know the IP addresses for our favorite sites?

## Extended Learning

- **What is DNS?** A deeper look into the DNS and what one sees when communicating with the DNS from the command prompt or terminal.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CL** - Collaboration

### Computer Science Principles

- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.

### Discussion Goal

Some answers to the discussion questions:

- The Internet is basically a network of computers sending messages to request information and computers replying to messages to satisfy information requests. Computers need to identify "from" and "to" for all messages. Computer speak in numbers, not names.
- All communication online is via IP addresses. However, we are more familiar with human readable names, most notably URLs like "Code.org" or "Google.com."
- We need a way to translate human-readable names into IP addresses.
- It is inefficient for everyone on the Internet to maintain a table of IP addresses.
- The DNS is NOT centralized, but it's not completely autonomous and distributed like routing, either. There is a hierarchical system of servers to maintain an authoritative table that, like a phone book, others can consult when they need to find an address.
- A properly functioning DNS system requires collaborative efforts among all users to ensure it is up to date and accurate.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.



# Lesson 13: HTTP and Abstraction on the Internet

External Tools | Exploration

## Overview

In this lesson students are introduced to another high-level protocols of the Internet, [HTTP](#). The lesson begins with a review of the layers of the Internet covered thus far, before transitioning to a video covering high-level protocols of the Internet, most notably HTTP. Students will investigate HTTP traffic generated within their own browser by accessing the browser's developer tools and visiting a variety of websites. A handout summarizing the structure of HTTP is provided to help students understand the components of the HTTP requests and responses they will observe. The lesson concludes with students sharing their findings with their classmates and a reflection on how the layers of the Internet make use of abstraction.

## Purpose

The main purpose of this lesson is round out understanding of one of the main Enduring Understandings and associated Learning Objective of the CSP Framework:

- 6.1 The Internet is a network of autonomous systems.
  - 6.1.1 Explain the abstractions in the Internet and how the Internet functions. [P3]

There are two main purposes to this lesson:

First, simply getting a basic understanding of what HTTP is and what it's for. HTTP like DNS is an ASCII-text based protocol - it's just two computers sending text messages to each other. What makes it a protocol are the rules of the "conversation" the two machines are having. In the case of HTTP, it is a call-and-response protocol for a client/server relationship, where a client requests a web page or other content (image, sound, video, etc.) from a server. The server looks for it and sends it back.

Second and perhaps more importantly, is understanding HTTP as a "high level" protocol that sits on top of **all** the other protocols and internet systems we've learned about in the course. That text message conversation between the computers is being broken up into TCP/IP packets, and all the data gets sent as bits over wires and airwaves, taking different paths, and it gets interpreted reassembled at the end.

We often talk about how the Internet works in "layers" and this is a perfect example of abstraction on the Internet, as one layer makes use of the functionality provided by the layer below it, without worrying about the details of how this functionality is achieved. And HTTP doesn't have to worry about anything other than the text protocol of HTTP works. The network software and devices on your and others' computers handle looking up addresses, breaking down data, packeting, routing, transmission and interpretation and reassembly. It's really amazing.

What we want students to understand is some degree of the "Big Picture". We want them to be able to recognize that these layers exist and to reasonably explain how they work together for common web page requests.


## Agenda

## Objectives

Students will be able to:

- Explain how layers of protocols allow the Internet to function.
- Use developer tools in a modern browser to explore the HTTP traffic associated with visiting common websites.
- Identify abstractions used in the development of Internet protocols.
- Describe how a protocol or layer of the internet acts as an "abstraction" for other layers.

## Preparation

 Practice accessing the Developer Tools for the browser used in your classroom (see [Worksheet](#) for links to directions)

## Links

For the Students

- [HTTP and Abstraction on the Internet](#) - Resource (PDF | DOCX)
- [HTTP in Action](#) - Worksheet (PDF | DOCX)
- [The Internet: HTTP and HTML](#) - Video ([download](#))
- [Unit 1 on Code Studio](#)

## Vocabulary

- **DNS** - The service that translates URLs to IP addresses.
- **HTTP** - HyperText Transfer Protocol - the protocol used for transmitting web pages over the Internet
- **IP Address** - A number assigned to any item that is connected to the Internet.
- **TCP** - Transmission Control Protocol - provides reliable, ordered, and error-checked delivery of a stream of packets on the internet. TCP is tightly linked with IP and usually seen as TCP/IP in writing.
- **URL** - An easy-to-remember address for calling a web page (like [www.code.org](#)).

**Getting Started (15 mins)**

**The Internet Protocol Stack**

**Activity (25 mins)**

**Show Video: "The Internet: HTML and HTTP."  
Investigate HTTP traffic on your computer**

**Wrap-up (10 mins)**

**Assessment**

**Extended Learning**

# Teaching Guide

## Getting Started (15 mins)

### The Internet Protocol Stack

Display the image shown at right, and/or distribute the **HTTP and Abstraction on the Internet - Resource** handout and have students follow along with the explanation for the first 2 pages.

#### Remarks

We're rounding out our understanding of how the many protocols and systems of the Internet work together to make the Internet function.

- We want to think of the protocols as working in "layers".
- This picture here (also on first 2 pages of **HTTP and Abstraction on the Internet - Resource**) is a simplified version of what's known as the "Internet Protocol Stack." We've studied each layer separately but now you can begin to see how they work together.

We imagine outgoing information going DOWN the stack and incoming information going UP the stack.

All of this happens in the network software on each computer, whether that computer is your phone, a laptop, or a server like a DNS or web server (routers are a special case that only look at the IP layer).

The **important takeaway** here is to understand that the system was constructed with **layers of abstraction** where each layer only needs to concern itself with its specific job, and then hands it off to another layer. This makes the very complicated task of digital network communication possible.

At the top layer, for example, DNS just thinks is "speaking DNS" to some other computer - **the DNS protocol does not even need to know how the other layers work** It just relies on them doing their jobs.

Answer any questions and move on to the activity.

**Transitional Prompt: "Have you ever seen the letters "HTTP" anywhere while using the internet?"**

- You see it in the URL of most websites **http://code.org** for example.

Today we will focus our attention on HTTP, which is a protocol that sits at the same "layer" as DNS - right above TCP.

## Activity (25 mins)

#### Remarks

- HTTP is an ASCII text based protocol.
- It's somewhat remarkable to note that many "high level" protocols, like HTTP, are just computers sending ASCII text messages back and forth. Each protocol simply defines the rules of the "conversation" between two machines.
- In the case of HTTP it is the protocol used for sending and receiving web pages and other web content.
- Today we'll look under the hood and see HTTP in action.

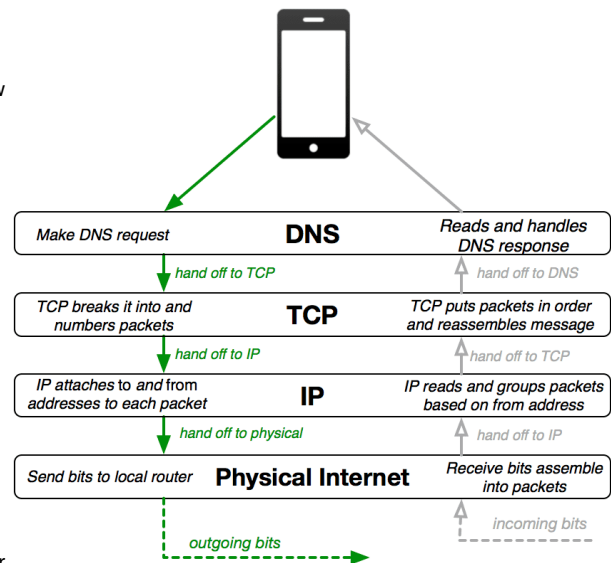
### Show Video: "The Internet: HTML and HTTP."

Either as a class or individually, students watch The Internet: HTML and HTTP Overview.

Tell students to pay particular attention to the portions about HTTP.

Video Link: **The Internet: HTTP and HTML - Video**

### Investigate HTTP traffic on your computer




#### Teaching Tip

It's not important to get too deep into the specifics. If students understand the basic idea of **layers of abstraction** -- that each layer relies on those below it, but uses them "abstractly" that's enough.

Put differently, higher-level layers depend on the functionality of lower-level layers, without worrying about precisely how those layers work. The **Enduring Understanding** from the CSP framework states: **"6.1 The Internet is a network of autonomous systems"** And these layers of abstraction (protocols) is a large part of what that statement is about.

#### Discussion Goal

This is largely a rhetorical question. Use a gateway to explaining that today we'll learn a little bit more about how HTTP works and look under the hood to see it in action.

💡  We can actually watch HTTP traffic on our computers through tools in the web browser!

### Activity:

Students should be placed in pairs and given **HTTP in Action - Worksheet**

- (optionally distribute **HTTP and Abstraction on the Internet - Resource** if you haven't yet).

Students will complete the worksheet while using the resource as a reference.

During the activity, students will:

- 💡
- Access the developer tools of their browser.
  - Monitor the HTTP traffic generated by visiting a variety of websites.
  - Record their findings, using the resource as a guide.

### 💡 Teaching Tip

This video includes some content relating to security which will not be covered until later in the course. Rather than focus on every protocol covered, keep the focus on the fact that all of the protocols discussed are text-based and rely on the lower-level protocols already covered in class in order to function.

### 💡 Teaching Tip

At the beginning of the activity, students will need to gain access to the Developer Tools of their web browser, and more specifically gain access to the "Network" tab of the tools. It is recommended you practice prior to the beginning of the lesson so that you can demonstrate to the class and provide assistance.

If students haven't previously encountered the terms "static" and "dynamic," as used to describe web pages, those concepts should be introduced before they begin the activity. This article gives a good explanation: **Wikipedia: Static web page**.

## Wrap-up (10 mins)

**Discuss:** If there is time, ask students to share their findings using a group-share-strategy. Otherwise use a closing discussion to once more discuss the way abstraction is utilized by the different layers of the Internet, relating the discussion to the material discussed today. Some possible prompts:

- What surprised you about the HTTP traffic you observed?
- What does it mean to say that high-level layers of the Internet use low-level layers "abstractly"?
- What other examples of abstraction have we seen in this course? Hint: Unit 1 is basically all about abstraction.

## Assessment

**Questions** (also on Code Studio):

1. Multiple Choice: HTTP is considered to be a high-level protocol because:
  - a) HTTP requests are given higher priority for fast delivery when being routed on the Internet.
  - b) HTTP requests are encoded using higher-frequency radio waves.
  - c) HTTP messages can be either requests or responses.
  - d) HTTP requests make use of abstractions provided by lower-level protocols.
2. Choose Two: Choose the two true statements about HTTP:
  - a) An HTTP request is sent from a client to request access to data stored on a server.
  - b) HTTP requests and responses have identical formats.
  - c) Displaying a web page will often require multiple HTTP requests in order to acquire all the necessary data.
  - d) An HTTP response code is only used when a server could not fulfill a request.
3. Free Response: The definition of HTTP makes use of the ASCII character set, without reference to how these characters are encoded. Explain why this is an example of abstraction.

**Worksheet:** If you like, you may collect and assess the Worksheet for completion.

## Extended Learning

- Research SMTP, a high-level protocol for formatting email messages.
- An excellent resource for explaining the many layers of protocols that make up the Internet: **Internet Techniques and Web Formats**.
- **Blown to Bits (www.bitsbook.com), Appendix**, The Internet as System and Spirit, pp. 309-316 (The Internet Spirit), then answer the following questions: The layers of protocols used in network communication is an example of abstraction. Can you give other examples of abstraction in everyday life? When you browse to a web page, maybe with some animated advertisements embedded on it, describe in detail what happens behind the scenes to display that page on your browser.

- **Blown to Bits (www.bitsbook.com), Chapter 3**, Ghosts in the Machine: Secrets and Surprises of Electronic Documents, pp. 73-77, then answer the following question about abstraction: Discuss how not knowing some basics of how a software tool or computer works, and the abstractions they use, could lead to bad outcomes.

## Standards Alignment

### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CPP** - Computing Practice & Programming
- ▶ **CT** - Computational Thinking

### Computer Science Principles

- ▶ **6.1** - The Internet is a network of autonomous systems.
- ▶ **6.2** - Characteristics of the Internet influence the systems built on it.



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 14: Practice PT - The Internet and Society

Practice PT | Research | Presentation

## Overview

This lesson is a capstone to the Internet unit. Students will research and prepare a flash talk about an issue facing society: either **Net Neutrality** or **Internet Censorship**. Developing an informed opinion about these issues hinges on an understanding of how the Internet functions as a system. Students will prepare and deliver a flash talk that should combine forming an opinion about the issue and an exhibition of their knowledge of the internet.

This lesson is good **practice** for certain elements of the AP Explore Performance Task.<sup>1</sup> The primary things practiced here are: doing a bit of research about impacts of computing (though here it's specifically about the Internet), explaining some technical details related to ideas in computer science, and connecting these ideas to global and social impacts. Students will practice synthesizing information, and presenting their learning in a flash talk.

<sup>1</sup>**Note:** This is NOT the official AP® Performance Task that will be submitted as part of the Advanced Placement exam; it is a practice activity intended to prepare students for some portions of their individual performance at a later time.

## Purpose

This lesson has dual purposes of honing "rapid research" skills and tying a temporary bow on the Internet Unit.

The act of doing "rapid research" is one that will come up over and over again in this course. We want to build students' confidence and skills in researching topic using a variety of sources. In the case of this lesson we want students to read articles on the issues but scan for the terms and vocabulary they know like: IP, DNS, HTTP, routing, packets, scaling, redundancy, reliability. We want students to be able to explain with some level of technical proficiency **how** these things work as well as the potential beneficial and harmful effects.

Net Neutrality and Internet Censorship are related issues having to do with organizations attempting to control internet traffic for a variety of reasons. There are many other large societal issues and dilemmas related to the Internet besides these that like: big data, surveillance, security and encryption. We address these issues in Unit 4: Big Data and Privacy. For this practice PT we want to keep the focus on issues that relate more directly to the systems and protocols

## Agenda

**Getting Started (2 mins)**

**Present the Practice PT**

**Activity (3 days)**

**Practice PT - Flash Talk: The Internet and Society.**  
**Suggested Timeline**

**Wrap-up**

**Delivery and Assessment Options.**

**Assessment**

**Extended Learning**

## Objectives

**Students will be able to:**

- Research a global impact of the Internet.
- Create and present a flash talk on a global impact of the Internet.
- Analyze the relationship of an Internet technology to the impact.

## Preparation

Review the Practice PT and decide how students should present (see Lesson Plan)

## Links

**For the Students**

- **Flash Talk: The Internet and Society** - Practice PT (PDF | DOCX)
- **Unit 1 on Code Studio**

## Vocabulary

- **DNS** - The service that translates URLs to IP addresses.
- **HTTP** - HyperText Transfer Protocol - the protocol used for transmitting web pages over the Internet
- **IP Address** - A number assigned to any item that is connected to the Internet.
- **Net Neutrality** - the principle that all Internet traffic should be treated equally by Internet Service Providers.
- **TCP** - Transmission Control Protocol - provides reliable, ordered, and error-checked delivery of a stream of packets on the internet. TCP is tightly linked with IP and usually seen as TCP/IP in writing.

# Teaching Guide

## Getting Started (2 mins)

### Remarks

At the beginning of this unit we looked at **Vint Cerf's RFC** entitled **"The Internet is for Everyone"** in which he laid out some challenges to the prospect that the Internet would be a large scale, open resource for everyone.

However, on the scale of human history, the Internet is still relatively new and the Internet has introduced some new and potentially difficult issues for people and society that have never existed before. We are still grappling with these issues and they often present moral and ethical dilemmas about what's best to do.

By the same token **a large number of people do not understand as much about the Internet and how it works** as you do. You have now learned quite a bit about how the Internet works as a large scale system, that involves layers of abstraction, protocols and algorithms. You have learned about some of the most fundamental protocols and structures that contribute to the functioning of the Internet.

## Present the Practice PT

Several major issues facing society today require a knowledge of the Internet and how it works in order to have a well-informed opinion about what's best or the right thing to do. Over the next few days you will research one of these issues and prepare a (very) short presentation about it - a Flash Talk

## Activity (3 days)

### Practice PT - Flash Talk: The Internet and Society.

Distribute and review: **Flash Talk: The Internet and Society - Practice PT**

This Practice PT has 3 main components to it:

1. Pick an issue and research it.
2. Prepare (write) a Flash Talk
3. Deliver it.

### Suggested Timeline

Here is a sample timeline for how this project might unfold.

#### DAY 1 - Start Research

- Review the **Flash Talk: The Internet and Society - Practice PT**
- Students choose a topic:
  - Net Neutrality
  - Internet Censorship
  - Advanced: Protocol Hacks and Vulnerabilities
- Students start research

#### DAY 2 - Start Flash Talk Prep

- Conclude research
- Prepare (write) script for flash talk

#### DAY 3 - Finish and Deliver

- Present flash talk (see options listed above)
- Submit materials for assessment

### Teaching Tip

#### Teaching Strategies:

- Pair each student with a research "buddy" to either collaboratively research or to bounce ideas off each other
- Encourage students to start by just simply looking for words and names of protocols that they recognize in the text of articles
- Students may or may not find the research organizer helpful - you might have another template for them to follow that provides more structure or guidance.

### Teaching Tip

If students choose the advanced option they are still responsible for knowing what the issue of Net Neutrality is about. Researching protocol hacks is a certain kind of fun, and students may get very "into" it. However, you may want to ensure to pair a student or group who chooses the "hacking" option with a group who researched Net Neutrality or censorship to make sure that the concepts get to both groups.

## Wrap-up

The wrap up should happen on **Day 3** and you can choose from one of the Delivery and Assessment Options described below.

The major thing students should be able to do is describe the relationship and connection between a societal issue, like Net Neutrality, with

aspects of the technical underpinnings that gave rise to the issue in the first place.

## Content Corner

### Delivery and Assessment Options.

You probably don't have time for every student to give their flash talk to the entire class. You might consider doing one of the following options:

- Put students in small groups of to give their flash talks to each other
- Have students trade what they wrote for the flash talk and present the other person's talk
- Only require that they write the talk (basically a speech or response to a reflection prompt) and turn it in
- Have students read each others' talks anonymously and evaluate it according to the rubric
- Collect the student's research organizer as well as a text-copy of the flash talk for assessment.

### Assessment

See rubric in the Practice PT document.

#### Purpose and Relationship to the Explore PT:

Note that preparing a flash talk is basically like writing a short speech. 2 minutes is about 300 words, which is the maximum word-length of a typical reflection prompt on the Explore Performance task. So you might want to reframe this as a simple prompt to respond to.

The Explore PT is also an individual task for students. You may want students to have a partner to work with here, but students do need some practice doing these things on their own.

The primary skills related to the Explore PT that students are practicing here are:

- Research on a topic related to computing
- Connecting Computing to socially relevant issues
- Explaining the technical details necessary for understanding the issues at play
- Writing short concise text that explains a complicated issue and associated technology

### Extended Learning

- **Blown to Bits (<http://www.bitsbook.com/>):** Students may find additional research help on their Global Impact of the Internet topic in the following chapters:
  - **Chapter 2:** Naked in the Sunlight
  - **Chapter 6:** Balance Toppled
  - **Chapter 7:** You Can't Say That on the Internet
  - **Chapter 8 (some explicit language) | Chapter 8 (clean/edited version):** Bits in the Air

### Standards Alignment

#### CSTA K-12 Computer Science Standards

- ▶ **CD** - Computers & Communication Devices
- ▶ **CI** - Community, Global, and Ethical Impacts
- ▶ **CL** - Collaboration

#### Computer Science Principles

- ▶ **6.3** - Cybersecurity is an important concern for the Internet and the systems built on it.
- ▶ **7.1** - Computing enhances communication, interaction, and cognition.
- ▶ **7.4** - Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.