Unit 2 Lesson 1

Bytes and File Sizes

Resources

Name(s)_________________________________ Period ______ Date ________________

Activity Guide - Bytes and File Sizes

What is a byte? A byte is a unit of data that is 8 bits long. A byte is the standard "chunk size" for binary information in most modern computers

Larger Chunks of Data: On modern computers the amount of information we can create and store has grown so large that we need new units of measurement to describe the size of our data. Use these websites for your research.

- **Stanford University - CS 101 - Kilobytes Megabytes Gigabytes**
- **Computer Hope - How much is 1 byte, kilobyte, megabyte, gigabyte, etc.?** http://www.computerhope.com/issues/chspace.htm

Unit	Number of Bytes (approx)	Example of File Type or Data Measured in this Unit
Kilobyte (KB)		
Megabyte (MB)		
Gigabyte (GB)		
Terabyte (TB)		
Petabyte (PB)		
Exabyte (EB)		

How big are the files I use every day? Try to determine the size of files you probably use every day. You can either research these answers online or check the size of actual files on your computer.
- PC/WINDOWS: Right-click and choose "Properties"
- MAC: Ctrl+click and choose "Get Info"

File type	Size as # of pages, minutes, seconds, or dimensions	Size of file in Bytes, KB, MB, GB, etc.
page of plain text (.txt)	About 500 words, or 2500 characters	2500 Bytes, 2.5KB
.jpg image		
animated .gif image		
.pdf file		
Audio file as .mp3		
movie file such as .mov or .mp4		

Test your knowledge!

The first 3 questions here are from: Stanford University - CS101
You can check the answers there.

1. Alice has 600 MB of data. Bob has 2000 MB of data. Will it all fit on Alice's 4 GB thumb drive?

2. Alice has 100 small images, each of which is 500 KB. How much space do they take up overall in MB?

3. Your ghost hunting group is recording the sound inside a haunted classroom for 20 hours as MP3 audio files. About how much data will that be, expressed in GB?

Here are a few more.

1. A salesperson is trying to sell you a phone that has 16 GB of memory saying, "that's enough space to record an hour of high quality video!" This salesperson is probably wrong, but in which direction? Would you have more than enough memory or not enough?

2. Shakespeare's complete works have approximately 3.5 million characters. Which is bigger in file size: Shakespeare's complete works stored in plain ASCII text or a 4 minute song on mp3? How much bigger?

3. **Tricky**: Assume your Internet connection can transmit 1 million ***bits*** per second. Approximately how long would it take you to download 1 Terabyte of data? (Hint: first figure out how many bits a terabyte is, second be prepared to wait a long time).

Unit 2 Lesson 2

Text Compression

Resources

Name(s)_______________________________________ Period _______ Date _______________

Activity Guide: Decode this message!

What's the original message?

Below is an encoded message. It's not necessarily a secret message but it does need to be decoded. Study the clues and key to reconstruct the original message.

Encoded Message:

★listen_to☀rain_★on☀window_pane

Key:

☀	_the_
☂	tter_
☃	Pi☂
☄	Pa☂
★	☃☄☃☄

Original Message:

____

____

____

____

____

____

Name(s)____ Period ______ Date ______________

Activity Recap: Decode this message!

Here is a breakdown of how much the “Pitter Patter” message was compressed.

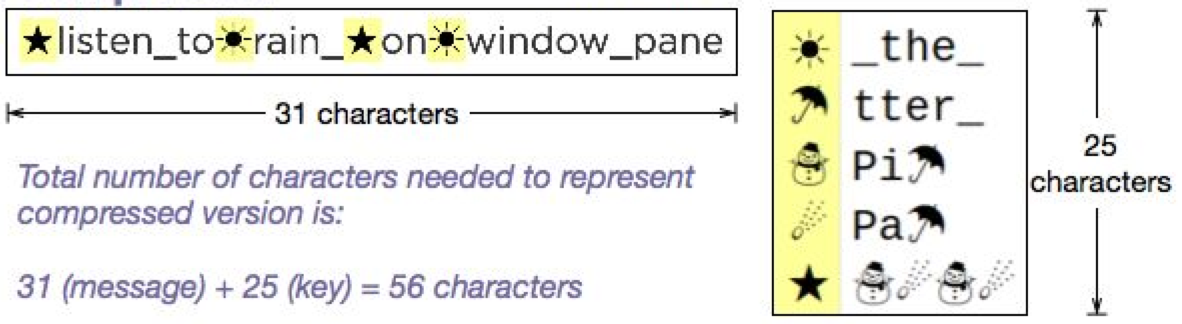
How Much? 40% compressed!

Original:	93 characters
Compressed:	56 characters
Difference:	**37 characters (~40%)**

Original Message 93 characters

Pitter_patter_pitter_patter_listen_to_the_rain_pitter_patter_pitter_patter_on_the_window_pane

Compressed 56 characters



Total number of characters needed to represent compressed version is:

31 (message) + 25 (key) = 56 characters

Name(s)___________________________________ Period ______ Date _____________

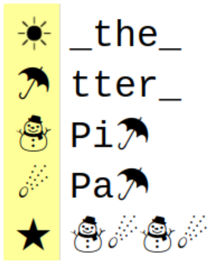
Activity Guide - Text Compression

Objectives

- Compress a piece of text using the Text Compression Widget
- Explain the factors that make compression challenging.
- Explain why the "best" compression is impossible or "hard" to identify.
- Create your own heuristic for compressing data.

Text Compression Tool

If you have not already done so, open up the Text Compression Tool in Code Studio and watch the video explaining how to use it. Then choose one poem and build a simple dictionary using the table below.



☀	_the_
☂	tter_
☃	Pi☂
☄	Pa☂
★	☃☄☃☄

Tips:

- Look for patterns that repeat and enter each in the "dictionary"
- Look for patterns of patterns - the dictionary can refer to itself (see right) making for a powerful amount of compression!
- Compare with others compressing the same text
- Try to develop a general strategy that will lead to "good" compression,

What's your best?

Copy and paste the best compression you made. (If you have a digital copy of this activity guide you could copy/paste a screenshot here, or just copy paste the text from the tool itself.)

Poem Name: ______________________________

Compressed Text:

Dictionary:

Compression Stats:

Reflection

Respond to these prompts.

1. What made compressing text hard to do?

2. Describe the thinking process you used in solving this challenge - what was your strategy for compressing the text. Could you explain it to someone who had never done this before?

Develop a Heuristic

Continue working on compressing your poem. As you do so, develop a set of rules, or a “heuristic” that generally seems to provide good results. Record the steps of your heuristic in some way that will allow you to exchange with another group. Make sure your rules are as clear as possible, so the other group will always know what to do.

Exchange Heuristics

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.

Test Heuristics

You should now have another group’s heuristic. Using the heuristic, attempt to compress the poems below. Record the compression rates you achieve.

Poem	**Compression Rate**
I Need a Dollar	
Pitter Patter	
A Tutor	
She Sells Sea Shells	
I Know an Old Lady	
Pease Porridge	

Record Your Conclusions

Use the data you collected to respond to the questions below.

1. 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?

2. 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?

3. If you send the compressed poem would your friend be able to read it? Why is the dictionary important?

Vocabulary

Compress: to decrease the number of bits used to represent a piece of information
Algorithm: a precise sequence of instructions designed to complete a task
Heuristic: a specific type of algorithm, usually used when exact solutions are difficult or impossible. Heuristics are generally simple to use and are designed to provide reasonably good results without guaranteeing a perfect solution.

Name(s)____ Period ______ Date _____________

Activity Guide - Text Compression Heuristics

This meant as a follow-on to the text compression activity.

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.

Develop a Heuristic

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.

Poem	Compression Rate

Record Your Conclusions

1. 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?

2. 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?

Unit 2 Lesson 3

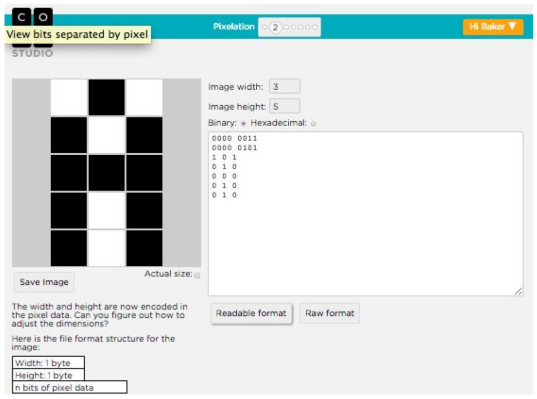
Encoding B&W Images

Resources

Name(s)_______________________________________ Period _______ Date ________________

Activity Guide - B&W Pixelation Widget

The B&W Pixelation Widget
The pixelation widget uses a file format as depicted below. For example, the 3x5 image of the letter "A," shown at right within the Pixelation tool, would be encoded as a simple stream of these bits, organized like this (color added for emphasis):


View bits separated by pixel

Pixelation

Hi Baker

Image width: 3

Image height: 5

Binary: ● Hexadecimal: ○

```
0000 0011
0000 0101
1 0 1
0 1 0
0 0 0
0 1 0
0 1 0
```

Save Image

Actual size:

Readable format

Raw format

The width and height are now encoded in the pixel data. Can you figure out how to adjust the dimensions?

Here is the file format structure for the image:

Width: 1 byte
Height: 1 byte
n bits of pixel data

$$\underbrace{00000011}_{\text{width}}\underbrace{00000101}_{\text{height}}\underbrace{000000011010100000010010}_{\text{pixel data}}$$

We can break it up into pieces like so:

purpose	size	example
width	1 byte	0000 0011
height	1 byte	0000 0101
pixel data	varies	0000000110101000001 0010

B&W Image File Format

	B&W Image File Format
metadata	Bits 0-7 (1 byte) = width
	Bits 8-15 (1 byte) = height
pixel data	Bits 16 - n = pixel data *0 = black (light off)* *1 = white (light on)*

Student Tasks

Log into Code Studio to get started with the pixelation widget. In code studio there are a few short tasks to help you get acquainted with the tool.

- **Create a small image:** Start by trying to recreate the 3x5 letter "A" depicted (shown above) using the pixelation widget.
- **Correct the error:** Oh no! An extra bit was inserted into an image during transmission! Track it down.
- **Make your own image of any size.**
  - Encode an image of anything you like.
  - Do not simply make an abstract pattern, like a checkerboard. It should depict something, perhaps your name written out, an icon or logo of some sort.
  - Optional: For fun, send your image bits to a friend using the Internet Simulator

When you are finished, complete the table below and answer the following questions:

Your image: Save your image and drag-and-drop or copy/paste it below.	**Your bits:** Copy/paste the bits that create your image below.

Questions

1. 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)?
2. 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?

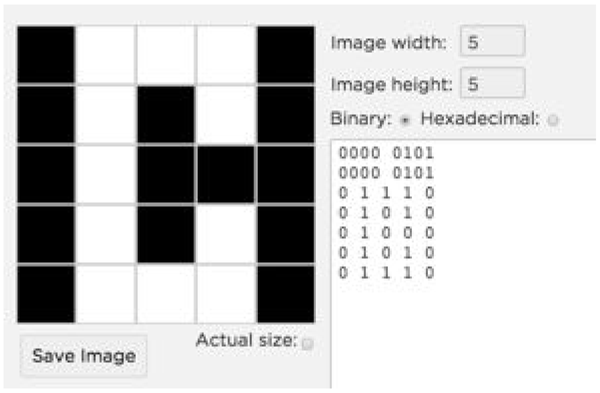
Name(s)_______________________________________ Period ______ Date _________________

Activity Guide - Extension: Magnify an Image

Directions

Add bits to magnify the image below (a small letter "C") by a factor of 2. The image should look the same, but twice as big! The bits are provided and a screen capture of the resulting image is shown.

Before	**After**

0000010100000101011100101001000001 01001110  Image width: 5 Image height: 5 Binary: ● Hexadecimal: ○
`0000 0101`
`0000 0101`
`0 1 1 1 0`
`0 1 0 1 0`
`0 1 0 0 0`
`0 1 0 1 0`
`0 1 1 1 0`	

What was challenging about magnifying the image? How did you interpret the instruction to "double the size" of the image?

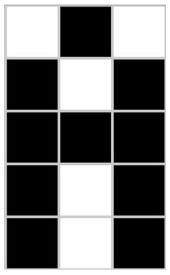
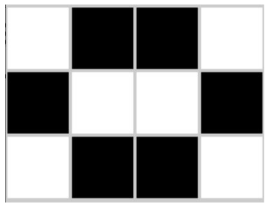
How many more bits did you need to double the size?

Name(s)_______________________________________ Period ______ Date ________________

Activity Guide - Invent a B&W Encoding Scheme

Develop an Encoding Scheme

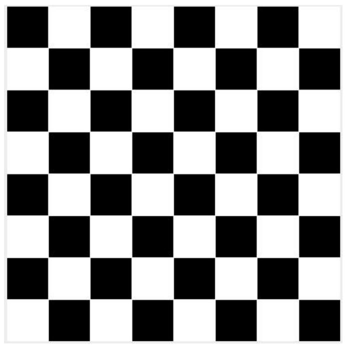
Look at the simple black-and-white images below. With your partner discuss how you might encode images like these in binary.

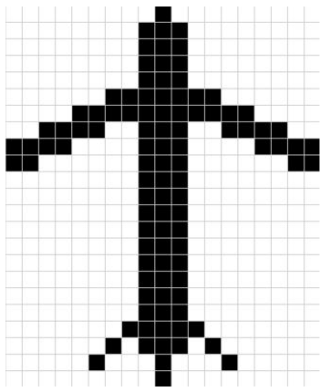



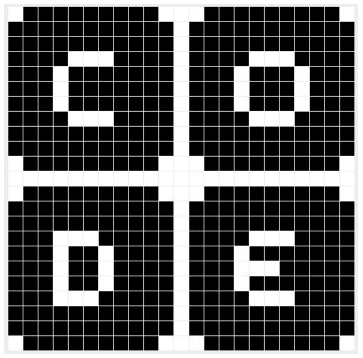
Record your ideas for your encoding in the space below. As you develop your encoding consider:

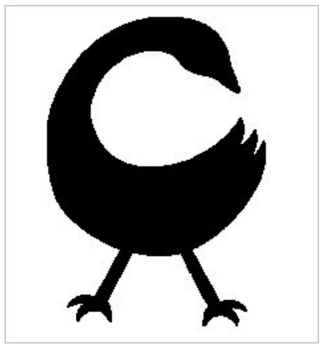
- What information will need to be included to reconstruct the image?
- How will that information be represented in binary?
- Try out your encoding with the sample images above. Is there any aspect of your encoding that is ambiguous or could lead to improperly structured images? Could you include more information in your encoding to solve this problem.

Now consider these larger images. Does your encoding scheme still work? Why or why not? What would you have to change to make it work?









Record any new ideas you have in the space provided below.

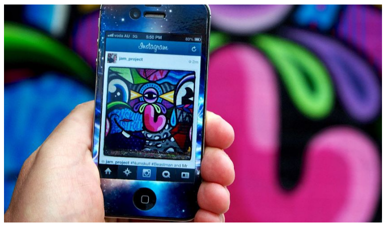
Unit 2 Lesson 4

Encoding Color Images

Resources

Name(s)____ Period ______ Date _____________

Worksheet - Video Guide for “A Little Bit about Pixels”



VIDEO OVERVIEW

In this video, Kevin Systrom, one of the co-founders of Instagram (a popular image sharing application), and Piper Hanson, a freelance photographer, explain pixels, RGB color, how image filters work, and that all image data is ultimately represented as bits, 1’s and 0’s.

Concepts

- Image sharing
- Digital images as data
- RGB color
- Screen resolution and pixel density
- Digital photo filters

THINK/DISCUSS

- Do pixels have to be physically represented as squares? What else could they be and why? Why are squares preferred?
- Think about your favorite image filters. How do you think the RGB values are manipulated to produce another image? Suggest possible algorithms.
- How do you suppose image sharing services could send the image data (bits) faster from one machine to another?
- Why can you never have a perfect digital reconstruction of a real-life image no matter how many pixels are used?
- There are 8 bits in one byte. How many bits and how many bytes are used to represent one pixel’s RGB components?

QUESTIONS

1. Why were images first broken down into pixels? What problem did this solve?

2. How many different values can be represented by one Red component of a pixel?

3. What are screen resolution and pixel density? How do they differ? Draw a diagram to support your answer.

4. RGB Values
    a. Why do RGB values range from 0 to 255?
    b. Can they be any other range?
    c. What happens if you increase the range?
    d. Is it possible to represent all of the colors in the world? Why or why not?

5. Describe a function that would take in any RGB value and double its intensity. What do you think happens if the input is 200, 220, 209?

6. Extra: Take out the calculator! How many different colors can be represented in the standard RGB color model?

Name(s)____ Period ______ Date _____________

Activity Guide - Encoding Color Images

Objectives:

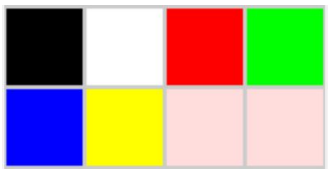
- Practice with the color pixelation tool.
- Use the Pixelation Widget to recreate specific colors.
- Recreate simple color images and describe the difference when using 6 or 12 bits per pixel.
- Encode colors with hex.

Directions:

Go to Code Studio - Find this lesson on Code Studio
The tasks below follow the guided sequence in Code Studio which has tutorial videos followed by specific tasks. The purpose of this activity is only to become familiar with the color pixelation tool, and to practice a little bit, in order to be comfortable enough to start the **favicon assignment.**

Step 1: 3-bit color

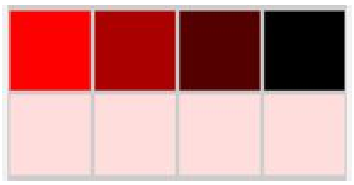
Tutorial Video: How to use the pixelation widget to control color.
Task 1: Fill in the last two pixels with the missing colors
Problem: fill in the last two pixels.



Optional Practice: change the pixel data so that the blue and yellow pixels exchange locations in the image. If you mess up, you can always re-copy-paste the bits from this example.

Step 2: 6-bit color

Tutorial Video: more bits per pixel for more colors
Task 2: Experiment with 6-bit color
Set bits/pixel to 6 and experiment with each of the R, G, and B values. At 6 bits/pixel, it means each color channel has 2 bits assigned to it, so there are 4 possible shades of each of R, G, and B. To start you are presented with the image at right.



Step 3: 12-bit color and hex

Tutorial Video: Using hex to type bits more quickly
Task 3: Experiment with Hex

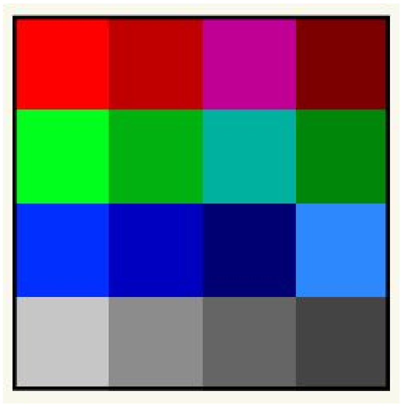
Make a 4x4 image with *bits-per-pixel set to 12* and the tool into *hexadecimal mode.*

NOTE: Using 12-bits-per-pixel (4096 colors!) is convenient because 12 bits maps to exactly 3 hex digits, one digit each for RGB. For example, to make a teal color (shown right) whose 12-bit value is: 001110101011 We can represent a 12-bit color in hex much more easily as hex: **3AB**. Here is the breakdown:

	R	G	B
binary	0011	1010	1011
hex	3	A	B

To make a darker shade, just choose a smaller hex digit for the channel of color you want to affect, such as 3A5 (which turns the amount of blue light down).

Recall: When writing a hex number we frequently preface it with a “#” symbol or “0x”. So hex: 3AB, would be written #3AB or 0x3AB. This is to avoid confusion when hex numbers have only decimal digits. 101 is a very different number from #101.



Create an image that looks something like picture shown at right in which

- row 1 - displays 4 different “reddish” colors (with red being the dominant color).
- row 2 - display should show 4 different “greenish” colors
- row 3 - should show 4 “bluish” colors
- row 4 - show only shades of gray

Questions:

1. Rewrite this string of numbers from raw format to readable format, and identify or label the sections of bits.

   000000100000001000000011010010101010

2. How many more colors are available with 12 bits-per-pixel than 6 bits-per-pixel?

3. What happens if you have you an image that has 6 bits-per-pixel and you change it to 12?

4. A digital artist comes up to you and says, “Help! I need a 12-bit color that’s just a little bit greener than **#79B.”** What would you suggest and why?

Name(s)_______________________________________ Period ______ Date ___________________

Activity Guide - Encoding Hexadecimal Numbers

Hexadecimal: a number system comprised of the familiar ten arabic numerals (0, 1, 2 … 9) as well as the first six English letters (A, B, C, D, E, F). Also referred to as a "base 16" number system, hexadecimal is used in the world of computing to help humans read and talk about large binary numbers. Since there are no symbols reserved for the numbers 10 through 15 in our familiar base 10 number system, the characters A through F are used to represent them.

Here are the first 16 numbers, in both binary and hexadecimal:

Number	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

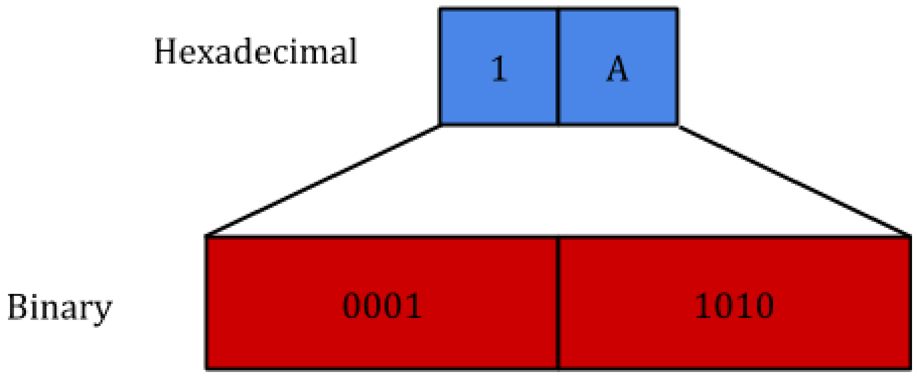
Converting from hexadecimal to binary is quite easy. One hexadecimal digit can represent any of 16 values. To do the same in binary you would need 4 bits, or 4 binary digits. Therefore, every hexadecimal digit can be replaced with its four bit equivalent in binary.

Converting Hexadecimal to Binary

1. Split the hexadecimal number up into two digits.
2. Convert each digit to decimal.

Example: Converting 1A in hexadecimal to binary

The first digit contains the number 1. The 4-bit binary representation of 1 is 0001.
The second digit contains "A" or 10. The 4-bit binary representation of 10 is 1010.
Therefore 1A in hexadecimal can be converted to 00011010 in binary.

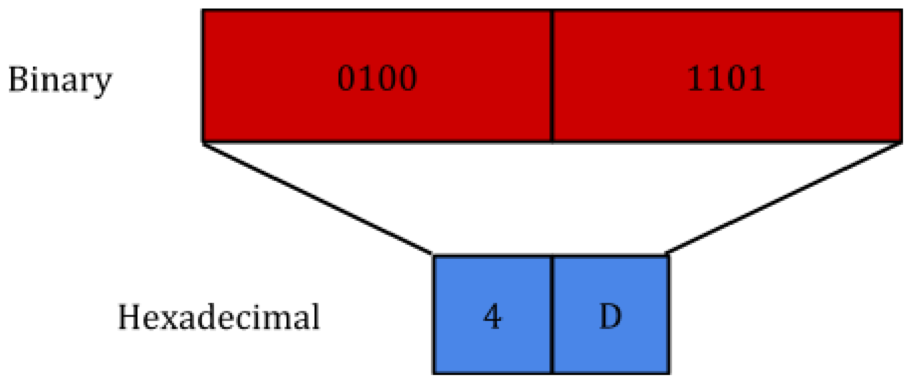


Convert: Convert the following hexadecimal numbers to their binary equivalents:

Hexadecimal	**Binary**
3	
A	
1C	
89	
DD0	
ABC7	

Example: Converting 01001101 in binary to hexadecimal

The first four bits are 0100 or “4,” which is just the character “4” in hexadecimal.
The second four bits are 1101 or “13,” which is just the character “D” in hexadecimal.
Therefore 01001101 in binary can be converted to 4D in hexadecimal.



Binary	Hexadecimal
0111	
1110	
01010001	
11000111	
000101011111	
101001011110	

Hexadecimal numbers are used because this conversion allows you to quickly communicate information about large chunks of binary digits. Perhaps the most common place one will see this is in naming colors to be displayed on a screen.

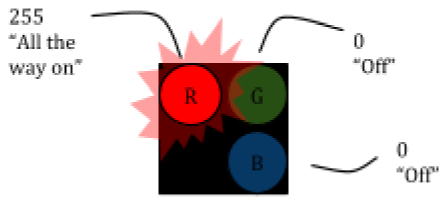
Representing Colors

While monitors typically can represent thousands, if not millions of colors, each individual pixel contains only three tiny lights, one each of red, green, and blue. Every color you see on a screen is created by setting different brightnesses for each of these little lights. Amazing!

Typically each of these little lights (red, green, blue) has its brightness dictated by a single byte. The largest number that can be written using one byte is 255, which indicates that a light should be turned to full brightness. 0 indicates that a light should be turned off.

Example: *If we wanted to make the reddest color our pixel could make, we would turn the red light all the way on, and the other two all the way off. This would be represented as*

red: 11111111 or 255
green: 00000000 or 0
blue: 00000000 or 0



A Single Pixel

But since these will usually be read as one 24-bit chunk, we display them below in the order they would be received, first the red byte, then the green byte, then the blue byte

Reddest Red *: 111111110000000000000000*

While this system makes a lot of sense for a computer to read, it can be tricky for a human to keep track of all those digits. This is where hexadecimal becomes really handy. The 24-bit sequence representing each pixel is 6 nibbles long, and so it can be represented as 6 hexadecimal digits.

Example: *Our reddest red can be written in hexadecimal as "#FF0000". To do the conversion yourself, just read the bits above in 4-bit nibbles and convert from binary to hexadecimal using the table above.*

Match the color to its hexadecimal equivalent:

🟩	#df0060
🟦	#ffffff
⬜	#00ffb6
🟩	#ff00ff
🟧	#000000
⬛	#0000ff
🟪	#ff9900
🟥	#00ff00

Name(s)____ Period _______ Date _____________

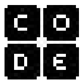
Activity Guide - Personal Favicon Project

Objectives

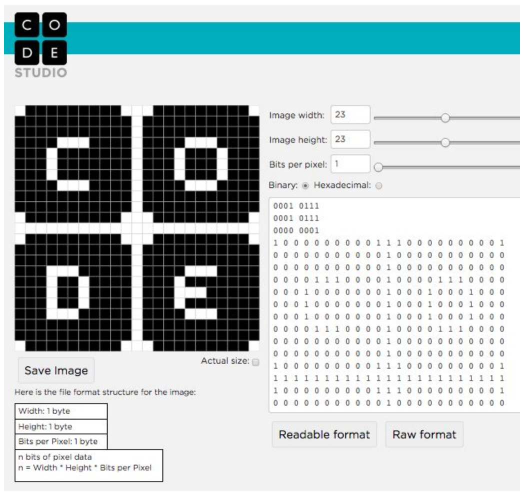
- Encode a 16 x 16 pixel image with at least 12 bits per pixel.
- Create and encode a color image of your own design.
- Explain the benefits of using hexadecimal numbers for representing long streams of bits.

Overview

A favicon is a small image, usually 16x16 pixels, that is typically shown in a web browser's address bar next to the title of the page or web address for a particular site. It is typically a small version of a company logo or some other symbol for the site. A favicon for Code.org is shown to the right.



Favicons are designed by artists and programmed into web pages by web designers. Below are some examples of favicons—you might recognize some!



```
0001 0111
0001 0111
0000 0001
1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
```

Here is the file format structure for the image:

Width: 1 byte
Height: 1 byte
Bits per Pixel: 1 byte
n bits of pixel data n = Width * Height * Bits per Pixel

Do a google search for ***Favicon*** and see what comes up.

Directions

1. Create a personal 16x16 favicon and encode it using the Pixelation Widget on the final level of this lesson in Code Studio.
2. The image you make should represent your personality in some distinctive way.
3. Optional: 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 & App Icon Generator: http://favicon-generator.org/editor/
    - Make Pixel Art: http://makepixelart.com/free/

Tip

Consider switching the Pixelation tool into HEX mode (base-16) instead of binary. This will enable you to more easily use a greater number of bits. If you use hex, you should consider using 12-bits-per-pixel (4096 colors!) because that maps to exactly 3 hex digits, which also maps easily to RGB—1 hex digit for each. For example, to make a teal color (shown right) whose 12-bit binary value is: 001110101011, we can represent it in hex much more easily as 3AB. Here is the breakdown:

	R	G	B
binary	0011	1010	1011
hex	3	A	B

To make a darker shade, just choose a smaller hex digit for the channel of color you want to affect, such as 3A5 (which turns down the amount of blue light).

Name(s)____ Period ______ Date __________________

Rubric - Personal Favicon Project

Rubric: Evaluate the success in creating a favicon according the following criteria.

Criteria	Yes	No	Comments
Favicon is 16 by 16 pixels			
Favicon is encoded in RGB color using at least 12-bits-per-pixel			
Favicon is a discernable image, and not merely a pattern			

Reflection Questions

1. What are the potential benefits of using a greater number of bits in designing your favicon? What are the potential drawbacks?

2. Your classmate claims, “Switching my favicon from binary mode to hexadecimal mode is be an example of compression”. Do you agree with classmate? Justify your response.

3. Explain what your personal favicon is a representation of and why you chose to create this particular image.

Unit 2 Lesson 5

Lossy vs. Lossless Compression

Resources

Name(s)_______________________________________ Period ______ Date _______________ **Unit 2 Lesson 5**

Activity Guide - File Formats and Compression

CODE

Formats Comparison

JPG, MP3, and PNG are three popular formats for representing information in a computer. These formats have some important key differences. Work with a partner to research the three formats and fill in the table below.

	JPG	MP3	PNG
List the sources (URLs) for your information here			
Type of file (image / audio / video)?			
What year was it created? Who (person / organization) created it?			
Is it free to use or do you need to buy a license?			
Lossy or lossless compression? How much can it compress?			
Advantages to this format or good situations for using it.			
Disadvantages to this format or bad situations for using it			

Assessment Questions

1. Which of the following is true of lossy and lossless compression techniques?
    a. Lossless compression throws away unimportant details that a human being will likely be unable to detect.
    b. Lossy compression is only possible on files that are at least one gigabyte in size before compression.
    c. Lossy compression techniques are no longer commonly used.
    d. Lossless compression is fully reversible, meaning the original file can be recreated bit for bit.

2. Which of the following is true of lossy and lossless compression techniques?
    a. Lossless compression is only used in situations where lossy compression techniques can't be used.
    b. Lossy compression is best suited for situations where some loss of detail is tolerable, especially if it will not be detectable by a human.
    c. Both lossy and lossless compression techniques will result in some information being lost from the original file.
    d. Neither lossy nor lossless compression can actually reduce the number of bits needed to represent a file.

3. In general, what are reasons that someone would choose to use lossy compression? When would they choose to use lossless compression? Write a brief response below.

Unit 2 Lesson 6

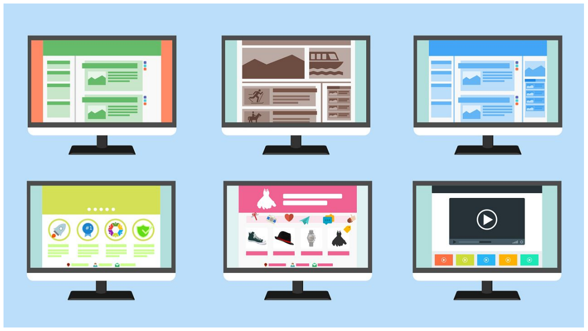
Rapid Research - Format Showdown

Resources

Name(s)_______________________________________ Period ______ Date _______________ **Unit 2 Lesson 6**

Rapid Research - Format Showdown

Project Overview: Format Showdown



Sharing text, images, video, and audio is increasingly how people communicate. For any of this to work we need shared file formats. Over the last few decades there have been many formats for each kind of file, and at any point in time there's usually many formats to choose from. Why so many choices, and which one is best?

For this project you will be researching a popular file format to understand what it does, how it works, and what makes it important. Then you will create a computational artifact sharing the results of your research and arguing why your format is the best!

One Pager and Computational Artifact

You will choose a file format that you wish to research. Based on your research you will create both a **one-pager** and a **computational artifact**. Important questions will include:

- The history of the format
- Key information about how it works
- Advantages and disadvantages of this format
- How it compares to other similar formats

General Process & Requirements

- Review the **one-pager template** (separate document) and the **rubric** (below)
- Choose a format using the guide
- Conduct your research by following the **research guide** (below)
- Complete the **one-pager**
- Design a **computational artifact** (instructions on one-page template)

Choose Your Format

Some Suggestions: Use the list below to help you begin your research. Just because you haven't heard of a format before doesn't mean that it isn't interesting or important!

Image Formats	Audio Formats	Video Formats	Other Formats
BMP JPEG 2000 GIF TIFF PDF RAW WebP HEIF	WAV AAC OGG FLAC ALAC WMA AIFF	MP4 OGG Quicktime AVI FLV	DOCX ODT

Want to find more?: This is just a short list of formats to get your research started. If you like you can skim this list of formats from Wikipedia to find more to check out. https://en.wikipedia.org/wiki/List_of_file_formats

Just pick one!: Don't spend more than 5-10 minutes deciding on a format. Pick one and then move on to the research guide below.

Conduct Your Research

You'll want to find **recently published articles** from **authoritative sources.** As you've already seen in the world of CS research may mean using sources like "Wikipedia" or even online forums to begin your search. From there, however, try to find your way to the organizations who actually created the various formats. They'll usually be the best source of information. You may encounter some technical language, but keep an eye out for familiar terminology and topics covered in this unit to be your guide.

Key Information to Find

- **Format Basics:** What type of file is it (audio / video / document / image etc.)? Where is (or was) this format normally used?
- **Format History:** Who created it? When? What else was around at the time? Is it still in use? Is it open source or do you need to pay for a license?
- **Compression:** Does it use compression? Lossy or lossless? What's the average file size? On average how much (as a percent) does it compress a file?
- **Advantages:** What situations is your format good at? What advantages does it present over similar formats?
- **Disadvantages:** What situations is your format bad at? Are there other formats that perform better?

Use the tables below to keep track of your information; you can also add more if you like. **You'll need to include at least 3 sources of information** but you can use more.

My Format: ____

Reference Name: **Year Published:**	**URL:**
Key Information	

Reference Name: **Year Published:**	**URL:**
Key Information	

Reference Name: **Year Published:**	**URL:**
Key Information	

Rubric - Format Showdown

Component	1	2	3	Score
Sources & Computational Artifact				
Sources	Research Guide includes references to fewer than three sources and the sources listed are not recent and authoritative.	Research Guide includes references to fewer than three sources or the sources are not recent and authoritative.	Research Guide includes references to at least three recent, authoritative sources.	
Computational Artifact	Artifact is incomplete and provides limited information about either the chosen format or "competitor" formats.	Artifact either fails to name "competitor" formats or provides limited information about why their format is the best. Information provided may also be irrelevant to comparing the indicated formats.	Artifact clearly names "competitor" formats and provides at least three pieces of information from research guide indicating why their format is the best.	
One-Pager				
Overview	Response is incomplete and does not provide a clear overview of the file format.	Response either is missing some components of the overview or does not clearly communicate some portions.	Response clearly explains the type of media, common usage of this format, and (if not overly complex) explains briefly how it works.	
History	Response is incomplete and does not provide a clear overview of the history of the format.	Response either is missing some components of the overview or does not clearly communicate some portions.	Response clearly explains the format history including who created it, when, similar formats, whether it's still used, and whether it is open source or licensed.	
Compression	Response is incomplete and does not provide a clear overview of compression within the format.	Response either is missing some components of the overview or does not clearly communicate some portions.	Response clearly explains compression within the format including whether it uses compression, is lossy/lossless, typical file sizes, and typical percent compression.	
Similar Formats and Advantages / Disadvantages	Response is incomplete and does not provide a clear overview of the advantages / disadvantages of the format.	Response either is missing some components of the overview or does not clearly communicate some portions.	Response clearly indicates similar formats as well as the advantages and disadvantages of the chosen format.	

Name(s)_______________________________________ Period ______ Date _______________ **Unit 2 Lesson 6**

File Format Showdown One-Pager Template <change this to your title>

***Note:** All text in Italics, including this text, is intended to be replaced by your responses, and deleted once you've completed your one-pager.*

Format Overview

What is your format? What kind of media (images/video/music/etc?) is it for? What does a typical file in this format contain? Where are/were common places this format was used? If it is not overly technical provide a brief description of how it works.

Format History

Who created it? When? What similar formats were around at the time? Is it still in use? Is it open source or do you need to pay for a license?

Compression

Does it use compression? Lossy or lossless? What's a typical file size for this format before and after compression? On average how much (as a percent) does it compress a file?

Similar Formats and Advantages / Disadvantages

What other similar formats might someone use for this kind of information? What situations is your format good at? What advantages does it present over similar formats? What situations is your format bad at? Are there other formats that perform better?

Sources

List all websites that you used to find any information you wrote here. Include the permanent URL. Identify the author, title, source, the date you retrieved the source, and, if possible, the date the reference was written or posted. You should number your sources, here is a template you can follow:

[1] Author's Last name, First name. "Title of Web Page." Title of Website, Publisher, Date, URL. Date retrieved.
[2] Author's Last name, First name. "Title of Web Page." Title of Website, Publisher, Date, URL. Date retrieved.
[3] ….

Computational Artifact Instructions

Design a video, audio recording, graphic, etc. that makes the case for why your format is the best among other similar formats. Your must only cite accurate information from your research guide above, but you're allowed to put your own spin on it. Think of this like a campaign ad or advertisement for your format.

In this section of your one pager include:

- *A list of other formats you'd specifically like to target in your computational artifact*
- *Three key points you'd like to make in your artifact explaining the benefits of your format*