

Unit 2 - Digital Information

This unit further explores the ways that digital information is encoded, represented and manipulated. In this unit students will look at and generate data, clean it, manipulate it, and create and use visualizations to identify patterns and trends.

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 2 student-facing activity guides [here](#). You can access a flat pdf of all the lessons in unit 2 [here](#).

Chapter 1: Encoding and Compressing Complex Information

Big Questions

- Are the ways in which digital information is encoded more laws of nature or man made?
- What kinds of limitations does the binary encoding of information impose on what can be represented inside a computer?
- How accurately can human experience and perception be captured or reflected in digital information?

Enduring Understandings

- 1.1 Creative development can be an essential process for creating computational artifacts.
- 1.3 Computing can extend traditional forms of human expression and experience.
- 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.

Week 1

Lesson 1: Bytes and File Sizes

Research

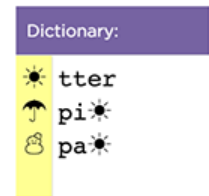
Students are introduced to the standard units for measuring the sizes of digital files: bytes, kilobytes, megabytes, gigabytes, etc. and research the sizes of files they make use of every day.



Lesson 2: Text Compression

Widget - Text Compression | Individual and Group Discovery

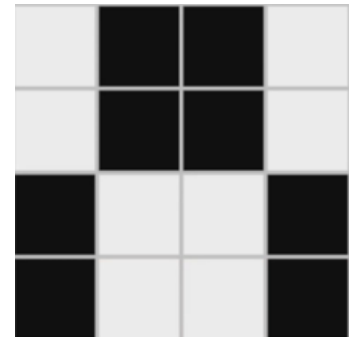
At some point we reach a physical limit of how fast we can send bits and if we want to send a large amount of information faster, we have to find a way to represent the same information with fewer bits - we must compress the data.



Lesson 3: Encoding B&W Images

Widget - Pixelation | Concept Invention | Individual Creation

Students explore methods for encoding digital images in binary which requires representing metadata such as width and height as well as pixel data. Students use the the Pixelation widget to encode simple B&W raster images.

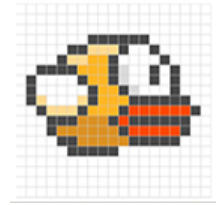


Week 2

Lesson 4: Encoding Color Images

Widget - Pixelation | Individual Creation

Students learn about the RGB color encoding scheme and use an updated version of the pixelation widget to encode color images. Hexadecimal notation is useful for representing larger groupings of binary digits.



Lesson 5: Lossy Compression and File Formats

Research

Students research real compression schemes used for images, text, or sound and determine what kind of compression it uses - lossy or lossless - explaining the theory behind it.



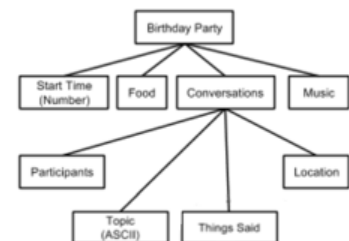
Week 3

Lesson 6: Practice PT - Encode an Experience

Practice PT | Unplugged | Individual Creation

Students break down an ambiguous type of information such as personal experience (attending a party, playing a game, etc) and invent a way to encode its sub-parts. The project includes a written reflection questions similar to those students will see on the AP Performance Tasks.

Diagram: Attending a Birthday Party



Chapter Commentary

Unit 2 Chapter 1 - What's the story?

The story here is about *representing* increasingly complex data and information as an entree to *manipulating* data and information in the next chapter. The lessons are essentially a tour through some of the more interesting forms of digital information representation - specifically, images and text. Encoding images in binary can quickly explode into a number of bits that's hard to keep in one's head all at once. It requires structuring data that includes **metadata**. Compression is the art and science of how to represent the same data with fewer bits, and there are two forms: **lossless** compression, which allows you to reconstruct the exact original bits from the compressed version; and **lossy** compression, common in images and sounds, which throws out information that is likely invisible or inaudible.

The small project that concludes the chapter, *Encode an Experience*, is about the **intersection of abstraction and data**. In a nutshell, students have to think: how can I represent everything here as a series of numbers? The top-down design approach we advocate for is a useful **thinking and problem-solving strategy** for progressively working

at finer and finer levels of detail. This approach is about understanding the spectrum of choices that are made when deciding how to **represent information as data**. Since so many different choices can be made, it explains the existence of so many different data formats for similar information that you encounter on a daily basis. For images you see .jpg, .gif, png. For text: .txt, .docx, .pdf, and so on. What are the differences between these things and, more importantly, why are there differences? Why can't we just settle on a standard image format or protocol? We explore these reasons through learning experiences that allow students to try their hand at it.

The *Encode an Experience* project has a few underlying purposes: 1) it shows how quickly human decision-making comes into play when figuring out how to represent information; 2) the structure students come up with will look like a tree of relationships between different components of information that make up the whole - this is similar to the layers of data abstraction in database designs, and a lot of publicly-available data is often broken up this way; and 3) the "top down" approach for breaking down information is a precursor to ideas about top-down program design we address in Unit 3 - Algorithms and Programming.

Our Approach to the Content

These lessons will, in many ways, feel a lot like the information representation problems encountered in Unit 1 Chapter 1, and the approach you take should be similar - the only difference is that these lessons are strictly about information representation, rather than being about the Internet. Ultimately the choices made about how to represent information affect how you are able to process or compute with it. We encourage students to "peek" out into the real world as you go through lessons in this chapter to relate the way we encode images and compress to text to the way it's done in the "real world".

This chapter leans heavily on two major widgets that allow students to play with concepts. The **Pixelation Widget** lets students enter binary information and the widget renders an image according the embedded image format. The black and white version simply encodes images with 1 bit per pixel - 0 is black, 1 is white - while the color version requires students to understand how the **RGB** color scheme works and why hexadecimal representation is so useful for looking at long strings of binary values. In the widget students must also include **metadata** about the image (width, height, amount of color information), which mimics the "real world" uncompressed image encoding scheme known as **bitmap (bmp)**.

The **Text Compression Widget** lets students play with a text encoding/compression scheme that mimics what's known as LZW or ZIP compression. It works by identifying repeated patterns in the original text and storing them in a "dictionary" of patterns for later recall. The challenge is to see how much students can compress an a piece of text - the catch is that there is no way to *actually* know what the "best" is. Compression is a type of computationally hard problem, and the best solution is to experiment, and come up a **heuristic** - a process that is likely to lead to a good-enough solution.

Chapter 2: Manipulating and Visualizing Data

Big Questions

- What is the relationship between data, information and knowledge?
- What are the best ways to find, see, and extract meaningful trends and patterns from raw data?
- Where and how does human bias affect the collection, processing and interpretation of data?

Enduring Understandings

- 1.3 Computing can extend traditional forms of human expression and experience.
- 3.1 People use computer programs to process information to gain insight and knowledge.
- 3.2 Computing facilitates exploration and the discovery of connections in information.
- 3.3 There are trade offs when representing information as digital data.
- 7.1 Computing enhances communication, interaction, and cognition.

- 7.3 Computing has a global affect -- both beneficial and harmful -- on people and society.

Week 4

Lesson 7: Introduction to Data

Unplugged | External Tools | Individual and Group Discovery

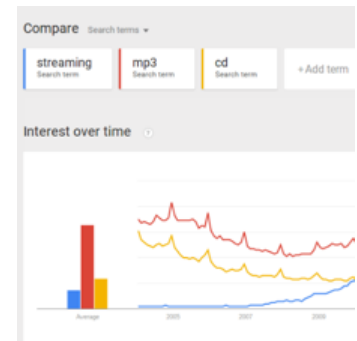
Students examine sources of data in the world around them how that data is collected. The Class Data Tracker project is introduced, and students predict what they will find after all the data has been collected.



Lesson 8: Finding Trends with Visualizations

External Tools | Research | Presentation

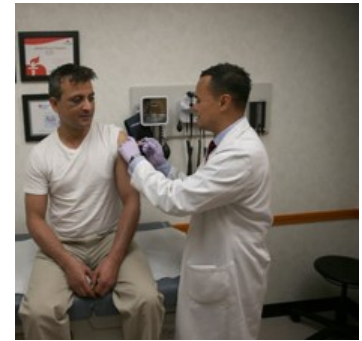
Students use the Google Trends tool in order to identifying patterns in historical search data. Students present their findings, differentiating between explanations of what the data shows versus plausible explanations for discovered patterns.



Lesson 9: Check Your Assumptions

Research | Class Discussion

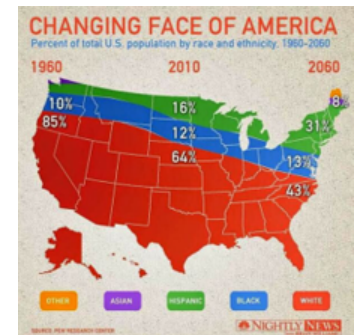
Students examine the assumptions they make when interpreting data and visualizations by first reading a report about the "Digital Divide" which challenges the assumption that data collected online is representative of the population at large. Students also evaluate a series of scenarios in which data-driven decisions are made based on flawed assumptions.



Lesson 10: Good and Bad Data Visualizations

Analyzing Artifacts | Group Discovery | Class Discussion

As a precursor to creating their own data visualizations, students examine collections of (mostly bad) data visualizations, rate them and discuss the characteristics of good v. bad visualizations.

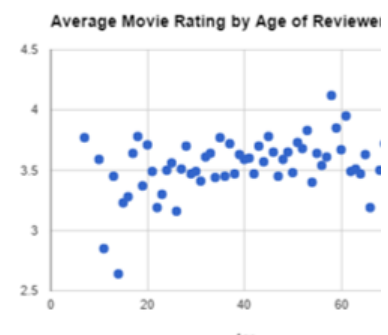


Week 5

Lesson 11: Making Data Visualizations

External Tools | Individual Skill Building | Tutorial

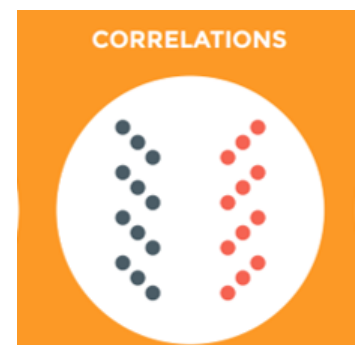
Students follow a guide to learn how to make scatter, bar, and line charts out of provided data using a spreadsheet tool (such as Google sheets or MS Excel).



Lesson 12: Discover a Data Story

External Tools | Collaborative Artifact Creation | Writing

Students collaboratively investigate some datasets (provided) to "discover a data story." Students choose one dataset, create a visualization, identify a trend, and accurately write about it.

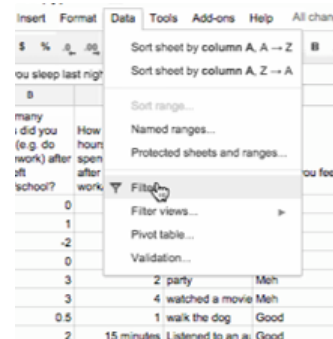


Week 6

Lesson 13: Cleaning Data

External Tools | Analyzing | Group Skill Building

Students begin working with the data that they have been collecting for the Class Data Tracker project by first "cleaning" it to prepare it for visualization and other analyses. Each team makes their own copy of the data to examine, correct errors, categorize ambiguous items, and perform other cleaning tasks.



Lesson 14: Creating Summary Tables

External Tools | Artifact Creation | Analyzing

Students learn how create summary tables (also known as pivot tables) from some raw datasets provided in a spreadsheet tool. Then students create and use summary tables to investigate data they've collected as a class.

Movie	Avg. Rating
Air Force One	3.615384615
Chasing Amy	4
Contact	4.05
...	...
Toy Story	3.615384615
Twelve Monkey	4
Willy Wonka	3.818181818

Lesson 15: Practice PT - Tell a Data Story

Practice PT | External Tools | Artifact Creation

Students continue to analyze their class tracker project data to discover, visualize, write about and present a trend or pattern they find. The writing prompts are reflective of prompts from the AP Explore Performance Task.



Chapter Commentary

Unit 2 Chapter 2 - What's the story?

The story of this chapter is about how data can be manipulated to extract or reveal new information. Up to this point we have been focused primarily on bits and what they can be used to represent. Now we're taking a big step back to do the inverse: we want to use tools meant for viewing, manipulating, and visualizing data in order to *extract* or find new information.

The lessons in this chapter often have two things going on at once. In the background, the class is daily collecting some data about themselves (the "Class Data Tracker project") in order to accumulate data to process later on. In the interim, students are learning about and developing skills with spreadsheet and visualization tools. The goal is

for students to learn a few basic skills, see lots of examples, and then apply what they know to the *Tell A Data Story* project at the end of the chapter.

A big part of the story here is for students to understand the *computer scientist's role* in working with data, which means emphasizing how to use tools to manipulate, compute, and visualize the data. We look at things like making sure that data type choices support the way we intend to process it later (e.g. don't collect text when you need a number). Data inevitably gets "dirty" during collection and needs to be cleaned. Computers are really useful for doing some aggregations and visualizations to look for patterns. Along the way, we need to understand how human bias can be introduced at each step so that we can accurately convey what patterns in the data are or are not telling us. These activities help build toward the enduring understanding that there are trade offs when representing information as digital data.

Our Approach to the Content

The lessons in this chapter lean heavily on external tools, especially **spreadsheets**. The benefit is that students will gain experience with **real tools and real data** for the first time. The pitfall is that, because the tools are external, they are not scaffolded or designed for learning. We have tried to provide tutorials and curated data sets to ease the burden as much as possible, but ultimately you're operating in the real world. While confined to the world of your classroom, the Class Data Tracker project should provide some **authentic** examples, scenarios, and sometimes headaches related to data collection and processing in the real world.

As the teacher it's important to keep in mind the goals of CS Principles because it can be enticing with these lessons to dig into "hardcore" data analysis techniques and statistics. While these are important, they are beyond the scope of CS Principles. Thus, we treat data analysis and statistics a bit like an electric fence: get close, but don't touch. Students should be able to extract interesting things as the result of letting the tools do the work. We provide some large sets of curated data that came from real sources. The data is big enough that you have to apply some computation to make sense of it. We show how to use spreadsheets to do basic aggregations (such as grouping, counting, clustering) and computations (such as average, median, etc.), without turning it into a lesson on statistics and data analysis. We want to build toward the enduring understanding that computing facilitates exploration and the discovery of connections in information.

The idea behind the *Class Data Tracker* project is that we have found that when students work with data that they collected themselves it is easier and **intrinsically motivating** for students to dig in. To accumulate enough data, we collect it in increments during the time they're building up other skills with data tools. You should connect the skills students are learning in the exercises to similar things they might do with the class tracker data for the *Tell a Data Story* project.



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: Bytes and File Sizes

Overview

In this lesson students are introduced to the standard units for measuring the sizes of digital files, from a single byte, all the way up to terabytes and beyond. Students begin the lesson by comparing the size of a plain text file containing “hello” to a Word document with the same contents. Students are introduced to the units kilobyte, megabyte, gigabyte, and terabyte, and research the sizes of files they make use of every day, using the appropriate terminology. This lesson foreshadows an investigation of compression as a means for combatting the rapid growth of digital data.

Purpose

The simple purposes of this lesson are:

1. Get terminology out in the open
2. Become somewhat conversant with file types and sizes
3. Grapple with orders-of-magnitude differences between things.

The 8-bit byte has become the de-facto fundamental unit with which we measure the “size” of data on computers, and in fact, today most computers only let you save data as combinations of whole bytes; even if you only want to store 1 bit of information, you have to use a whole byte to do it. And many computer systems will require you store even more than that. Messages sent over the Internet are also typically structured as messages with byte-offsets.

Paralleling the explosion of computing power and speed, the sheer size of the digital data now created and consumed every day is staggering. Units of measure (terabytes) that previously seemed unfathomably large are now making their way into personal computing. This rapid growth of digital data presents many new opportunities and also poses new challenges to engineers and programmers. The implications of so-called **Big Data** will not be investigated until later in the course, but it's good and interesting to be thinking about the size of things now.

Agenda

Getting Started (10 mins)

Terminology - Byte

Compare sizes of plain text v. MS Word doc

Activity (30 mins)

Rapid Research: Bytes and File Sizes

Wrap-up

Review worksheet

Foreshadow Compression

[View on Code Studio](#)

Objectives

Students will be able to:

- Use appropriate terminology when describing the size of digital files.
- Identify and compare the size of familiar digital media.
- Solve small word problems that require reasoning about file sizes.

Preparation

- ☐ You should verify that you know how to look at the sizes of files on computers that your students are using (see activity).
- ☐ For the getting started activity might want a Word processing program (such as MS Word) and plain text editor (such as Notepad or TextEdit) open and ready.
- ☐ The teaching remarks and content corners in this lesson contain lots of little bits of history that you might choose to share at various points in the lesson.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- [Activity Guide KEY - Bytes and File Sizes - Answer Key](#)

For the Students

- [Bytes and File Sizes - Activity Guide](#)

[Make a Copy](#) ▼

Teaching Guide

Getting Started (10 mins)

Remarks

As we embark on a new unit about Data and Digital Information we need to get familiar with terminology about data and different types of data files.

Terminology - Byte

Recall that a single character of ASCII text requires 8 bits. The technical term for 8 bits of data is a Byte.

A **byte** is the standard fundamental unit (or “chunk size”) underlying most computing systems today. You may have heard “megabyte”, “kilobyte”, “gigabyte”, etc. which are all different amounts of a bytes. We're going to learn more about them today.

Compare sizes of plain text v. MS Word doc

Introduction:

Recall In a previous lesson (Unit 1 - Sending Formatted Text) we learned that in addition to the actual text of a document, it is usually necessary to store the formatting information that allows the text to be displayed correctly. We might wonder just how much extra information, i.e. how many extra bytes, we need to store when we include all of this formatting. Let's find out!

If a single ASCII character is one byte then if we were to store the word “hello” in a plain ASCII text file in a computer, we would expect it to require 5 bytes (or 40 bits) of memory.

What about a Microsoft Word document that contains the single word “hello”?

Predict: “How many more bytes will a Word document require to store the word “hello” than a plain text document?”

- Give students a chance to write down a prediction or ask for predictions and write them on the board.

Demonstrate or lead students through discovering for themselves the size of a word-processing document.



Here are some files you can download to use.

- Plain text document **hello.txt**
- MS word document **hello.docx**

To find the actual size of a file on your computer, do one of the following:

- PC/Windows: Right-click and choose “Properties”
- Mac: Ctrl+click and choose “Get Info”

In general, the Word Doc should be **thousands of times larger** than the plain text. For the files above:

- hello.txt - 5 bytes

Content Corner

Why is a Byte 8 bits?

The 8-bit byte was not always standard. Computers used many different “byte” sizes over the course of history, depending on hardware and how addressable memory worked. However, much of the early computing world relied on representing data and computer instructions encoded in ASCII text where every character is 8 bits. Thus, 8-bits was such a common chunk-size for representing information that it stuck and they gave it its own name - **byte**.

There are various accounts about why it was called a “byte” but most point to early days at IBM where “bite” was used to refer to groups of 8-bits that a computer was processing, as in it could “bite” off 8 bits at time. The spelling was changed to “byte” to avoid confusion with “bit”.

Bytes became the fundamental unit with which we measure the “size” of data on computers, and in fact, today most computers only let you save data as combinations of whole bytes; even if you only want to store 1 bit of information, you have to use a whole byte to do it.

Teaching Tip

If you wish, it might be more fun to create these files in front of your students, saving them on the desktop for a quick demo. To make a plain ASCII text file you’ll need to use the correct program:

- PC/Windows: use Notepad
- Mac: use TextEdit (Note: TextEdit needs to be switched into plain text mode from rich text. Go to Format → Make Plain Text)

- hello.docx = 21,969 bytes

Look back at predictions to see how close they were.

- The big difference in file size between .txt and .docx is due to the extensive formatting information included along with the actual text in .docx.

Transitional Remark

Modern data files typically measure in the thousands, millions, billions or trillions of bytes. Let's get a little practice looking at files and how big they are.

Content Corner

NOTE: A 5-byte file is so small that some computers won't allocate a chunk of memory that small. For example you might see something like this:

Kind: Plain Text Document
Size: 5 bytes (4 KB on disk)

Which indicates that even though the file is 5 bytes, it's taking up 4 Kilobytes of memory on your computer.

Activity (30 mins)

Rapid Research: Bytes and File Sizes

Activity Guide: Bytes and File Sizes - Activity Guide

- Put students in pairs to find answers or work individually.

Distribute: Activity Guide: Bytes and File Sizes - Activity Guide

- Introduces the terminology
- Refers to websites for students to use as reference
 - **Stanford CS 101 website**
 - **Computer Hope**



- Has questions and space for students to write answers to questions like:
 - How many bytes are in a Megabyte?
 - Give an example of a file type that is measured in Gigabytes
 - What is the typical size of a .jpg image, .mp3 audio etc.
- Allow students time to finish this activity either individually or in pairs by conducting online research.



- There are 6 practice questions on the 2nd page of the activity guide.

Content Corner

There are some discrepancies in common usage of the kilo, mega, giga prefixes.

From the **Stanford CS 101 website**:

It's convenient within the computer to organize things in groups of powers of 2. For example, 2^{10} is 1024, and so a program might group 1024 items together, as a sort of "round" number of things within the computer. The term "kilobyte" above refers to this group size of 1024 things. However, people also group things by thousands -- 1 thousand or 1 million items.

*There's this problem with the word "megabyte" .. does it mean $1024 * 1024$ bytes, i.e. 2^{20} which is 1,048,576, or does it mean exactly 1 million, $1000 * 1000$. It's just a 5% difference, but marketers tend to prefer the 1 million, interpretation, since it makes their hard drives etc. appear to hold a little bit more. In an attempt to fix this, the terms "kibibyte" "mebibyte" "gibibyte" "tebibyte" have been introduced to specifically mean the 1024 based units (see wikipedia kibibyte article). These terms do not seem to have caught on very strongly thus far.*

If nothing else, remember that terms like "megabyte" have this little wiggle room in them between the 1024 and 1000 based meanings. **For purposes of CS Principles the distinction is not important - "about a million bytes" is a fine, close-enough interpretation for "megabyte".**

Wrap-up

Review worksheet

Share: Provide students an opportunity to clear up any remaining confusion and share interesting pieces of information they came across.

Review answers to the questions on the Activity Guide.

Foreshadow Compression

Remarks

As you have seen data file size can grow very quickly in size. In the modern world there is a lot of data around us and usually we want it transmitted over the internet.

There is a problem though: If you want to transmit a lot of data you are limited by the speed of your internet connection. Even if you have a fast Internet connection there is a *physical limit* to how fast you can transmit bits.

What if the data you want to send is big enough that it takes an unreasonable amount of time to transmit it, even with a really fast internet connection. Assuming you can't make the Internet connection any faster, could you still transmit the data faster somehow?



The answer is yes and it's probably something you've done, or do every day!

Teaching Tip

Note that answers to 3 of the 6 questions on the activity guide can be found on the Stanford CS 101 page linked to in the activity guide.

Perfect accuracy is not important for some sections in this activity, but using the correct terminology and achieving a rough estimate of size (one million bytes vs. one billion) is important. Encourage students to practice using terms like megabyte, gigabyte, and terabyte to gain comfort with them.

Teaching Tip

Time permitting you could do the warm up activity from the next lesson (Text Compression) here. That warm up activity asks students to write down common abbreviations they use when sending text messages to friends and family, and then asks why they do that. The answer is compression: to save time and space.

Assessment

Use the last 3 questions on the activity guide for assessment.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- **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 2: Text Compression

Overview

At some point we reach a physical limit of how fast we can send bits and if we want to send a large amount of information faster, we have to find a way to represent the same information with fewer bits - we must **compress** the data.

In this lesson, students will use the Text Compression Widget to compress segments of English text by looking for patterns and substituting symbols for larger patterns of text. After some experimentation students are asked to come up with a process (or algorithm) for arriving at a "good" amount of compression despite the fact that there is no way to know what is best or optimal. In developing a so-called "heuristic" approach to this problem, students will grapple with the tradeoffs in compressing data and begin to develop a sense of computing problems that are "hard" to solve.

Purpose

This is a big lesson that covers a lot of bases. It should easily take 2 or more days of class. First and foremost it covers two or three topics directly from the CSP framework.

1. lossless compression

The basic principle behind compression is to develop a method or protocol for using fewer bits to represent the original information. The way we represent compressed data in this lesson, with a "dictionary" of repeated patterns is similar to the **LZW compression scheme**, but it should be noted that LZW is slightly different from what students do in this lesson. Students invent their own way here. LZW is used not only for text (zip files), but also with the GIF image file format.

2. heuristics

The lesson touches on computationally hard problems and heuristics but please note that **computationally hard problems and heuristics will be revisited later on**. A general "hand-wavy" understanding is all that's needed from this lesson.

We do want students to see, however, that there is no single correct way to compress text using the method we use in this lesson because a) there is no known algorithm for finding an optimal solution, and b) we don't even know a way to verify whether a given solution is optimal. There is no way to prove it or derive it beyond trying all possibilities by brute force. **This is an example of an algorithm that cannot run in a "reasonable amount of time"** - one of the CSP learning objectives.

3. Foreshadowing programming behaviors

[View on Code Studio](#)

Objectives

Students will be able to:

- Collaborate with a peer to find a solution to a text compression problem using the Text Compression Widget (lossless compression scheme).
- Explain why the optimal amount of compression is impossible or "hard" to identify.
- Explain some factors that make compression challenging.
- Develop a strategy (heuristic algorithm) for compressing text.
- Describe the purpose and rationale for lossless compression.

Preparation

- ☐ Test out the Text Compression Widget
- ☐ Review the teaching tips to decide which options you want to use

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **Activity Recap - Decode this Message - Activity Recap** [Make a Copy](#)

For the Students

- **Decode this message - Activity Guide** [Make a Copy](#)
- **Activity Guide - Text Compression - Activity Guide** [Make a Copy](#)
- **Video: Text Compression with Aloe Blacc - Video (download)**
- **Activity Guide - Text Compression Heuristics - Activity Guide** [Make a Copy](#)

Vocabulary

- **Heuristic** - a problem solving approach (algorithm) to find a satisfactory solution

Lastly, the Text Compression Activity is an **important lesson to refer back to when students start programming**. The activity engages students in thinking and problem solving behaviors that foreshadow skills that are particularly useful for programming later down the line. In particular, when students recognize patterns that repeat, and then represent those patterns as abstract symbols, and then further recognize patterns within those patterns, it is very similar to the kinds of abstractions we develop when writing **functions and procedures when programming**. Decoding the message in the warm-up activity is very similar to tracing a sequence of function calls in a program.

Agenda

Getting Started (5-7 mins)

Warm up: Abbr In Ur Txt Msgs (5-7 mins)

Activity (45 mins)

Decode this Mystery Text (10-15 mins)

Use theText Compression Widget

Discuss properties and challenges with compression.

Activity 2 (30 mins)

Develop a heuristic for doing compression

What's best?

Wrap-up (20 mins)

Recap Questions

Compression in the Real World (.zip)

Assessment

Extended Learning

where finding an optimal or exact solution is impractical or impossible.

- **Lossless Compression** - a data compression algorithm that allows the original data to be perfectly reconstructed from the compressed data.

Teaching Guide

Getting Started (5-7 mins)

Warm up: Abbr In Ur Txt Msgs (5-7 mins)

Prompt:



- "When you send text messages to a friend, do you spell every word correctly?"
 - Do you use abbreviations for common words? List as many as you can.
 - Write some examples of things you might see in a text message that are not proper English.

Give students a minute to write, and to share with a neighbor?

- "Why do you use these abbreviations? What is the benefit?"
 - Possible answers:
 - to save characters/keystrokes
 - to hide from parents/teachers
 - to be cool, clever, funny
 - to "speak in code"
 - to say the same thing in less space

Discussion Goal

As a warm up to thinking about Text Compression, connect to ways that most people already compress text in their lives, through abbreviations and acronyms with which most people have some experience in text messages.

Motivate some ideas about why someone would want to compress text.

What's this about? - Compression: Same Data, Fewer Bits

- Today's class is about **compression**
- When you abbreviate or use coded language to shorten the original text, you are "compressing text." Computers do this too, **in order to save time and space**.
- The art and science of compression is about figuring out how to represent the SAME DATA with FEWER BITS.
- Why is this important? One reason is that storage space is limited and you'd always prefer to use fewer bits if you could. A much more compelling reason is that there is an upper limit to how fast bits can be transmitted over the Internet.
- What if we need to send a large amount of text faster over the Internet, but we've reached the physical limit of how fast we can send bits? Our only choice is to somehow capture the same information with fewer bits; we call this **compression**.

Transition:

Let's look at an example of a text message that's been compressed in a clever way.

Activity (45 mins)

Decode this Mystery Text (10-15 mins)

- **Distribute or Display** the Activity guide: **Decode this message - Activity Guide**
- Put students into partners or work individually.
- **Task:** What was the original text?
- Give students a few minutes to decode the text. The text should be a short poem (see activity recap below)

Student Activity Guide

Activity Recap

Student Activity Guide	Activity Recap
<p>Distribute or Display Activity guide: Decode this message - Activity Guide</p> <div> <div> <p>Encoded Message:</p> <div>★listen_to★rain_★on★window_pane</div> <p>Original Message:</p> <div></div> </div> <div> <p>Key:</p> <div> <div>★_the_</div> <div>☂tter_</div> <div>☂Pi☂</div> <div>☂Pa☂</div> <div>★☂☂☂☂</div> </div> </div> </div>	<p>(Display or draw yourself) Activity Recap: Activity Recap - Decode this Message - Activity Recap</p> <div> <p>How Much? 40% compressed!</p> <div> <p>Original: 93 characters</p> <p>Compressed: 56 characters</p> <p>Difference: 37 characters (~40%)</p> </div> </div> <div> <p>Original Message 93 characters</p> <div>Pitter_patter_pitter_patter_listen_to_the_rain_pitter_patter_pitter_patter_on_the_window_pane</div> </div> <div> <p>Compressed 56 characters</p> <div> <div>★listen_to★rain_★on★window_pane</div> <div>31 characters</div> <div>Total number of characters needed to represent compressed version is:</div> <div>31 (message) + 25 (key) = 56</div> </div> <div> <div> <div>☂_the_</div> <div>☂tter_</div> <div>☂Pi☂</div> <div>☂Pa☂</div> <div>★☂☂☂☂</div> </div> <div>25 characters</div> </div> </div>

Recap: How much was it compressed?

To answer, we need to compare the number of characters in the original poem to the number of characters needed to represent the compressed version.

Let's break it down.

- **Display** or **Demonstrate** yourself ideas from: **Activity Recap - Decode this Message - Activity Recap** (shown in table above)
- **Important Note:**
 - The compressed poem is not *just* this part: ★listen_to★rain_★on★window_pane If you were to send this to someone over the Internet they would not be able to decode it.
 - The full compressed text includes BOTH the compressed text *and* the key to solve it.
 - Thus, you must account for the total number of characters in the message *plus* the total number of characters in the key to see how much you've compressed it over the original.

Transition

Now you're going to get to try your hand at compressing some things on your own.

Use theText Compression Widget

Code Studio levels

Lesson Vocabulary & Resources

1

(click tabs to see student view)

Text Compression

Student Overview

Widget: Text Compression

Student Overview

Check Your Understanding

4

5

6

7

(click tabs to see student view)



Video: Text Compression with Aloe Blacc -

Video



- Video explains compression
- Demonstrates the use of the Text Compression Tool.
- NOTE: This video pops up automatically when students visit the text compression stage in Code Studio.
- Divide students into groups of 2
- Assign each pair one of the poems provided and challenge them, as a pair to compress their poem as much as possible.
- Deliver or put simple instructions on the board so students can follow.
 - **Challenge:** compress your assigned poem as much as possible.
 - Compare with other groups to see if you can do better.
 - Try to develop a general strategy that will lead to a good compression.
- After some time, have pairs that did the same poem get together to compare schemes. As a group their job is to come up with the best compression for that poem for the class.
- Optionally: you may hand out **Activity Guide - Text Compression - Activity Guide** and have students complete it individually. It may work well as an out-of-class activity or assessment.

Discuss properties and challenges with compression.

Ask groups to pause to discuss the questions at the end of the activity.

Prompts:

- "What makes doing this compression hard?"
 - Invite responses. Some of these issues should surface: You can start in lots of different ways. Early choices affect later ones. Once you find one set of patterns, others emerge.
 - There is a tipping point: you might be making progress compressing, but at some point the scale tips and the dictionary starts to get so big that you lose the benefit of having it. But then you might start re-thinking the dictionary to tweak some bits out.
- "Do we think that these compression amounts that we've found are the the best? Is there a way to know what the best compression is?"
 - We probably don't know what's best.



Content Corner

The video explains a little bit about compression in general - the difference between lossless compression and lossy compression. Today's class is about **lossless compression** we'll do lossy compression in a class or two after looking at image encoding.



Teaching Tip

Teacher's Choice whether to show the video to the whole class or let students watch it from within Code Studio. There are benefits and drawbacks to each.

Option to Consider: Get students into the text compression tool BEFORE showing the video. You might find students are more receptive to some of the information in the video if they have tried to use the tool first.

Communication and Collaboration: To develop communication and collaboration between students, include one of the following scenarios in class:

- Have students who were assigned the same poem compare results, or seat them in the same area of the room.
- Have a little friendly competition - but be careful not to let "bad" competition seep in - to see which pair can compress a poem the most. Use a poem that none of the students have compressed yet.
- For each poem, have the group(s) who did it figure out the best in the class, and record it on the board or somewhere that people can see.
 - Have a class goal of getting the compression percentages for the four poems as high as possible.
 - The groups with the best compression percentages may be asked to share their strategy with the class.

Students may be reluctant to share if they feel they don't have the best results, but students should see others' work and offer advice and strategies.

- There are so many possibilities it's hard to know. It turns out the only way to guarantee perfect compression is brute force. This means trying every possible set of substitutions. Even for small texts this will take far too long. The "best" is really just the best we've found so far.
- *"But is there a process a person can follow to find the best (or a pretty good) compression for a piece of text?"*
 - Yes, but it's imprecise -- you might leave this as a lingering question that leads to the next student task.

Activity 2 (30 mins)

Develop a heuristic for doing compression

Distribute or Display: Activity Guide - Text Compression Heuristics - Activity Guide

In computer science there is a word for strategies to use when you're not sure what the exact or best solution to a problem is.

Vocabulary: heuristic a problem solving approach (typically an algorithm) to find a satisfactory solution where finding an optimal or exact solution is impractical or impossible.

Instructions:

- Continue working on compressing your poem using the Text Compression Widget. As you do so, develop a set of rules, or a "heuristic" that generally seems to provide good results.
- **Record your heuristic** as a list of steps that someone else unfamiliar with the problem could follow and still end up with decent compression.
- **Trade your heuristics with another group.** Are they clear and specific enough that you always know what to do? If not, provide feedback to one another and improve your heuristics to provide clearer instructions.
- Using another group's heuristic, attempt to compress one or more of the poems in the tool. Record the amount of compression you achieve.

Teaching Tip

You may elect to not do this heuristic activity and instead get the key take-aways (see Activity Goal below) across through discussion following the previous activity.

Activity Goal

The point here is to establish:

- There is no real way to determine for sure that you've got the *best* compression besides trying everything possible by brute force.
- Heuristics are techniques for at least making progress toward a "good enough" solution.
- Following the same heuristic might lead to different results.

What's best?

Share Findings:

Have one member of each group give a summary of their heuristic and the results on each of the poems. If time is limited, these presentations can be done between groups instead in front of the entire class. The discussion questions below could also be done group to group.

Reflection Prompts (from the Activity Guide)



"Do you think it's possible to describe (or write) a specific set of instructions that a person could follow that would always result in better text compression than your heuristic? Why or why not?"

- Some compression programs (like zip) do a great job if the file is sufficiently large and has reasonable amounts of repetition.
- However, it is also possible to create a "compressed file" that is larger than the original because the heuristic does work in every single case.

"Is there a way to know that a compressed piece of text is compressed the most possible? If yes, describe how you could determine it. If no, why not?"

- Stress that there is no perfect solution.

- The size and shape of the data will determine what the “best” answer is and we often cannot even be sure it is the best answer (only that it is better than other answers we have tried.)

Wrap-up (20 mins)

Recap Questions

“What did all groups’ processes for compression have in common?”

- Pattern Recognition
- Abstraction (patterns referring to other patterns)

“Will following this process always lead to the same compression? (i.e. two people following the process for the same poem, will result in the same compression?)”

- No. It’s imprecise, but still OK. The text still gets compressed, no matter what.
- Since there is no way to know what’s best, all we need is a process that comes up with some solution, and a way to make progress.

Terminology: Verify students know or use an **exit ticket* on this vocabulary:

- lossless compression v. lossy compression
- heuristic

Compression in the Real World (.zip)

Zip Compression

- There is a compression algorithm called **LZW compression** upon which the common “zip” utility is based. Zip compression does something very similar to what you did today with the text compression widget.
- Here is an **animation of lzw in action**. You can see the algorithm doesn't compress it the most, but it is following a heuristic that will lead to better and better compression over time.
- Do you want to use zip compression for real? Most computers have it built in:
 - **Windows:** select a file or group of files, right-click, and choose “Send To...Compressed (zipped) Folder.”
 - **Mac:** select a file or group of files, ctrl+click, and choose “Compress Items.”
- Warning: if you try this results may vary.
 - Zip works *really* well for text, but only on large files. If you try to compress the simple hello.txt file we used in a previous lesson, you'll see the resulting file is actually *bigger*.
 - Zip is meant for text. It might not work well on non-text files very well because they are already compressed or don't have the same kinds of embedded patterns that text documents do.

Teaching Tip

- You **do not** have to review or demo LZW compression in depth here. It is an interesting real-world application of the activity done in class.
- While details of LZW compression are not part of the AP course content, but the idea of lossless compression is.
- Recommendation: demonstrate zip quickly.
- Have a large text file at the ready, such as the **plaintext version of Hamlet**
- Use the .zip utility on your computer to compress into a zip file and then compare the file size to the original. (We learned how to do this in the previous lesson).

Assessment

Questions:

- If you send the compressed poem, would your friend will be able to read it? Why is the dictionary important?
 - Your friend would only be able to read it if she knew how it was encoded. The dictionary is necessary because it tells her how to decompress the information that she has.

- Why do you want to compress anything? What’s the point?
 - It is useful for sending things faster or for smaller storage. It allows for optimization of limited resources.
- For a piece of text, what is a “good” amount of compression? Is there a way to know when you’ve compressed it the most? Explain how you would know, or why you can’t know.

Case Study: A simple message has been compressed below:

Compressed Poem	Dictionary
_o-★_t_★_o-back	bi bu the _o-g_g_ t__★ll_
<div> Compressed poem size: 17 bytes Dictionary size: 26 bytes Total: 43 bytes </div>	

- What was the original message?
 - the_big_bug_bit_the_bull_but_the_bull_bit_the_big_bug_back
- Approximately what was the percentage of compression? (count bytes in original vs. total bytes in compressed version)
 - approximately: 25% compression

Extended Learning

Real World: Zip Compression

- Experiment with zip using text files with different contents. Are the results for small files as good as for large files? (On Macs, in the Finder choose “get info” for a file to see the actual number of bytes in the file, since the Finder display will show 4KB for any file that’s less than that.)
 - Warning: results may vary. Zip works really well for text, but it might not compress other files very well because they are already compressed or don’t have the same kinds of embedded patterns that text documents do.

Challenge: Research the LZW algorithm

- .zip compression is based on the **LZW Compression Scheme**
- While the idea behind the text compression tool is similar to LZW (zip) algorithm, tracing the path of compression and decompression is somewhat challenging. Learning more about LZW and what happens in the course of this algorithm would be an excellent extension project for some individuals.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CL** - Collaboration
- ▶ **CPP** - Computing Practice & Programming
- ▶ **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.
- ▶ **4.2** - Algorithms can solve many but not all computational problems.



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: Encoding B&W Images

Overview

In this lesson, students will begin to explore the way digital images are encoded in binary. The class begins by asking students to invent their own image encoding protocol in order to familiarize themselves with some of the subtle complications of encoding images, namely the need for other data, called metadata, that describes properties of the image necessary for rendering it. Students will learn about pixels, raster images, and what an image file format is. Students will encode binary image data using a widget in Code Studio.

Purpose

The main purpose of this lesson is for students to exhibit some creativity while getting some hands-on experience manipulating binary data that represents something other than plain numbers or text. Connections to abstraction in data can be made here. Connections can be made back to file sizes and file formats here as well - e.g. how many bytes does it take to store an image v. text? If you want to broach the subject, the concept of data compression can come in here too - it is interesting to think about how a black and white image might be compressed. You should be aware that this lesson largely acts a stepping stone to the *next lesson* which addresses how RGB colors are represented in binary.

Image file types have some similarities to data packets we saw in the Internet unit -- because images must include metadata, or data about the data. The data of a black-and-white image is the list of bits that represent whether each pixel is on or off. To create the image, however, we must also know how wide and tall the image is in order to recreate it accurately. This necessitates the creation of a file format which clearly defines how this metadata will be encoded, since it is crucial for interpreting the subsequent data of the image. It is similar to how an internet packet doesn't only contain the data you need to send, but must also include metadata like the to and from addresses and packet number.

Digital images can be stored in many formats, but one of the most common formats is "raster". Raster images store the image as an array of individual pixels, each of which has a particular color. Higher-quality images can be obtained by decreasing the size of the pixels (resolution). While full color will be addressed in the next lesson, an important idea here is that images on computer screens are created with *light* by illuminating pixels on the screen. This is why it is typical in a black and white image for the value 1 to represent white - it means turn the light on - and 0 represents black - light off. If you were drawing on paper you might do the inverse.

[View on Code Studio](#)

Objectives

Students will be able to:

- Explain how images are encoded with pixel data.
- Describe a pixel as an element of a digital image.
- Encode a B&W image in binary representing both the pixel data (intensity) and metadata (width, height).
- Create the necessary metadata to represent the width and height of a digital image, using a computational tool.
- Explain why image width and height are metadata for a digital image.

Preparation

- (Optional) Graph or grid paper for drawing pixel images by hand

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **Activity Guide KEY - Encode a B&W Image** - Answer Key
- **Teaching Tips & Tricks Video - Encoding Images** - Video ([download](#))

For the Students

- **B&W Pixelation Widget** - Activity Guide [Make a Copy](#)
- **Extension: Magnify an Image (optional)** - Activity Guide [Make a Copy](#)
- **B&W Pixelation Tutorial** - Video ([download](#))
- **Invent a B&W image encoding scheme** - Activity Guide [Make a Copy](#)

Vocabulary

- **Image** - A type of data used for graphics or pictures.

Agenda

Getting Started (10 mins)

Invent An Encoding Scheme for B&W Images

Activity (40 mins)

Video: The Pixelation widget

Use the Pixelation Widget

Wrap-up (10 mins)

Assessment

Extended Learning

- **metadata** - is data that describes other data. For example, a digital image may include metadata that describe the size of the image, number of colors, or resolution.
- **Pixel** - short for "picture element", the fundamental unit of a digital image, typically a tiny square or dot that contains a single point of color of a larger image.

Teaching Guide

Getting Started (10 mins)

Remarks

Back in the Internet Unit you encoded a line-drawing image as a list of numbers that made up the coordinates of the points in the image. That works for line drawings, but how might you encode a different kind of image? Today we're going to consider how you might use bits to encode a photographic image, or if you like: how could I encode vision?

Today, we're going to start to learn about images, but we're going to start simple, with black and white images.

Invent An Encoding Scheme for B&W Images

• **Distribute Invent a B&W image encoding scheme - Activity Guide**

- Separate students into pairs, and hand each student a copy of the activity guide.
- Students should work the first two pages
- Give groups time to work

Discuss:

Ask students to **share-out** their file format to identify commonalities and patterns. (Use: **Teaching Strategies for the CS Classroom - Resource** for ideas about how to share out.)

- As a class, address students' questions that arise from the concept invention activity.
- Use the questions below to spur conversation. If the concept of metadata, or data about data, arises naturally, then address it here.

Prompts:

- How have you encoded white and black portions of your image, what do 0 and 1 stand for in your encoding?
- Are your encodings flexible enough to accommodate images of any size? * How do they accomplish this?
- Is your encoding intuitive and easy to use?
- Is your encoding efficient?

Activity Goal

The purpose of this little concept invention activity is to be creative and to get the mind moving. There is no exact right answer that we're going for here.

There are many clever and interesting ways this could be done. Most students will likely end up saying that each pixel should be represented with either a 0 or a 1.

But what we really want to draw out is the idea of "metadata". Simply encoding the pixel data is not enough. We also need to encode the width and height of the image, or the image could not be recreated - other than through trial and error

Remarks



- **Vocabulary:** each little dot that makes up a picture like this is called a pixel. Where did this word pixel come from? It turns out that originally the dots were referred to as "picture elements", that got shortened to "pict-el" and eventually "pixel".
- What we've discovered is that the data for our image file must contain more than just a 0 or 1 for every pixel. It must contain other data that describes the pixel data.
- This is called metadata. In this case the metadata encodes the width and height of the image.
- We've seen forms of metadata before. For example: an internet packet. The packet contains the data that needs to be sent, but also other data like the *to* and *from* address, and packet number.

Content Corner

There is some mystery about the etymology of the word "pixel". You can read more about it on the **Wikipedia: pixel page**

Activity (40 mins)

Introduction

- The pixelation widget in Code Studio will allow us to play with these ideas a little more.
- This widget follows a particular encoding scheme for images that you'll have to follow.

Video: The Pixelation widget

- Show the tutorial video: **B&W Pixelation Tutorial - Video**
- NOTE: This video pops up the first time you visit the pixelation widget in Code Studio. You might prefer to have students watch it there on their own.



Use the Pixelation Widget

Distribute: Activity Guide - **B&W Pixelation Widget - Activity Guide**

Activity Guide

Page 1:

- Explains the encoding scheme and a bit about how the tool works.
- Describes the 3 student tasks to get familiar with the tool:
 1. Create a small image: Start by trying to recreate the 3x5 letter "A" depicted (shown above) using the pixelation widget.
 2. Correct an error: Oh no! An extra bit was inserted into an image during transmission! Track it down.
 3. Make your own image of any size of anything you like.

The second page asks students to:

- Copy/paste a copy of their personal creation
- Copy/paste the bits that are used to encode it
- Written reflection questions:
 - What are the largest dimensions (width and height) of an image we can make with the pixelation widget?
 - How many total bits would there be in the largest possible image we could make with the pixelation widget?
 - How many bits would it take to represent the smallest possible image (i.e. an image with one pixel)?
 - What would happen if we didn't include width and height bits in our protocol? Assume your friend just sent you 32 bits of pixel data (just the 0s and 1s for black and white pixels). Could you recover the original image? If so, how?

Teaching Tip

- You may not need or want to use the first page of the activity guide. It is a reference for students, but the tasks for students are given in Code Studio.
- Similarly for the second page, if you don't intend to collect it for assessment purposes, you can use the questions as group discussion or wrap-up questions.

Wrap-up (10 mins)

Review:

- The image file protocol we used contains "metadata": the width and height. **Metadata is "data about the data"** that might be required to encode or decode the bits.
- For example, you couldn't render the B&W image properly without somehow including the dimensions. Prompts:
 - What other examples of metadata have we seen in the course so far?
 - What other types of data might we want to send that would require metadata?

(Optional) Prompt:

"Did you think about compression at all while doing this exercise? Can you think of a way that you might represent an image of pixel data with fewer bits? What would have to change about the encoding strategy?"

- For an answer to this see the **"Color by Numbers"** Activity from CS Unplugged (csunplugged.org).
- It uses something called "run-length encoding"

Assessment

Check students responses on: **B&W Pixelation Widget - Activity Guide**

- Check to make sure that the bits they submitted actually produce the image as claimed.
- Score the digital artifact as you see fit, with points for creativity and perceived effort.
- The following questions can be found in the Activity Guide and also appear on Code Studio
- Answers these questions can be found here: **Activity Guide KEY - Encode a B&W Image - Answer Key**
 - Using the B&W file format from the pixelation widget
 - What are the largest dimensions (width and height) of an image we can make with the pixelation widget?
 - How many total bits would there be in the largest possible image we could make with the pixelation widget?
 - How many bits would it take to represent the smallest possible image (i.e. an image with one pixel)?
 - What would happen if we didn't include width and height bits in our protocol? Assume your friend just sent you 32 bits of pixel data (just the 0s and 1s for black and white pixels). Could you recover the original image? If so, how?

Extended Learning

- Check out the **"Color by Numbers"** from CS Unplugged (csunplugged.org) which uses a different clever encoding scheme for B&W images.
- ◦ Do the **Extension: Magnify an Image (optional) - Activity Guide** activity (double the size of an image on the Pixelation Tool).
- Have students research raster graphics in anticipation of the subsequent lesson.
- Attempting to communicate with possible intelligent life beyond our solar system has been a dream for humans and the goal of scientists for many years. Questions about messages to send, as well as how to send messages deep into space to unknown recipients have been debated. In 1974, scientists sent the Arecibo message to the star cluster M13 some 25,000 light years away. Read about the message they sent using 1,679 binary digits (https://en.m.wikipedia.org/wiki/Arecibo_message).
 - How would you change the content of the message? What would you delete and add? Why would your change be significant in a communication to other intelligent beings?
 - Sketch the segment of the design you would alter. Remember, you must retain the original number of bits.
 - List the details in this article that you understand more deeply because of what you have learned in this class up to this point.

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CL** - Collaboration
- ▶ **CPP** - Computing Practice & Programming
- ▶ **CT** - Computational Thinking

Computer Science Principles

- ▶ **1.1** - Creative development can be an essential process for creating computational artifacts.
- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **1.3** - Computing can extend traditional forms of human expression and experience.
- ▶ **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.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.
- ▶ **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 4: Encoding Color Images

Overview

In this lesson students are asked to consider how color is represented on a computer and to imagine how it might be encoded in binary. Students then learn about how color is actually represented on a computer - using the RGB color scheme - and create their own images in an new version of the pixelation widget that allows you use more than 1 bit per pixel to represent color information. After grappling with the prospect of possibly many bits just to represent a single pixel, students are shown how using hexadecimal allows us to represent many bits with fewer characters. Students use a new version of the pixelation tool to encode an image with color and create a personal favicon.

Purpose

The main purpose here, similar to the B&W pixelation activity is for students to get hands-on and "down and dirty" with bits. A major outcome will also be understanding the relationship between hexadecimal (base-16) and binary (base-2), and how useful it is to use hex to represent groups of 4 bits. It's important to realize that using hex is not a form of data compression, it's simply a different view into the bits.

The most common color representation scheme - RGB - typically uses 24 bits (3 bytes) with 8 bits each for Red, Green and Blue intensities. And one of the most common ways you see these colors represented is in hexadecimal. The pixelation widget, with its ability to choose how many bits represent the color value for each pixel, can be a very useful tool for showing the utility of hex representations for bits.

The process of rendering color on a computer screen by mixing red, green and blue light is an important concept of this lesson. The results are not always intuitive, because mixing pigment and mixing colored lights (like what's on a computer screen) lead to different results.

Another important objective of this lesson is to understand how (uncompressed) image file sizes can become quite large. For example, even a relatively small image of 250x250 pixels is a total of 62,500 pixels, each requiring up to three bytes (24 bits) or color information, resulting in a total of 1.5 million bits to store one image! Thus, interesting connections to compression *can* be made here, but note that lossy compression and image formats like .jpg are covered in the next lesson.

Agenda

Getting Started (5 mins)

[View on Code Studio](#)

Objectives

Students will be able to:

- Use the Pixelation Tool to encode small color images with varying bits-per-pixel settings.
- Explain the color encoding scheme for digital images.
- Use the Pixelation Tool to encode an image of the student's design.
- Explain the benefits of using hexadecimal numbers for representing long streams of bits.

Preparation

- (Optional) Consider demonstrating the color pixelation widget instead of showing the video.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **Teaching Tips & Tricks Video - Encoding Images** - Video ([download](#))
- **Activity Guide KEY - Encoding Color Images** - Answer Key
- **Video Guide KEY for "A Little Bit about Pixels"** - Answer Key
- **Activity Guide KEY - Hexadecimal Numbers (optional)** - Answer Key

For the Students

- **A Little Bit about Pixels** - Video ([download](#))
- **Worksheet - Video Guide for "A Little Bit about Pixels" (optional)** - Worksheet [Make a Copy](#)
- **Encoding Color Images** - Activity Guide [Make a Copy](#)
- **Hexadecimal Numbers (optional)** - Activity Guide [Make a Copy](#)

Prompt: How might you encode colors?

Activity (40 mins)

Video: A Little Bit about Pixels
Color Pixelation Widget

Activity 2 (30-40 mins)

Personal Favicon Project

Wrap-up

Submit Favicon
Gallery Walk

Assessment

Extended Learning

- **Personal Favicon Project** - Activity Guide
[Make a Copy ▾](#)
- **Rubric - Personal Favicon Project** - Rubric
[Make a Copy ▾](#)

Vocabulary

- **Hexadecimal** - A base-16 number system that uses sixteen distinct symbols 0-9 and A-F to represent numbers from 0 to 15.
- **Pixel** - short for "picture element", the fundamental unit of a digital image, typically a tiny square or dot that contains a single point of color of a larger image.
- **RGB** - the RGB color model uses varying intensities of (R)ed, (G)reen, and (B)lue light are added together in to reproduce a broad array of colors.

Teaching Guide

Getting Started (5 mins)

Prompt: How might you encode colors?

Use a **getting started strategy** to address these questions (for ideas consult: **Teaching Strategies for the CS Classroom - Resource**)

- In the previous lesson we came up with a simple encoding scheme for B&W images. What if we wanted to have color?
- Devise an encoding scheme for color in an image file. How would you represent color for each pixel?
- How many different colors could you represent? Do you have a particular order to the colors?

Pair and share ideas



- Discuss some of the difficulties of representing color
- Compare and contrast the different schemes students come up with.

Discussion Goal

It is likely that many students will come up with an idea like making a list of colors and just assigning a number to each one. That is fine and reasonable.

Some students may already be aware of a numeric RGB color scheme. If they can describe that here, that is fine as well.

Regardless of their encoding, students should be thinking about the number of bits they will allocate to the encoding and how that will affect the number of colors that can be encoded.

Activity (40 mins)

Remarks

The way color is represented in a computer is different from the ways we represented text or numbers. With text, we just made a list of characters and assigned a number to each one. As you are about to see, with color, we actually use binary to encode the physical phenomenon of LIGHT. You saw this a little bit in the previous lesson, but today we will see how to make colors by mixing different amounts of colored light.

Video: A Little Bit about Pixels

- Show the video: **A Little Bit about Pixels - Video**
- Kevin Systrom, founder of Instagram, explains pixels and RGB color.
- (Optional) complete the video worksheet: **Video Guide KEY for "A Little Bit about Pixels" - Answer Key**

Discuss:

Following the video, you might address any questions (or give students time to complete the video worksheet)

Important ideas from this video include:

- Image sharing services are a universal and powerful way of communicating all over the world.
- Digital images are just data (lots of data) composed of layers of abstraction: pixels, RGB, binary.
- The RGB color scheme is composed of red, green, and blue components that have a range of intensities from 0 to 255.
- Screen resolution is the number of pixels and how they are arranged vertically and horizontally, and density is the number of pixels per a given area.
- Digital photo filters are not magic! Math is applied to RGB values to create new ones.



Color Pixelation Widget

- Distribute the Activity Guide: **Encoding Color Images - Activity Guide**
- Direct students to work in Code Studio.



- There are 3 tutorial videos that appear in Code Studio that guide students through the
- This activity guides students through a few levels to get used to representing pixel data with more than one bit per pixel. It works up to full 24-bit RGB color and will present hexadecimal as a convenient way to represent binary information for humans to read.

Guide: Encoding Color Images

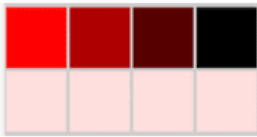
Each of the items below are presented to students on the activity guide and in Code Studio.

Step 1: 3-bit color



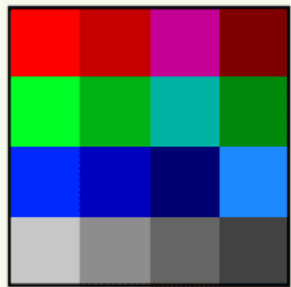
- **Color Pixelation widget tutorial video - Part 1 - Video:** How to use the pixelation widget to control color.
- Task 1: Fill in the last two pixels with the missing colors

Step 2: 6-bit color



- **Color Pixelation widget tutorial video - Part 2 - Video:** more bits per pixel for more colors
- Task 2: Experiment with 6-bit color

Step 3: 12-bit color and Hex



- **Color Pixelation widget tutorial video - Part 3 - Video:** Using hex to type bits more quickly
- Task 3: Experiment with Hex

Teaching Tip

If you are comfortable **you might consider demonstrating the pixelation tool** for each of the 3 steps in the activity guide rather than having students watch the tutorial video. Demonstrating might be a more efficient and interactive/engaging way bring students through each step.

Activity 2 (30-40 mins)

Personal Favicon Project

Students will create a 16 by 16 pixel personal favicon in RGB color using the Pixelation Tool. This project will likely require some time to complete, and should serve as a practice with hexadecimal numbers, metadata, and the underlying encoding of images in a raster file.

- Distribute the Activity Guide: **Personal Favicon Project - Activity Guide** and review the criteria for the project.
- Students will need a decent amount of work time to create their favicon. You might get them started in class and then assign it as homework.

Personal Favicon

(From the activity guide)

Directions

- Create a personal 16x16 favicon and encode it using the Pixelation Widget on the final level of this lesson in Code Studio.
- The image you make should represent your personality in some distinctive way. You will be using this favicon in future lessons and web sites that you make, so be creative and thoughtful.
- After you have finished your favicon, share it with others in the class by sending them the bits with the Internet Simulator Widget!



Requirements

- The icon must be 16x16 pixels.
- You must use the Pixelation Widget to encode the bits of color information.
- The image must be encoded with at least 12 bits per pixel.

Things to think about

- A simple design with a few basic colors is probably the best solution. How could you use more colors?
- Plan ahead: Sketch your design before starting to encode the bits. You might want to use a tool to help you draw small images. Suggestions:
 - Favicon Maker: <http://www.favicon.cc/>
 - Make Pixel Art: <http://makepixelart.com/free/>

Wrap-up

Submit Favicon

You should ask students to submit a .png version of their favicon, blown up to a larger size. And ask them to send you the bits that made up the image.

Gallery Walk

- With the images you can make a class favicon “quilt” by printing them out.
- And you can copy/paste the bits into the pixelation tool to verify that image is correct.

Assessment

Questions:

Content Corner

RGB color model - Additive Light

Computer screens emit light, so when you mix RGB colors, you are really mixing light together. This is counterintuitive for many students who have grown up mixing paints in school. When you mix paint it absorbs light.

It is illustrative to look at how you make black and white with paint vs. light:

- To make black: with paint, mix a full spectrum of colors together; with light, turn off all the lights.
- To make white: with paint, don’t use any paint (assuming canvas is white); with light, turn on all lights for a full spectrum of color.

This can make mixing colors a little bizarre too:

- With paint, mix full red and full blue to make Purple
- With light, mix full red and full blue to make Pink

The Pixelation Tool is in RGB mode, as long as the number of bits per pixel is a multiple of 3 (3, 6, 9, 12, etc.) This allows for the same number of bits to be allocated to each color channel. Other bits-per-pixel settings will set the image to grayscale, with more bits allowing finer control over the shade of gray.

Hexadecimal Numbers:

When working through the Activity Guide for the color version of the Pixelation Tool, students will be introduced to the concept of hexadecimal numbers, so-called because there are 16 unique symbols that can appear in each place value, 0-9, A, B, C, D, E, and F.

MISCONCEPTION ALERT

It is important to note that hexadecimal numbers are used to aid humans in reading longer strings of bits, but they in no way change the underlying data being represented. Instead, they allow us to read 4 bits at a time rather than 1, and so allow us to more easily parse binary information. Hexadecimal representation is NOT a form of compression, since the underlying binary representation is not changing at all. Rather it is a more convenient way of representing that binary information when humans need to read and interact with it.

You may wish to separately address this topic as a class. Students can practice with the **Hexadecimal Odometer** and can complete this **Hexadecimal Numbers (optional) - Activity Guide** if you deem more practice necessary.

- How many bits (or bytes) are required to encode an image that is 25 pixels wide and 50 pixels tall, if you encode it with 24 bits per pixel?
 - To help students understand how quickly the bit size of images expands as the image is enlarged, start with smaller numbers (5 X 10) and then incrementally increase the width and height to illustrate the concept.
- Imagine that you have an image that is too dark or too bright. Describe how you would alter the RGB settings to brighten or darken it. Give an example.

Extended Learning

- If you had to send your favicon using the sending bits widget, it would probably take a long time. Could you compress your image? How? Describe in broad strokes the kinds of things you could do.
- Read **Blown to Bits (www.bitsbook.com), Chapter 3**, Ghosts in the Machine, pp. 95-99 (Hiding Information in Images), then answer the following questions:
 - Besides hiding information sent to others, what other uses can steganography have for everyday users? For example, what uses would steganography have for an American businessman in China?

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **CL** - Collaboration
- ▶ **CPP** - Computing Practice & Programming
- ▶ **CT** - Computational Thinking

Computer Science Principles

- ▶ **1.1** - Creative development can be an essential process for creating computational artifacts.
- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **1.3** - Computing can extend traditional forms of human expression and experience.
- ▶ **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
- ▶ **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.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.
- ▶ **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 5: Lossy Compression and File Formats

Overview

This lesson is mostly an investigation of different kinds of file formats that exist in the real world. The lesson begins with students exploring a mock “lossy” text compression scheme as a way to learn about “lossy” compression. Then we do a jigsaw “rapid research” activity in which pairs of student research a real image, text, or sound encoding file format and determine what kind of compression it uses and the theory behind it. This lesson also sets the stage for the practice Performance Task (Encode a Complex Thing) that follows this lesson.

Purpose

The main purpose of this lesson is straightforward: understand what lossy compression is and when/why it might be used. It's mostly used in visual or audio formats where a loss in precision is undetectable to human eyes and ears. Beyond that we, want to continue to build students' skills and comfort with rapidly doing research online, reporting back, and verifying that the information they got was good. This is good life skill but will also serve students well for the Explore Performance task. The hope with this lesson is that students will have greater insight into these technical articles that they know a bit about the binary make up of things -- many of the image file format articles actually show the binary file format and what bits mean what.

In particular, students might discover, or you might point out that the BMP image format is basically the image encoding format used in a previous lesson, and that the GIF image format and ZIP compression scheme are versions of the text compression scheme we used as well. In the case of GIF, it uses a dictionary of up to 255 different colors and each pixel is stored as small number that refers to the dictionary.

Agenda

Getting Started (10 mins)

Quick Discovery: Lossy Text Compression
Lossless vs. Lossy compression

Activity

Jigsaw research.
Share results.

Wrap-up (5 mins)

Assessment

Assessment Possibilities

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Explain the difference between lossy and lossless compression.
- Identify common computer file types and whether they are compressed or not, and whether compression is lossy or lossless.
- Read a technical article on the web and sift its contents for targeted information.

Preparation

- ☐ Copies of File Formats Rapid Research worksheet for students

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **KEY - File Formats Rapid Research - Answer Key**

For the Students

- **File Formats Rapid Research - Worksheet**
[Make a Copy](#)
- **Lossy Text Compression App** - App Lab
- **Text Compression** - Video (lossy in second half)

Vocabulary

- **Lossless Compression** - a data compression algorithm that allows the original data to be perfectly reconstructed from the compressed data.
- **Lossy Compression** - (or irreversible compression) a data compression method that uses inexact approximations, discarding some data to represent the content. Most commonly seen in image formats like .jpg.

Teaching Guide

Getting Started (10 mins)

Quick Discovery: Lossy Text Compression

- With a partner, go to the **Lossy Text Compression App - App Lab**.
- Answer the following questions:
 - What is happening in the app?
 - Should this “count” as text compression? Why or why not?
 - What do you think “lossy” refers to?

Group discussion (brief)

- Verify that students saw the text was being reduced by keeping the first letter of every word and throwing out all the vowels.
- Get some opinions about whether it should count as text compression.
 - Opinions might vary, but it is true that the amount of text was reduced.
 - However, the work of reconstructing was left to human intelligence and intuition.

Lossless vs. Lossy compression

Remarks

- When we did text compression a few lessons ago, that kind of compression is known as “**lossless**” compression because in doing the compression, and in reconstructing the original text, nothing was lost; every character that was part of the original text could be recovered.
- “**Lossy**” compression -- yes, that’s the official word -- does something else. Lossy compression schemes are ones in which “useless” or less-than-totally-necessary information is thrown out in order to reduce the size of the data.
 - The lossy text compression app did that, and for the most part, you could probably make out what the text was supposed to say.
 - But it’s not perfect. If you saw the word “fd” it could be “food”, “feed”, “feud”, or “fad”. By reading it in context, you might know what it was supposed to be, but there’s no real way to know what the original word was. The original word is lost.

Transition:

We’ve been looking at image file formats. And we’ve also seen text compression. Both of those attempted to render perfectly every piece of information.

Both the image file format and the text compression scheme we used were lossless. Lossy compression schemes usually take advantage of the fact that a human is supposed to interpret the data at the other end, and human brains are good at filling the gaps when information is missing.

Activity

Today you and a partner will do some rapid research and reporting on some of the most common file formats. Use the web as your research tool.

Optional:

- The **Text Compression - Video (lossy in second half)** video from a previous lesson also discusses lossy v. lossless.
- You might show that part of the video here before diving in.

Jigsaw research.

- Distribute **File Formats Rapid Research - Worksheet**.
- Assign pairs or small groups one of the file format types listed in the table. It's OK if two groups research the same type.
- Each pair/group should research the file format assigned to it and fill in one row of the table.



Share results.

Ask for a volunteer to read what he found for the file type he was assigned. Ask if anyone else who researched that type has anything to add (or clarify) about what the first person said. Do this for each of the file types.

Wrap-up (5 mins)

- There was a question at the bottom of the worksheet that asked if you had ever heard of any other file type that you were curious about. What were those?
- Do a whip around and write what students say on the board. Types might include: .doc, .pdf, .docx, .mp4, .mov, .html, etc.
- All of these are specialized file formats in which some person or group decided how to organize (and in some cases, compress) the bits that make up the file type. There is nothing magical about them.

Assessment

Assessment Possibilities

Matching: Match the encoding type with the data type and compression. (In Code Studio.)

BMP	?	Compressed Image - Lossy
WAV	?	Uncompressed Image
JPEG	?	Compressed Files - Lossless
GIF	?	Uncompressed Sound
MP3	?	Compressed Image - Lossless
ZIP	?	Compressed Sound - Lossy
PNG	?	Compressed Image - Lossless (256 color limit)

Content Corner

- Students might discover, or you might point out, that the **BMP** image format is basically the image encoding format used in a previous lesson.
- The **GIF** image format and **ZIP** compression scheme are versions of the text compression scheme we used as well. In the case of GIF, it uses a dictionary of up to 255 different colors and each pixel is stored as small number that refers to the dictionary.
- The bit layouts of BMP and GIF should be understandable for students.

Teaching Tip

You can use any sharing strategy you like. The goal is for every student to have her file format table filled in for the first two columns (data type and compression type). Knowing how they work is also good, but some are rather complicated. It might have to be left a mystery.

Content Corner

The file extension you often see on a file (for example: myPhoto.jpg) is really just an indicator to the computer of how the underlying bits are organized, so the computer can interpret them. If you change the name of the file to myPhoto.gif, that **does not** magically change the underlying bits; all you've done is confuse the computer. It won't be able to open the file because it will attempt to interpret the file as a *GIF* when really the bits are in *JPG* format.

Extended Learning

- GIF and PNG are both lossless image compression formats. Which one is better?
- Read **Blown to Bits (www.bitsbook.com), Chapter 3**, Ghosts in the Machine, pp. 88-90 (Reducing Data, Sometimes Without Losing Information), then answer the following question:
 - Do you think the need for file compression will always be needed, considering the advances in data storage, the speed of computers, and speed of the Internet?
- Read **Blown to Bits (www.bitsbook.com), Chapter 3**, Ghosts in the Machine, pp. 90-94 (Technological Birth and Death), then answer the following questions:
 - Data formats are constantly changing. What challenges does this present for historians? For a given document, movie, or audio file, what are all the component pieces that need to be preserved along with it?
 - There is concern about Microsoft's de-facto ".doc" format. Do similar concerns exist for cloud services such as Cloud Data formats and Cloud APIs? What are some such APIs and what will the dangers be if those de-facto standards are adopted?

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

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

Computer Science Principles

- ▶ **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 6: Practice PT - Encode an Experience

Overview

In this 2-day lesson, students will design their own way to encode a personal experience (such as attending a party, playing a game, etc). The project begins with students doing some top-down design to figure out the components and subcomponents of an experience that are encodable as binary information. Students then select a portion of the experience to flesh out into a more detailed design. The project includes a written reflection questions similar to those students will see on the AP Performance Tasks. While students will complete this project individually, they will exchange feedback with a classmate at one point of the project.

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

In terms of Big Ideas in AP CSP this lesson is very much about Abstraction. Abstraction is the practice of temporarily ignoring details to focus only on the most significant or relevant portions of a problem. In the instance of binary information, we know that it's just a sequence of bits underlying even seemingly complex data structures, but we don't have to worry about that all the time. The ability to rely on high-level encodings and temporarily ignore lower-level details is the key to building the complex systems that we use and interact with every day.

The main purposes of this lesson are to:

- Put a bow on thinking about the digital (binary) representation of information.
- Practice creating an abstraction of their own design
- Practice writing in response to an open prompt
- Submit a written project

The course focuses so much on the digital representation of information because it is probably the most fundamental law of computing. So much of computer science is abstract, but it is all grounded in the laws and limitations of having to represent *everything* in binary. Internalizing this fact, and internalizing the levels of abstraction that result, we believe is central to what makes a person "good with computers" or "natural" with them. Similarly, being able to take a high-level, human idea and break it down into something that *could be* computed on is the fundamental essence of what it means to do computing work.

[View on Code Studio](#)

Objectives

Students will be able to:

- Break a complex piece of information down into its component parts such that it could be represented on a computer.
- Choose appropriate binary encodings for specific pieces of information and justify those choices.
- Complete a project with written response in a format similar to the AP® performance tasks.

Preparation

- ☐ Determine how you want to collect the project - digitally or on paper - and prepare to explain that to students
- ☐ Determine what (if anything) you want to print for distribution to students
- ☐ Review the "birthday party" example from the activity guide

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Students

- **Encode an Experience** - Activity Guide [Make a Copy](#)
- **Encode and Experience Templates** - Project Templates [Make a Copy](#)

Vocabulary

- **Abstraction** - a simplified representation of something more complex. Abstractions allow you to hide details to help you manage complexity, focus on relevant concepts, and reason about problems at a higher level.

Certainly, it helps a person explain and grapple with abstract computing ideas later on this course. For example, when programming you have to manage complexity in code by breaking things down into smaller procedures and routines. You have to make many choices about how to represent and store the information your program needs. The person who implicitly understands the difference between choosing to use the number 5 instead of the character "5" is more likely to make the right choice.

It's useful to practice the mechanics of producing a project like this on a tight timeline, especially one that includes both design and written elements. It's up to you how much you want to mimic the AP Performance Task process, but this relatively small-scoped project would be a very useful barometer for you and your students to see what it takes to take a project from initial understanding to actual submission of an artifact.

Agenda

Getting Started

Activity (2 days)

Introduce the Project: Encode an Experience

Distribute Activity Guide and Project Templates

Day 1

Day 2

Wrap-up

Assessment

Extended Learning

Teaching Guide

Getting Started

Remarks

Throughout this unit, we have been building layers of encodings on top of the foundation of bits.

First we learned to develop binary numbers, then ASCII text, then formatted text, and finally color images. High-level encodings are actually quite removed from the underlying bits from which they are made.

In the world of computer science, we call this **abstraction** - a mental tool that allows us to ignore low-level details when they are unnecessary. This ability to ignore small details is what allows us to develop complex encodings and protocols.

For example, the encoding for an image doesn't need to know that the RGB values in each pixel are actually 8-bit numbers, and an encoding for formatted text does not care how the ASCII symbols that comprise it are actually represented. As long as there is some way to encode numbers and characters, these high-level encodings will function.

We also do this with the layers of Internet Protocols that we designed. The DNS protocol doesn't need to care or know how the bits are physically being transmitted to and from it, or even how the request is routed to it. That all has to happen of course, but DNS only has to focus on the the higher-level task of mapping a domain name to a number.

Discussion Goal

Introduce the term “abstraction” (see paragraph below), and frame the coming project as an opportunity for students to develop their own layers of abstraction.

Teaching Tip

You do not have to give all of these examples if you think students get it, or you have already covered these ideas with some detail. The point is to convey the central importance that developing abstractions and understanding the effects and trade-offs that result from having to represent *all* information in binary has on pretty much anything having to do with computers.

Activity (2 days)

Introduce the Project: Encode an Experience

Today we are starting a project in which you are going to apply this idea of abstraction as you set out to build your own encoding and layers of abstraction for a complex piece of information.

The Big Question is: **How can you take something complex like a human experience, and break it down so that it *could* be represented in a computer?**



For example: how might you digitally encode the experience of attending a birthday party?

Distribute Activity Guide and Project Templates

- Distribute the **Activity Guide: Encode an Experience - Activity Guide**. The activity guide contains:
 - A description of the project
 - An example of encoding a birthday party
 - Submission guidelines and Rubric
- (Optional) Distribute the **Encode and Experience Templates - Project Templates**. You can collect student work on paper, or digitally free form, or use these templates. This document contains:
 - A page for drawing a diagram
 - A template for the detailed encoding
 - Space for the written reflection
- Review the project guidelines

- Review the "Birthday Party" Example
- Answer questions. Students will need time to understand the extent of the project and expectations for their final product.

Practice PT: Encode an Experience

A proposed schedule of the steps of this project is provided below, as well as more thorough explanations of how to conduct the various stages.

Day 1

- Review Project Guidelines and Rubric.
- Step 1: Choose the experience to encode: Brainstorm ideas.
- Step 2: Break down the topic -- Create the top-down diagram of the experience chosen.
- Consult a peer -- Present ideas and provide feedback to each other on progress thus far.
- Start Step 3: Detailed encoding

Day 2

- Finish Step 3: Develop a detailed encoding .
- Respond to reflection Question: There are trade-offs in the digital representation of information
- Submit project

Selecting an Experience to Encode

Provide students with time to develop a topic of their encoding. Encourage them to pick something they really like, are interested in, or know a lot about. The

most important thing is to pick a category of experience rather than a single instance. For example, "taking a trip to the grand canyon" is better than "that time I went to the grand canyon". You might need to encourage or remind students that ultimately we're trying to figure out if there is a way to encode this kind of experience in such a way that it could be represented in a computer.

Making the diagram

If you intend to collect this digitally this is an opportunity to have students use a digital tool to make the diagram. The diagrams in the activity guide itself were made simply with the drawing tool provided in Google Docs (Insert - > Drawing). Students could do this directly in the google doc templates provided. Alternatively you can print out the templates and have students draw it by hand.

Either way, in the interest of time *some amount* of hand-drawing and sketching is probably a good idea. Draw by hand first, then commit to digital form if you're going that route.

Choosing a Peer Reviewer

While the project will be completed individually, students should consult a peer during the process to receive feedback and brainstorm ideas. Either assign pairs or allow students to pick their own.

Peer Consultation After students have outlined their encoding, they should meet with their peer reviewers, present their work so far, and provide feedback regarding their progress. Potential Questions to address:

- Do you think this experience is a good choice? Why or why not?

Teaching Tip

If you are comfortable doing it, you should consider demonstrating the birthday party example from the activity guide for the class **BEFORE** distributing the activity guide to students.

Understanding the requirements of the project might be much easier to understand once students have seen a more fully worked example.

You could simply display the example from the activity guide, or do a more interactive, inquiry-based approach asking students for examples and drawing the diagram on the board as you go. It doesn't need to end up exactly like the one provided in the activity guide, but you could probably coerce students toward it.

Tip from the Field

A CSP teacher writes in to suggest the following riff on how to get students thinking in terms of data and top-down design.

Tweak the directions slightly to state that students are going to develop an online form which collects information from the user to help them encode their experience. This gets students thinking about these components as "containers" for storing specific instances of the component, rather than storing simply the name of the component itself. This is also a nice preview to chapter 2 of the unit where students start thinking about data collection and analysis.

Teaching Tip

If students are having difficulty developing their encoding, or with any part of the project, encourage them to talk with their peer reviewer. Develop the expectation that prior to asking you for help, students will have consulted one another.

- Have I identified the basic elements correctly? Did I miss any?
- Do you think I will be able to encode this data? What challenges do you think I will have?
- Do you have any suggestions for the next steps?

Creating the Detailed Encoding Students incorporate feedback from their peer reviewer to develop their encoding. This project can be completed entirely digitally, using the templates provided, but if students will be visually arranging their tables on a poster or on paper, they should be given access to needed supplies.

Written reflection and submission It is likely students will need a decent amount of time to write a response for the written portion of the project.

Communicate to students how they will be submitting their projects and ensure they have the tools necessary to produce and submit their projects.

Wrap-up

Self-assess: It can be a useful exercise to have students briefly assess themselves using the rubric they were provided at the beginning of the project. Ask them to identify points where they could improve, and remind them that this rubric is very similar in structure to the one that will be used on the actual AP Performance Tasks they will see later in the year.

Teaching Tip

Gallery Walk: If students create visual representations of their encodings, then you may want to provide time to present their work to their classmates in a gallery walk. Students should hopefully appreciate the diversity of interests and encodings created.

Feel free to exclude the wrap-up activity in the interest of time. Neither is an essential portion of the Performance Tasks and they are included only to provide a more natural conclusion to the project within your class.

Assessment

- **Rubric:** Use the provided rubric in the activity guide, or one of your own creation, to assess students' submissions.

Extended Learning

- Ask students to each examine a classmate's submission and identify potential additions or improvements to the encoding.
- Locate the most recent Performance Task Descriptions: <http://media.collegeboard.com/digitalServices/pdf/ap/ap-computer-science-principles-performance-assessment.pdf>
- Locate the most recent Performance Task Rubrics: <http://www.csprinciples.org/home/about-the-project>

Standards Alignment

CSTA K-12 Computer Science Standards (2011)

- ▶ **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



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: Introduction to Data

Overview

In this kickoff to the Data Unit, students begin thinking about how data is collected and what can be learned from it. To begin the lesson, students will take a short online quiz that supposedly determines something interesting or funny about their personality. Afterwards they will brainstorm other sources of data in the world around them, leading to a discussion of how that data is collected. This discussion motivates the introduction of the Class Data Tracker project that will run through the second half of this unit. Students will take the survey for the first time and be shown what the results will look like. To close the class, students will make predictions of what they will find when all the data has been collected in a couple weeks.

Purpose

This lesson introduces many of the lessons and themes that will run through the unit. Students are introduced to the Class Data Tracker and the fact that they will be collecting and analyzing their own data in a couple weeks. They also begin thinking about the many ways data impacts their lives and how it can be used. While the primary goal of this lesson is to get ideas and processes in place for the rest of the unit, there are many places where students can start asking interesting questions about where and how data is collected, who is collecting it, and how they are using it.

Agenda

Teacher Setup (10 mins)

Teacher Setup Guide for Data Tracker Project

Getting Started (20 mins)

Pop “Quiz”

Activity (25 mins)

Introduce Class Data Tracker survey.

Wrap-up

Assessment

Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Develop a hypothesis about student behavior over time, based on a small sample of data.
- Describe sources of data appropriate for performing computations.

Preparation

- ☐ Review the **Data Tools Resources** for this lesson (including Excel support)
- ☐ Teacher Setup for Google Forms (see Teacher Setup in Teaching Guide)

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **Class Data Tracker Setup Guide** - Google Form Setup (includes template)

[Make a Copy](#)

For the Students

- **How Much of a Left and Right-Brained Person Are You?** - [Link](#)

Teaching Guide

Teacher Setup (10 mins)

Teacher Setup Guide for Data Tracker Project

This lesson requires a one-time special setup in order to create a form for data collection with the students in your class. Once you have it setup you will use it for several weeks.

 **Please use: Class Data Tracker Setup Guide - Google Form Setup (includes template)**

Please click the link to see the full **Class Data Tracker Setup Guide - Google Form Setup (includes template)**.

In a nutshell the guide has you:

1. Make a copy of a Google Form (**short link to template**)
2. Share a link to the form with your students

There are also **notes on editing the survey questions** if you want to -- we chose the questions so that certain properties would emerge later on. If you want to change the questions just ensure that you'll get the same properties or later lessons might not work.

After the setup you should have:

- A copy of a Google form in your Google Drive
- A spreadsheet that will collect responses from the form in your Google Drive
- A link that students should use to fill out the survey

Students fill out this form **every day** or as frequently as possible over the next few weeks. We will look at the results more fully in Unit 2 Lesson 12. You can place the form and spreadsheet documents wherever you like in your Google Drive. They are yours now.

Getting Started (20 mins)

Opening Remarks

Transitioning to a new chapter. These remarks are meant to help you make a bridge between the *Encode and Experience* activity and this chapter about manipulating and visualizing data.

- The last project you did (Encoding an Experience) was about organizing and structuring digital data to represent complex information.
- You did it by thinking about bits.
- In reality we typically don't have to break digital data down all the way to bits in order to work with it, but understanding that digital data at its root is just bits **gives you insights** into working with larger data sets.
- We are about to embark on a new series of lessons where you will work with real data sets and learn how to use tools to explore and extract information and knowledge from the data.
- One way we think about it is learning how **tell stories with data**. We start today!

Pop “Quiz”

Before saying anything, point students to this online quiz and have them complete it: **How Much of a Left and Right-Brained Person Are You? - Link**

Share Results: Allow students to share and compare their percentages of left and right-brainedness. It should be mildly amusing. The point of this little exercise will be revealed after the discussion.

Remarks

Discussion Goal

Get students to start thinking about where they interact with and produce data in their lives, by looking at their past experiences with online quizzes and surveys, to bridge the gap to a long-term class data collection activity.

This unit will address the topic of data more deeply. In computing, we're interested in where data comes from, what structure or formats it comes in, and most importantly, what kind of knowledge or information we can extract from that data using computational tools.

Prompt:

"People say there is data all around us. What do you think that means? Brainstorm as many examples of data as you can think of."

For each one, try to answer:

- Who is generating the data?
- Where is the data being stored or saved? Who owns it?

Discussion

- Give students 2 minutes to jot down ideas before sharing with a neighbor.



- Do a whip-around to get ideas out in the air, perhaps writing them on the board.
- Student responses will vary widely and may be related to:
 - cell phone data plans
 - science experiments
 - GPS tracking
 - online shopping data
 - taxes or accounting info
 - sports data

Discussion Goal

- Make sure to point out that, for most of these examples, people are generating the data through their own actions, though sometimes they might not be aware of it.
- In most cases this data is stored somewhere else, and by someone else.
- The point to make is not necessarily a concern for privacy (yet) but simply the fact that there is lots of data gathered by individuals and organizations, which makes it possible to compute with/on.
- Some knowledge could be extracted from that data.

Transitional Remarks

Good, you identified all kinds of places that data comes from. In this unit we'll be looking at lots of those same examples and learning a bit about how to use, manipulate and visualize data with computational tools.

In Computer Science, sometimes we can have the computer itself generate data for us. Later in the course when we get to programming, we'll write programs that generate a lot of data.

But there are other kinds of data that can't be generated by the computer. In particular, data about people and how they act in the real world is hard to capture without just asking them. So that's what a lot of tools online do. They try to capture people's responses to things because the data, in aggregate, might contain useful information that could be extracted.

That "dumb" online quiz you took at the beginning of class is an example. These quizzes ask people to reveal things about themselves, their preferences, likes and dislikes. This is data! While these online quizzes are probably innocuous, some interesting things about people could probably be discovered if the data were analyzed.

As a class, we're going to do something similar...

Activity (25 mins)

Setup Reminder: Make sure you have prepared the Google form, and have the share link ready ahead of time. See notes above.

Remarks

As our first adventure into data, each of you is going to complete a short survey. Surveys are one of the best ways to collect data from people, and they are functionally no different from an online poll, funny quiz, or anything else that asks you for your opinion. We're going to use our own survey, so that we can collect and see all the data.

Activity Goal

Introduce the class data tracker project. The class will collect data about themselves so that students can see trends and patterns in the class's behavior over time.

Introduce Class Data Tracker survey.

Distribute: Share the survey link with your students and have them complete it once.

Display the Initial Responses: Once everyone has filled out the survey, show them a glimpse of the results. You can find the results from your survey by clicking the Responses tab next to the Questions tab at the top of the form you made.

QUESTIONS

RESPONSES

Display the responses on the board. Scroll through them, giving students a chance to see the data. Try not to get hung up on issues of formatting, like a student who responded “seven hours” instead of “7” or “7 hours.”

You may want to show the raw spreadsheet view instead of, or in addition to, the default “dump” of responses shown in the form.

Briefly Discuss: Have students look at the results from the survey and discuss what they notice.

- What do you notice?
- What was surprising?
- What do the results tell you about you and your answers?
- What other information would you like?
- What kind of questions would we need to ask to find out more information?

Explain:

You are going to complete this survey every day in class for the next several weeks. By the end, we should have several hundred entries. You’ve seen the questions and have taken a quick glimpse at the results. What do you think we might be able to find out in a few weeks?

Prompt:



*“Write down one or two **hypotheses** (predictions) about what we might be able to find out about our class, assuming that everyone fills out this survey every day for a few weeks.”*

Transition to wrap up.

Teaching Tip

If necessary, introduce (or review) the term hypothesis with your students. The CSP Framework has a learning objective that reads: 3.1.1 Find patterns, and test hypotheses about digitally processed information to gain insight and knowledge. [P4] We will come back to these hypotheses when we look at the data in earnest a few lessons down the line.

Many will have probably seen the word in a science class. The **Merriam-Webster Dictionary** says a hypothesis is “an idea or theory that is not proven but that leads to further study or discussion.”

Wrap-up

Share:



Do a quick share-out of students’ hypotheses about what the class data will show in a few weeks.

“What kinds of predictions did you make?”

- Student responses will focus on different aspects of the data.
- Anything related to time spent doing things outside of school and how it makes you feel is fair game.

Remarks

- Those are all interesting ideas.
- Many of them will require us to perform some computations on the results to find the answers, or spot other trends or patterns.

Discussion Goal

Foreshadow the class data tracker project and the rest of the Unit.

In student's hypotheses: try to focus on hypotheses that hinge on a relationship between two elements of the data For example:

- people who get more sleep tend to feel better
- predictions about trends or other patterns (e.g., I think most people will go to the movies to relax, but only on weekends).

- Over the coming weeks, we'll collect this data, and over that time, you'll learn some things about how to process and visualize data like this, so you can see for yourself what kinds of knowledge the data holds.

Welcome to data.

Assessment

TBD

Extended Learning

TBD

Standards Alignment

Computer Science Principles

- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.
- ▶ **7.3** - Computing has a global affect -- both beneficial and harmful -- on people and society.



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: Finding Trends with Visualizations

Overview

Students use the Google Trends tool in order to visualize historical search data. They will need to identify interesting trends or patterns in their findings and will attempt to explain those trends, based on their own experience or through further research online. Afterwards, students will present their findings to ensure they are correctly identifying patterns in a visualization and are providing plausible explanations of those patterns.

Purpose

The two main purposes of this lesson are:

1. Navigating and using a real data tool (Google Trends, see below) that is external to the course
2. Getting acquainted with talking and writing about data. In particular we want to:
 - Draw a distinction between describing **what** the data shows and describing **why** it might be that way
 - In other words: describe connections and trends in data separate from drawing conclusions.
 - We want students to get in the habit of separating the **what** from the **why** when it comes to talking and writing about data

As a bit of foreshadowing, the next lesson looks deeper into assumptions that people make about data that can lead to unintentional consequences and even exacerbate some of society's divisions.

Agenda

Getting Started (5 mins)

Survey Reminder

Introduce: Data Stories

Activity (30 mins)

Exploring Google Trends

Wrap-up (5-20 mins)

Share Data Stories

Assessment

Assessment Possibilities

[View on Code Studio](#)

Objectives

Students will be able to:

- Use Google Trends to identify and explore connections and patterns within a data visualization.
- Accurately describe what a data visualization of a trend is showing.
- Provide plausible explanations of trends and patterns observed within a data visualization.

Preparation

- ◻ Use the **Google Trends** tool to familiarize yourself.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Students

- **Exploring Trends** - Activity Guide
[Make a Copy](#)
- **Google Trends** - Link

Teaching Guide

Getting Started (5 mins)

Survey Reminder

Survey Reminder: Give students a few minutes to fill out the class tracker survey that you started in **Lesson 7 - Introduction to Data**.

Introduce: Data Stories

Remarks

Yesterday we started to collect data about ourselves so that we could learn about trends and patterns in our behavior. Today we're going to look at another tool that has collected a lot more data about you, me, and everyone else in this room. We're going to start thinking about how to **tell stories with data**, what data we need, and how best to use and present it.

Discussion Goal

Quickly connect today's activity to previous day (collecting data about ourselves) but then move into actually using Google Trends.

Activity (30 mins)

Exploring Google Trends

Distribute: Activity Guide - Exploring Trends - Activity Guide

As a class or individually students should read the summary at the top of the activity guide, which explains what information they will be looking at and how to use the Google Trends tool.



Students will use Google Trends a tool which visualizes data taken from Google search histories all around the world from the past several years.

- Students will work individually or in pairs to identify topics they wish to examine in greater detail.
- They should spend some time just exploring the tool, but eventually they will need to choose a single topic or set of topics that they will use to answer the questions that appear on the bottom of the activity guide.

Tell a Story

Students should find a trend or set of trends they think is particularly interesting or personally relevant and try to tell a story from the data they see. Students will write down:

- A description of what they were trying to look for
- An accurate description of what the visualization is showing
- A plausible explanation of why that trend might have happened.

Content Corner

Search trends are used in a variety of fields in order to understand what topics are most popular across the country and world.

- Search trends are also powerful predictors.
- Medical professionals may use this information to trace an outbreak of the flu.
- Businesses, media outlets, and advertisers keep a close eye on trending topics in order to understand how potential customers are thinking.

The fact that a global "conversation" is now happening online and computational tools exist to capture and visualize that conversation enables entirely new ways of identifying, understanding, and predicting patterns in culture and society at large.

Wrap-up (5-20 mins)

Share Data Stories

Once students have developed their charts and responded to the questions, have them share their “data stories” with each other.

Each group or individual should only take a minute or so to present their chart and story, after which the class might ask questions or add their own interpretations of the chart. Good questions include:

- Is the story the students told supported by the chart?
- Are there other ways to interpret the chart?
- Are there additional terms you’d also like to see shown on the chart?

Remarks

It’s exciting to be able to look at so much data in such a concise way, and it certainly feels like we’ve seen a lot of good stories here. As we start thinking more about how we use data, however, we’ll need to make sure that the assumptions we’re making about our data are correct.

(Optional) Collect: student activity guides. Students may want to revise their stories after the next lesson. Hold onto their activity guides or ask them to do the same so that they can update their assumptions if necessary after the next lesson.

Assessment

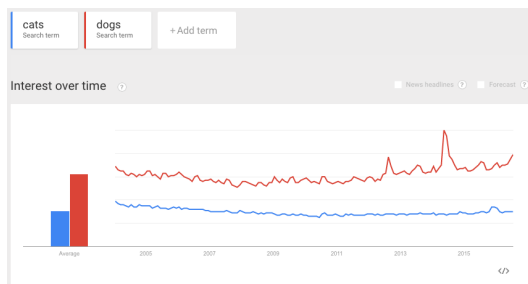
Assessment Possibilities

Score Activity Guides

- Collect and Grade
- Have students do peer review

Multiple Choice (in Code Studio)

At right is an image from Google Trends that plots Cats and Dogs.



Click to Enlarge

Choose

the most accurate description of what this data is actually showing based on what you know about how Google Trends works.

- People like dogs more than cats
- People search for "dogs" more frequently than "cats"
- There was a sharp increase in the dog population sometime between 2014 and 2015
- The popularity of dogs as pets is slightly increasing over time, while the popularity of cats is relatively flat

Free Response:

What would you investigate further?

Teaching Tip

Demonstrate the Tool: You may wish to demonstrate how to use Google Trends in front of the class before asking them to use it themselves. You could use the following steps:

- Ask students to recommend a search term to display.
- As a class, speculate as to what the trend might be showing.
- Add a second term to your visualization and discuss quickly what it might be showing.
- Demonstrate the ability to select different time periods, regions, etc.

Note: The front page of **Google Trends** shows a collection of stories compiled by others. To actually use the tool yourself, you need to enter text into the **“Explore” trends**. For more help with Google Trends, you can see the **Google support page on the subject**.

Discussion Goal

Provide students a chance to share their findings. Ensure students are accurately describing trends in the charts and that their stories or explanations for these trends are reasonable.

Teaching Tip

For sharing, you may want to: *Bring the whole class together.* Have individuals share with an elbow partner or in a small group.

It is likely that students are going to want to play with Google Trends individually, so having them share with a small group might be less intimidating. However, with small groups you will need to circulate and be vigilant about ensuring that students are emphasizing the right things and asking critical questions. You might find it easier to ensure that students see the right kinds of critical questioning as a whole class.

For the graph above, give a plausible explanation or hypothesis for the spike in dog searches that occurred between 2014 and 2015 that would lead to further investigation or research.

Standards Alignment

Computer Science Principles

- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.



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 9: Check Your Assumptions

Overview

This lesson asks students to consider carefully the assumptions they make when interpreting data and data visualizations. The class begins by examining how the Google Flu Trends project tried and failed to use search trends to predict flu outbreaks. They will then read a report on the Digital Divide which highlights how access to technology differs widely by personal characteristics like race and income. This report challenges a widespread assumption that data collected online is representative of the population at large. To practice identifying assumptions in data analysis, students are provided a series of scenarios in which data-driven decisions are made based on flawed assumptions. They will need to identify the assumptions being made (most notably those related to the digital divide) and explain why these assumptions lead to incorrect conclusions.

Purpose

In this lesson we look deeper into why we separate the **what** from the **why** when looking at data. The main purpose here is to raise awareness of the assumptions that we (all people) make when looking at data and try to call them out. Some of these assumptions lie hidden beneath the surface and we want to shed some light on them by looking at some examples from the news. This is a useful mode of reflection that will serve students well when doing reflective writing on the performance tasks.

Analyzing and interpreting data will typically require some assumptions to be made about the accuracy of the data and the cause of the relationships observed within it. When decisions are made based on a collection of data, they will often rest just as much on that set of assumptions about the data as the data itself. Identifying and validating (or disproving) assumptions is therefore an important part of data analysis. Furthermore, clear communication about how data was interpreted should also include an account of the assumptions made along the way.

Agenda

Getting Started (15 mins)

Survey Reminder

Video: Google Flu Trends

Google Flu Trends Failure

Activity (25 mins)

The Digital Divide and Checking Your Assumptions

Part 1: The digital Divide

Part 2: Checking Your Assumptions

Wrap-up (15 mins)

[View on Code Studio](#)

Objectives

Students will be able to:

- Define the digital divide as the variation in access or use of technology by various demographic characteristics.
- Identify assumptions made when drawing conclusions from data and data visualizations

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **KEY - Digital Divide and Checking Assumptions** - Answer Key

For the Students

- **Digital Divide and Checking Assumptions** - Activity Guide [Make a Copy](#)
- **Google Trends Video** - Video

Assessment

Assessment Possibilities

Extended Learning



Teaching Guide

Getting Started (15 mins)

Survey Reminder

Survey Reminder: Give students a few minutes to fill out the class tracker survey that you started in **Lesson 7 - Introduction to Data**.

Video: Google Flu Trends

Show this **Google Trends Video - Video** video, which describes how Google used the trending data students saw earlier in the unit to predict outbreaks of the flu.

Thinking Prompt: What are the potential beneficial effects of using a tool like Google Flu Trends?



Discuss: Students should share their responses in small groups or as a class. In general, responses should be centered around the following ideas.

Discussion Goal

- Introduce the idea that incorrect assumptions about a dataset can lead to faulty conclusions.
- Earlier prediction of flu outbreaks could limit the number of people who get sick or die from the flu each year.
- More accurate and earlier detection of flu outbreaks can ensure resources for combating outbreaks are allocated and deployed earlier (e.g., clinics could be deployed to affected neighborhoods).

Google Flu Trends Failure

Distribute:

Share one or more of these articles with the class. They detail why Google Flu Trends eventually failed and should serve as a basis of discussion for some of the potential negative effects of large-scale data analysis.

- **Wired - What Can We Learn from the Epic Failure of Google Flu Trends?**
- **NYTimes - Google Flu Trends: The Limits of Big Data**
- **Nature - When Google got flu wrong**
- **Time - Google's Flu Project Shows the Failings of Big Data**
- **Harvard Business Review - Google Flu Trends' Failure Shows Good Data > Big Data**

Thinking Prompt:

"Why did Google Flu Trends eventually fail? What assumptions did they make about their data or their model that ultimately proved not to be true?"

Discuss:

Once students have read one of the articles, **review the key points** from your article. The most important points about Google Flu trends can be found below:

- Google Flu Trends worked well in some instances but often over-estimated, under-estimated, or entirely missed flu outbreaks. A notable example occurred when Google Flu Trends largely missed the outbreak of the H1N1 flu virus.
- Just because someone is reading about the flu doesn't mean they actually have it.
- Some search terms like "high school basketball" might be good predictors of the flu one year but clearly shouldn't be used to measure whether someone has the flu.
- In general, many terms may have been good predictors of the flu for a while only because, like high school basketball, they are more searched in the winter when more people get the flu.

Teaching Tip

Reading Strategy: Most of these articles are somewhat more sophisticated in their analysis of the problems with Google Trends than is necessary for discussion. You may wish to read one of these articles together as a class and just touch on the key points outlined below.

- Google began recommending searches to users, which skewed what terms people searched for. As a result, the tool was measuring Google-generated suggested searches as well, which skewed results.

Transitional Remarks

The amount of data now available makes it very tempting to draw conclusions from it. There are certainly many beneficial results of analyzing this data, but we need to be very careful. To interpret data usually means making key assumptions. If those assumptions are wrong, our entire analysis may be wrong as well. Even when you're not conducting the analysis yourself, it's important to start thinking about what assumptions other people are making when they analyze data, too.

Activity (25 mins)

The Digital Divide and Checking Your Assumptions

Distribute: Activity Guide - **Digital Divide and Checking Assumptions - Activity Guide**

Part 1: The digital Divide

This activity guide begins with a link to a report from Pew Research which examines the “digital divide.” Students should look through the visualizations in this report and record responses to the questions found in the activity guide.

Discuss:

In small groups or as a class, students should discuss the answers they have recorded in their activity guides. Key points for the following discussion include:

- Access and use of the Internet differs by income, race, education, age, disability, and geography.
- As a result, some groups are over- or under-represented when looking at activity online.
- When we see behavior on the Internet, like search trends, we may be tempted to assume that access to the Internet is universal and so we are taking a representative sample of everyone.
- In reality, a “digital divide” leads to some groups being over- or under-represented. Some people may not be on the Internet at all.

Part 2: Checking Your Assumptions

Students should complete the second half of the activity guide. They are presented a set of scenarios in which data was used to make a decision. Students will be asked to examine and critique the assumptions used to make these decisions. Then they will suggest additional data they would like to collect or other ways their decision could be made more reliably.

Wrap-up (15 mins)

Discuss: In small groups or as a class, students should share their responses on the activity guide. Use this opportunity to reinforce a group understanding of what kinds of assumptions are being made to interpret the data. Some possible types of assumptions are listed below.

Discussion Goal

Students should practice identifying when data is being interpreted and what assumptions are made to do so, by sharing their work from the activity guide.

- The data collected is representative of the population at large (e.g., ignoring the “digital divide”).
- Activity online will lead to activity in the real world (e.g., people expressing interest in a candidate online means they will vote for him or her in real life).
- Data is being collected in the manner intended (e.g., ratings are generated by actual customers, instead of business owners or robots).
- Many other assumptions regarding data are possible.

Closing Remarks

Would anyone like to revise the explanation they gave for their google trends research in the previous lesson?

Has what you've learned today changed your perspective on the "story" you thought the data was telling?

In this course, we will be looking at a lot of data, so it is important early on to get in the habit of recognizing what assumptions we are making when we interpret that data.

In general, it is a good idea to **call out explicitly your assumptions and think critically about what assumptions other people are making when they interpret data.**

We may not become expert data analysts in this class, and even organizations like Google can make mistakes when interpreting data. Sometimes, the best we can do is just be honest with ourselves and other people about what assumptions we're making, correct our wrong assumptions where we can, and keep an eye out for the assumptions other people are making when they try to tell us "what the data is saying."

Teaching Tip

Leading the Discussion:

The answer key to the activity guide contains possible assumptions that could be made in each data scenario presented. In most instances, there will be many other possible assumptions. The focus here should be primarily on building a habit of checking assumptions before jumping to conclusions about trends in data.

Assessment

Assessment Possibilities

Score or peer review the activity guide

- There is an answer key to the questions listed in the activity guide.

Multiple Choice (also in Code Studio)

Which of the following is the most accurate description of what is known as the "digital divide".

The digital divide is about how...

- ...people's access to computing and digital technology increases over time through a process of dividing and growing quickly - it is often likened to the biological processes of cell growth
- ...people's access to computing and the Internet differs based on socioeconomic or geographic characteristics.
- ...people's access to computing technology is affected by the fact that newer devices that use new protocols makes it more difficult for them to communicate with older devices and technology
- ...the amount of data on the Internet is growing so fast that the amount computing power and time we have to process it is lagging behind

Performance Task-style reflection question (also in Code Studio)

Consider the following statement from the CS Principles course framework: 7.4.1C The global distribution of computing resources raises issues of equity, access, and power. Briefly describe one of these issues that you learned about in the lesson and how it affects your life or the lives of people you know. Keep your response to about 100 words (about 3-5 sentences).

Extended Learning

Share this article with students criticizing inaccurate or misleading ways of using Google Trends to write news stories. <https://medium.com/@dannypage/stop-using-google-trends-a5014dd32588#.dd7bifrl5>

Standards Alignment

Computer Science Principles

- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.
- ▶ **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 10: Good and Bad Data Visualizations

Overview

This is a pretty fun lesson that has two main parts. First students warm up by reflecting on the reasons data visualizations are used to communicate about data. This leads to the main activity in which students look at some collections of (mostly bad) data visualizations, rate them, explain why a good one is effective, and also suggest a fix for a bad one.

In the second part of class students compare their experiences and create a class list of common faults and best practices for creating data visualizations. Finally, students review and read the first few pages of **Data Visualization 101: How to design charts and graphs** to see some basic principles of good data visualizations and see how they compare with the list the class came up with.

Purpose

An important skill is the ability to critically evaluate information. As our world is increasingly filled with data, more and more the information from that data is conveyed through visualizations. Visualization is useful for both **discovery** of connections and trends and also **communication** - both are potentially aspects of the Explore Performance Task. In this lesson we will focus on the communication aspects of visualization.

Interpreting data visualizations is not typically thought of as a core computer science skill, but it is certainly an important one in an age of digital data. Computing has enabled massive amounts of information to be automatically collected, aggregated, analyzed, and visualized. Visualizations are useful in helping humans understand large amounts of data quickly, and they are useful communication tools when presenting findings about a collection of data. Not all visualizations are created equal, however, and in many cases the type of visualization used may distract or even mislead the reader.

As both creators and consumers of data visualizations, students need to be on the lookout for these common pitfalls. This will allow them to be savvier readers of data visualizations, and more effective communicators when creating visualizations of their own.

Agenda

Getting Started (10 mins)

Fill out class tracker survey

Think-Pair-Share: Why Make Visualizations?

Activity (25 mins)

[View on Code Studio](#)

Objectives

Students will be able to:

- Identify an effective data visualization and give justification.
- Collaborate to investigate and evaluate a data visualization.
- Suggest an appropriate visualization for some data.
- Evaluate a data visualization for effectiveness of communication.
- Identify a poor data visualization and give justification.

Preparation

☐ Print a copy of **Data Visualization**

Scorecard - Worksheet for each student.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Students

- **Data Visualization 101: How to design charts and graphs** - Link
- **Data Visualization Scorecard** - Worksheet [Make a Copy](#)
- Data Visualization Collection A & B (in Code Studio)

Review and Rate Data Visualizations

Debrief: What makes a good/bad data visualization?

Make a table of good v. bad visualization characteristics

Wrap-up (15 mins)

Data Visualization 101 discussion

Assessment

Assessment Possibilities

Extended Learning

Teaching Guide

Getting Started (10 mins)

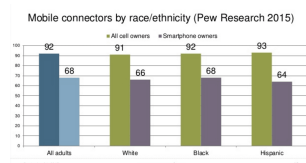
Fill out class tracker survey

Survey Reminder: Give students a few minutes to fill out the class tracker survey that you started in **Lesson 7 - Introduction to Data**.

Think-Pair-Share: Why Make Visualizations?

Remarks

Yesterday you looked at a bunch of data from the Pew Research Center that was all presented visually in graphs and charts. The question is: why? Why did they choose to make a bunch of charts and graphs rather than just showing the raw data itself?



Prompts:

Here are two related prompts to respond to:

- "Why did Pew Research choose to make a bunch of charts and graphs rather than just showing the raw data itself?"
- "List a few advantages and disadvantages (at least 2 for each) of using visualizations to communicate data"

Pair: Have students share with an elbow part.

Share: Draw out responses from the a whole class to generate common themes.



Here are a few notes to help guide the discussion:

Why visualize?"

- Student responses should focus on *communication*. You choose to make a chart because you think it's a better or more effective way to communicate information than using raw data.

Advantages / Disadvantages

- Advantages: pictures allow you to compare things more easily, easier to see trends or patterns, can focus on, or highlight, particular aspects of the data that are important
- Disadvantages: easy to mislead or miscommunicate, removes details that might be important or valuable, sometimes very dense - takes a while to study to understand what it means.

Discussion Goal

The goal here is to do a quick thinking prompt to activate prior knowledge about visualizations. There are two key points to draw out:

- Visualization is about *communication*. The goal of any data visualization is to transform data into useful *information*.
- There are *both* advantages and disadvantages to data visualizations - some of the disadvantages might not be as clear at this point but they will be after this lesson. The disadvantages mostly stem from the fact that human error and bias can be introduced when trying to communicate.

Activity (25 mins)

Remarks

Making even a small visualization may have been surprisingly challenging and varied.

In fact, even experienced data analysts can end up obscuring their message when they make data visualizations.

To better understand some of the skills we just read about, we are going to evaluate a collection of data visualizations to determine how well they communicate their message.

Review and Rate Data Visualizations

Pair: Partner students who will work through the worksheet together.

Assign: There are two different collections of data visualizations. Each pair of students should be assigned to evaluate one of **either**:

- Data Visualization **Collection A**

or

- Data Visualization **Collection B**

Links to the separate collections can be found in Code Studio

Distribute: Worksheet - **Data Visualization Scorecard - Worksheet**

(There is a link to the worksheet in Code Studio, but this is one you probably want printed out.)



Transition to Good and Bad Visualizations on Code Studio

The worksheet asks pairs of students to **collaborate in reviewing the data visualizations**:

- Give a rating from “Great” to “Horrible” for 15 different data visualizations.
- Choose the best (or favorite one) and explain why it effectively tells a story or communicates some data.
- Choose the worst (or a bad one) and explain why it’s ineffective.
 - **Optional:** Suggest (via a sketch) a better visual that could represent the same data.

Share:



After completing the worksheet, have each group share the best and worst image from their set with another group. Groups should focus on how they would fix the worst visualization they chose. Share and exchange ideas about different ways to visualize the data.

💡 Teaching Tip

Compare with Different Groups: Collection A and Collection B have different sets of visualizations. There are benefits to discussing with both a group that used the same collection of visualizations and with a group that used a different one. Time allowing, encourage groups to share their findings with groups who used both collections of visualizations.

Debrief: What makes a good/bad data visualization?

Have a discussion with two main parts...

Part one: Share out best/worst

- Ask student pairs to share the graphic they rated the worst and best.
- Focus on one or two and have all students look at it (bring up on a projector or have all students bring it up on screen).
- Ask students to justify or give reasons why they rated the graphic highly or poorly.

Part 2: Discuss graphic number 5 - The Changing Face of America

Graphic number 5 in both collections are two different displays of the same data. For reference, here are snapshots of both graphics.

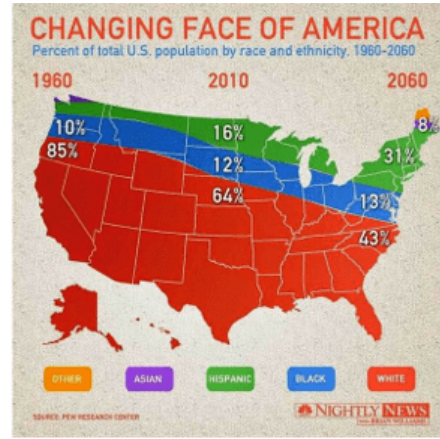
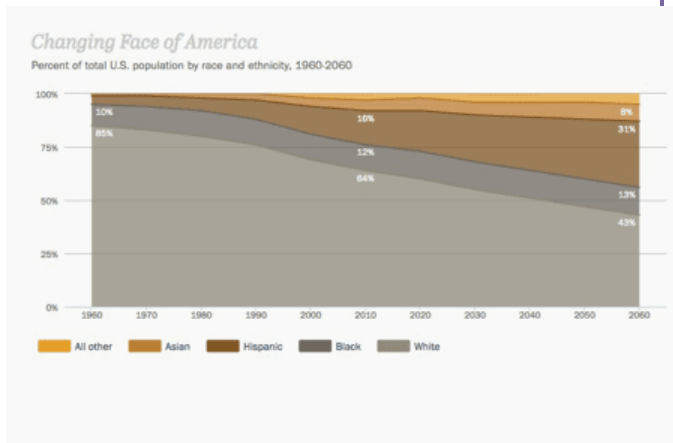
You might **display these** on the screen, **or have students look together** by sitting next to each other and opening up the graphic from each collection.

💡 Teaching Tip

To be creative with the share-out, you could:

- Have students vote publicly on each one in some way
- Have the groups that looked at Collection A all get together and figure out a best/worst to show to the other group(s) that looked at Collection B, and vice versa.

Discussion Goal



Prompt:

"Can everyone please look at graphic number 5 in their collection. It's called "The Changing Face of America."

Whip around:

- How did different groups rate this graphic?
- What data is presented?
- What is the difference between the two visualizations?

Students should recognize that the two graphics are attempting to represent the same data.

Make a table of good v. bad visualization characteristics

Prompts:

- Following the principles of good data visualization, which one would we say is better?
- What makes the good one good and the bad one bad?

As students respond, steer the discussion toward generating general characteristics of good and bad visualizations. Make a simple chart that everyone can see.

Something like this...

Good	Bad
<ul style="list-style-type: none"> simple easy to read a basic graph that makes a simple point ...etc... 	<ul style="list-style-type: none"> complicated confusing colors too much text ...etc...

Teaching Tip

Strategies for bringing out characteristics of good/bad visualizations:

- fill in a chart at the front of the room as students talk
- Have students write ideas on post-its and attach to a centrally located chart as basis of discussion
- Have students add ideas and comments to a shared online document

Wrap-up (15 mins)

Data Visualization 101 discussion

Remarks

We're going to be making some of our own visualizations of data very soon. To help us do that, we're going to look at some helpful tips for effectively communicating with data visualization.



Distribute: Data Visualization 101: How to design charts and graphs - Link. Students should read the first 4 pages of this document.



Discuss: What are the key take-aways from this guide?

Some key ideas that should come up:

- Choosing the right way to visualize data is essential to communicating your ideas.
- There are stories in data; visualization helps you tell them.
- Before understanding visualizations, you must understand the types of data that can be visualized and their relationships to each other.
- Certain chart types are right for certain situations, depending on the data.

Remarks

The Data Visualization 101 guide is a resource for you (students).

The rest of the guide goes into some specifics of different chart types.

You should keep this guide at your side as you review visualizations data, and when you develop your own in the future.

Further Discussion Points: What else did we learn about data visualization today?

- What are the benefits of visualizing data?
- Can we characterize common mistakes in visualizations to which we gave low ratings?
- Can we characterize common strengths in effective visualizations?
- Not all visualizations were charts; what other types are there?
- As you embark on making your own visualization, what do you want to keep in mind so that you can avoid rookie mistakes?

Assessment

Assessment Possibilities

- **Assessment Idea:** show students a visualization and have them analyze it, using the table of characteristics of good/bad visualizations to justify their opinion.

Performance Task-style reflection question

Choose the visualization that you thought was the best or worst (pick one) from the ones you saw in class and do the following:

- *Describe the visualization so the reader knows which one you are talking about (example: "Collection A #2 -- Average divorce rates in America")*
- *Say whether this was the best or worst visualization for you and why. Justify your opinion by citing principles of visualizations that you have learned about. Use the visualization 101 guide as a resource.*
- *Try to keep your response to around 100 words (about 3-5 sentences).*

Extended Learning

Discussion Goal

Establish some rules of thumb for visual displays of data. Students should recognize that some types of charts are more appropriate than others, depending on the nature of the data or the message the author is trying to convey.

Teaching Tip

Reading Strategies: Students can read individually, in partners, or as a whole class. The guide is not particularly long, but you'll want all students to have had a chance to look through those pages before the discussion. Let students know ahead of time that you'll be discussing the reading and ask them to pick one or two key points as they are going through.

If you want additional sources of data visualizations, consider the following sources:

- Daily Infographic: <http://www.dailyinfographic.com/>
- Infographics Archive: <http://www.infographicsarchive.com/>

Standards Alignment

Computer Science Principles

- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **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 11: Making Data Visualizations

Overview

Now that students have had the chance to see and evaluate various data visualizations, they will learn to make visualizations of their own. This lesson teaches students how to build visualizations from provided datasets. The levels in Code Studio provide a detailed walkthrough of how to use Google Sheets to create several different kinds of charts. While this lesson focuses on the Google Sheets tool, other tools may be substituted at the teacher's discretion, and MS Excel support is coming soon to the lesson.

The main activity teaches students to build different chart types (scatter, line, and bar charts) from a single data set. It should be emphasized to students that the purpose of this lesson is to explore and experiment with creating different types of visualizations, not to build the perfect chart. Students will have a chance to create and customize their own charts. At the end of class, students compare their custom visualizations with those of their classmates.

Purpose

Being able to create meaningful data visualizations is extremely important in order to effectively communicate information about large data sets. It's also important to be able to use visualizations to simply "look" at data that is too complex to make sense of by looking at the raw data alone. Any computer scientist working with data should have some skills and facility with producing visualizations of the data to get a sense of what it contains. Visualizing the data allows you to see patterns, trends or relationships you might otherwise not.

The most important piece of this lesson is not learning to create the prettiest chart; it's about using charts to "tell the story" of what's really going on in the data. Different charts are more or less appropriate for communicating this story, depending on the data. The point of having students explore different chart types is to help them build visualizations that reveal trends or connections in the data that are too hard to see by just looking at a data table in a spreadsheet.

Agenda

Getting Started

Survey Reminder

Using visualization to discover connections and patterns

Make a Quick Visualization

Activity

Make scatter, line, bar, and custom charts

[View on Code Studio](#)

Objectives

Students will be able to:

- Select the appropriate type of data visualization to discover trends and patterns within a dataset.
- Create a bar, line, and scatter chart from a dataset using a computational tool.
- Use the settings of a data visualization tool to manipulate and refine the features of a data visualization.

Preparation

- ▣ **Data Tools Resources** (including Excel support)

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Teachers

- **KEY - Making Data Visualizations Target Charts** - Answer Key

For the Students

- **Data Visualization 101: How to design charts and graphs** - Link
- **MovieRating_avgRatingByAgeByGender.csv** - Data Set ([download](#))

Wrap-up (10 mins)

Compare with a partner

Assessment

Assessment Possibilities

Extended Learning

Teaching Guide

Getting Started

Survey Reminder

Survey Reminder: Give students a few minutes to fill out the class tracker survey that you started in **Lesson 7 - Introduction to Data**.

Using visualization to discover connections and patterns

Prompt:

*"Do you **have to** use a computer to create a data visualization? What are some reasons that you need to use a computer to manipulate data?"*



Briefly Share and discuss responses

- Do a quick think-pair-share (or other strategy)

Transitional Remark

Taking data from its raw state to the point where you can create a meaningful visualization involves several steps. Today we're going to use visualization in attempt to discover things in the data we might not otherwise see.

It takes practice to create good visualizations. Today, we'll get our feet wet by learning to create charts using Google Sheets.

Discussion Goal

We want to motivate students' desire to create some visualizations on their own. Build on the "Good/Bad Visualizations" lesson. Some responses students might give:

- A large data set is too big to understand by looking at a table in a spreadsheet.
- Creating a data visualization with a computer is faster and more accurate than creating one by hand.

Make a Quick Visualization

Remarks

When trying to understand data, having a visualization, or picture of it, is often much more effective at communicating information than the raw data itself.

Making a good visualization of data is often challenging but can be fun and very creative, and we're about to start making our own. Let's try one, quickly.

warm up goal

This is intended to only be a brief activity to illuminate issues around making visualizations and how much variety there can be, and letting the students be creative and share with each other.

Scenario:

Here is some data: *On some survey 2,000 people were asked, "What do you do when you're bored?". Here are the most common responses by age group.*

age	most common response	number	out of
18 and under	texting	157	500

Teaching Tip

Please note: this data is completely fabricated and is only intended to serve the purposes of the warm up. It is intentionally slightly ambiguous. If students ask questions seeking clarification that's a good sign, but you might have to simply respond: "Well, this is the data we have".

There are no right or wrong answers here as long as students attempt to represent the data in a different way *somehow*.

age	most common response	number	out of
19-64	watching TV	247	1200
65+	reading	54	300
all ages	talking with friends	451	2000

For example: of the 1200 people surveyed between the ages of 19-64, 247 said "watching TV" which was the most common of any other responses to the question for that age group.

Prompt:

- "Take a few minutes by yourself and try to make a visual, graphical, explanation of this data. Try to communicate something about through drawing while remaining true to the results of the data."
- Give students 3-5 minutes to draw.

Compare and Discuss:

- Have students compare what they drew with an elbow partner and point out similarities and differences.

Prompts:

- "In this exercise what was challenging?"
- "What kinds of things were visually effective at communicating information?" <-- ALT: "What were the characteristics of the visualizations that effectively communicated this information visually?"

Activity

Make scatter, line, bar, and custom charts



Transition to Making Data Visualizations on Code Studio

- The "Activity Guide" for this lesson is all laid out in Code Studio.
- Put students into pairs and send them to Code Studio.
- The steps students go through are laid out below.
- Please note the purpose and teaching tips on this lesson for perspective.

While students are working, circulate the room to help and encourage.

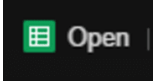



- Use **KEY - Making Data Visualizations Target Charts - Answer Key** which shows what students should be trying to create. It also explains a few common mistakes that students might make.

Teaching Tip

Remember the point here is not to make the prettiest chart, but choose the chart type that makes the most sense for the data you've got and the story you're trying to tell.

You can also point out to students that finding "no correlation" or "no relationship" is actually just as interesting as finding a strong correlation or relationship. For example, if you examine the difference between men and women in average rating of Star Wars, you will see virtually no difference! That's interesting!

Teacher Code Studio Reference

Teacher Code Studio Reference	
Students are asked to make a copy of the data set in their Google Drive. (Students must be logged into Google Drive for this step to work.) When they open the link to the CSV file, they can click the “Open” button next to the green Google Sheets logo, which will make a copy of the CSV in their personal Drive folder.	
Students follow step-by-step instructions to create a scatter plot showing the average movie rating by age of reviewer	
Students follow step-by-step instructions to create a line chart showing the average movie rating by age of reviewer, broken down by gender.	
Students follow step-by-step instructions to create a bar chart showing the number of ratings by age of reviewer, broken down by gender.	
Students experiment with creating their own charts on the same data set NOTE: they'll get a chance to explore many different data sets in the next lesson. It should be emphasized that the purpose of this part of the lesson is to freely explore the chart tool and discover connections in the data; students should not fixate on creating the perfect chart.	

Wrap-up (10 mins)

Compare with a partner

With partners or in small groups, have students discuss the following prompt. Once students have shared with each other, have students report back to the class about the charts they made and what they learned.

Prompt: What was the most interesting visualization you were able to create? What did it help you discover about the data?

Assessment

Assessment Possibilities

- Score or review a written response to the reflection prompt from the wrap up (also found in code studio)
- Make a simple rubric (a checklist basically) for the steps of the activity that students were supposed to go through:
 - Scatter Plot
 - Line Chart
 - Bar Chart
 - Optional: something on their own

Extended Learning

If you want additional sources of data visualizations, consider the following sources:

- Daily Infographic: <http://www.dailyinfographic.com/>
- Infographics Archive: <http://www.infographicsarchive.com/>

Standards Alignment

Computer Science Principles

- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **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 12: Discover a Data Story

Overview

In this lesson, students will collaboratively investigate some datasets and use visualization tools to “discover a data story.” The lesson assumes that students know how to use some kind of visualization tool - in the previous lesson we used the charting tools of a basic spreadsheet program. Students should be working with a partner but without much teacher hand-holding. Most of the time should be spent with students poking around the data and trying to discover connections and trends using data visualization tools. It is up to them to discover a trend, make a chart, and accurately write about it.

Purpose

Being able to look at large sets of data and use visualization as a tool for discovery is a common task that many people who work with data do on a daily basis. A computer scientist should have decent facility with using tools opening and browsing large datasets, and doing some cursory exploration to see what’s there. The computer scientist should be familiar enough with the tools to, over time, develop some instincts about data, how it’s collected, the kinds of formats it comes in, and how that affects what can or cannot be done to visualize it.

Agenda

Getting Started (10 mins)

Fill out class tracker survey

Visualization as a discovery tool

Quick Investigation of a sample dataset

Activity (40 mins)

Discover a Data Story

Wrap Up (10 mins)

Share your data stories

Assessment

Assessment Possibilities

[View on Code Studio](#)

Objectives

Students will be able to:

- Collaboratively investigate a dataset.
- Create a visualization (chart) from provided data.
- Identify possible trends or connections in a data set by creating visualizations of it.
- Accurately communicate about a visualization of their own creation.

Preparation

- ▢ Print a copy of **Activity Guide - Discover a Data Story - Activity Guide** for each student (or provide link)
- ▢ Print a copy of **Data Visualization 101: How to design charts and graphs - Link** for each student (or provide link) NOTE: this may have already been printed for lesson 10

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Students

- **Activity Guide - Discover a Data Story - Activity Guide** [Make a Copy](#)
- **Data Visualization 101: How to design charts and graphs - Link**
- **Rubric - Discover a Data Story - Rubric** [Make a Copy](#)
- **Data sets folder - Folder**

Teaching Guide

Getting Started (10 mins)

Fill out class tracker survey

Survey Reminder: Give students a few minutes to fill out the class tracker survey that you started in **Lesson 7 - Introduction to Data**.

Visualization as a discovery tool

Remarks

In the previous lesson, we learned how to use a data visualization tool to create a visualization. Sometimes data in its raw state is simply too big to be able to look at and derive any meaning. Even when the data is summarized in a table, it can be difficult to “see” what the data shows.

Today we're going to see how visualizing data can be a useful tool for discovery. In today's activity, you and a partner will investigate some sets of data on your own and use visualization to discover a connection or trend.

Quick Investigation of a sample dataset

For today's work there are several datasets for you to choose from.

- We're going to **take 5 minutes** to poke around in one of the datasets to see how it's structured.
- Then we'll come back together to get some terms straight before discovering further.



Go to Code Studio

1. Find the link to the **“Personality” dataset** and open the folder.
2. Find and open the **README file**.
3. Find and open the **rawData.csv file**.
4. Find and open **one other .csv file** - there are a few.

Discuss: What's in the folder for a dataset?

After students have had a few minutes to poke around, make sure the group understands what these files are.

You can use a think-pair-share or a simple whole group discussion to get the details out.

Ask the questions below, explanations are provided for you

“What's the README file?”

- Most datasets, when you download them, contain a README file.
- The README file is just a plain text document that gives some background information about the dataset, how it was collected, and what the column headings mean.
- The README is a good first stop when trying to understand exactly what a dataset contains.

“What's the rawData.csv file?”

- For the datasets we provide, each folder contains a “raw” dataset, which is the original data, as it was collected.
- Recall that **.csv** stands for “comma-separated values.” CSV is a common, plain text format for distributing datasets.

“What's in the other CSV files?”

- The other files are what we call **“summary tables.”**
- These are tables that were created by running some computations on the raw data to do things like count, average, sum, compare, and categorize the data in interesting ways.
- It is likely that these summary tables will be the data you use to create your visualizations.

Activity (40 mins)

Discover a Data Story

Pair: Put students in pairs or small groups to explore the datasets

Remarks

With your partner explore the datasets and choose one you'd like to learn more about. Make sure you

- Read the README to understand the raw data that was collected
- Look at the summary tables provided for your dataset.
- Repeat these steps with additional datasets
- Choose one to explore more deeply.



Discover a Data Story

Distribute: Activity Guide - **Activity Guide - Discover a Data Story - Activity Guide**

There is a link to this guide in Code Studio.

You may choose to have students make their own digital copies of this document and work on it there as well.

The activity guide asks students to:

- Pick a dataset
- Use visualization tools to “discover a data story”
- Prepare one (or two) to present
- Respond to prompts

Wrap Up (10 mins)

Share your data stories



Have students share their data stories with each other or with the whole class. A pair should:

- Show the visualization they made.
- Explain what it shows.
- Explain the possible story it tells.

Assessment

Assessment Possibilities

Use the rubric to score the activity guide

You may choose to collect the second page of the Activity Guide and score it using the **Rubric - Discover a Data Story - Rubric** provided.

Teaching Tip

A Note on distributing the Activity Guide

The first section of the activity guide contains the instructions above. It's suggested that students start exploring the datasets **before you distribute the activity guide** so they don't lose momentum.

You might choose to **assign the datasets to groups**. This cuts down on student choice, but might save time if students are taking a while to settle on which dataset they want to use.

While students are working:

- Remind students of the existence of the guide: **Data Visualization 101: How to design charts and graphs - Link**.
- Most of the students' time should be spent on working collaboratively to visualize data in different ways.
- Encourage and remind students that an “interesting” finding doesn't necessarily mean finding something world-changing or mind-blowing. The data is so big and hard to “see” that **simply making a clear chart that gives some kind of view into the data is interesting**

Teaching Tip

For student sharing, there are a number of different things you could do, depending on your needs and classroom dynamic. Here are a few suggestions.

- Have groups that used the same dataset share with each other.
- Have each group share with one or two groups who used a different dataset.
- Highlight one or two pairs' work by asking them to present to the whole class.

Note: Collecting and scoring the Activity Guide is optional.

- The intent of this activity is NOT to make a huge project out of it.
- The goal is simply to come away with some artifact that you might assess.
- It might be sufficient for students to share what they created in class rather than submitting the worksheet.

Personal Reflection: Collaboration

This prompt is also provided on Code Studio

(NOTE: The following is modification of one of the prompts given on the AP Create Performance task.)

Prompt: *Describe the development process of discovering your data story and creating a visualization. Describe the difficulties and/or opportunities you encountered along the way, and describe the collaborative process between you and your partner.*

Please limit your response to about 200 words.

Standards Alignment

Computer Science Principles

- ▶ **1.1** - Creative development can be an essential process for creating computational artifacts.
- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **1.3** - Computing can extend traditional forms of human expression and experience.
- ▶ **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 13: Cleaning Data

Overview

In this lesson, students begin working with the data that they have been collecting since the first lesson of the chapter in the class "data tracker." They are introduced to the first step in analyzing data: cleaning the data. Students will follow a guide in Code Studio, which demonstrates the common techniques of filtering and sorting data to familiarize themselves with its contents. Then they will correct errors they find in the data by either hand-correcting invalid values or deleting them. Finally they will categorize any free-text columns that were collected to prepare them for analysis. This lesson introduces many new skills with spreadsheets and reveals the sometimes subjective nature of data analysis.

Purpose

The main purpose here is have students independently apply some of the data manipulation skills (in spreadsheets) that they've learned over the past few lessons, to a new dataset that is relatively uncurated. This is the beginning of the process of "extracting knowledge from data": look at the data and clean it up so that you can process it using computational tools.

Using computational tools to analyze data has made it much easier to find trends and patterns in large datasets. When preparing data for this kind of analysis, however, it's important to remember that the computer is much less "intelligent" than we might imagine. Small discrepancies in the data may prevent accurate interpretation of trends and patterns or can even make it impossible to use the data in computation in the first place. Cleaning data is therefore an important step in analyzing it, and in many contexts, it may actually take the largest amount of time.

Agenda

Getting Started (5 mins)

Survey Reminder - Last one!

Discuss: Why we need to clean data

Activity (40 mins)

Clean Your Data

Wrap Up (5 mins)

Reflection: Is data analysis objective?

Assessment

Assessment Possibilities

[View on Code Studio](#)

Objectives

Students will be able to:

- Filter and sort a dataset using a spreadsheet tool.
- Identify and correct invalid values in a dataset with the aid of computational tools
- Justify the need to clean data prior to analyzing it with computational tools.

Preparation

- Prepare data collected from survey to share with students. Ensure that a "Teacher only" master copy is kept safely somewhere.
- Student partners will carry through the next lesson and Practice PT. You may wish to select these pairs beforehand.
- Review the **Data Tools Resources** for this lesson (including Excel support)

Teaching Guide

Getting Started (5 mins)

Survey Reminder - Last one!

Survey Reminder: Do one more entry in the class data tracker. You'll be using this data today. Give students a few minutes to fill out the class tracker survey that you started in **Lesson 7 - Introduction to Data**.

💡 Teaching Tip

If you need to prepare the data ahead of time, you might not be able to squeeze in one last entry.

Discuss: Why we need to clean data

🎤 Remarks

We have been collecting data about ourselves for several days. Now it's time to look at that data and see if we can find any interesting patterns or trends within it.

💬 Discussion Goal

Introduce the activity of the day and motivate the need to clean data before using it for analysis.

Prompt:

"Before we get started, what challenges do you think we'll encounter as we begin to peek into the data we've been collecting?"



Discuss:

Ask students to share their ideas with small groups or as a class.

- While there are presumably many challenges that will be mentioned, likely some of the comments will be related to the state of the data that was collected - in other words, how "clean" it is for analysis.

🎤 Transitional Remarks

There are many challenges associated with analyzing data. Today we're going to look at one that a lot of people don't often think about. When we collect data, it's usually "dirty," which means that, for one reason or another, it's not ready for analysis. We're going to investigate what this looks like and learn to use some tools to help us look at and "clean" the data.

Activity (40 mins)

Clean Your Data

Place Students in Pairs:

Students will clean and categorize their data in pairs. They will be using this data that they cleaned later in the unit for the practice PT.

Sharing the Data:

Pairs are going to need their own copy of the data collected from the survey. You should make your own master copy that will not be changed. To share the data with students, you can:

- Send a copy by email.
- Post a link to a Google Spreadsheet (make sure it's "View Only").
- **Note:** Instructions in Code Studio explain to students how they can "Make a copy" of a Google Sheet for themselves. If you are using a different spreadsheet tool, you should still share a copy.



Transition to Code Studio: Cleaning Data

Students will be guided through a series of activities that walk them through filtering, sorting, cleaning, and categorizing data.

The activity should be done in three parts.

1. Familiarizing Yourself with the Data:

Students learn how to sort and filter in a spreadsheet tool. There is no need yet to actually change any of the values. They simply should learn how these tools work in the spreadsheet tool you are using. Students can move on when they know how to filter and sort data.

2. Cleaning the Data:

Ignore “freeform text” responses for now -- for example, the “What did you do to relax?” column -- and focus attention on values that should be numeric or single words. Students will use sorting and filtering to find invalid values and will either fix or delete them. Students can move on when they have cleaned all “non-freeform” columns.

3. Categorizing Data:

Now focus attention on “freeform text” columns. Students will need to manually create new columns that categorize the inputs. This is a necessary step in order to perform computation with the data but it won’t feel very “algorithmic.” They will need to make choices, which is fine and will be addressed in the wrap up. Students can move on when they have cleaned all “freeform” columns by creating new columns of categories.

💡 Teaching Tip

- You may wish to work through these set of activities as a class.
- When using Google Sheets or other online spreadsheet tools, it is possible for two students to clean the same dataset at the same time.
- Students should consult with their partners as they make their categorizations. Remind them that the goal is to have something they could analyze or chart later.

Wrap Up (5 mins)

Reflection: Is data analysis objective?

Prompt: (Also found on Code Studio)

"In order to analyze data with a computer, we need to clean the data first. Based on your experience today, would you say that data analysis is a perfectly objective process? Why or why not?"



Discuss:

Students should share their ideas in small groups before discussing as a class. The key ideas to touch on are:

- Data cleaning usually requires a human to make decisions about the data.
- There often will not be one “right” way to clean the data and different people will do it differently.
- Any categorizing in particular is quite subjective.

NOTE: Make sure to save the cleaned up data:

Pairs should save their data somewhere they can both access it. They will be using it in the following lesson.

💬 Discussion Goal

Students should reflect on the often subjective nature of cleaning data. Even as data is being cleaned to be used by computers, there will often be a “human element” to how it is cleaned.

Assessment

Assessment Possibilities

- Score or review a written response to the reflection prompt from the wrap up “Is data analysis objective?” (also found in code studio)

- Make a simple rubric (a checklist basically) for the steps of the activity that students were supposed to go through:
 - Used sorting in a spreadsheet
 - Used filtering to help identify outliers for cleaning
 - **Added a column** to categorize some form of free form text.

Multiple Choice (also on code studio)

Which of the following is the most accurate statement about cleaning and filtering data?

- Using computing tools to filter and clean raw data makes it impossible to analyze or draw accurate conclusions
- Filtering and cleaning data is a fully automated process that should not require human input or intervention
- Filtering and cleaning data is a human process that does not require the use of computers
- Filtering and cleaning data is necessary to ensure that data is in a form that is better for computers to process

Standards Alignment

Computer Science Principles

- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.
- ▶ **7.1** - Computing enhances communication, interaction, and cognition.



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: Creating Summary Tables

Overview

In this lesson students learn how create their own summary tables from raw data. A summary table typically represents one or more aggregations (groupings of items) and computations that are performed on the raw dataset. In most spreadsheet programs, a summary table is called a pivot table. In the lesson, students learn how to make pivot tables in Google Sheets using a provided dataset. Then students turn to the data they've collected as a class and, with their partner, use pivot tables to investigate it further.

Purpose

Making a summary (pivot) table is often considered an advanced technique. Once you get used to it, however, it's an extremely powerful computational tool that is available in most spreadsheet software. The purpose here is to acquaint students with using such a tool and to expose this power. Also creating summary tables is a direct tie to the CSP Framework essential knowledge statement: *3.1.3C Summaries of data analyzed computationally can be effective in communicating insight and knowledge gained from digitally represented information.*

The other purpose here is that creating a summary table is a good example of making a computational artifact for the **Explore Performance Task**. For that performance task students might find some raw data while doing research and might create a new artifact that is a summary table of the data that reveals some interesting aspect of it. Using a tool like a spreadsheet to make summary tables let's you explore data in deep ways, quickly and easily.

Being able to manipulate data is an important skill for computer scientists. Being able to create summary tables from larger datasets represents a form of computational thinking. To make a good summary table, one must have a good sense of the data, be able to hypothesize about what might be interesting to look at, and then have the skills to use a computational tool to create it. While seemingly mundane, a spreadsheet is an extremely powerful tool for working with data. Understanding the features of a spreadsheet tool, and what kinds of computations it can perform, can save you a lot of time and energy from either doing such things "by hand" or writing your own program to do it.

Agenda

Getting Started

The need to create summary tables of raw data

Activity (90 mins)

[View on Code Studio](#)

Objectives

Students will be able to:

- Create a pivot table with at least one aggregation and one calculation when given a set of data.
- Describe the benefits a summary table has over a raw dataset.
- Collaboratively investigate a dataset by creating summary tables.
- Explain the meaning of a summary table they created.

Preparation

- ☐ Review **Data Tools Resources** (including Excel support)
- ☐ Familiarize yourself with the tutorials about making pivot tables in Code Studio.
- ☐ Ensure sutudents have access to the dataset they cleaned in the previous lesson.

Vocabulary

- **Aggregation** - a computation in which rows from a data set are grouped together and used to compute a single value of more significant meaning or measurement. Common aggregations include: Average, Count, Sum, Max, Median, etc.
- **Pivot Table** - in most spreadsheet software it is the name of the tool used to create summary tables.
- **Summary Table** - a table that shows the results of aggregations performed on data from a larger data set, hence a "summary" of larger data. Spreadsheet software typically calls them "pivot tables".

Transition to Code Studio

Making Pivot Tables Part 1 - The Basics

**Making Pivot Tables Part 2 - Manipulation and
Visualization**

Free play - make a summary table of the class tracker data

Wrap Up

Share and compare

Assessment

Assessment Possibilities

Teaching Guide

Getting Started

The need to create summary tables of raw data



Remarks

In the previous lesson we cleaned up the data we've been collecting. Now the question is: what can we do with it? Look at this table. It was created from the over 65,000 rows of data in the movie rating dataset we saw a few lessons ago....

Teaching Tip

As an alternative, you could show the little summary table above and ask the students:

- How long do you think it would take you to calculate the values in this table from the raw dataset of ~65,000 rows?

Based on what students know so far, they should guess relatively large amounts of time (dozens of minutes, or even hours). You can then reveal that today we'll learn how to make a table like this in roughly 10 seconds.

	Women		Men	
	<i>Number</i>	<i>Avg. Rating</i>	<i>Number</i>	<i>Avg Rating</i>
All Movies	16,716	3.54	48,819	3.53
Star Wars	102	4.23	284	4.37
Abyss, The	20	4.00	82	3.55

This is an example of a *summary table*. A lot of work and computation went into this. Notice that this is actually new data that was computed from the raw data. This is way beyond filtering and sorting. Computing this by hand for ~65,000 ratings (or writing formulas in a spreadsheet) would be pretty painstaking.

But we can use computing tools to create summary tables like this for us in a flash. Most data manipulation tools, like spreadsheets, allow you to quickly group, categorize, count, and average things. Making a summary table is a computational technique for exploring the data; let's try it.

Activity (90 mins)

Transition to Code Studio

Put students back together with their data cleaning partner.

- Students should go through the tutorial individually on their own computer, but should be seated next to their partner.

There are 3 levels in Code Studio that students go through.

Look at the levels in Code Studio for full details.

Here is a synopsis of what the students are being asked to do:

Making Pivot Tables Part 1 - The Basics

The first tutorial walks students through the entire process of making simple pivot table using a provided data set.

Here are the steps they go through.

Getting Started - Copy the Data

- A data set of movie ratings is provided.

Your First Summary Table

- Select all the data and make a new Pivot Table

Add Rows and Values to Your Table

- Organize the summary by listing each movie on its own row and show the average rating for it in each column.

Summarize by: COUNT

- Change the value from the Average rating the COUNT of the number of ratings for each movie.

Add Another Field to Values

- Add another column so that the table shows both the average rating and the count side by side.

Teaching Tip

As you circulate the room, keep in mind the **key ideas** we want students to have:

- Summary tables (pivot tables) provide a way to visualize data
- summary tables allow you to see things in the data you might otherwise not see.
- Summary tables allow you to manipulate and create new data.
- A summary table helps you look at your data in new ways.
- A summary table can be a first step toward a good visualization.

Making Pivot Tables Part 2 - Manipulation and Visualization

Students learn about a few more advanced features of making pivot tables and build up toward making a chart (visualization) based on a pivot table that they made, still using the movie rating data.

Here are the steps they go through:

Adding Columns

- Add more columns (values) to the table to show more stuff for each movie

Filtering Pivot Tables

- You can filter for values in a pivot table, just a like a spreadsheet - only show values that meet some criteria

The Next Step - Manipulating the Pivot Table

- Copy the pivot table to a new spreadsheet in order to manipulate the values further -- once you have the basic table you want, manipulating it further in "pivot table mode" can be cumbersome since the computer needs to recompute the data every time you do anything.

Moving on: Visualizing Summary Tables

- Make a chart of the pivot table you just made. See the examples in the tutorial on code studio.

Free play - make a summary table of the class tracker data

The entire lesson builds toward students being able to make a pivot table of their own data - the data they cleaned previously.

PLEASE NOTE: FREE PLAY is OPTIONAL

This free play should be considered optional or a bonus for this lesson.

If students have finished the tutorial they are ready to start the performance task in the next lesson.

Wrap Up

Question: Did anyone find the potential makings of a data story today?

Share and compare

- Have pairs of students share the pivot tables that they made with another pair, or with you, or with the whole class.
- This might be an opportunity for them to do peer-review of other groups' tables (see assessment below).
- Students should be able to describe what their table is showing, and preferably point out some insight they had.

Recall the key ideas of summary tables:

- **Summary tables (pivot tables) provide a way to visualize data.**
 - Yes, it's still a table, but by aggregating and summarizing information from a large dataset, summary tables allow you to see things in the data you might otherwise not see.
- **Summary tables allow you to manipulate and create new data.**
 - Even for our simple movies example here, the raw data didn't contain the average rating for every movie, or count how many ratings there were. We had to compute it, and the pivot table let us do that quickly and easily.
- **A summary table helps you look at your data in new ways.**
 - Think: how could data be grouped? What could be calculated? Once you know how to make a summary table you can begin to look at raw data and ask questions that you know might be possible to answer.
- **A summary table can be a first step toward a good visualization**
 - Often it's difficult to make a meaningful chart or graphic out of raw data. You often want to summarize it first, then chart it!

Foreshadow:

- In the next lesson, you and your partner will dig deeper into the data to find your own data story and make visualizations to tell it!

Assessment

Assessment Possibilities

Note: Formally assessing the pivot tables that come from this lesson should be considered optional. These partners will be making more pivot tables and charts from them for the Practice Performance Task in the next lesson.

Multiple Choice: (Also found in code studio)

Which of the following statements are true about pivot tables?

Select two answers.

- Pivot tables are used to quickly remove errors and inconsistencies from a dataset.
- Pivot tables are used to quickly perform aggregate computations and groupings on a set of raw data
- Pivot tables are used because they automatically detect and highlight potential trends or patterns in the underlying raw data
- Pivot tables are used to generate a summarized view of a large dataset which is helpful for gaining insight

Standards Alignment

Computer Science Principles

- ▶ **1.1** - Creative development can be an essential process for creating computational artifacts.
- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **3.2** - Computing facilitates exploration and the discovery of connections in information.



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 15: Practice PT - Tell a Data Story

Overview

For this Practice PT students will analyze the data that they have been collecting as a class in order to demonstrate their ability to discover, visualize, and present a trend or pattern they find in the data. Leading up to this lesson, students will have been working in pairs to clean and summarize their data. Students should complete this project individually but can get feedback on their ideas from their data-cleaning partner.

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

Students in this lesson will be telling their own story with a set of data about themselves. The hope is that using personal data will both motivate the exploration of the dataset and provide students with intuitions about the kinds of patterns or trends to explore. This Practice PT reflects many of the practices students will need to use on the actual AP® Performance Tasks, in particular the Explore PT. On that PT, students will need to create an artifact with a computational tool and explain both how it was created and what it is showing.

While students will not be required to create a chart or even necessarily visualize data for the PT, creating a data visualization would make for a strong computational artifact. This activity is designed to provide practice with one way to complete that aspect of the PT. Additionally students should leave this Practice PT familiar with many of the Learning Objectives related to the challenges of manipulating and analyzing data, as they will have now gone through the lifecycle of collecting, cleaning, analyzing, and visualizing data themselves.

AP® is a trademark registered and/or owned by the College Board, which was not involved in the production of, and does not endorse, this curriculum.

Agenda

Getting Started

Introduce the aims and goals of the Practice PT.

Activity (up to 3 days)

Create Individual Copies of the Data

Identify a Story

Visualize Your Story

Complete Written Responses

[View on Code Studio](#)

Objectives

Students will be able to:

- Create summaries of a dataset using a pivot table.
- Manipulate and clean data in order to prepare it for analysis.
- Explain the process used to create a visualization.
- Design a visualization that clearly presents a trend, pattern, or relationship within a dataset.
- Create visualizations of a dataset in order to discover trends and patterns.
- Draw conclusions from the contents of a data visualization.

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the Students

- **Tell a Data Story** - Rubric [Make a Copy](#)

Wrap Up

Submit Practice PT

Sharing Work

Assessment

Rubric

Teaching Guide

Getting Started

Introduce the aims and goals of the Practice PT.

Remarks

Throughout this unit we have been collecting data about ourselves in the hope that we'll be able to find some interesting trends and patterns in that data. Today we're going to finally be able to take a close look at the data we've collected. Your job will be to use your new skills at cleaning, summarizing, and visualizing data to "tell a story" using the data we collected. We hope that, with so many different perspectives in the class, a lot of interesting stories on the same dataset will emerge.

Activity (up to 3 days)

A proposed schedule of the steps of this project is included in the Teaching Tip as well as more thorough explanations of how to conduct the various stages.

Practice PT: Tell a Data Story

Distribute:

Distribute copies of **Tell a Data Story - Rubric** rubric and overview to students and review as a class. You may wish to read through the guidelines of the project together.

Alternatively, consider distributing the overview earlier in the unit to provide students an opportunity to preview and prepare.

Below are the steps of the PT that are laid out in the activity guide:

Create Individual Copies of the Data

Students will have been cleaning and summarizing a shared copy of their data thus far. Now they should make separate copies of their data to complete this project. In Google Sheets, one student will need to go to "File" → "Make a copy" In Excel, one student can email the other a copy of their cleaned data

Identify a Story

Students should already have some experience summarizing their data with a pivot table and visualizing it with charts. They should continue to iteratively use these tools to identify an interesting trend, pattern, or relationship within their data. Some good things to remind students:

- There's no need to tell a complex story. Simple relationships are still valuable to understand.
- The absence of a trend or pattern can still be interesting. If the amount of sleep you get doesn't have a clear impact on mood, that's interesting to know.

Visualize Your Story

Pacing Suggestion

Below is a suggested timeline for completing the PT entirely in class. It's possible that this could be done in a single day to get most of the work done, with written responses being done for homework.

Day 1

- Students create individual copies of their data.
- Students summarize and visualize data, looking for an interesting story to tell.

Day 2

- Students identify a story in their data.
- Students design a visualization showing their data story.

Day 3

- Students complete written responses and submit their Practice PT.

Students should once again refer to the **Data Visualization 101** guide for tips on how to make clear visualizations. Their chart will have accompanying explanations, but it should be able to “stand on its own” to communicate the story students have found. Some good things to remind students of:

- A fancy chart may actually be worse than a simple and clear one.
- Creating multiple charts is totally appropriate if they will better communicate the story.
- Experiment with different chart types. The chart type used to discover the story may not actually be the best one for visualizing the story.

Complete Written Responses

In the Practice PT, students will find responses modeled after those that will appear in the actual AP® Performance Tasks.

Wrap Up

Submit Practice PT

Students will need to submit their visualization and written responses. Direct students to check the rubric prior to submission to ensure they have all the necessary components.

Sharing Work

As an optional addition to this project, have students share their findings. The visualizations can be placed around the room for a gallery walk, added to a single shared folder, or presented to the class. This is a good opportunity to see how different groups cleaned and interpreted the same dataset.

Assessment

Rubric

Use the provided **Tell a Data Story - Rubric** rubric, or one of your own creation, to assess students’ submissions.

Standards Alignment

Computer Science Principles

- ▶ **1.2** - Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- ▶ **3.1** - People use computer programs to process information to gain insight and knowledge.
- ▶ **7.3** - Computing has a global affect -- both beneficial and harmful -- on people and society.



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**.