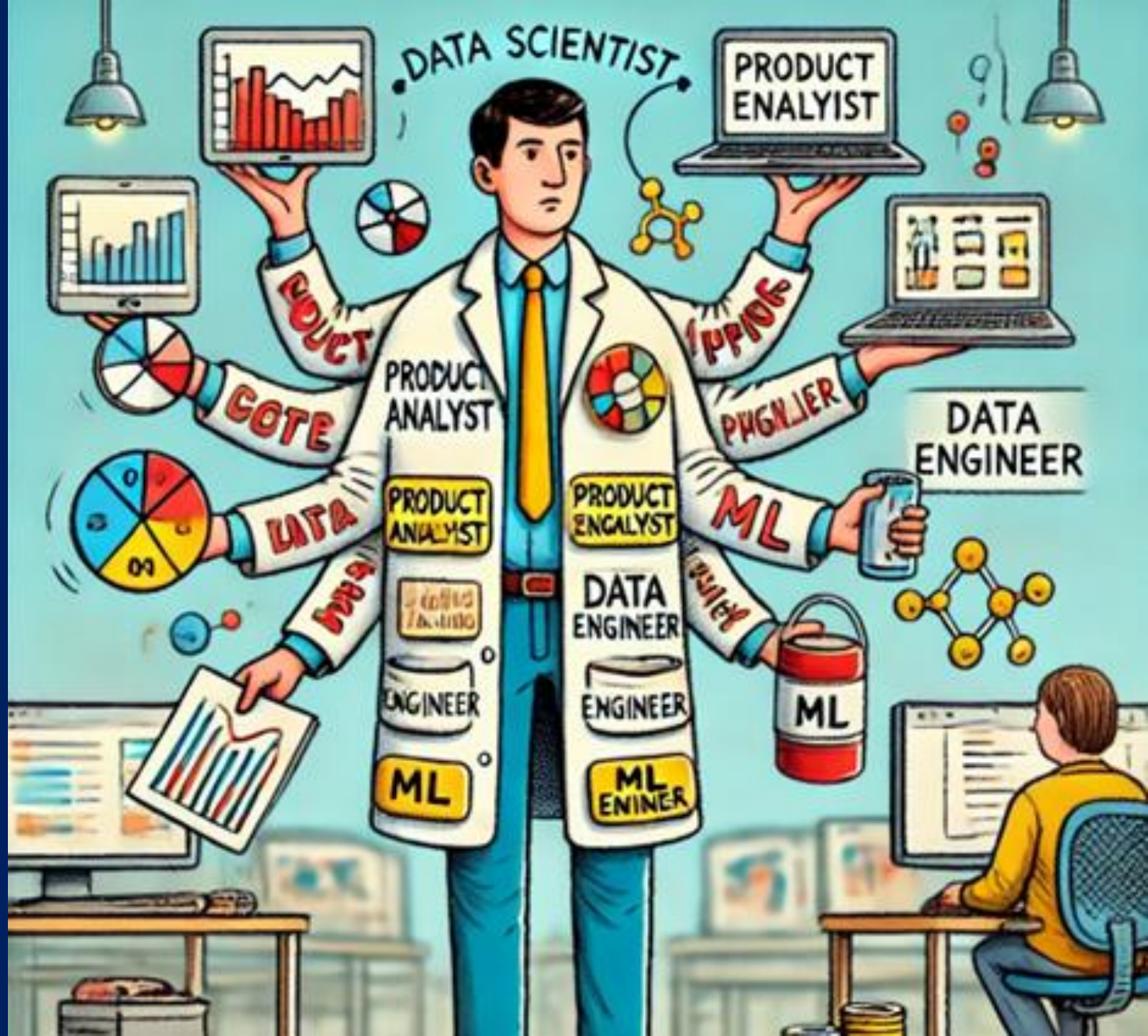
CS 51 Computer Science Principles

Module 3: Data, Internet, Computer and Programming

Unit 2: Data and Information

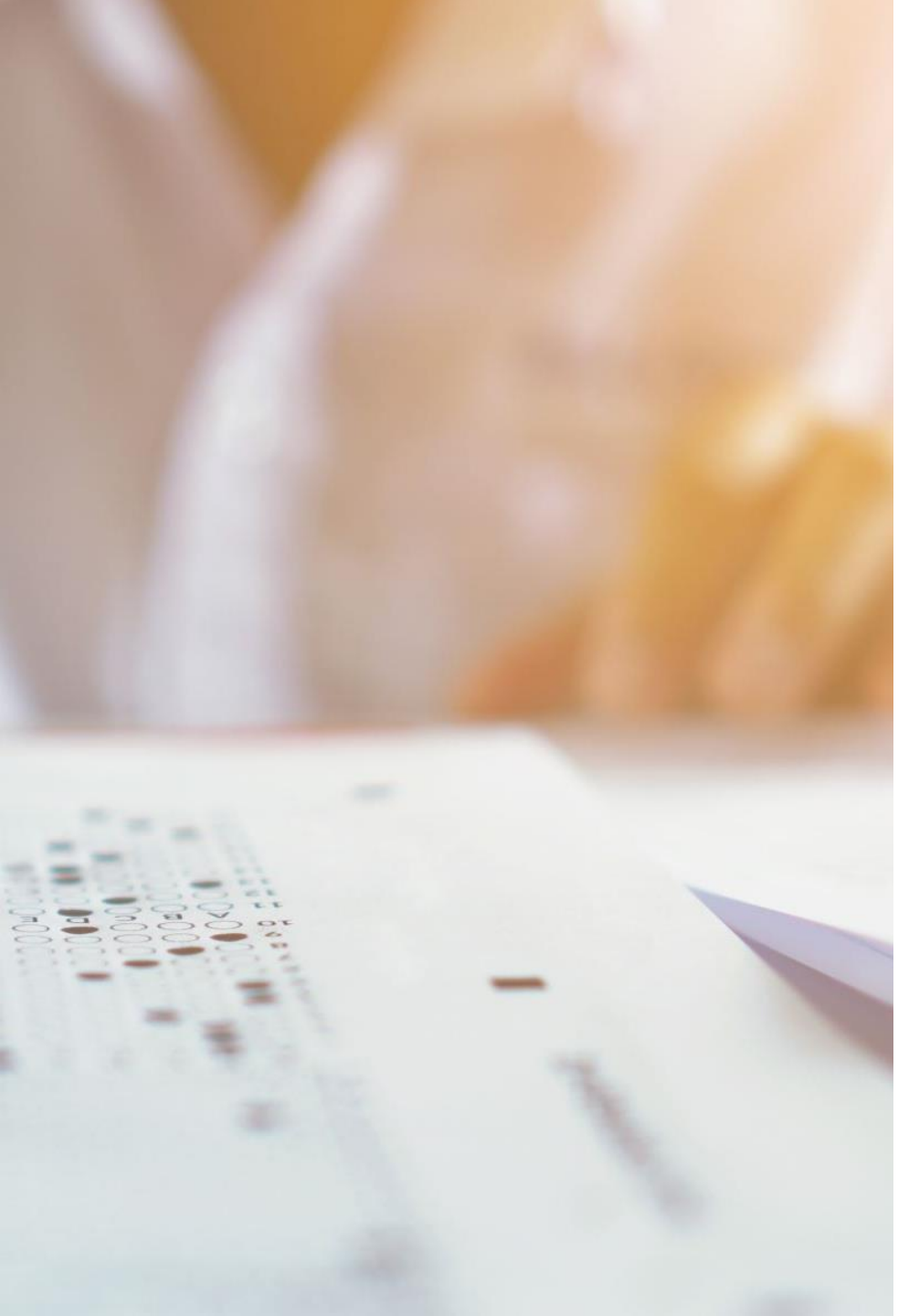
LECTURE 2 DATA TYPES

DR. ERIC CHOU
IEEE SENIOR MEMBER



The One Thing You Need to Know About this Big Idea:

- This unit is all about how computers represent data, and how they can store and process ever-increasing quantities of it.



Unit Overview:

Exam Weighing:

- 17-22% of the AP Exam
- Practically, this translates to about 20 questions on the test.

Data

LECTURE 1

Data

- In its simplest form, **data** is a collection of facts. You can collect data from all sorts of mediums, from lab experiments and sensors to photos and videos. It's used in almost every profession and makes up a big part of the world around us.
- Data values can be stored in variables, lists of items, or standalone constants.
- Computing devices represent data(text, numbers, audio, images) **digitally**, meaning that the lowest-level components of any value are **bits**.

Bits

- **Bit** is shorthand for **binary digit** and is either 0 or a 1. This system of representing data is called **binary** (base 2) which uses only combinations of the digits zero and one.

```
name = "Mike Smith"        # represented internally as 0's and 1's
grades = [87, 91, 75]      # represented internally as 0's and 1's
```

- Thus, **digital data** is simply some sequence of 0's and 1s representing some information.

Binary Numbers

LECTURE 2

Number Bases

- A **number base** is the number of digits or digit combinations that a system uses to represent values.
- Most of the time, we work with the decimal system (also known as base-10) which only uses combinations of 0-9 to represent values.
- However, computers function using **machine code**, which generally uses the **binary system** (also known as base-two).
- The binary system only uses combinations of 0 and 1 to represent values, and can be a little difficult to understand at first.

Number Bases

- If we think way back to elementary school, we'll remember the place value system that the decimal system uses. There's the ones' place, then the tens' place, then the hundreds' place and the thousands' place—and so on.
- For example, take the number 5,729. This number has a 5 in the thousands place, a 7 in the hundreds place, a 2 in the tens place and a 9 in the ones place. That means that 5,729 is made up of 5 thousands + 7 hundreds + 2 tens + 9 ones.

Base 10 (Decimal numbers)

- Computers store all information in the form of binary numbers. To understand binary numbers, let's first look at a more familiar system: decimal numbers.
- Decimal numbers is base 10. It uses 10 digits {0,1,2,3…,9}. Why do we prefer base 10? There are many reasons but one simple reason is simply because we have ten fingers.

Base 10 (Decimal) uses 10 digits: {0,1,2,3…,9}.

Base 2 (Binary) uses 2 digits: {0,1}.

Base 8 (Octal) uses 8 digits: {0,1,2,3,4,5,6,7}

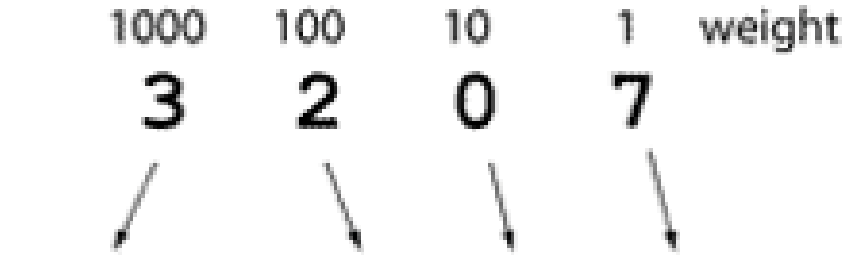
Base 16 (Hexadecimal) uses 16 digits: {0,1,2,3,4…,9,A,B,C,D,E,F}

Base 10 (Decimal)

Each Digit in a Number Has a Weight

you multiply the weight times the digit, then add them up

decimal (base 10): each digit has a weight that is a power of 10



Or, $3000 + 200 + 0 + 7 = 3207$

Powers of 10

$10^0 = 1$
$10^1 = 10$
$10^2 = 100$
$10^3 = 1000$

POLYNOMIAL EVALUATION

Symbolic **representation** (*What we write*)

547_{10}

Subscript used to indicate the radix (number base)

Digit values (coefficients)

$= 5\times10^2 + 4\times10^1 + 7\times10^0$

Positional weights (powers of the radix)

$= 5\times100 + 4\times10 + 7\times1$

$= 500 + 40 + 7$

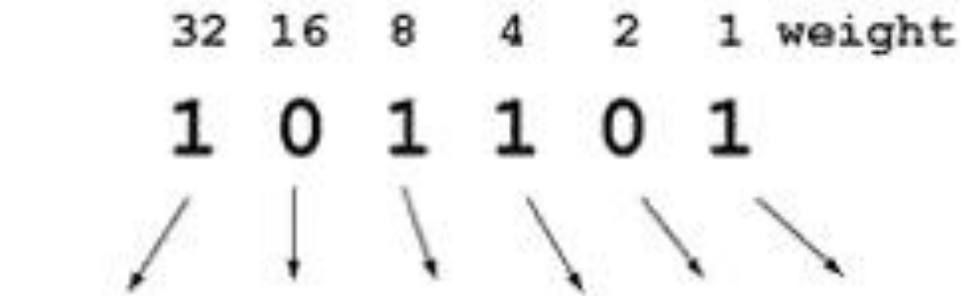
$= 547$

Numeric **interpretation** (*What we understand*)

The result of polynomial evaluation is always a decimal number, regardless of the radix used in the original representation.

Base 2 (Binary)

binary (base 2): each digit has a weight that is a power of 2



1x32 + 0x16 + 1x8 + 1x4 + 0x2 + 1x1

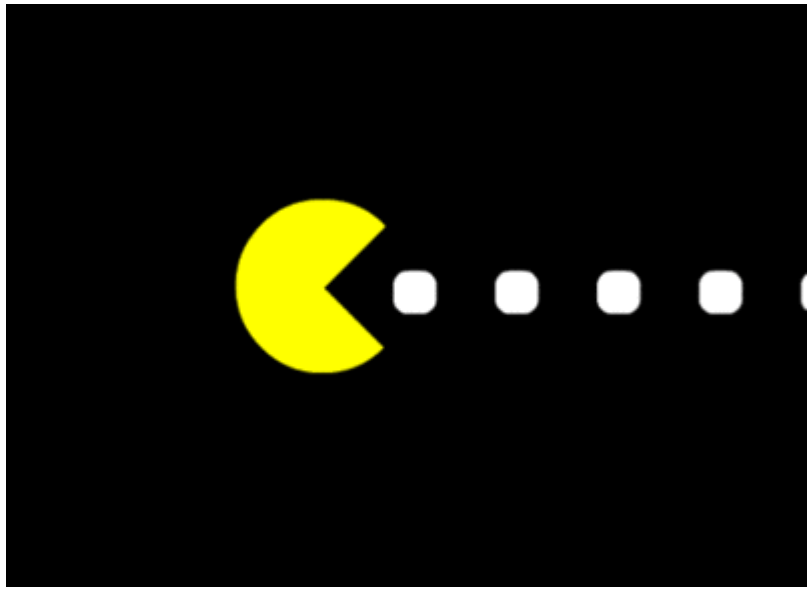
Or, 32 + 0 + 8 + 4 + 0 + 1 = 45

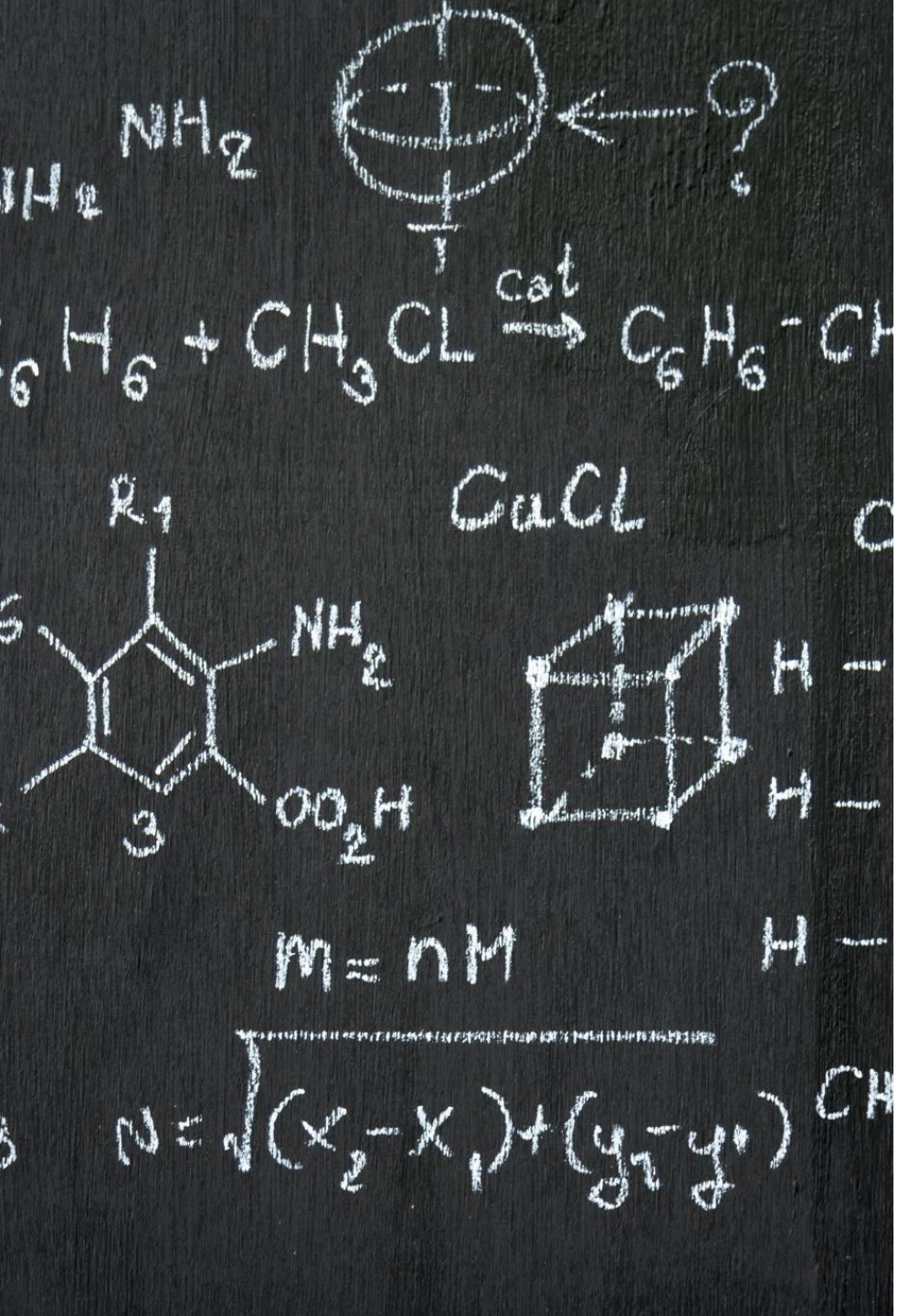
Powers of 2

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$

Number Bases

- The binary number 0101 has a 1 in the fours place and a 1 in the ones place, which means that the number represented is made up of 1 four plus 1 ones, or 4 +1, which = 5.
- These binary digits are also known as bits. If you put eight of these **bits** together, you get a **byte**.





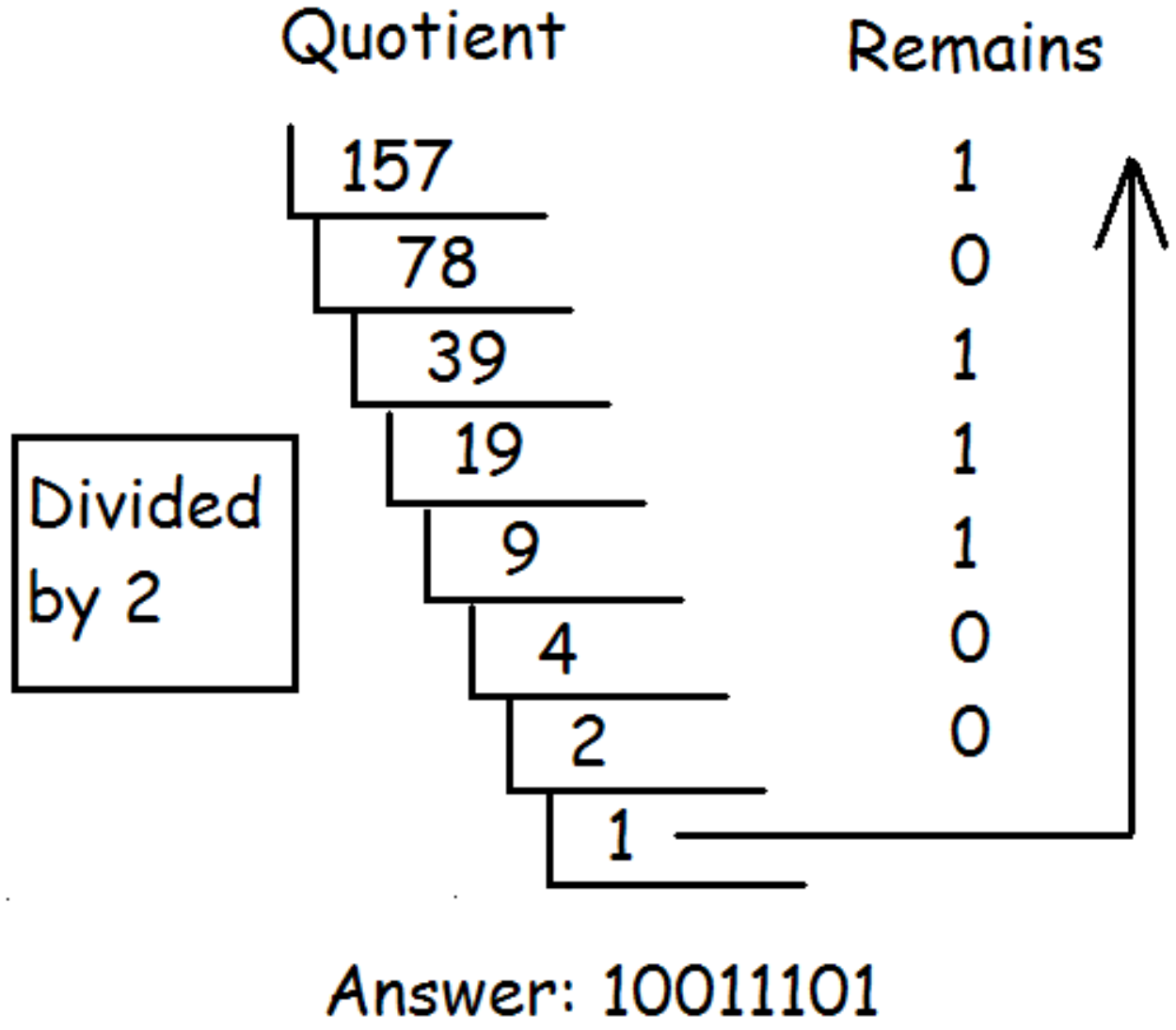
Bits and Bytes

1 **bit** is a single bit of information, a 1 or 0(Only two possible values)

1 **byte** is 8 bits, an 8-bit word

- 256 possible values from 0-255 **base 10**
- or 00000000 to 11111111 **base 2**

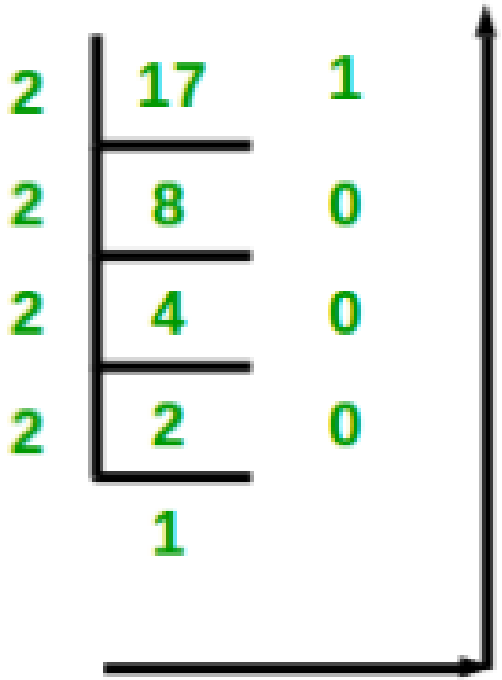
For example, 10100110 is a single byte.



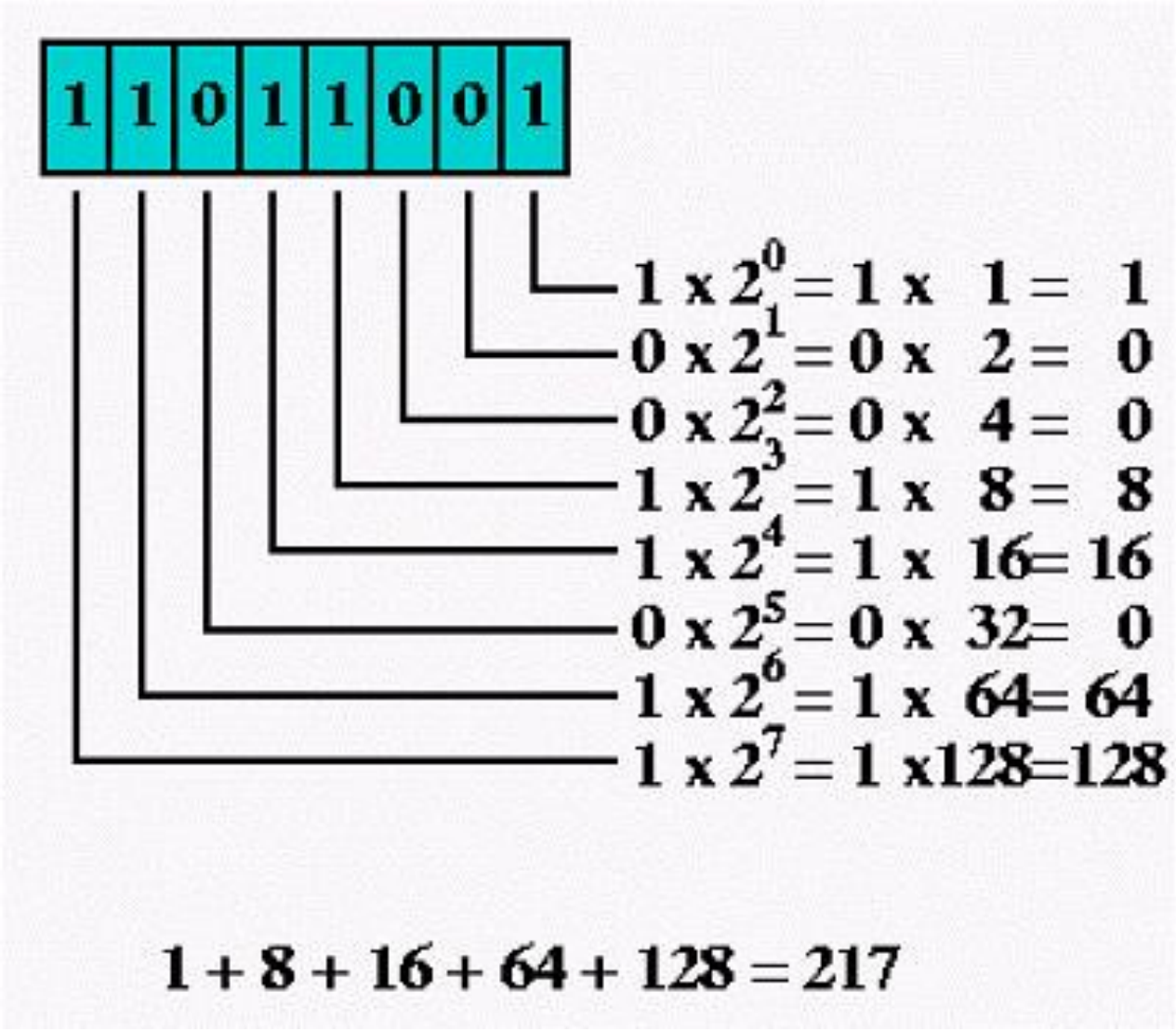
Decimal to Binary

- Repeatedly modulo 2 and the divide by 2. Stop at 0. The remainder list read top to bottom are the digits from left to right.

Decimal number : 17



Binary number: 10001



Decimal to Binary Conversion

Convert decimal number 57_{10} to binary

Method 1
Descending Powers of Two and Subtraction

1. Write the power of two

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$

2. Write the number as a sum of powers of two

$57 = 32 + 25$
$= 32 + 16 + 9$
$= 32 + 16 + 8 + 1$
$= 1 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 1$
$= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0$
$= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

3. Extract the coefficients of the powers of two

$$57_{10} = 111001_2$$

Method 2
Short Division by Two with Remainder

1. Divide continuously by two until we reach 0

2:57 (remainder 1)
2:28 (remainder 0)
2:14 (remainder 0)
2:7 (remainder 1)
2:3 (remainder 1)
2:1 (remainder 1)

1 1 1 0 0 1

2. Write the remainders in reverse order (last is first)

$$57_{10} = 111001_2$$

x-engineer.org

Hexadecimal

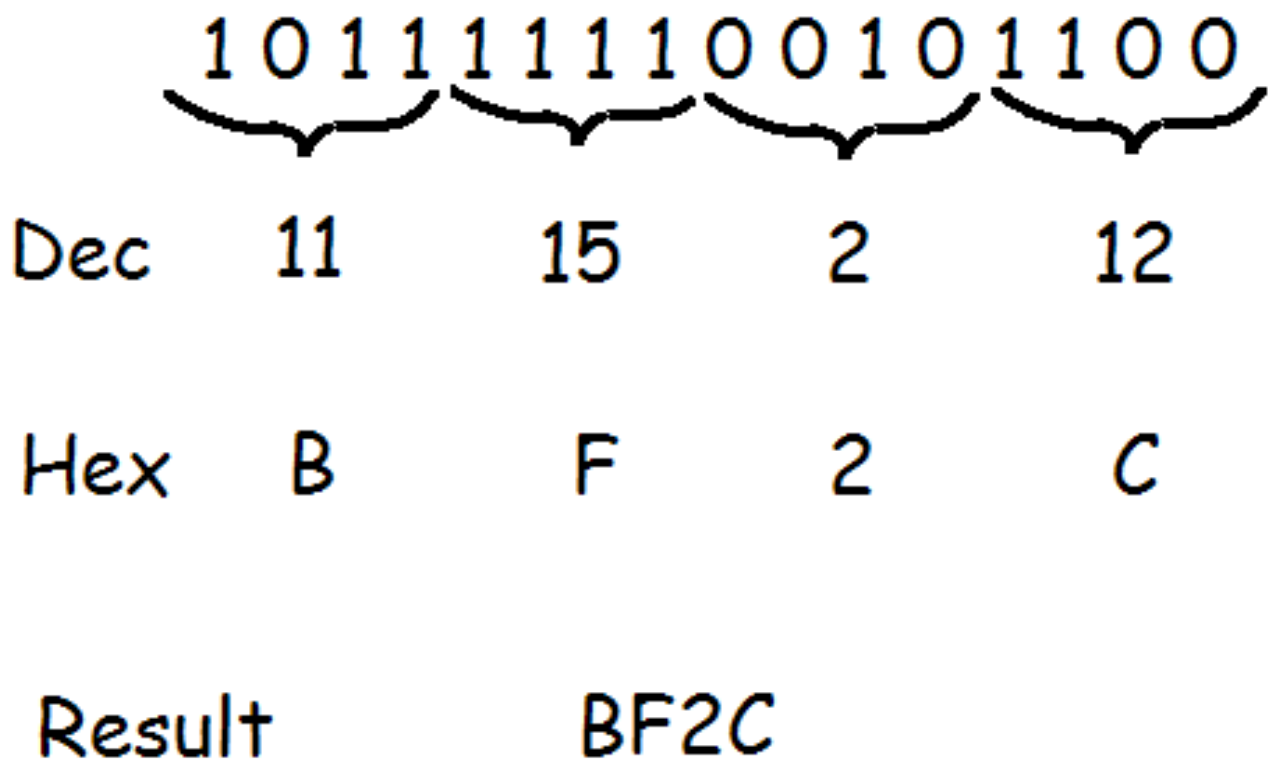
- These binary strings can get rather long rather quickly. In order for *people (*not computers, which use binary!) to work with these values more easily, they use the hexadecimal, or base-16 system. A good example of this are the RGB color codes, which are written in **hexadecimal**.
- The hexadecimal system uses both letters and numbers to represent values. You represent 0-9 as you would in the base-10 system, but then you use the letters A to F to represent 10-15. The place values change as well, to multiples of 16.

Power of 16	16^3	16^2	16^1	16^0
Value represented (base 10)	4096	256	16	1
Amount of value	1	6	6	1

This number is made up of one 4096 + six 256s + six 16s + one 1, which totals to equal 5,729.

Hexadecimal

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



Binary to Hex

Counting

Let’s count in Base 10.
0,1,2,3,4,5,6,7,8,9,__,___,…,18,19,__,__.
Answer: 10,11…..,20,21. Good job!

Let’s count in Base 5.
0,1,2,3,4,__,__,__,__,__,___.
Answer:10,11,12,13,14,20.

Let’s keep going!
40,41,42,43,44,___,___,___.
Answer:100,101,102.

Quick Quiz

What's the answer in base 8?

$$6_{10} + 3_{10} = 11_8$$

What's the answer in base 5?

$$2_{10} + 6_{10} = 13_5$$

What's the answer in base 12?

$$9_{10} + 15_{10} = 20_{12}$$

Abstraction

LECTURE 3

Abstraction

- **Abstraction** is the process of reducing complexity by focusing on the main idea.

- By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea.

Procedural Abstraction

- One type of abstraction we saw was **procedural abstraction**, which provides a name for a procedure(function) and allows it to be used only knowing what it does, not how it does it.

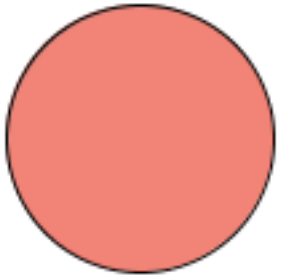
```
import random
print(random.randrange(10)) # random number from 0 - 9
```

- We don't need to know how the randrange() function is implemented to be able use it.

Data Abstraction

- Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color.
- Sequences of 0's and 1's represent a string of characters, a numeric grade and a color in the examples below. The color example is done using Processing.

```
name = "Mike Smith"
grade = 87
c = color(255, 123, 110)
fill(c);
ellipse(250, 80, 100, 100)
```



Analog Data and Bit Representations

- Analog data is data that is measured continuously. Its key characteristic is that analog data values change smoothly, rather than in discreet intervals.
- For example, imagine that you're listening to a flute solo your friend is playing, and you're at a part where the music is getting louder and louder. The volume of the music is changing smoothly and would be considered an example of analog data.
- Other examples include the time recorded on an analog clock or the temperature measured on a physical thermometer, like the one pictured below.

Encoding

- A computer cannot store “letters” or “pictures”. It can only work with bits(0 or 1).
- To represent anything other than bits, we need rules that will allow us to convert a sequence of bits into letters or pictures. This set of rules is called an **encoding scheme**, or **encoding** for short. For example,

```
01100010 01101001 01110100 01110011
b        i        t        s
```

ASCII

- ASCII(American Standard Code for Information Exchange) is an encoding scheme that specifies the mapping between bits and characters. There are 128 characters in the scheme. There are 95 readable characters and 33 values for nonprintable characters like space, tab, and backspace and so on.
- The readable characters include a-z, A-Z, 0-9 and punctuation. In ASCII, 65 represents A, 66 represents B and so on. The numbers are called code points. But ASCII only uses 8 bits. It does not have enough bits to encode other characters and languages(Chinese, French, Japanese).

ASCII

- Similar to ASCII, Unicode provides a table of code points for characters: "65 stands for A, 66 stands for B and 9,731 stands for ☃". UTF-32(Unicode Transformation Format), UTF-16 and UTF-8 are three encodings that use the Unicode table of code points.
- Of the three, UTF-8 is by far the most widely used encoding. It is the standard encoding for all email and webpages(HTML5).

ASCII Sample

To **encode** something in ASCII, follow the table from right to left, substituting letters for bits.

To **decode** a string of bits into human readable characters, follow the table from left to right, substituting bits for letters.

encode

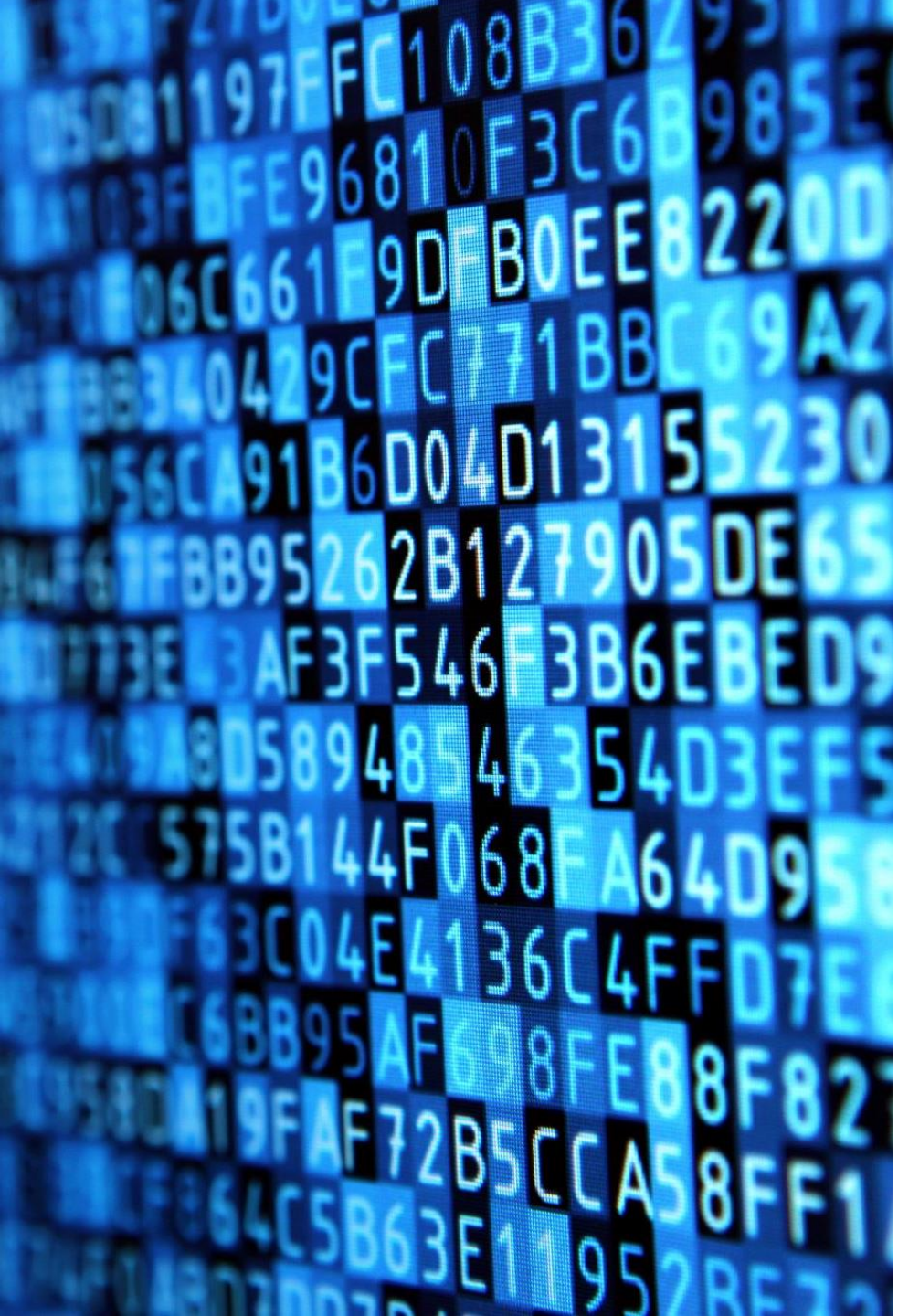
bits	character
01000001	A
01000010	B
01000011	C
01000100	D
01000101	E
01000110	F

decode

Dec	Hex	Char
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9

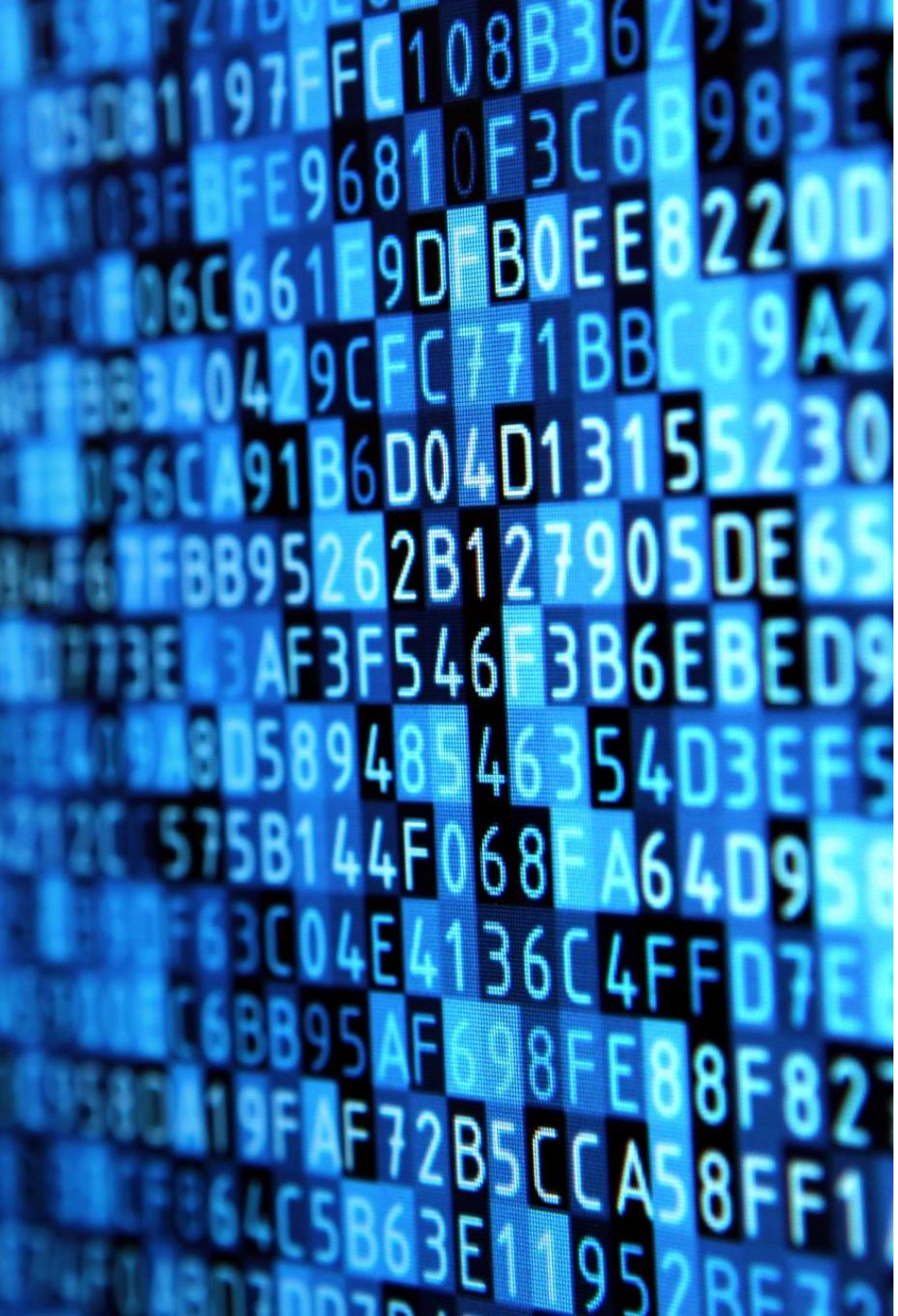
Dec	Hex	Char
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z

Dec	Hex	Char
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z



Unicode

- So, how many bits does Unicode use to encode all these characters? *None.* Because Unicode is not an encoding.
- *Unicode* first and foremost defines a table of ***code points*** for characters. That's a fancy way of saying *"65 stands for A, 66 stands for B and 9,731 stands for ☃"*.
- To represent 1,114,112 different values, two bytes aren't enough. Three bytes are, but three bytes are often awkward to work with, so four bytes would be the comfortable minimum.
- **UTF-32(Unicode Transformation Format)** is such an encoding that encodes all Unicode code points using 32 bits. That is, four bytes per character.



UTF-32

- To represent 1,114,112 different values, two bytes aren't enough. Three bytes are, but three bytes are often awkward to work with, so four bytes would be the comfortable minimum.
- **UTF-32(Unicode Transformation Format)** is such an encoding that encodes all Unicode code points using 32 bits. That is, four bytes per character.
- UTF-32 very simple, but often wastes a lot of space. For example, if A is always encoded as 00000000 00000000 00000000 01000001 and B as 00000000 00000000 00000000 01000010 and so on, documents would bloat to 4x its necessary size.

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

UTF-16 and UTF-8

UTF-16 and UTF-8 are variable-length encodings. If a character can be represented using a single byte (because its code point is a very small number), UTF-8 will encode it with a single byte. If it requires two bytes, it will use two bytes and so on. UTF-16 is in the middle, using at least two bytes, growing to up to four bytes as necessary.

Sequences of Bits

- What does a list of numbers, an image and an audio file have in common? They are all just lists of numbers!
- The same sequence of bits may represent different types of data in different contexts.

```
import numpy as np
import matplotlib.pyplot as plt
nums = np.loadtxt("numbers.txt")
print(nums)          # array([210 190 225 ..., ])
print(nums.size)     # 783126
```

- These numbers can represent anything. For example, it could represent the number of characters in a list of 783,126 tweets that were tweeted over some one-minute interval.
- Or this same list could represent something else entirely. See the next slide!

List of Numbers is An Image

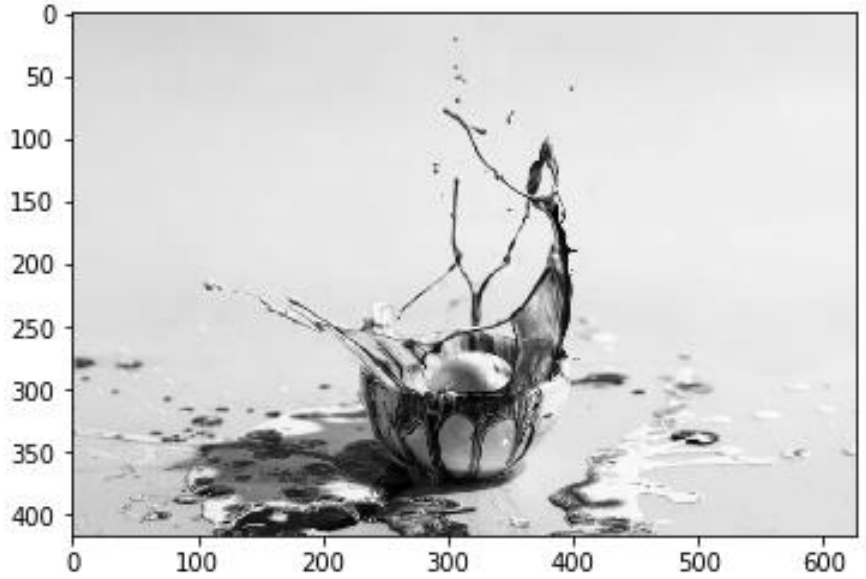
- The list of numbers in the previous example is now an image! We'll discuss how to write the code below in the next lecture.

```
import numpy as np

import matplotlib.pyplot as plt

nums = np.loadtxt("numbers.txt", dtype="uint8")

print(nums)        # array([210 190 225 ..., ]), same list of numbers

nums = nums.reshape(417, 626, 3)

fig, ax = plt.subplots()

ax.imshow(nums)
```

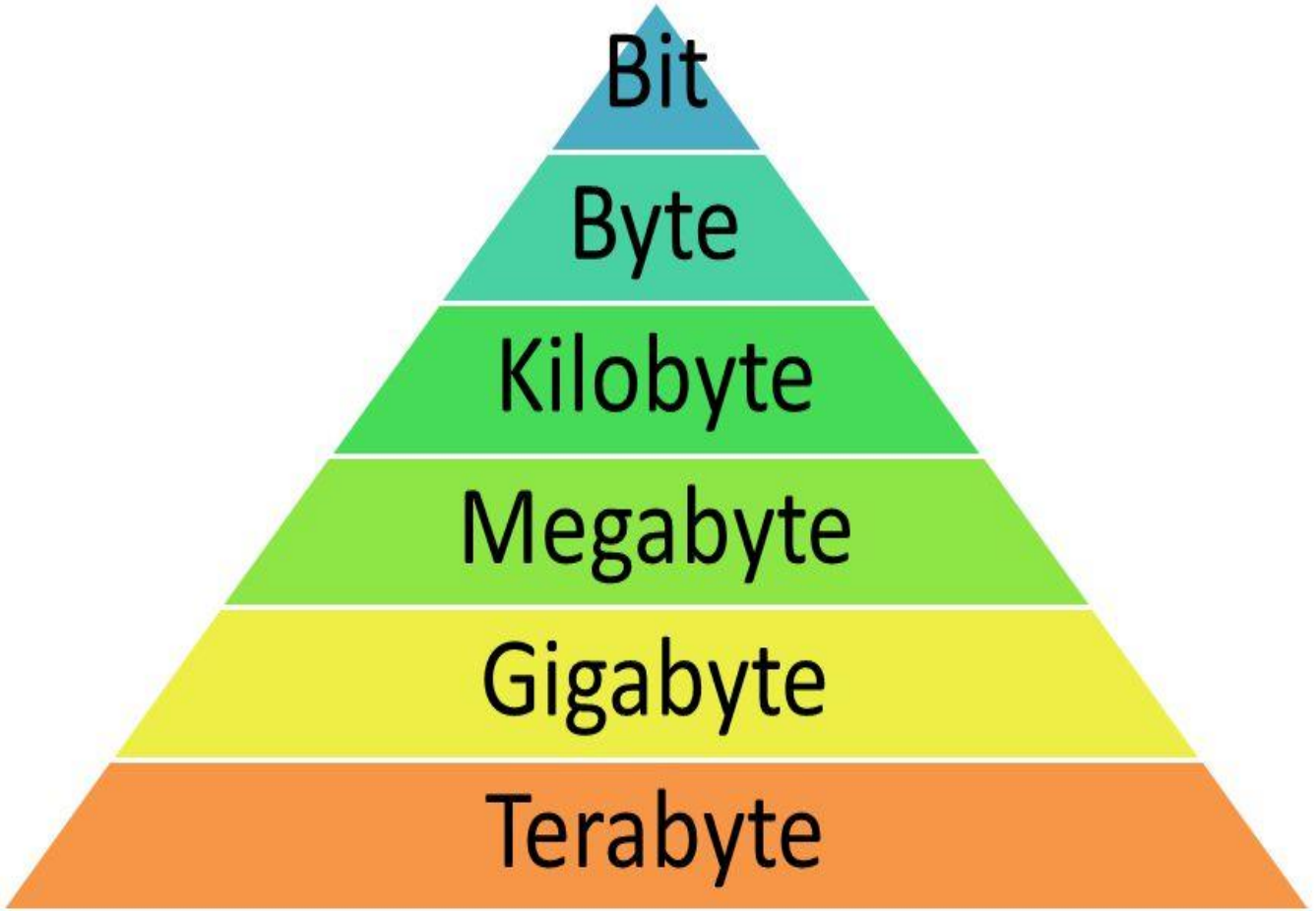
List of Numbers is An Image

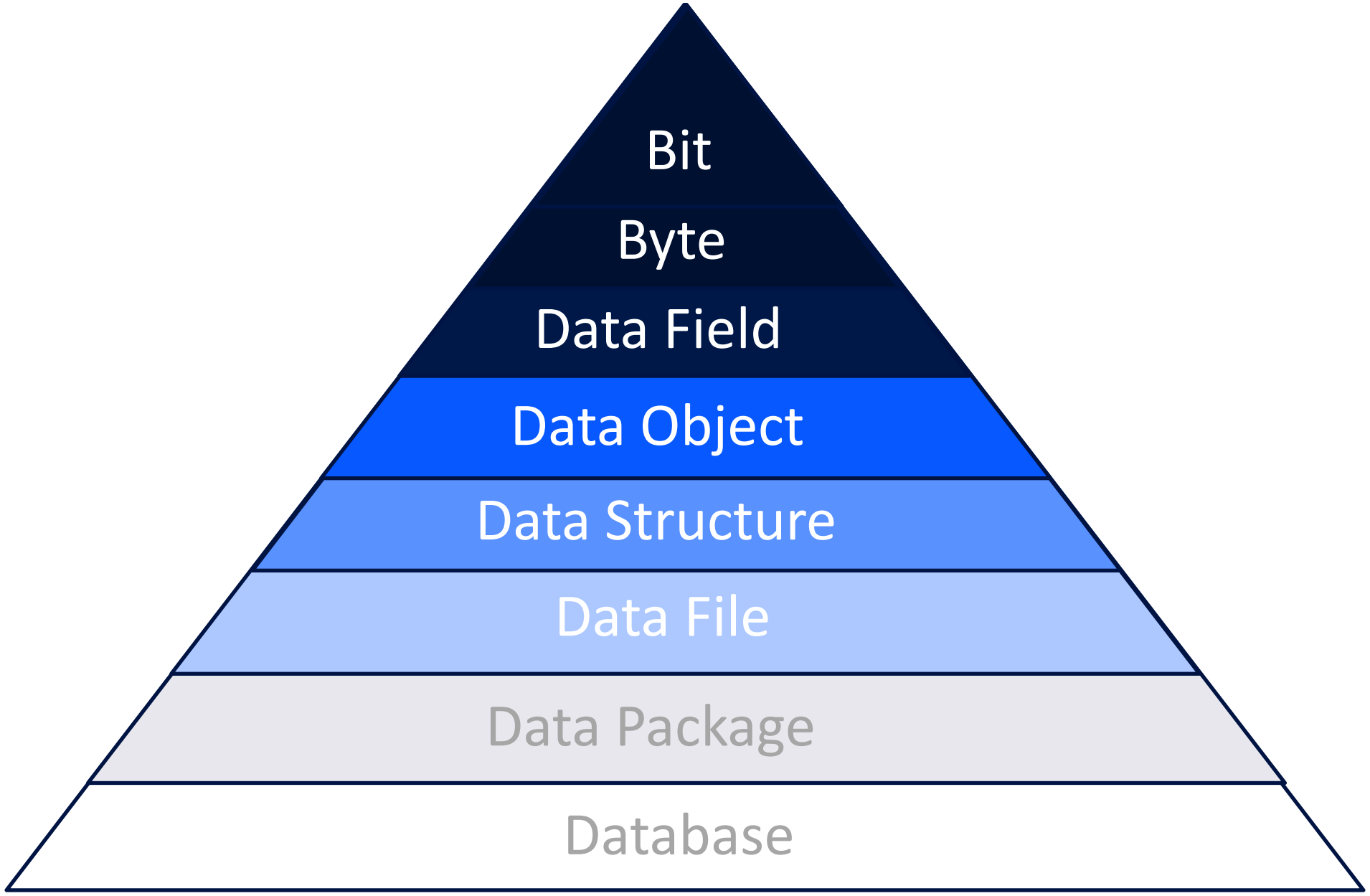
The following output is produced by running the above code on the Jupyter Notebook

<matplotlib.image.AxesImage at 0x1045a8710>



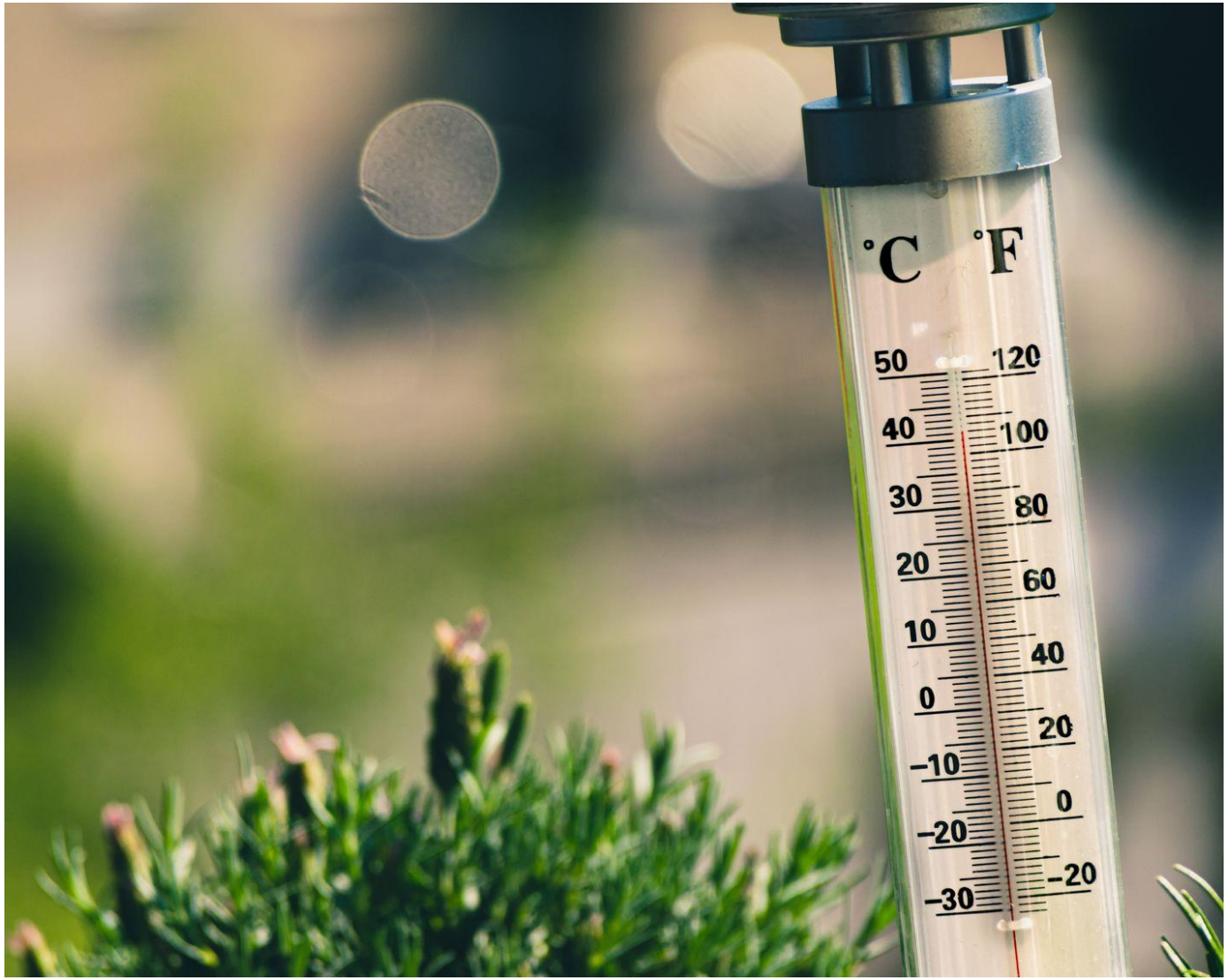
Types of Units of Storage




Bit
Byte
Data Field
Data Object
Data Structure
Data File
Data Package
Database

Analog and Digital Representation

LECTURE 4


°C °F
50 120
40 100
30 80
20 60
10 40
0 20
−10 0
−20 −20
−30

Analog Data and Bit Representations

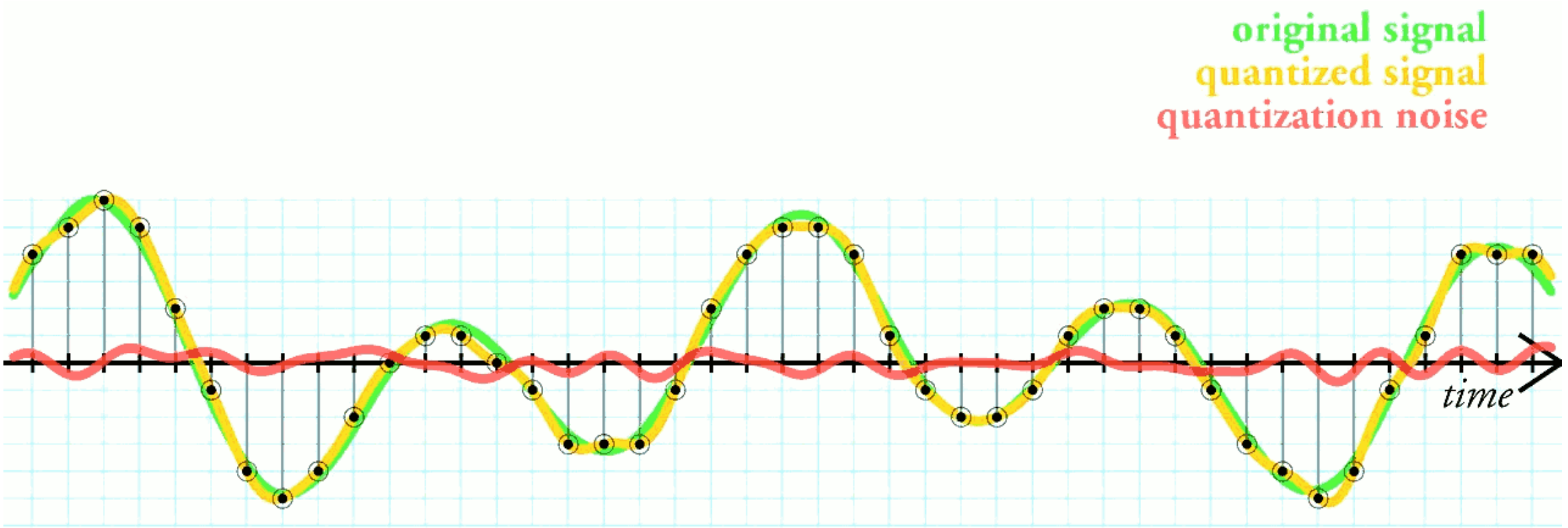
- In both of these devices, data is being recorded constantly: the clock hands and thermometer mercury are (at least in theory) always moving.
- Now, let's say you wanted to measure all of this data (the volume of the flute music, the temperature of a room, the current time) digitally, using a device like a sound recorder. Once collected, this information would be known as digital data.

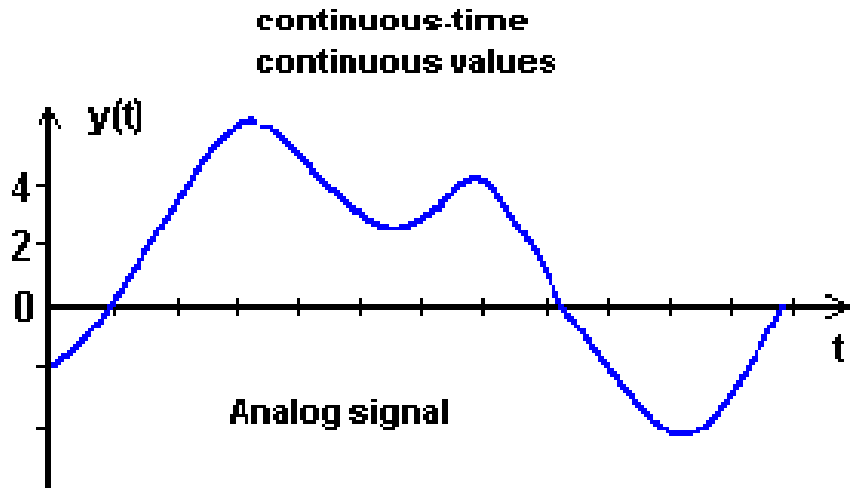
Sampling Techniques

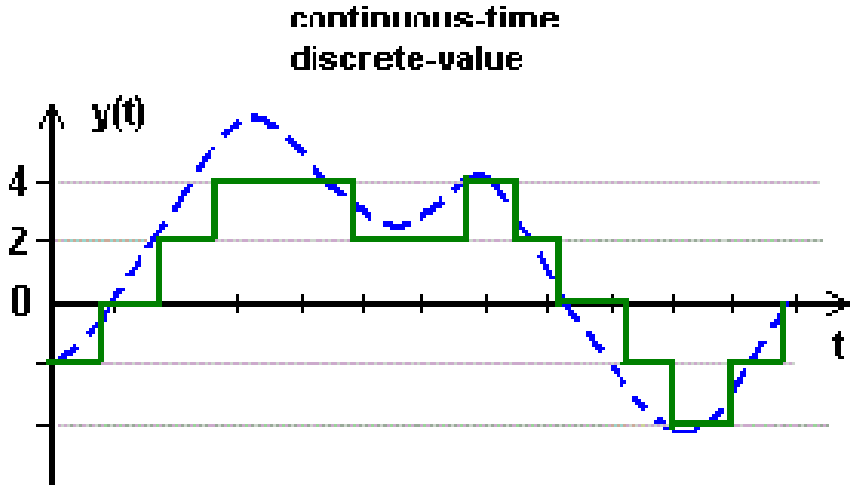
- Analog data can be represented digitally by using a **sampling technique**. The values of the analog signal are measured and recorded at regular intervals. (The intervals are known as samples.) These samples are then measured to figure out how many bits will be needed to store each of them.
- Digital data must be formatted in a finite set of possible values, while analog data can be infinite. Think about a (basic) digital clock, for example. It only changes every minute, as opposed to the constantly changing hands of an analog clock. Another example would be watching a video of an event versus watching the event in person: when you watch the video, you're watching a large number of images put together. In contrast, the event is happening continuously when you're at the venue.

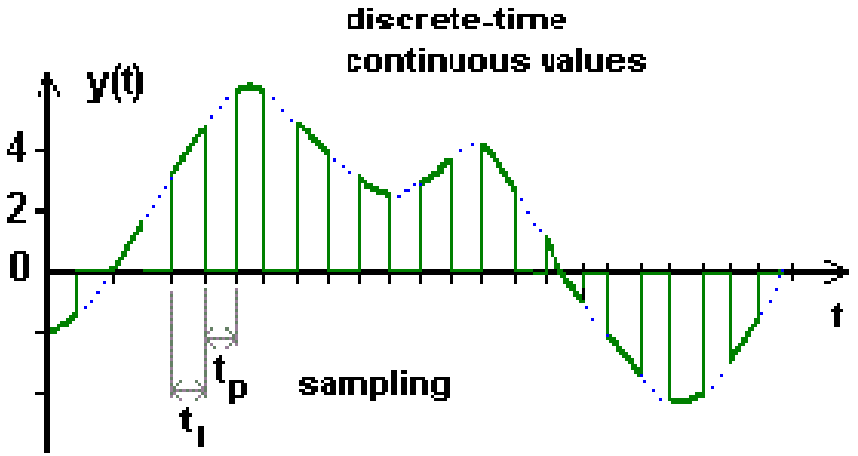
Sampling Techniques

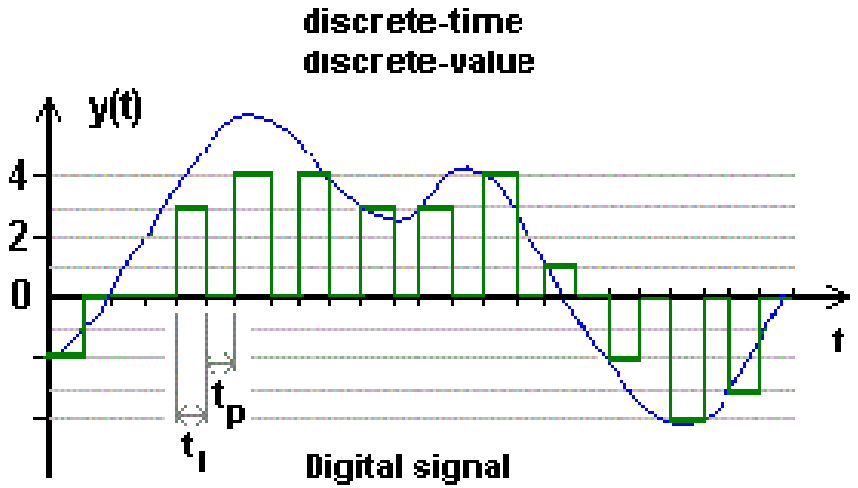
- This divide between continuous and finite means that digital data, while it can come very close, can only be used to approximate analog data rather than perfectly represent it.
- Digital data is a simplified representation that leaves out extra details. Going back to the digital clock example, you usually don't need to know the time down to the millisecond; oftentimes, the hour and minute are enough.
- Due to this simplification, using digital data to approximate real-world analog data is considered an example of abstraction.


original signal
quantized signal
quantization noise
time


continuous-time
continuous values
y(t)
4
2
0
t
Analog signal


continuous-time
discrete-value
y(t)
4
2
0
t


discrete-time
continuous values
y(t)
4
2
0
t
t_p
t_I
sampling


discrete-time
discrete-value
y(t)
4
2
0
t
t_p
t_I
Digital signal

Overflow Errors

- In many low-level programming languages, such as C and C+, numbers are represented by a certain number of bytes (anywhere from one to eight). This limits the range of values and operations that can be done on those values.
- Let's say that a number is represented by one byte, or eight bits.

Power of 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value represented (base 10)	128	64	32	16	8	4	2	1
Amount of value								

Overflow Error

- Here's a chart representing a byte, or eight bits.

❓ What's the maximum value this byte can store?

- The byte can represent (128 + 64 + 32 + 16 +8 + 4 + 2 + 1). If you add the numbers all up, you get 255 in base-10, represented as 1111 1111. The easier way to calculate this value is to go to the next power up from the last one on the chart—In this case 2 to the 8th, which is 256—and then to subtract one.
- What if you tried to represent 256 or higher on this chart? You can't because there's no way to store the value. Trying to store a larger number would lead to an **overflow error**.

Overflow Error

- This overflow error may present itself in several ways. The number may display negative when it should be positive, or vice versa. It might also display a completely different value, such as 0. Overflow errors don't usually cause the program to stop working, so they can be hard to detect.
- Higher-level programming languages, such as Python, work around this problem and remove the range limitation. In such languages, the largest number you can represent depends solely on how big your computer's memory is! (The AP CSP test also uses this standard.)

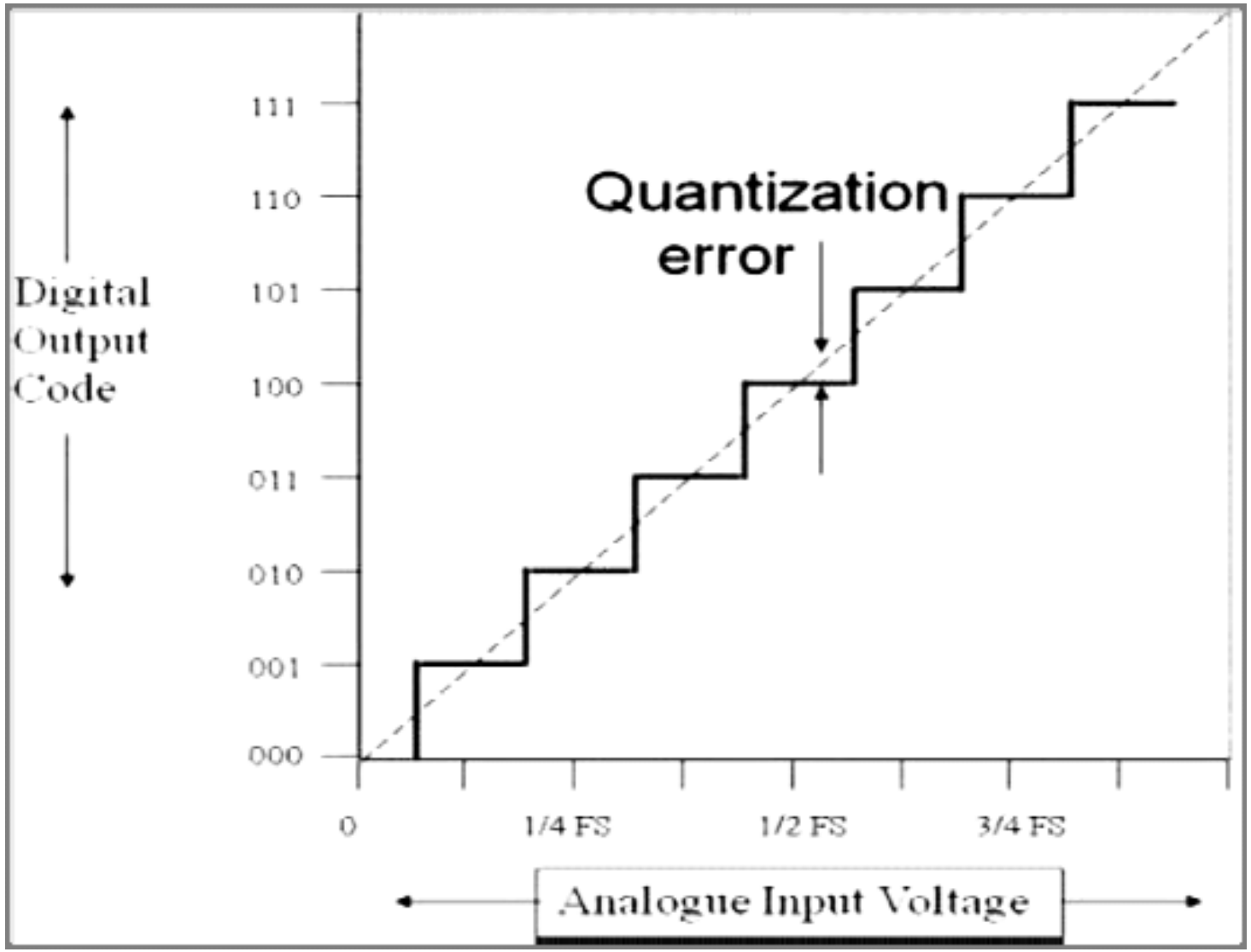
Quantization Effects

- Another way that number storage makes use of abstraction is through rounding. Because you have a limited amount of bits to store numbers, the computer will sometimes round or cut off your number. This can be most prominently seen when you're working with very small numbers or repeating decimals.

```
What do you want your first number to be?100
What do you want your second number to be?3
33.333333333333336
```

Quantization Error

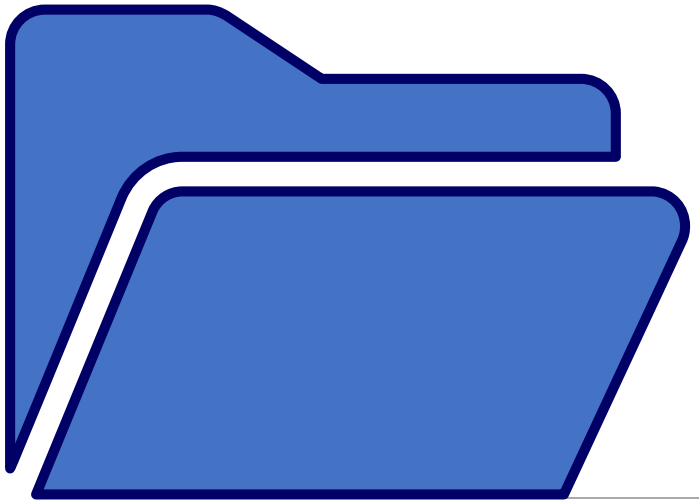
- Here's a basic dividing program I wrote on my computer in Python. As you can see, the answer to 100/3 has a finite end in the program, even though it's meant to be a repeating decimal.
- This is an example of abstraction because the number represented by the computer is a simplified version of the full value. Although it usually doesn't matter for most calculations, including the ones you'll do in school, **rounding or round-off errors** can sometimes cause issues if you need more precision.


Digital
Output
Code
111
110
101
100
011
010
001
000
Quantization
error
0
1/4 FS
1/2 FS
3/4 FS
Analogue Input Voltage

File Types

LECTURE 5


AVI
JPG
TXT
MP3
PDF
XLS
PNG
MOV
WAV
DOC
W
EPS
PPT
ZIP
CSS
[...]
RAR
HTML
EML
RAW
CSV
TTF
Aa



Text

FILE TYPE 1: TEXT

Data Representation

- 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.
- 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 [8 x 5]) of memory.

String

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

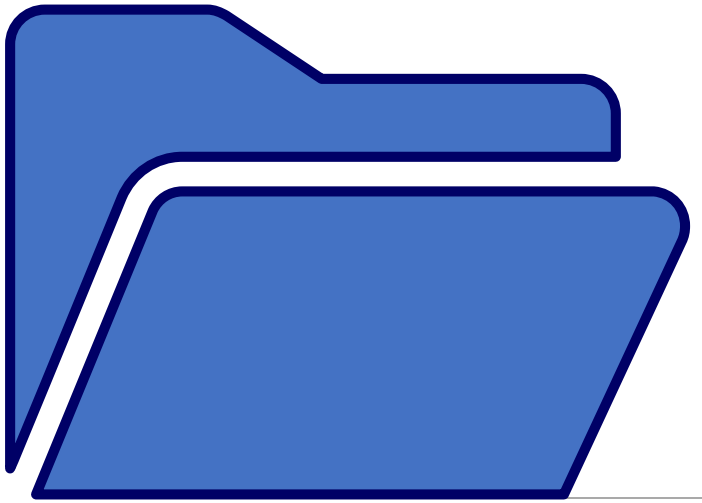
Why? (Formatted Data Versus Unformatted)

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

hello.txt - 5 bytes
hello.docx = 21,969 bytes

Formatted Data

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.



B&W Images

FILE TYPE 2: BW IMAGES

Objectives

- You will explore the way digital images are encoded in binary.
- You will exhibit some creativity while getting some hands-on experience manipulating binary data that represents something other than plain numbers or text.

Vocabulary

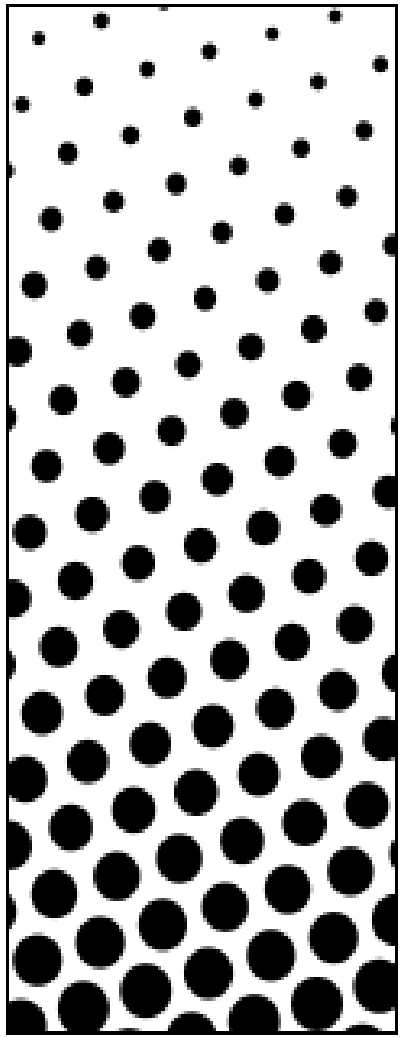
- **Image** - A type of data used for graphics or pictures.
- **metadata** - is data that describes other data. For example, a digital image may include metadata that describe the size of the image, number of colors, or resolution.
- **Pixel** - short for "picture element", the fundamental unit of a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

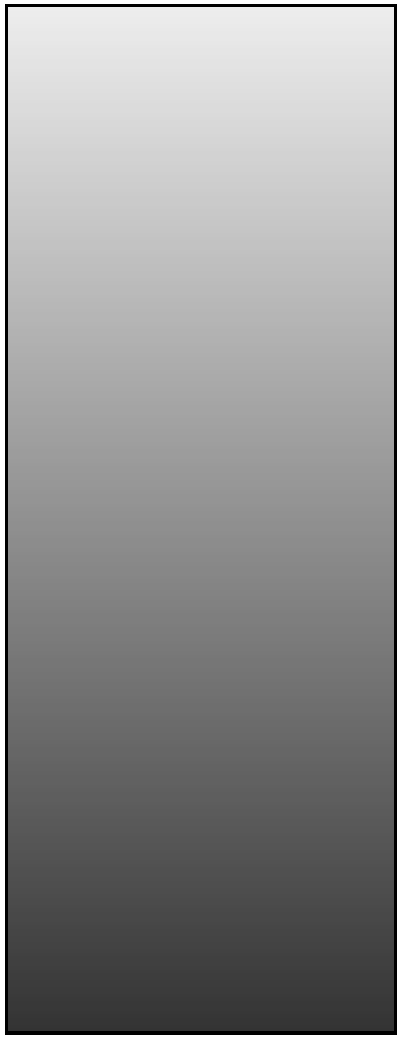
How many bytes does it take to store an image vs text?

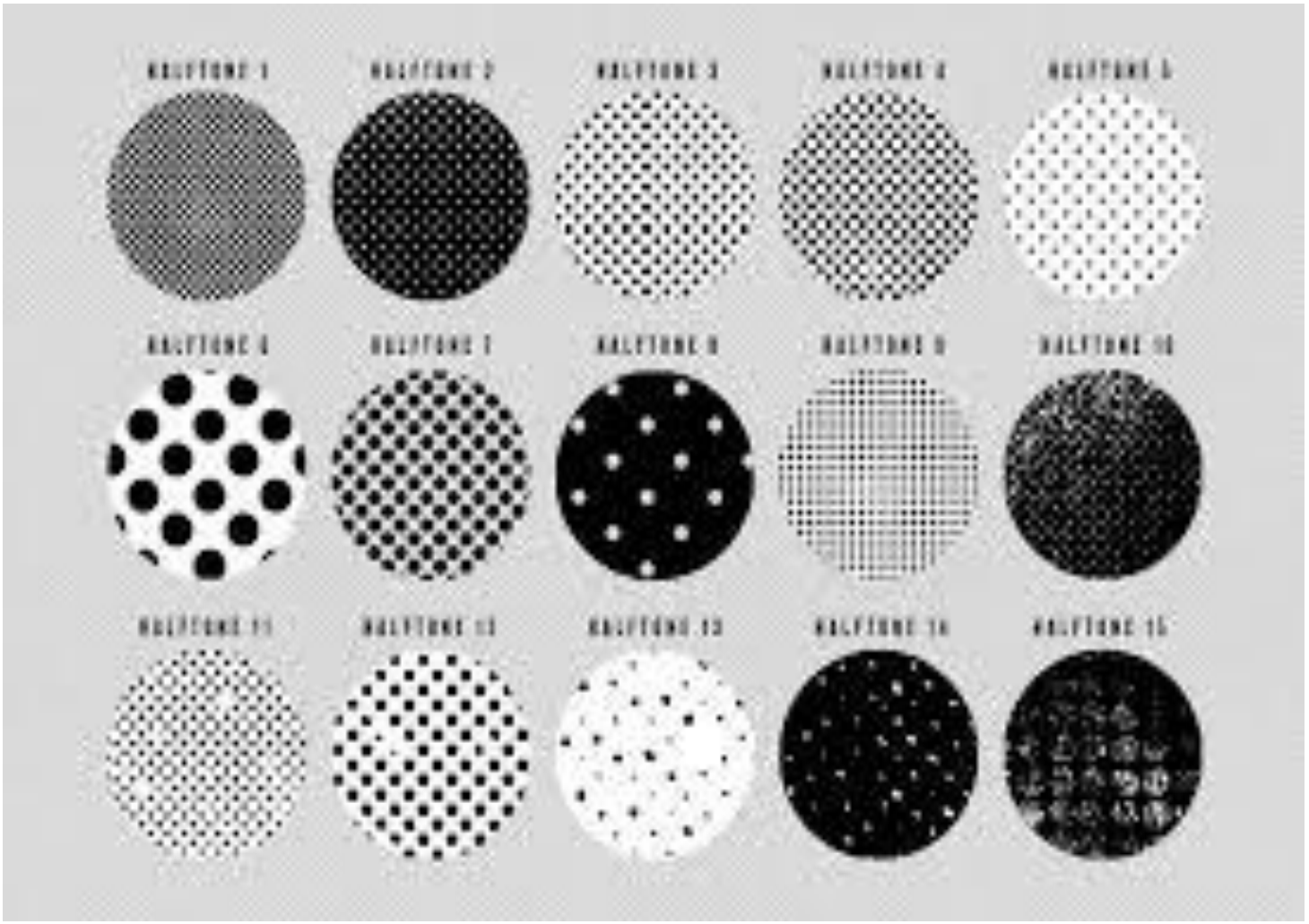
- Back in the Internet Unit you encoded a line-drawing image as a list of numbers that made up the coordinates of the points in the image. That works for line drawings, but how might you encode a different kind of image?
- Today we're going to consider how you might use bits to encode a photographic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

Activity Guide

- Invent a B&W Encoding Scheme
- Get in Pairs







B&W Encoding

- 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?

B&W Encoding

- 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

Pixel

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".

Code Studio Excercise

- 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.

How to mix Light

- Remember you are mixing LIGHT, not PAINT!
- What makes White?
- What makes Black?

Color Model

- **White** is ALL COLORS
  Think of a prism, or a rainbow
- **Black** is Absence of Color – no color
  Think of outer space, or the lights off!

Video

- Watch video (3 min):

B&W Pixelation Tutorial – Video

Activity Guide

- B&W Pixelation Widget
  and
- **Walk Around Card:** Pixelation Widget

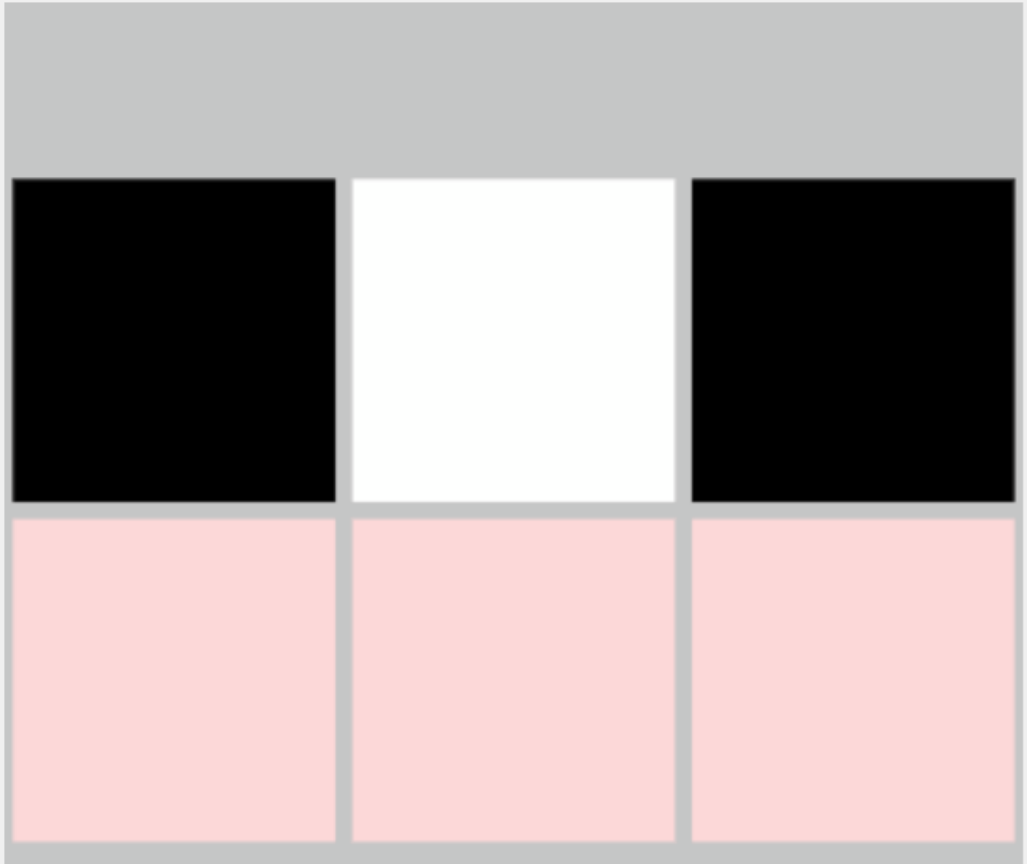
Image File Format:

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

Image width: 3

Image height: 2

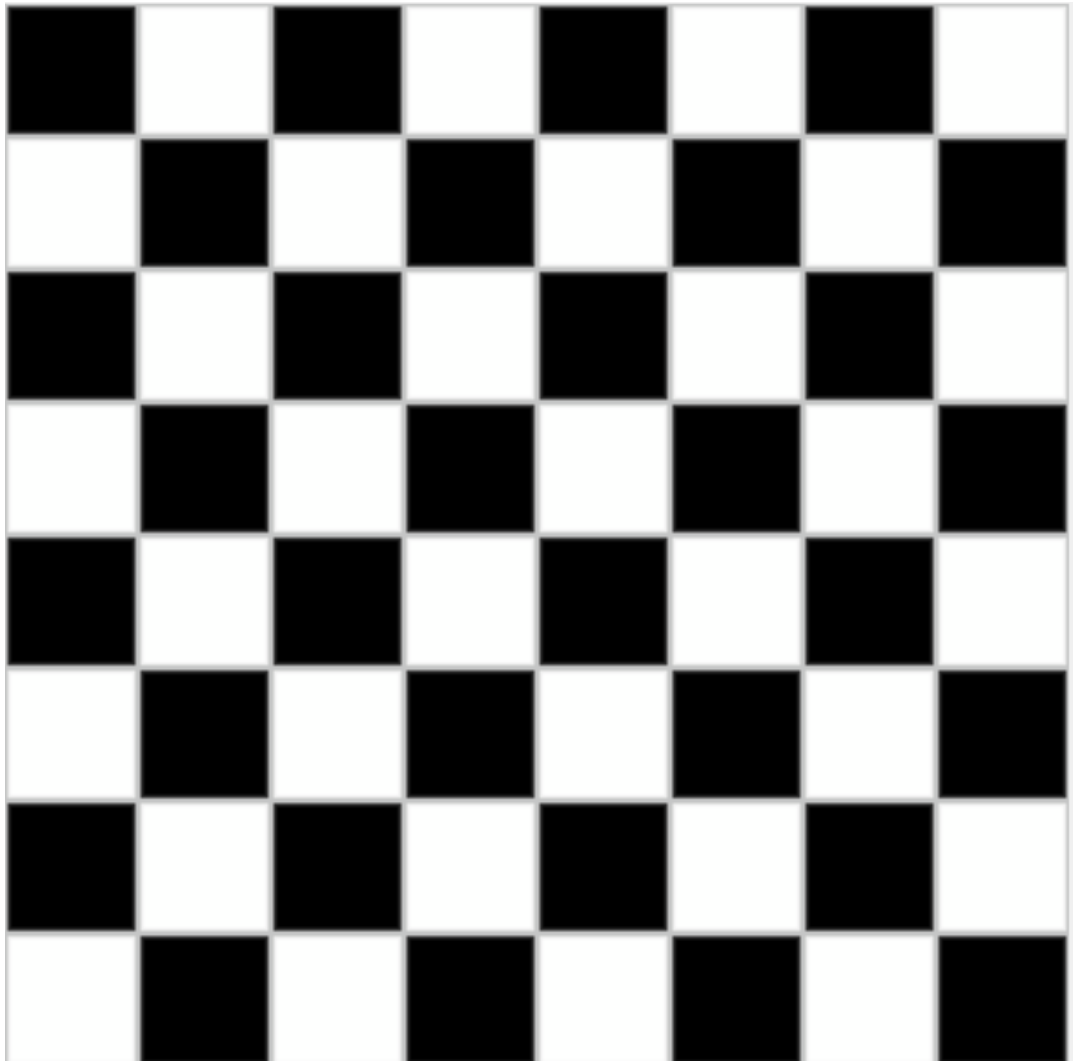
Binary: Hexadecimal:



```
0000 0011
0000 0010
0 1 0
```

Width

Height



```
0000 1000
0000 1000
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
```

Readable format

Metadata

- Let’s make sure that the bits you submitted actually produce the image as claimed.
- You get Scored on the digital artifact with points for creativity and perceived effort.
- 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.

Metadata (Optional)

- 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 **metatdata**. 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. [Address, Size, Packet Number]

Metadata (Optional)

Metadata is typically defined as “data about data.” Although the term itself was coined relatively recently, people have applied the concept of metadata for thousands of years to organize, describe and retrieve recorded information.

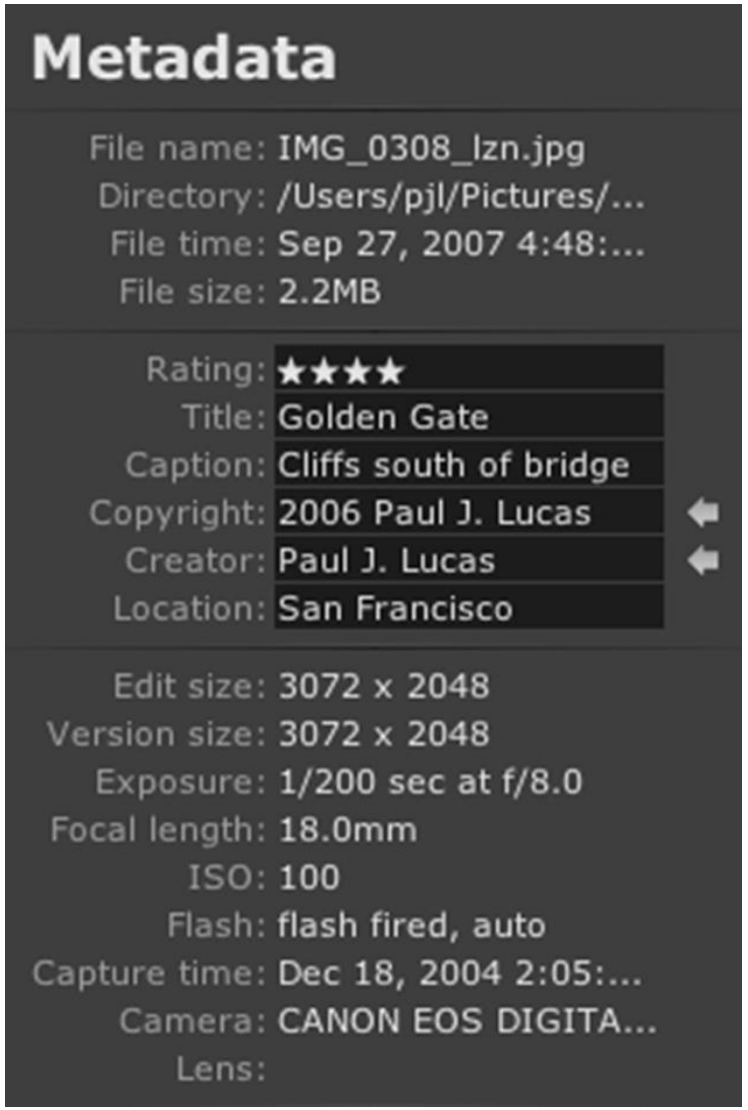
Metadata is ubiquitous. Think of any book on your shelf: its title, author, or even the paper stock used in your particular copy. Each of these bits of information is an example of metadata, and each is helpful for describing your book as an object, i.e., a discrete piece of knowledge.

Metadata (Optional)

Not coincidentally, metadata is at the heart of **library and archival science**. Professionals in these fields generally divide metadata into three or four basic types.

- **Descriptive metadata** is used for discovery and identification, and usually contains elements like title, author, or keywords.
- **Structural metadata** tells us how the various parts of an object fit together, such as page and chapter order.
- **Administrative metadata** describes the management of a resource, including provenance, ownership and rights management, and the technical features of an object. (Some specialists, however, consider technical metadata to be its own distinct category.)

The latest NISO standards also include markup languages as a major metadata type. The latest NISO standards also include markup languages as a major metadata type. Markup languages like **HTML** and **XML** integrate metadata and flags for structural and semantic features into actual content, improving navigation and increasing interoperability.


Metadata
File name: IMG_0308_lzn.jpg
Directory: /Users/pjl/Pictures/...
File time: Sep 27, 2007 4:48:...
File size: 2.2MB
Rating: ★★★★
Title: Golden Gate
Caption: Cliffs south of bridge
Copyright: 2006 Paul J. Lucas
Creator: Paul J. Lucas
Location: San Francisco
Edit size: 3072 x 2048
Version size: 3072 x 2048
Exposure: 1/200 sec at f/8.0
Focal length: 18.0mm
ISO: 100
Flash: flash fired, auto
Capture time: Dec 18, 2004 2:05:...
Camera: CANON EOS DIGITA...
Lens:

Metadata Type	Example Properties	Primary Uses
Descriptive metadata	Title Author Subject Genre Publication date	Discovery Display Interoperability
Technical metadata	File type File size Creation date/time Compression scheme	Interoperability Digital object management Preservation
Preservation metadata	Checksum Preservation event	Interoperability Digital object management Preservation
Rights metadata	Copyright status License terms Rights holder	Interoperability Digital object management
Structural metadata	Sequence Place in hierarchy	Navigation
Markup languages	Paragraph Heading List Name Date	Navigation Interoperability

HTML <meta> Tag

Example

Describe metadata within an HTML document:

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
</head>
```

HTML <meta> Tag

- The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be **machine parsable**.
- Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- The metadata can be used by **browser**s (how to display content or reload page), search engines (keywords), or other web services.
- HTML5 introduced a method to let web designers take control over **the viewport** (the user's visible area of a web page), through the <meta> tag.

Run-Length Encoding (RLE)

Check out COLOR BY NUMBERS

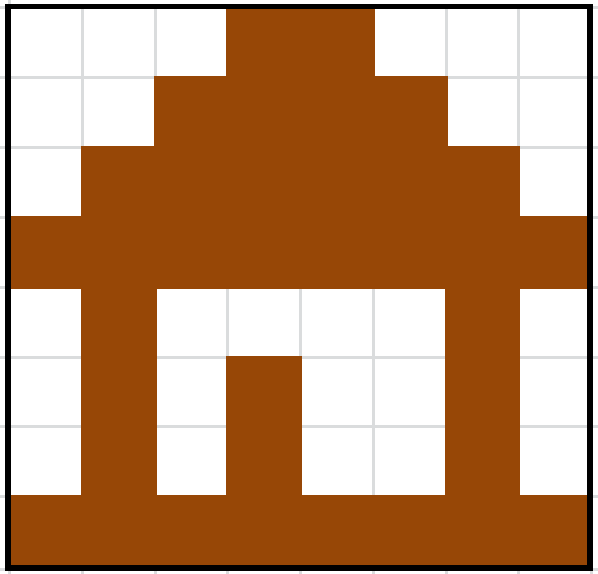
http://csunplugged.org/image-representation/#Colour_by_Numbers

It uses something called "run-length encoding"

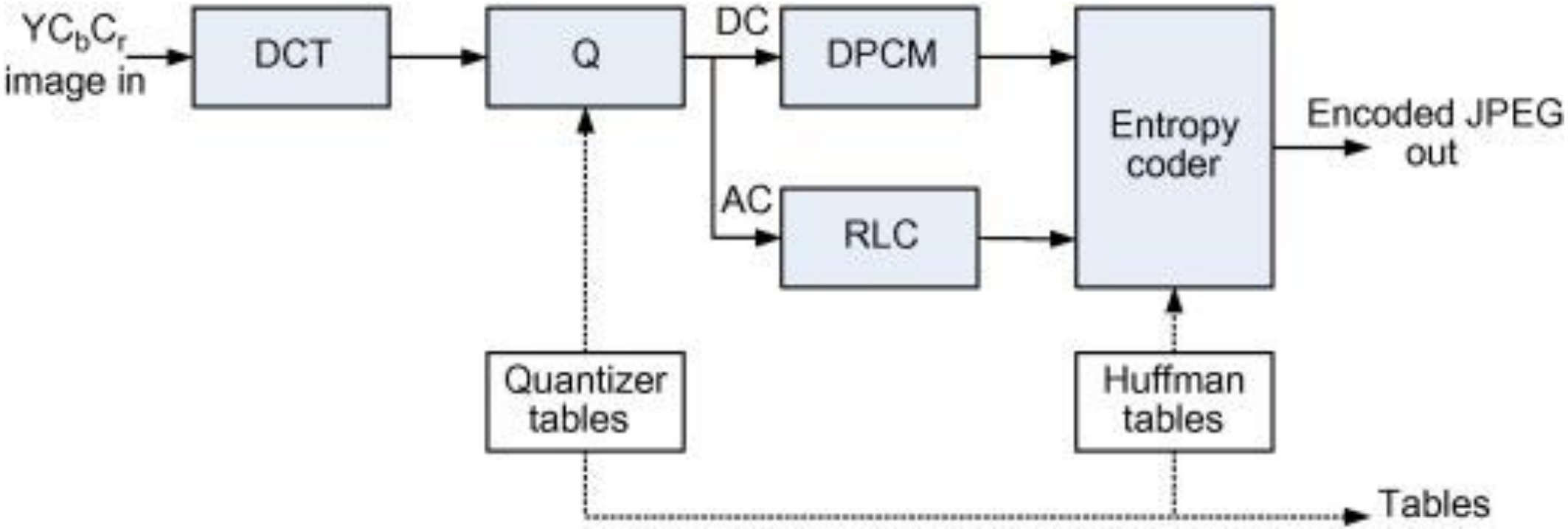
Run-length encoding

Run-length encoding (RLE) is a very simple form of lossless data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.

Consider the following picture with brown pixels(b) on white background(w)



w	w	w	b	b	w	w	w	3w2b3w
w	w	b	b	b	b	w	w	2w4b2w
w	b	b	b	b	b	b	w	1w6b1w
b	b	b	b	b	b	b	b	8b
w	b	w	w	w	w	b	w	1w1b4w1b1w
w	b	w	b	w	w	b	w	1w1b1w1b2w1b1w
w	b	w	b	w	w	b	w	1w1b1w1b2w1b1w
b	b	b	b	b	b	b	b	8b



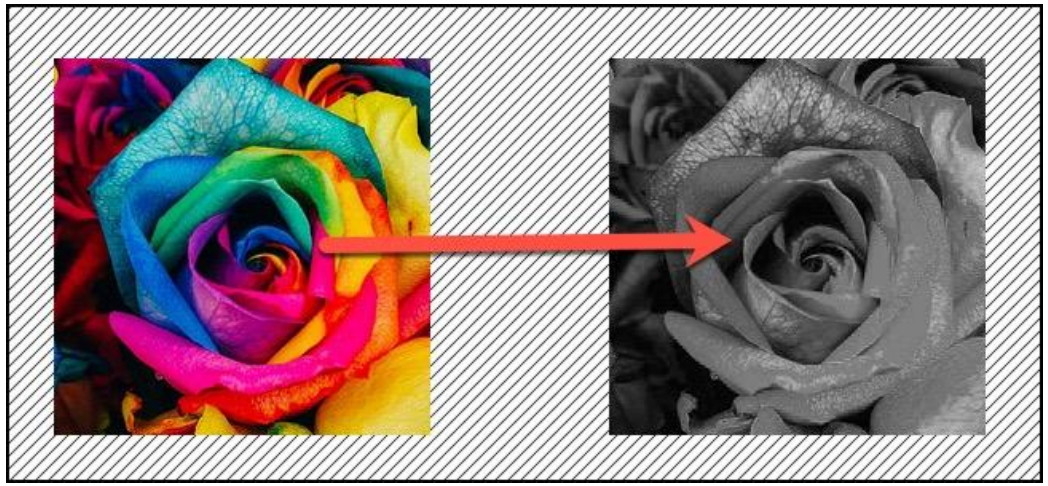
Conversion from Color Image to Black and White

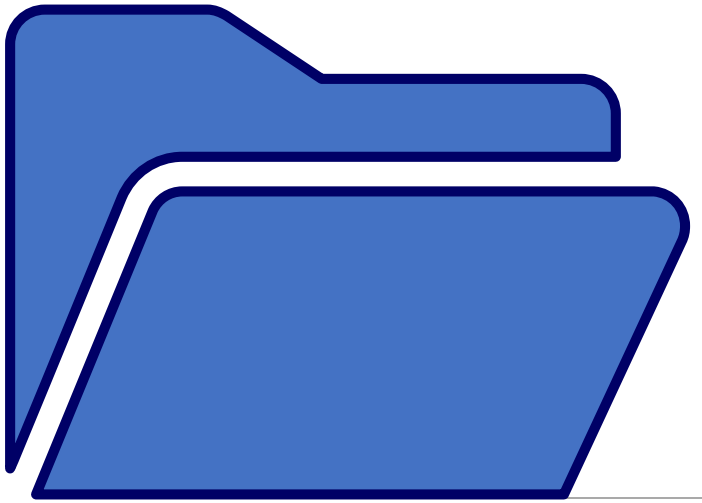
Related Information

https://www.cambridgeincolour.com/tutorials/color-black-white.htm

Online Tool:

https://manytools.org/image/black-and-white-or-sepia/





Color Images

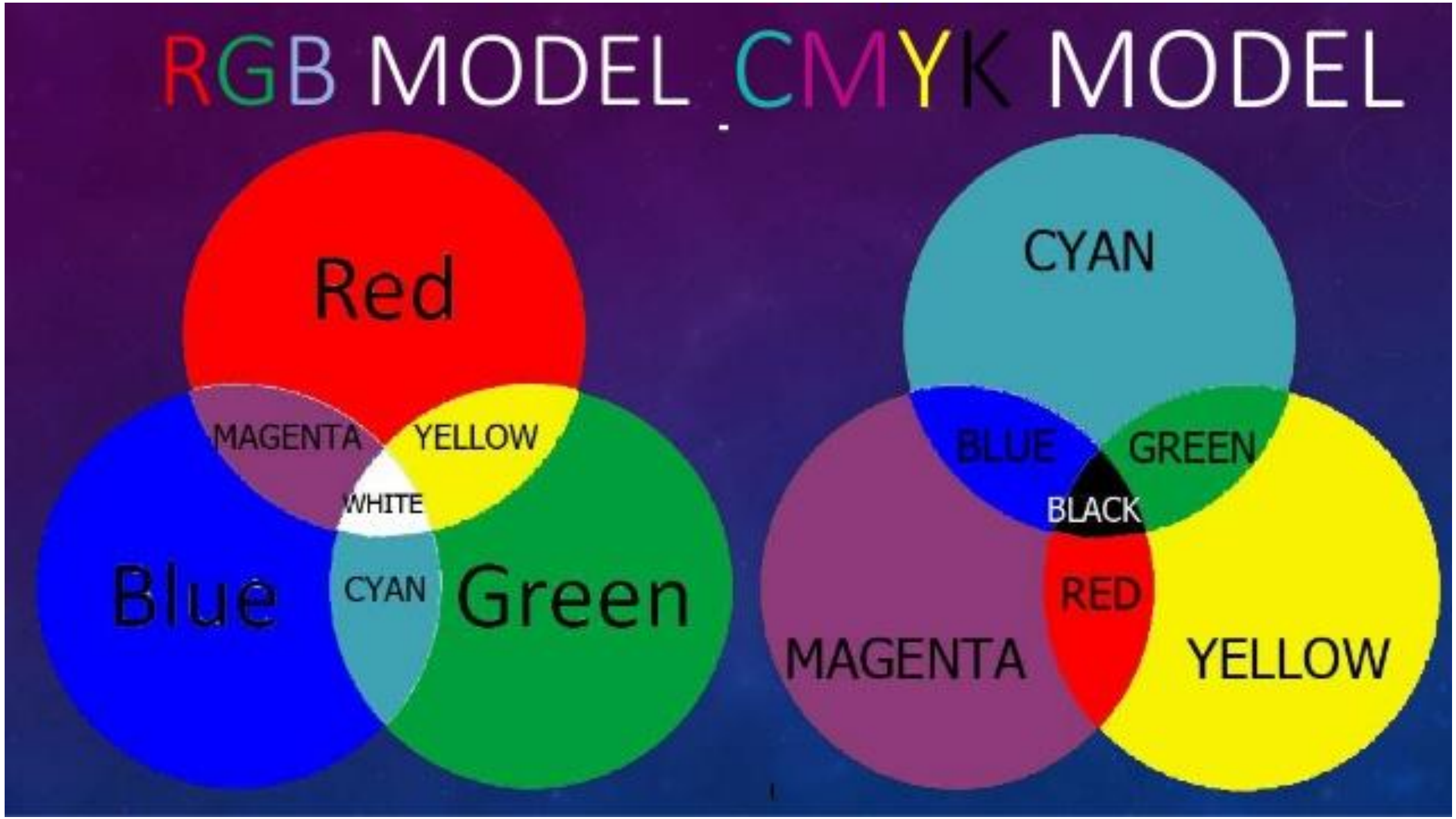
FILE TYPE 3: COLOR IMAGES

Vocabulary

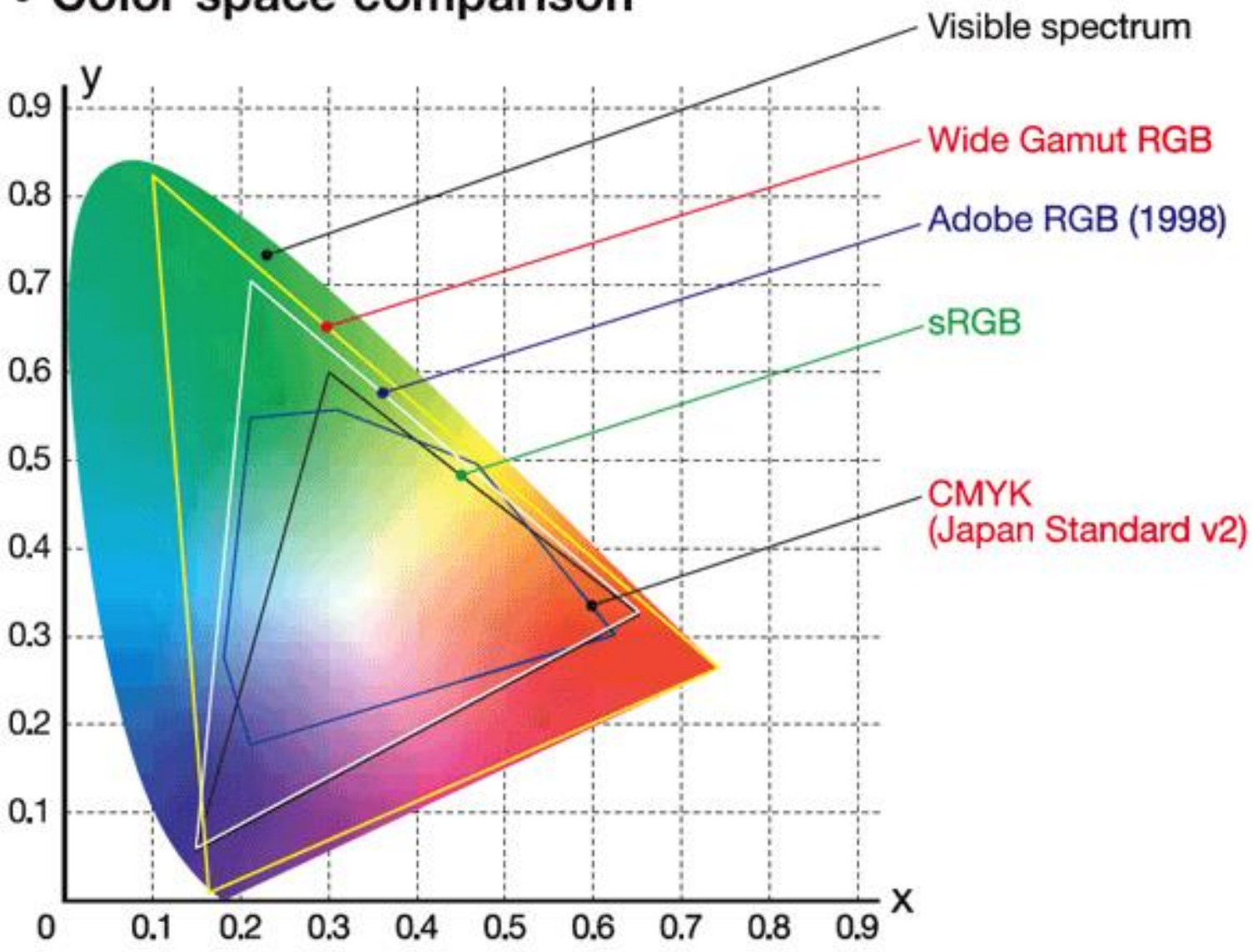
- **Hexadecimal** - A base-16 number system that uses sixteen distinct symbols 0-9 and A-F to represent numbers from 0 to 15.
- **Pixel** - short for "picture element", the fundamental unit of a digital image, typically a tiny square or dot that contains a single point of color of a larger image.
- **RGB** - 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.

Vocabulary

- **Favicon** - a small image, usually 16 by 16 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.


RGB MODEL CMYK MODEL
Red
Blue
Green
MAGENTA
YELLOW
CYAN
WHITE
CYAN
BLUE
GREEN
BLACK
RED
MAGENTA
YELLOW

• Color space comparison



Encoding Color

- How might you encode colors?
- In the previous lesson we came up with a simple encoding scheme for B&W images. What if we wanted to have color?

Color

- 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.

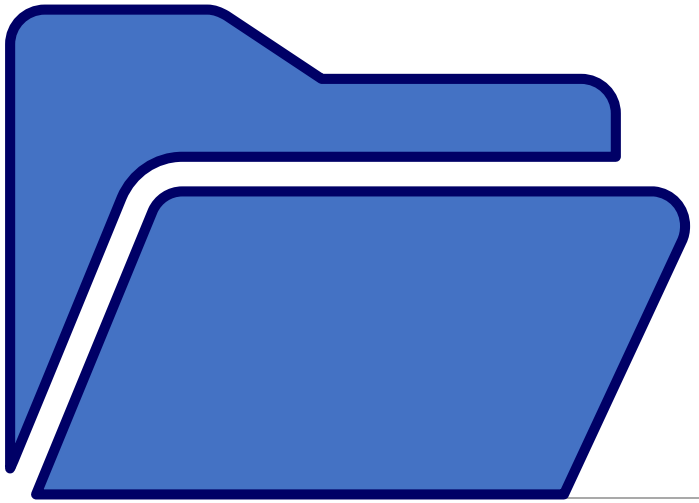
Video

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.

Tools

- 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/



Audio

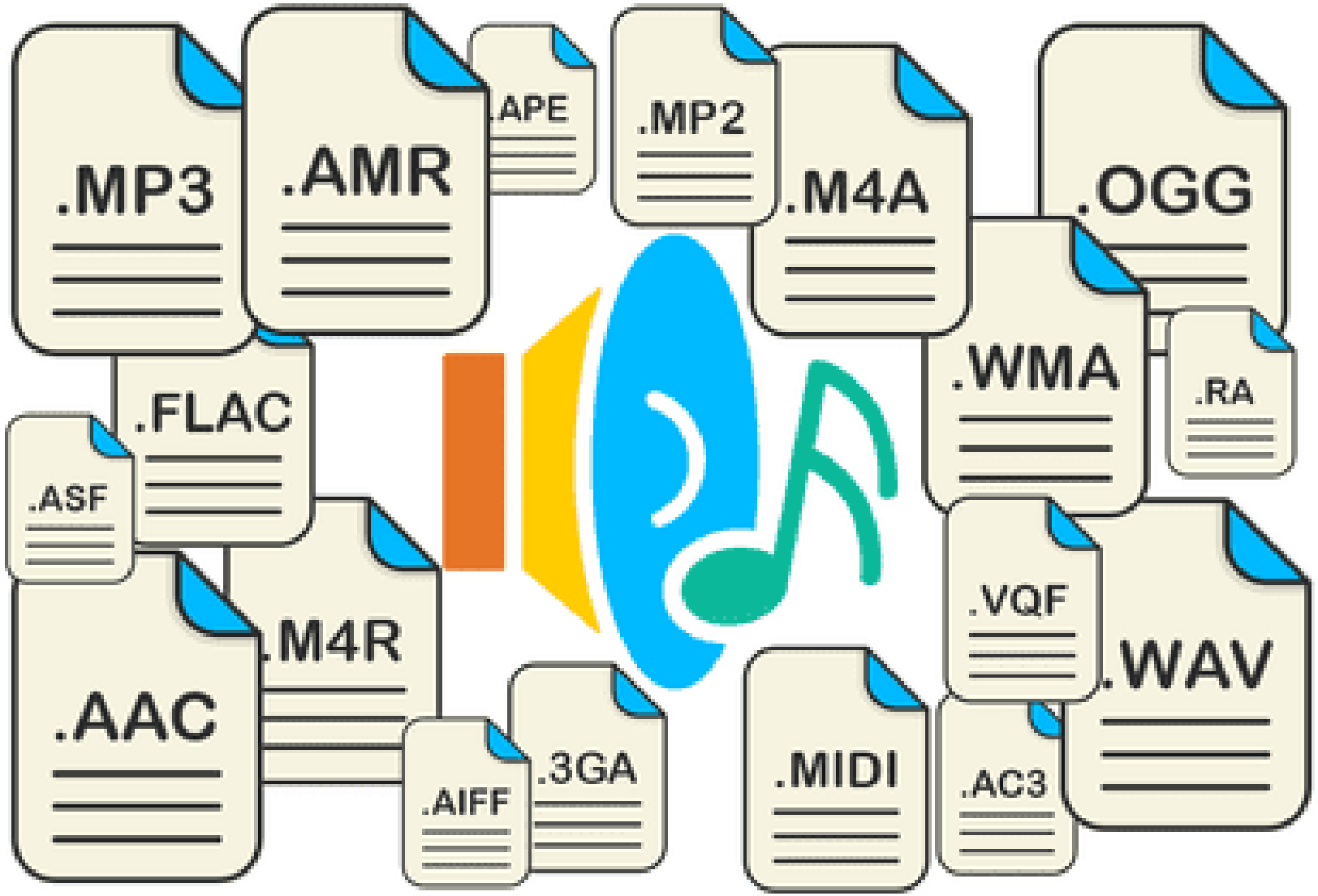
FILE TYPE 4: AUDIO

Vocabulary

- **Audio** - sound, especially when recorded, transmitted, or reproduced.
- **Sound** - vibrations that travel through the air or another medium and can be heard when they reach a person's or animal's ear.
- **Speech** - the expression of or the ability to express thoughts and feelings by articulate sounds.
- **Frequency** - frequency is the number of waves that pass a fixed place in a given amount of time.

Vocabulary

- **Amplitude** - n physics, the maximum displacement or distance moved by a point on a vibrating body or wave measured from its equilibrium position.
- **Period** - the interval of time between successive occurrences of the same state in an oscillatory or cyclic phenomenon, such as a mechanical vibration, an alternating current, a variable star, or an electromagnetic wave.
- **Wavelength** - the distance between successive crests of a wave, especially points in a sound wave or electromagnetic wave.


.MP3
.AMR
.APE
.MP2
.M4A
.OGG
.FLAC
.WMA
.RA
.ASF
.M4R
.AAC
.VQF
.WAV
.AIFF
.3GA
.MIDI
.AC3

Sound Vs Audio

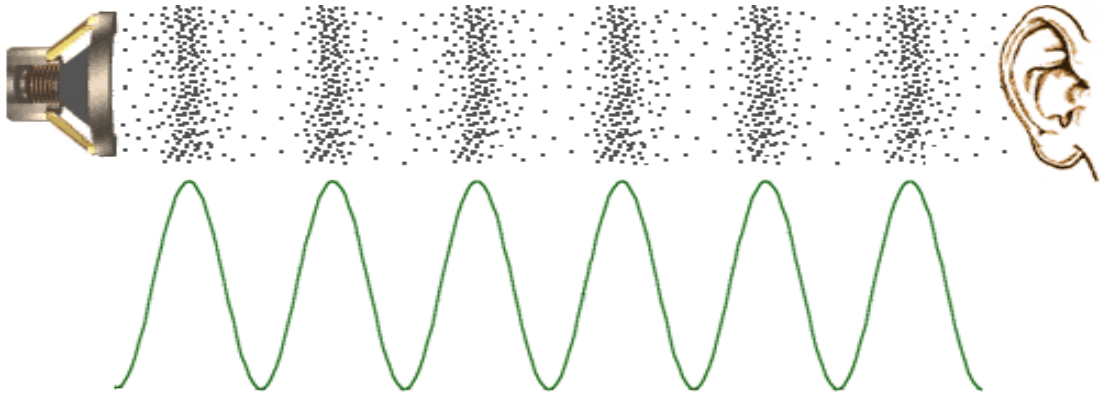
- **Audio** means the ***reproduction of sound***.

Classes of Sound:

- Voice
  - Defined as talking.
- Music
- Sound Effect:
  - Voice or Music; but often created by natural events like thunderclap, wind and door slamming.

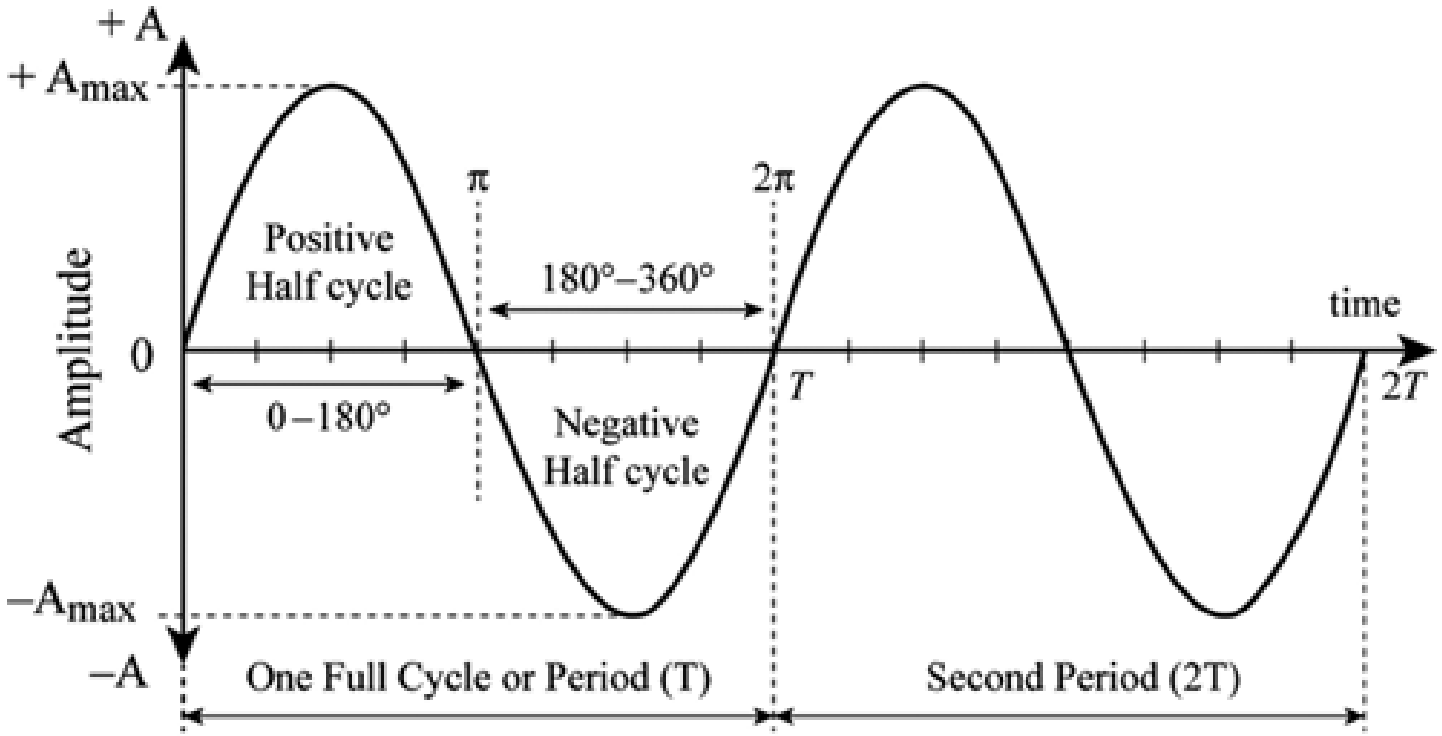
How do We Hear?

- Sound waves are variations of **pressure in a medium** such as air.
- Sound created by the **vibration** of an object, which causes the air surrounding it (medium) to vibrate.
- Vibrating air causes the human eardrum to vibrate, which the brain interprets as sound.



Properties of Sound

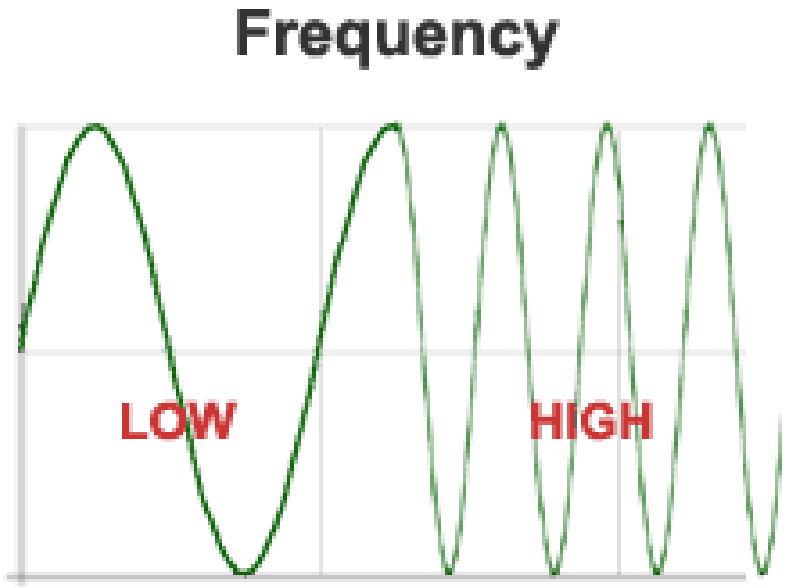
- frequency
- wavelength
- period
- amplitude
- speed



Frequency

Frequency: Number of times the wavelength occurs in one second.

- Measured in **Hertz** (Hz), or cycles per second.
- The faster the sound source vibrates, the higher the frequency, the higher the pitch
- Example: singing in a high-pitched voice forces the vocal chords to vibrate quickly.



Characteristics of digital sound

- Three main characteristics :
  - **Frequency**
    - defines the number of samples per second (or per other unit) taken from a continuous signal to make a discrete signal.
    - For time-domain signals, it can be measured in hertz (Hz).
  - **Sound resolution / Amplitude measurement**
    - Number of bits used to represent a sample.
  - **Channel**
    - Mono or stereo

Sound channel

- Whether you want mono or stereo sound will affect the size of the file.
- Mono means sound will be playing from one channel whereas stereo means two channels.
- Therefore, stereo sound will require larger storage space than mono sound.

Calculate audio data size

- The formula to calculate audio data size:
  - C = number of channels (mono = 1 , stereo = 2)
  - S = sampling rate in Hz (cycles per second)
  - T = Time (seconds)
  - B = bytes (1 for 8 bits, 2 for 16 bits)

File Size = C * S * T * B

Calculate audio data size

- Calculate a 30 seconds 16-bit, 44.1 kHz stereo music
  - **Step 1**

    44,100 x 2 bytes (or 16-bits) = 88,200 bytes
  - **Step 2**

    88,200 x 2 (for stereo) = 176,400 bytes
  - **Step 3**

    176,400 x 30 seconds = 5,292,000 bytes

Benefits of using digital audio

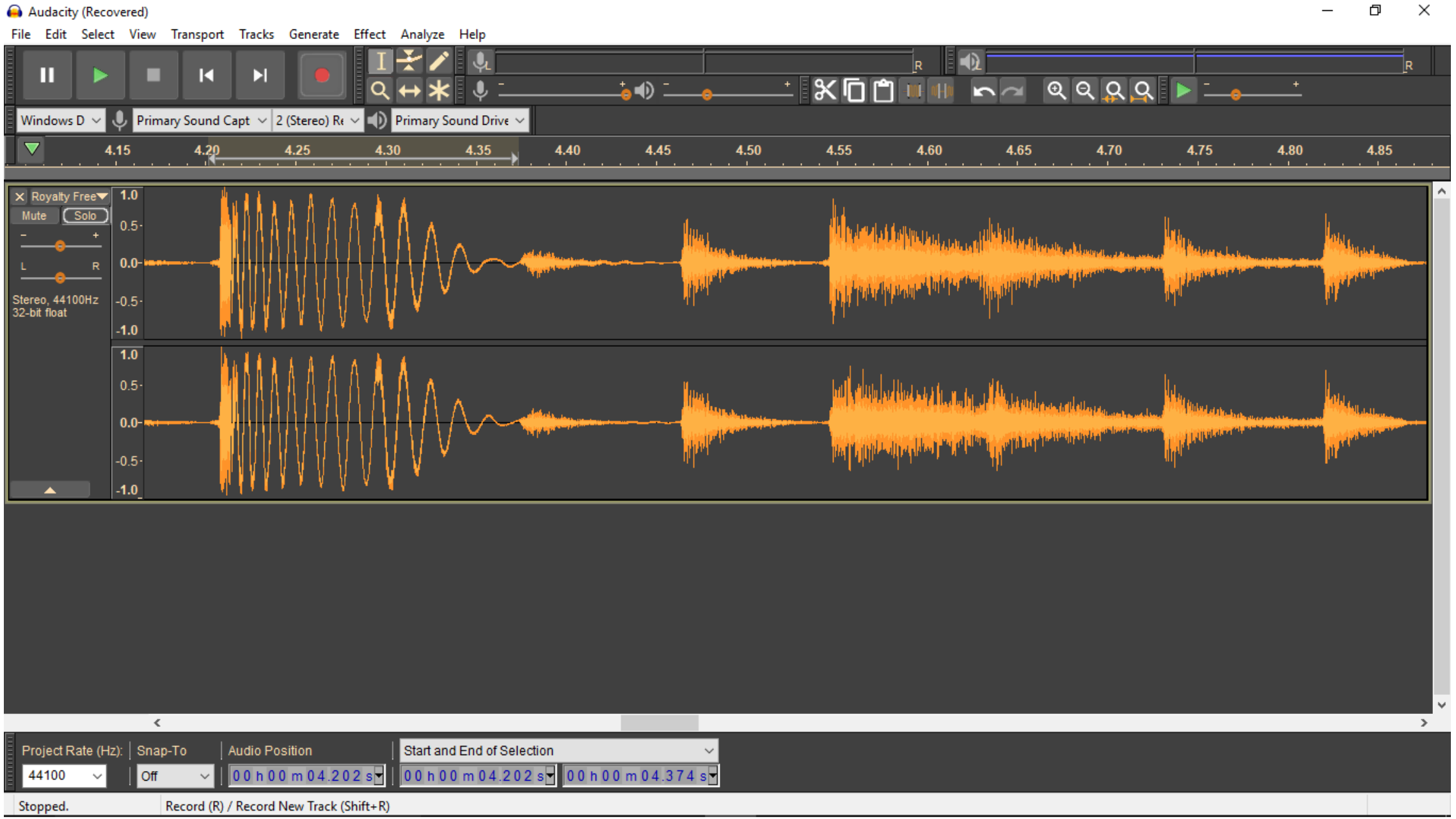
- Sound can be permanently stored in inexpensive CD.
- Consistent sound quality without noise or distortion.
- Duplicate will sound exactly the same as the master copy.
- Digital sound can be played at any point of the sound track. (random access)
- It can also be integrated with other media.
- Can be edited without loss in quality.

Audio file formats

Extension	Use (Note: popular formats)
wav	WAV Audio
aiff	Audio (common for Macintosh)
aac	Audio (Compressed)
ra	Real Audio (stream)
mov	QuickTime video
mp3	MP3 Audio

Sound in Multimedia Application

- It captures attention.
- It increases the associations the end-user makes with the information in their minds.
- Sound adds an exciting dimension to an otherwise flat presentation.
- Example usage of sound in multimedia application.
  - Background music
  - Sound effects
  - Voice over or narration



Editing Digital Recording

- There are abundance of sound editor available such as SoundForge (commercial), Goldwave (shareware), and **Audacity**(freeware).
- The basic sound editing operations that most commonly needed are:
  - Trimming
  - Splicing and Assembly
  - Volume adjustment
  - Format conversion
  - Resampling and Down-sampling
  - Fade-ins and Fade-outs
  - Equalization
  - Time Stretching
  - Digital Signal Processing
  - Reversing Sound

Editing Digital Recording

Trimming

- Removing “dead air” or silence space from the front of recording to reduce file size.

Splicing and Assembly

- Cutting and Pasting different recording into one.

Volume adjustment

- If you combining several recordings into one there is a good chance that you won’t get a consistent volume level. It is best to use a sound editor to normalize the combined audio about 80% – 90% of the maximum level. If the volume is increased too loud, you will hear a distortion.

Editing Digital Recording

Format conversion

- Saving into different file formats.

Resampling and Downsampling

- If you have recorded your sounds at 16-bit sampling rates, you can downsample to lower rates by downsampling the file to reduce the file size.

Fade-ins and Fade-outs

- To smooth the beginning and the end of the sound file by gradually increasing or decreasing volume.

Equalization

- Some program offer digital equalization capabilities to modify the bass, treble or midrange frequency to make the audio sounds better.

Editing Digital Recording

Time stretching

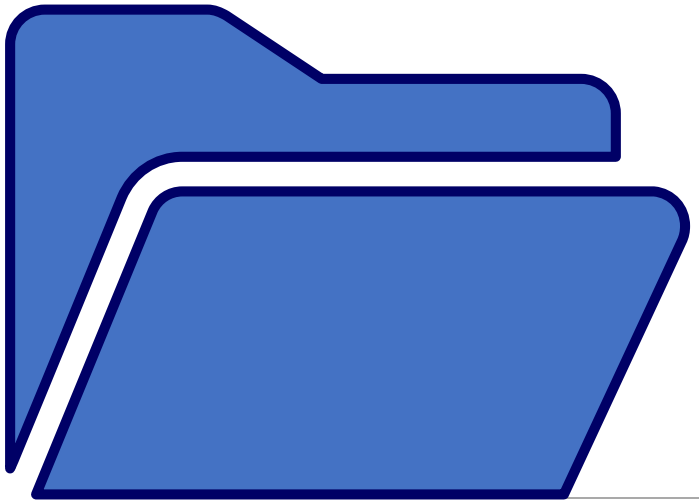
- Alter the length (in seconds) of a sound file without changing its pitch.

Reversing sound

- Spoken dialog can produce a surreal effect when played backward.

Digital Signal Processing (Special Effect)

- To increase pitch, robot voice, echo, and other special effects.



Video

FILE TYPE 5: VIDEO

Vocabulary

https://vimeo.com/blog/post/glossary-of-common-video-terms

Aspect Ratio: is the relationship between the width and the height of your video dimensions expressed as a ratio. The most common aspect ratios for video are 4:3, 16:9 and 1.85:1. Check out the diagram below for an approximation of those ratios.

Bit rate (also known as data rate) is the amount of data used for each second of video. In the world of video, this is generally measured in kilobits per second (kbps), and can be constant and variable. For more information, check out our lesson on Video Compression.

Codec is the method a computer uses to determine the amount of change between frames of a video. For more information, check out our lesson on Video Compression.

Resolution is a measure of the number of pixels a video contains both horizontally and vertically. Some common resolutions are 640x480 (SD) 1280x720 (HD), 1920x1080 (HD). Sometimes these are referred to just by their vertical dimension such as, 480p, 720p or 1080p. For more information, check out our lesson on The Basics of Image Resolution.

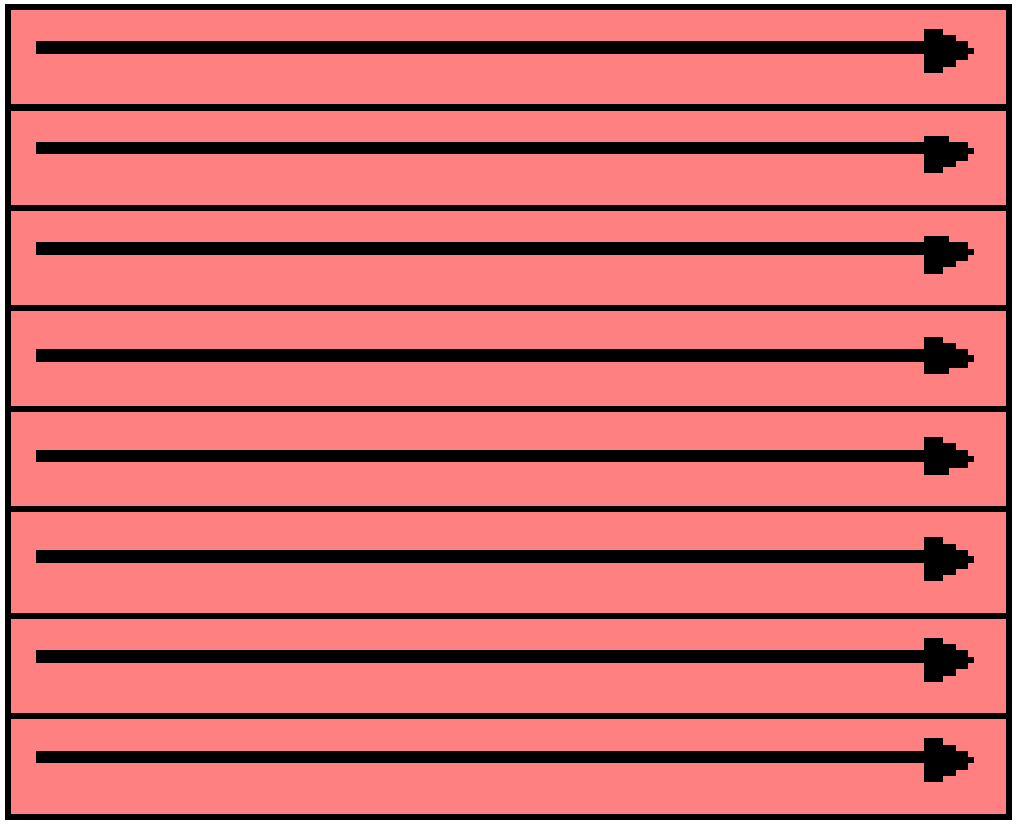
Vocabulary

https://vimeo.com/blog/post/glossary-of-common-video-terms

Video provides the magical elixir that keeps Vimeo's heart beating, its engine running, and its soul soaring.

Frame Rate is the rate at which a shutter opens and closes, or a sensor captures video during one second. Typical frame rates are 24, 25, and 29.97, 30 and 50 and 60. For more information, check out our lesson on Frame Rate Vs. Shutter Speed.

Interlaced



Progressive Scan
(Non-interlaced)

Evolution to Next Generation TVs

- Design: Zero Bezel, Curved TV, and Cinemascope
- Picture Quality: Glasses-Free 3D and High-Color Gamut
- Display: OLED, Transparent, Flexible, and 4K8K

Emergence of New Categories: 3D and Smart TVs

- Screen Size Increased from 55" to 110"
- Design: Super Narrow Bezel
- Picture Quality: 3D and UHD
- Smart TV

New Display Concept

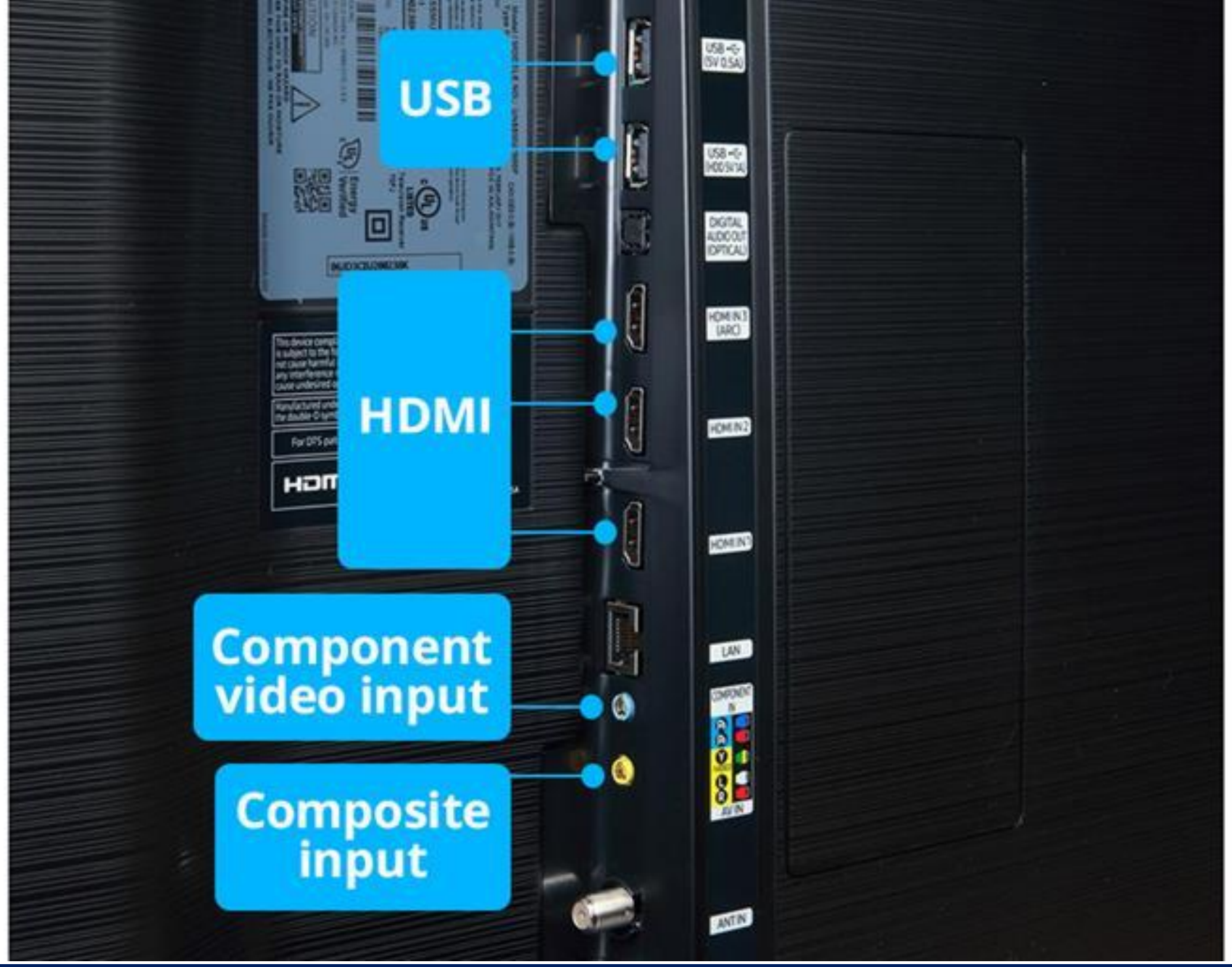
- Analog to Digital
- Screen Size Increased to 55"
- Design: Slim and from CCFL to LED
- Picture Quality: SD, HD, and UHD

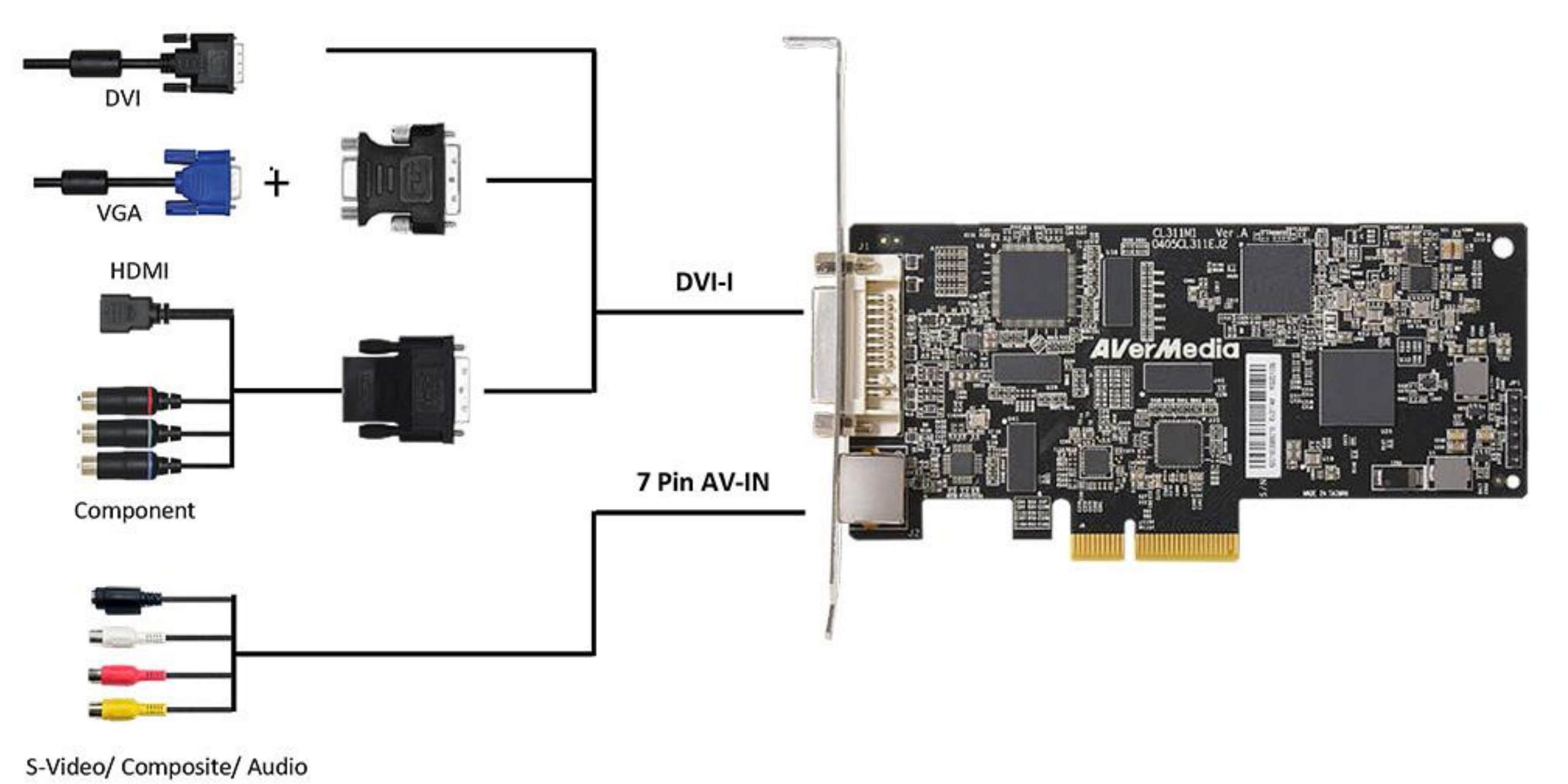

AVI
ASF
MOV
AVCHD
DivX
FLV
MPEG
MP4
WMV
H.264
MKV
SWF

Video File Support						
File Extention	**Container**	**Video Codec**	**Resolution**	**Frame Rate (fps)**	**Bit Rate (Mbps)**	**Audio Codec**
*.avi		DivX 3.11 / 4 / 5 / 6	1920×1080 (WMV v7,v8, MSMPEG4 v3 : 1280×720)	6~30	30	Dolby Digital LPCM ADPCM (IMA, MS) AAC WMA Dolby Digital Plus MPEG(MP3) DTS (Core)
*.mkv						
*.asf		MPEG4 SP/ASP				
*.wmv						
*.mp4	AVI	H.264 BP/MP/ HP				
*.3gp	MKV					
*.vro	ASF	Motion JPEG				
*.mpg	MP4					
*.mpeg	3GP	Microsoft MPEG-4 v3				
*.ts	MOV					
*.tp	FLV	Window Media Video v7,v8,v9				
*.trp	VRO					
*.mov	VOB	MPEG2				
*.flv	PS					
*.vob	TS	MPEG1				
*.svi	SVAF					
*.m2ts		VP6	640×480		4	
*.mts						
*.divx		MVC		24/25/30	60	
*.webm	WebM	VP8	1920×1080	6~30	20	Vorbis

Comparison of YouTube media types

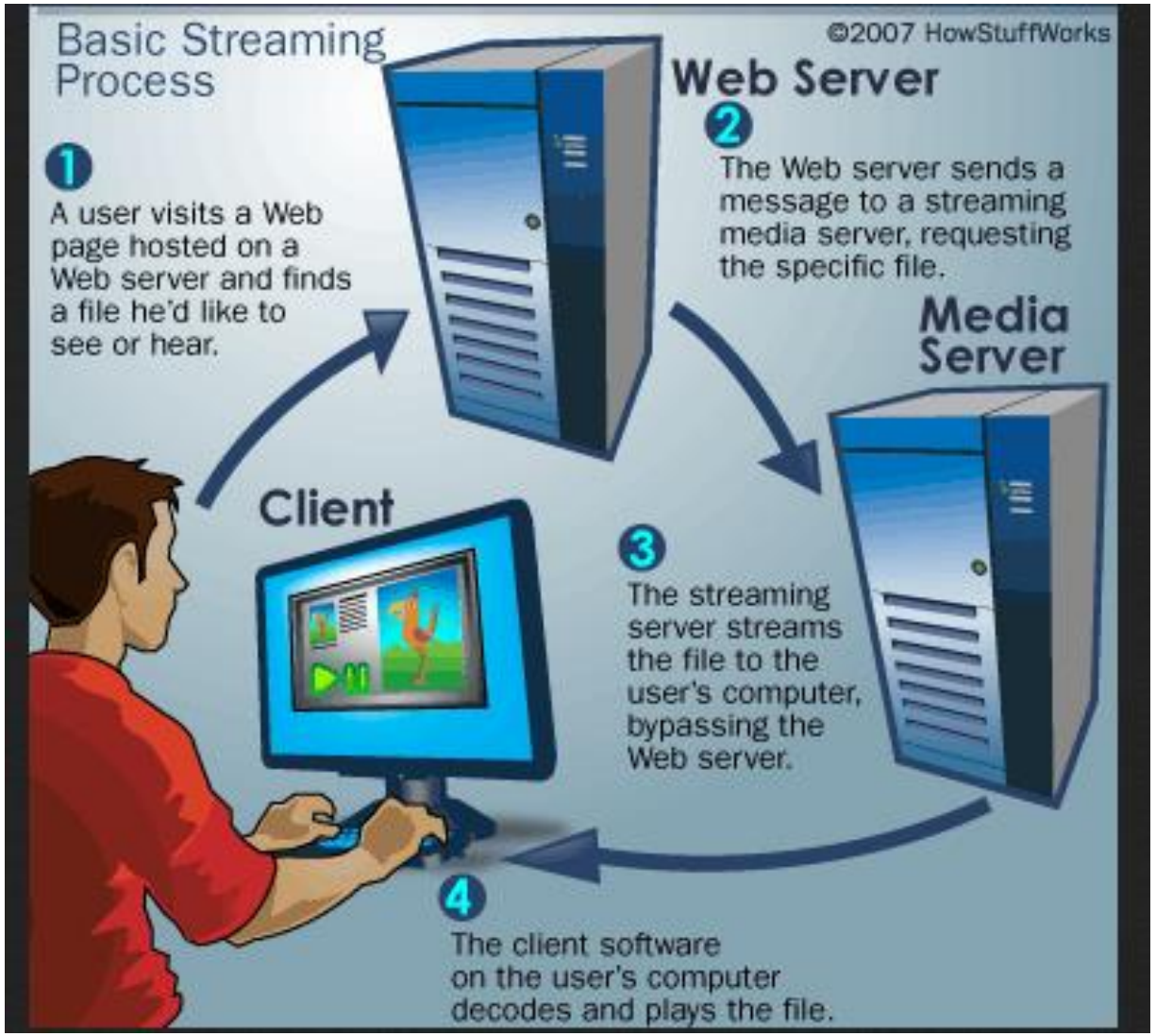
		Standard	Medium	High	720p	1080p	Mobile	Old formats (pre Feb 2009)		
								Standard	High	Mobile
fmt value		34	18	35	22	37	17	0, 5	6	13
Container		FLV	MP4	FLV	MP4		3GP	FLV		3GP
Video	**Encoding**	MPEG-4 AVC (H.264)					MPEG-4 Visual	H.263		
	Max Res Aspect ratio	4:3		16:9			11:9	4:3		11:9
	Max Resolution	640×480	480×360	854×480	1280×720	1920×1080	176×144	320×240	480×360	176×144
Audio	**Encoding**	AAC						MP3		AMR
	Channels	2 (stereo)						1 (mono)		
	Sampling rate (Hz)	44100						22050	44100	8000

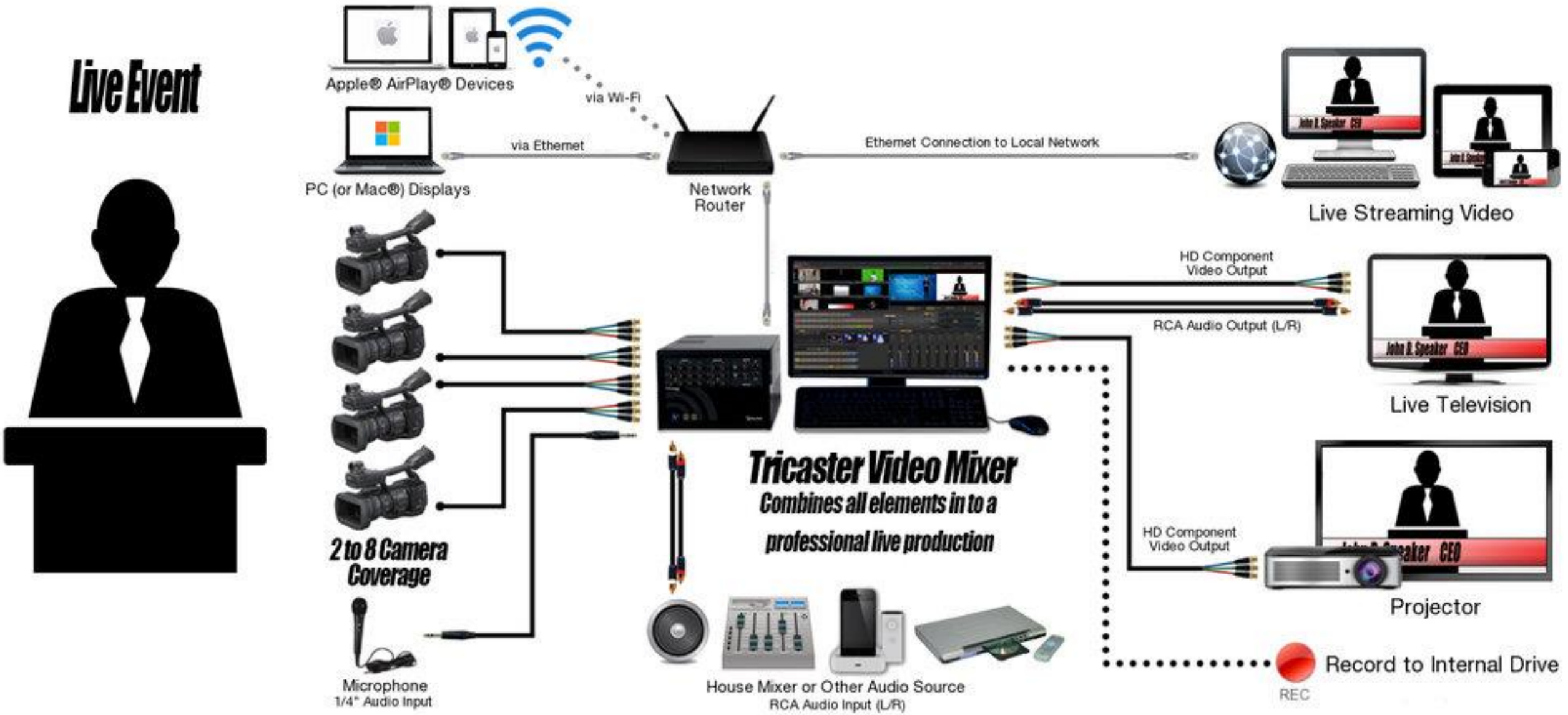



DVI
VGA
+
HDMI
Component
DVI-I
7 Pin AV-IN
S-Video/ Composite/ Audio
AVerMedia


Video
Streaming
YOU
Tube


How Streaming Video & Audio Work
Basic Streaming Steps
©2007 HowStuffWorks
1 Record
2 Digitize, compress and encode
3 Upload (to a server)
4 Stream (to a user's computer)
5 Decode

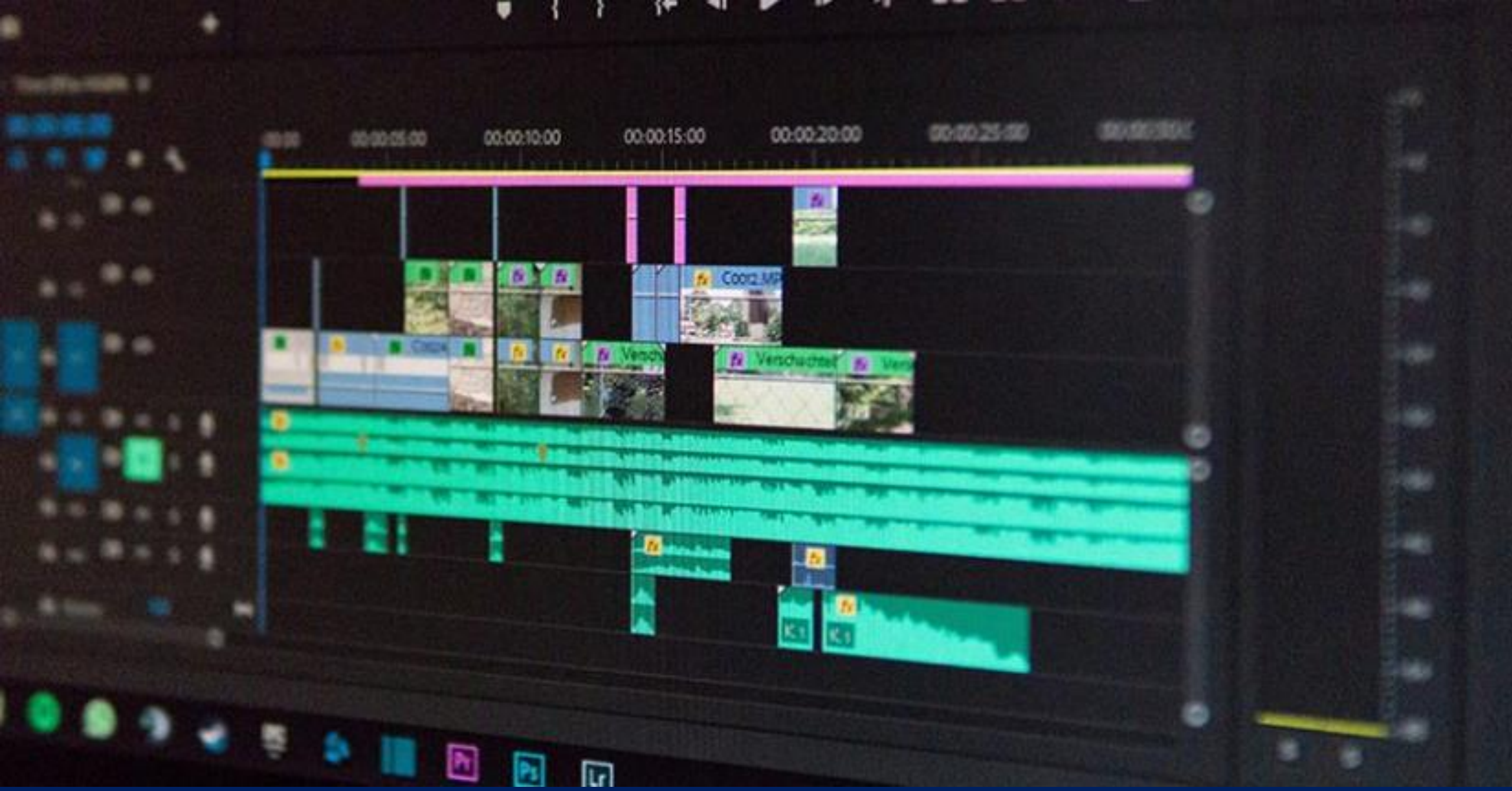

Basic Streaming Process
©2007 HowStuffWorks
Web Server
1 A user visits a Web page hosted on a Web server and finds a file he'd like to see or hear.
2 The Web server sends a message to a streaming media server, requesting the specific file.
Media Server
Client
3 The streaming server streams the file to the user's computer, bypassing the Web server.
4 The client software on the user's computer decodes and plays the file.



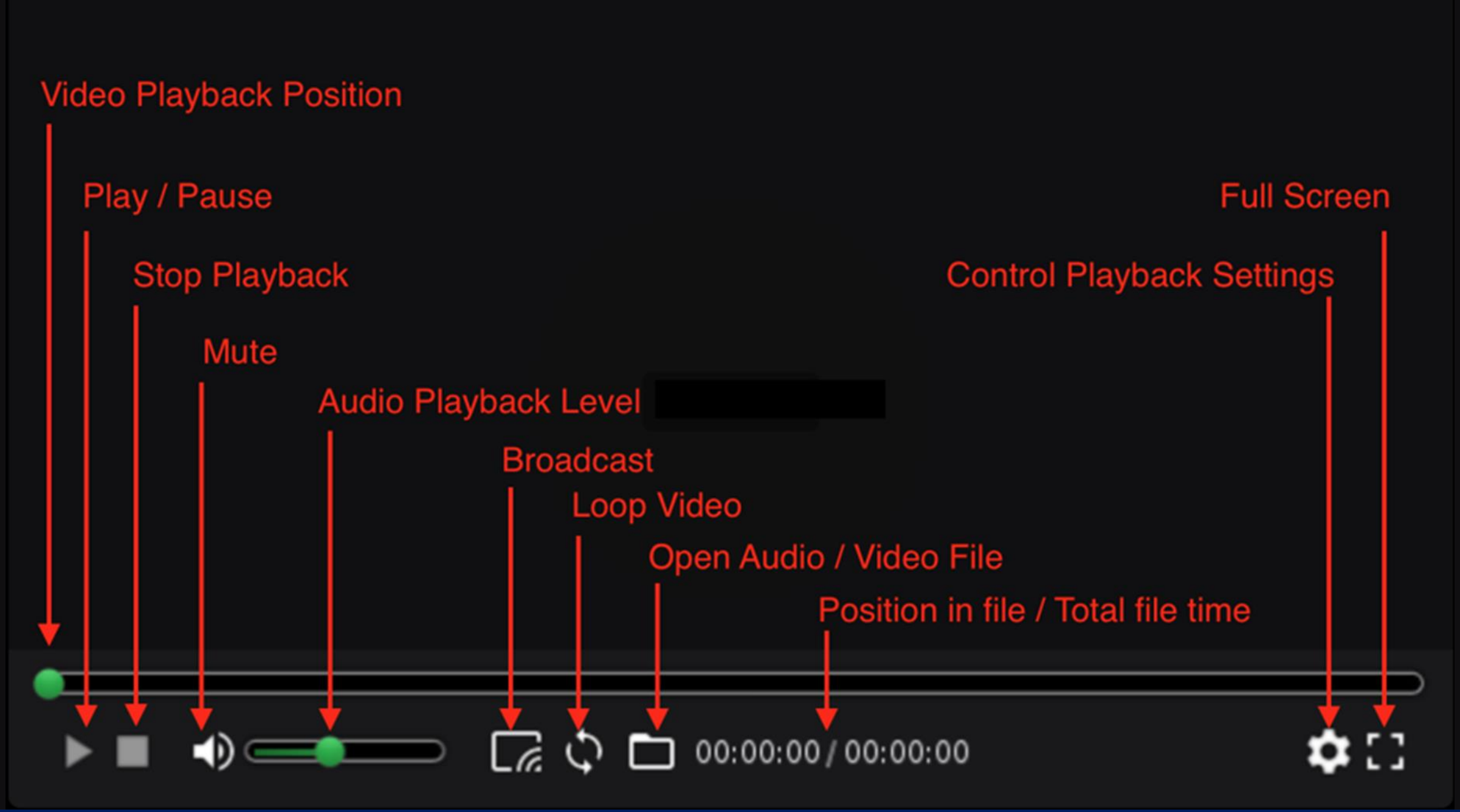
For workflow reference only. Example system configuration shown.

What is Digital Video Production?

- Video Recording
- Video Editing
- Video Hosting
- Video Engineering
- Video/Audio Playback


00:00:10:00
00:00:15:00
00:00:20:00

Video Hosting?


Video Playback Position
Play / Pause
Stop Playback
Mute
Audio Playback Level
Broadcast
Loop Video
Open Audio / Video File
Position in file / Total file time
Full Screen
Control Playback Settings
00:00:00 / 00:00:00

Roles and Responsibilities of a Typical Video Engineer

A Typical Video Engineer role is in charge of researching, creating, and keeping up programming modules identified with handling and enhancement of incoming videos.

- Experience with gushing conveyance sorts (HLS, MPEG-DASH)
- Build video transcoding work processes, upgrade video encoding profiles, and streamline the execution of the organization's restrictive video preparing motor
- Developing video preparing calculations utilizing C and C++;

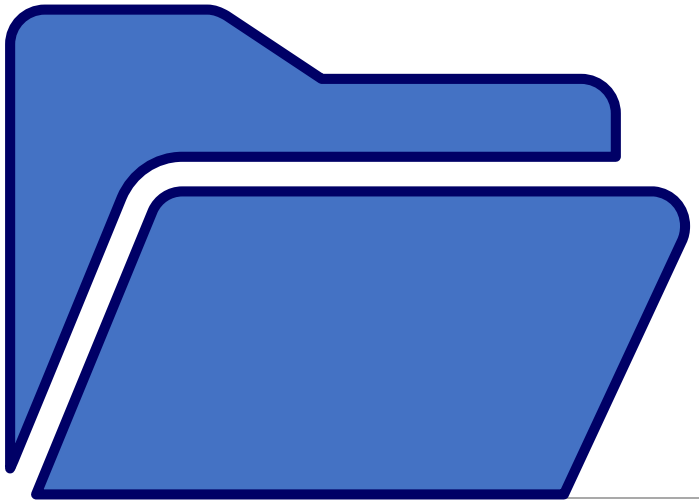
Roles and Responsibilities of a Typical Video Engineer

- Create tweaked video pressure profiles utilizing open-source video handling instruments, including: FFmpeg, libav and other comparable devices
- Implement calculations to distinguish scene softens up discretionary video content in an exceptionally performant way;
- Assist colleagues with planning machine vision applications to identify certifiable articles in video information and upgrades web customer video unraveling processor execution;
- Research programming modules identified with preparing and improvement of countless video records; and,
- Maintain actualized programming modules to guarantee media administrations work processes are ideal and persistently creative.

Demonstration

Creation of a short video clip (2mins):

1. Transcript
2. Audio
3. Video
4. Audio/Video Integration
5. Background Music or Animation
6. Video Rendering
7. Video Sharing



Graphics

FILE TYPE 6:GRAPHICS

Vocabulary

- **Vector Graphics** - Graphics that are based on mathematical formulas and are comprised of paths connected by anchor points that define lines, shapes, and curves.
- **Scalable** - Able to change size easily without loss of quality.
- **Resolution independent** - Regardless of how much the image is enlarged or reduced, the image definition and quality remain the same.
- **EPS** - Encapsulated Postscript Developed by Adobe, but supported by most programs. Most common interchange format for the print industry due to its portability.

Vocabulary

- **Meta graphic** - graphic formats that can contain both vector and raster data.
- **SVG - Scalable Vector Graphics** - Open standard developed by the W3C. All purpose vector format. Works well with web page design.
- **WMF - Windows Metafile** - Microsoft created format for raster and vector. Common format for Windows Clipart.

Vector Graphics

- Vector graphics is the creation of digital images through a sequence of commands or mathematical statements that place lines and shapes in a given two-dimensional or three-dimensional space. In physics, a **vector** is a representation of both a quantity and a direction at the same time.
- In vector graphics, the file that results from a graphic artist's work is created and saved as a sequence of vector statements.
- For example, instead of containing a bit in the file for each bit of a line drawing, a vector graphic file describes a series of points to be connected. One result is a much smaller file.

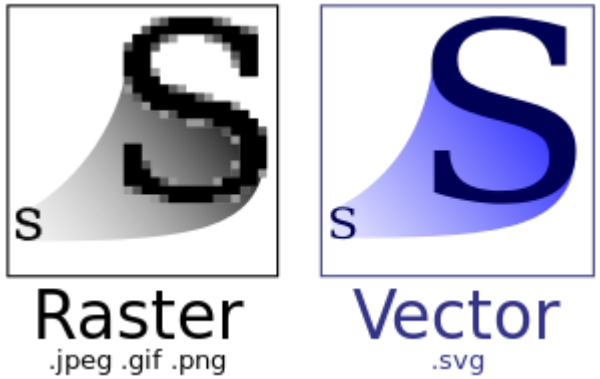
Vector Graphics

- At some point, a vector image is converted into a raster graphics image, which maps bits directly to a display space (and is sometimes called a *bitmap*). The vector image can be converted to a raster image file prior to its display so that it can be ported between systems.
- A vector file is sometimes called a **geometric** file. Most images created with tools such as Adobe Illustrator and CorelDraw are in the form of vector image files. Vector image files are easier to modify than raster image files (which can, however, sometimes be reconverted to vector files for further refinement).
- Animation images are also usually created as vector files. For example, Shockwave's Flash product lets you create 2-D and 3-D animations that are sent to a requestor as a vector file and then rasterized "on the fly" as they arrive.

Vector Graphic (or Icon) Library

Vector Graphics Libraries

- https://www.flaticon.com/
- https://publicdomainvectors.org/en/tag/svg
- https://www.clipartmax.com/
- https://www.creativebloq.com/graphic-design/best-places-free-vector-art-1012985



What Is an SVG File?

How to open, edit, and convert SVG files?

- A file with the SVG file extension is most likely a Scalable Vector Graphics file. Files in this format use an XML-based text format to describe how the image should appear.
- Since text is used to describe the graphic, an SVG file can be scaled to different sizes without losing quality — in other words, the format is *resolution independent*. This is why website and print graphics are often built in the SVG format, so they can be resized to fit different designs in the future.
- If an SVG file is compressed with GZIP compression, the file will end with the .SVGZ file extension and may be 50 percent to 80 percent smaller in size.
- Other files with the .SVG file extension that aren't related to a graphics format may instead be Saved Game files. Games like "Return to Castle Wolfenstein" and "Grand Theft Auto" save the progress of the game to an SVG file.

How to Open an SVG File?

- The easiest and quickest way to open an SVG file to view it (not to edit it) is with a modern web browser like Chrome, Firefox, Edge, or Internet Explorer — nearly all of them should provide some sort of rendering support for the SVG format. This means you can open online SVG files without having to download them first.
- If you do already have an SVG file on your computer, the web browser can also be used as an offline SVG viewer. Open those SVG files through the web browser's *Open* option (the **Ctrl+O** keyboard shortcut).
- SVG files can be created through Adobe Illustrator, so you can, of course, use that program to open the file. Some other Adobe programs that support SVG files (so long as the SVG Kit for Adobe CS plug-in is installed) include Adobe Photoshop, Photoshop Elements, and InDesign programs. Adobe Animate works with SVG files, too.

C:\Users\Jon\Downloads\car.svg - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

car.svg

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->
<svg
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:cc="http://web.resource.org/cc/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:svg="http://www.w3.org/2000/svg"
   xmlns="http://www.w3.org/2000/svg"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
   xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"
   width="900"
   height="600"
   id="svg2"
   sodipodi:version="0.32"
   inkscape:version="0.44+devel"
   version="1.0"
   sodipodi:docbase="/home/d/ink/inkscape/share/examples"
   sodipodi:docname="car.svgz"
   inkscape:output_extension="org.inkscape.output.svgz.inkscape"
   inkscape:export-filename="/home/hrum/Desktop/carrr/killxl_1.png"
   inkscape:export-xdpi="100"
   inkscape:export-ydpi="100"
   sodipodi:modified="true">
  <defs
     id="defs4">
    <linearGradient
       inkscape:collect="always"
       id="linearGradient4399">
      <stop
         style="stop-color:black;stop-opacity:1;"
```

eXtensible Markup Language file | length : 527,014 lines : 7,416 | Ln : 1 Col : 1 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

How to Open an SVG File?

- Some non-Adobe programs that can open an SVG file include Microsoft Visio, CorelDRAW, Corel PaintShop Pro, and CADSoftTools ABViewer.
- Inkscape and GIMP are two free programs that can work with SVG files, but you must download them in order to open the SVG file. Picozu is also free and supports the SVG format, too, but you can open the file online without downloading anything.
- Since a Scalable Vector Graphics file is really a text file in its details, you can view the text version of the file in any text editor. See our Best Free Text Editors list for our favorites, but even the default text reader in your operating system would work, like Notepad in Windows.

How to Open an SVG File?

- For Saved Game files, the game that created the SVG file most likely uses it automatically when you resume the gameplay, which means you probably can't manually open the SVG file through the program's menu. However, even if you do manage to get the SVG file to open through an *Open* menu of some sort, you have to use the right SVG file that goes with the game that created it.
- If the game itself won't open the SVG file, try GTA2 Saved Game Editor, or open the SVG file in a text editor to see if there's something there that's of use.

Activity

- Download a .svg file and open it with Chrome Browser and Inkscape.

How to Convert an SVG File?

- The easiest way to convert an SVG file to either PNG or JPG, the two most common image formats, is to use our own SVG file converter:
- Converting an SVG file with an online tool like ours is usually the quickest and easiest way to get your file into the format you want. No need to already have an expensive program installed or to download unfamiliar software.
- If you need to convert it to a different format, like PDF or GIF, and your SVG is pretty small, then a third-party online tool like Zamzar will do the trick.
- Autotracer.org is another online SVG converter which lets you convert an SVG (from your device or through its URL) to some other types of formats like EPS, AI, DXF, PDF, SK, etc.

9 Great Programs to Convert Images to Other Formats

- If you have a larger SVG file, any software programs mentioned above in the *How to Open an SVG File* section should be able to save/export the SVG file to a new format, too.
- For example, if you're using Inkscape, after you open/edit the SVG file you can then save it back to SVG with any changes you make, but can also save it to a different file format like PNG, PDF, DXF, ODG, EPS, TAR, PS, HPGL, and many others.

More Information on SVG Files

The Scalable Vector Graphics format was created in 1999 and is still being developed by the World Wide Web Consortium (W3C).

Like you've already read above, the entire contents of an SVG file is just text. If you were to open one in a text editor, you would see only text like in the example above. This is how SVG viewers are able to show the picture — by reading the text and understanding how it should be displayed.

Looking at that example, you can see how easy it is to edit the dimensions of the image to make it as large as you want without affecting the quality of the edges or color. Since the instructions for rendering the image can be easily altered in an SVG editor, so too can the image itself.