

Computer Science Principles

Unit 2: Data and Information



LECTURE 2: DATA REPRESENTATION (U1)

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.



Binary Numbers

LECTURE 1



Binary Numbers

- 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.
- As we've discussed before, computers run on data, but how do these computing devices store this data?
- There are many different ways to answer this question. You've seen how data can be stored in spreadsheets and charts for humans to use. When coding, data is also stored in variables, lists, or as a constant value for programs to work with. (We'll discuss variables, lists, and constants more in the next guide!)



Binary Numbers

- At their core, computers store data in **bits**, or binary **digits**.
- Before we talk about how binary is used, we first have to understand what it is and how it works. To do that, we need to talk about number bases.



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.

Here's a chart demonstrating this system:

Power of 10	10^3	10^2	10^1	10^0
Value represented (base 10)	1000	100	10	1
Amount of value	5	7	2	9



Number Bases

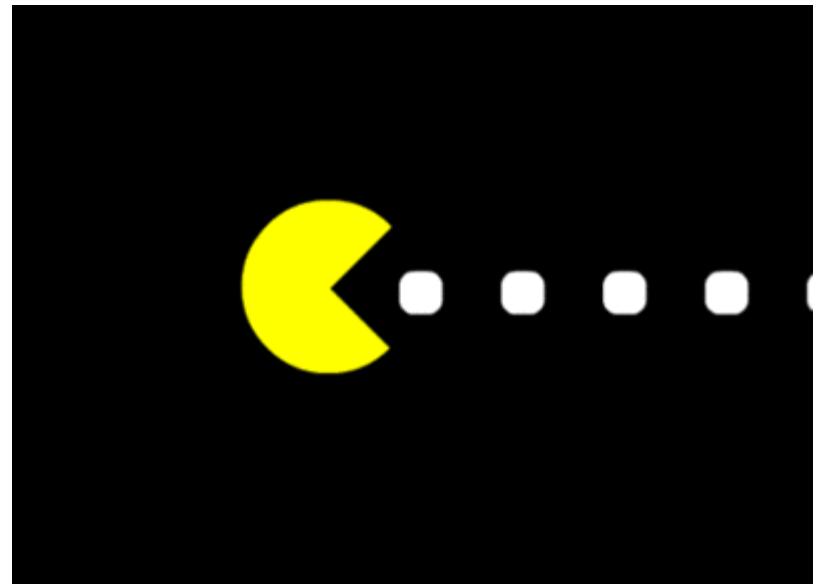
- Each place value represents a different power of ten and can only hold a value of up to 9. After that, the place reverts to zero, and the next place up increases by 1. For example, if you add 1 to 9, the number becomes 10, with a 1 in the tens place and a 0 in the ones place. If you add 1 to 29, the number becomes 30: the tens place increased by a value of 1.
- In the binary system, each place can only hold a value of up to 1. The name of the place values also changes because of this. The system goes from the ones' place to the twos' place to the fours' place to the eights' place—and so on.
- Let's look at an example of how we'd represent a value in binary, using the number 5.

Power of 2	2^3	2^2	2^1	2^0
Value represented (base 10)	8	4	2	1
Amount of value	0	1	0	1



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**.





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.



Bit Representations

- Bits are used to represent everything you see and work with on a digital screen, including numbers, colors, and even sound waves!
- The same sequence of bits can represent different types of data depending on the context.
- For example, take this chart representing the ASCII code, the foundation of the modern Unicode system that converts text to binary format.

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	11000000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[END OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(88	58	1011000	130	X					
41	29	101001	51)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					





Bit Representations

- Fortunately, programs are designed to know what the binary value they're reading is meant to represent, and there usually aren't mix-ups. However, it's important to have consistent communication standards between *different* programs and computers, to prevent any possible confusion.

 Check out [this super cool video](#) from TED-Ed about how Binary works!

- So if binary is the language that computers use, why aren't we constantly dealing with these zeros and ones when we use our devices? The answer lies in a concept known as abstraction.



Abstraction

LECTURE 1



Abstraction

- **Abstraction** is the process of reducing complexity by focusing on only the ideas that a user needs to know. This is done mainly by hiding irrelevant details from the user.
- How does this work?
- A non-computer science related example of abstraction is pressing the "start" button on your oven. You don't have to know how the oven works to use it. You also don't have to deal with all the little details of starting the oven, such as turning on the gas or controlling the fans inside. The start button is an abstraction because it only focuses on the main idea of the task you're trying to accomplish: beginning the heating process.





Abstraction

- This same concept applies to your computer. Your computer stores data in bits, but programs running on your computer interpret it through variables and lists and can then display it through spreadsheets and charts.
- It does this because it's difficult for humans to understand a string of zeros and ones. You also don't need to understand these strings in order to use your computer or code a program.
- Just like how the start button hides all the specific details of running an oven, code statements and screen displays hide the internal processes of a computer, including its binary representations.



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 discrete 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.





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.



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

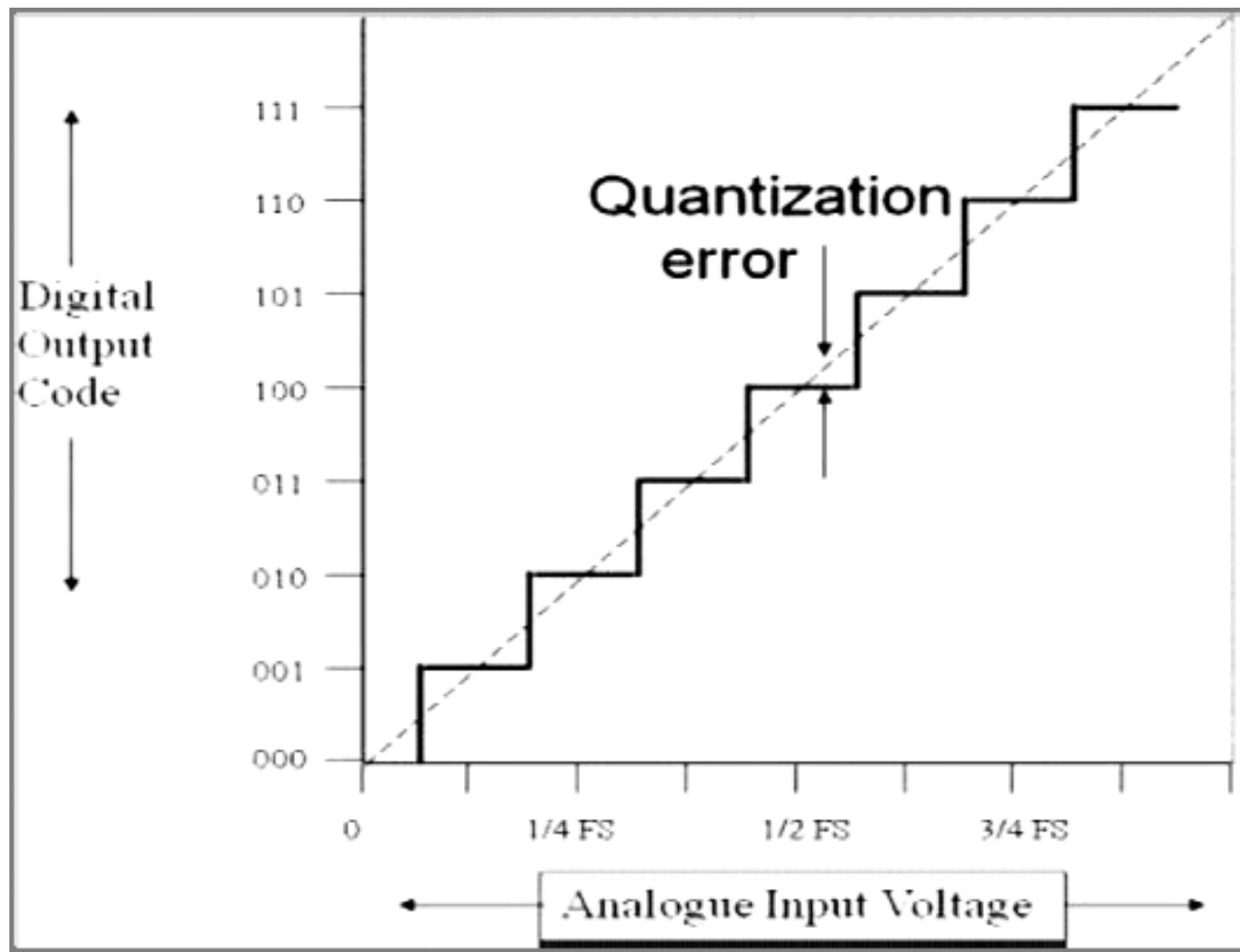
- Another way that number storage makes use of abstraction is through rounding. Because you have a limited amount of bits to store numbers, the computer will sometimes round or cut off your number. This can be most prominently seen when you're working with very small numbers or repeating decimals.

| What do you want your first number to be?100
| What do you want your second number to be?3
| 33.33333333333336

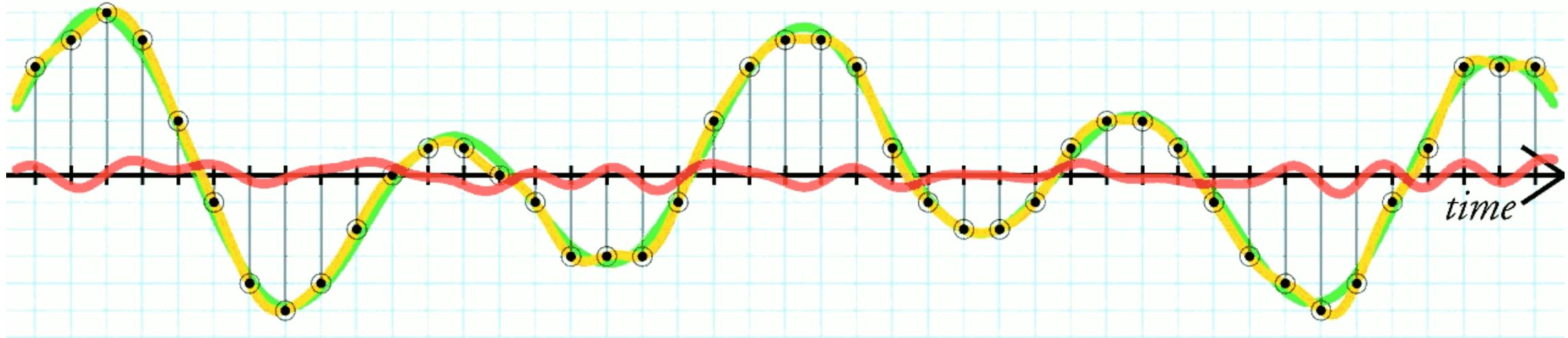


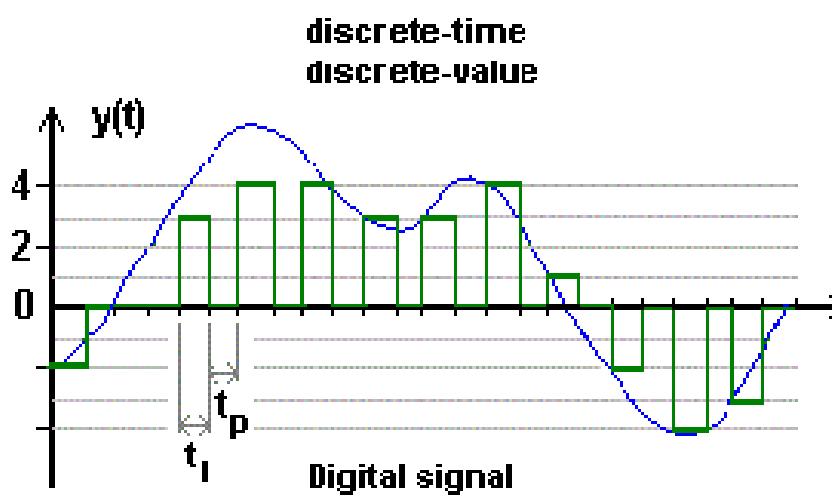
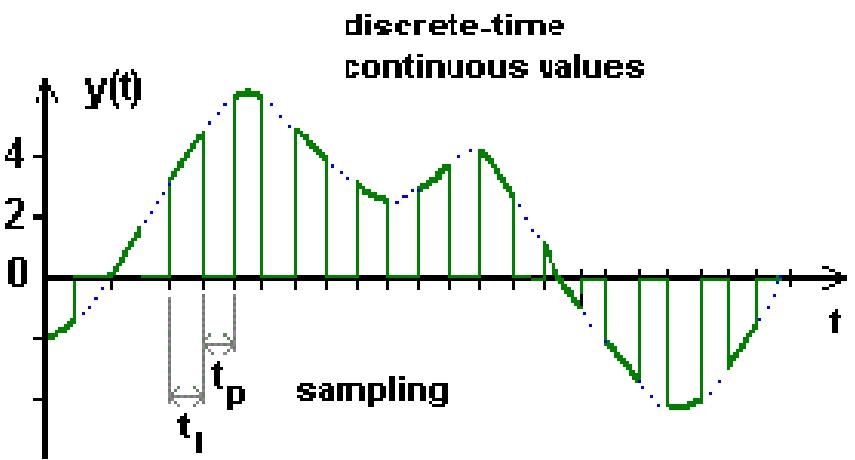
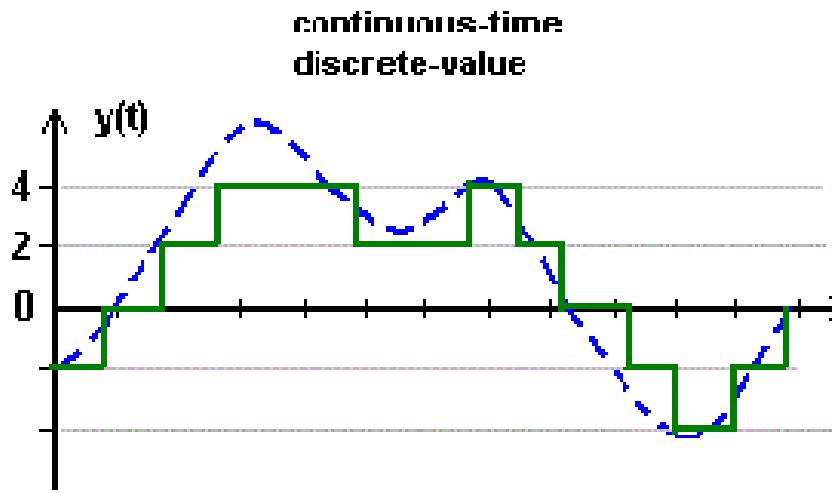
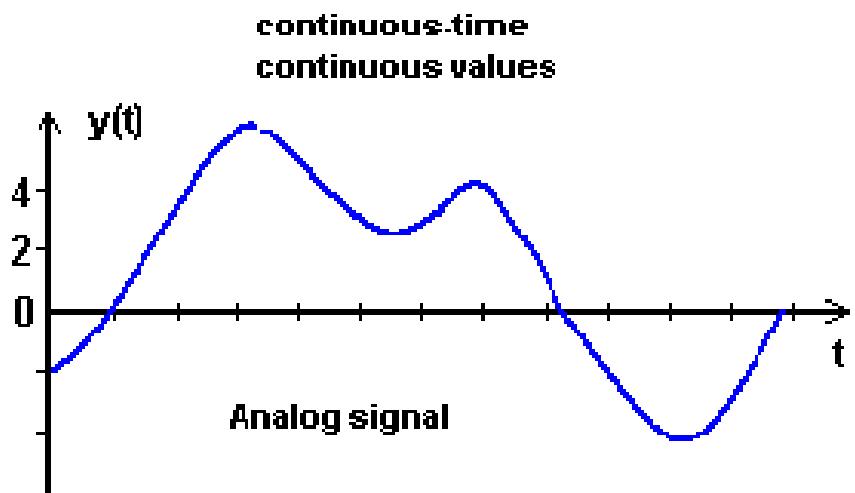
Quantization Error

- 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.



original signal
quantized signal
quantization noise







Number System

LECTURE 1

base number

power, exponent, index, order

base

4

$b = b \times b \times b \times b$

place value number system

The number of units of the base number to the power of 5	The number of units of the base number to the power of 4	The number of units of the base number to the power of 3	The number of units of the base number to the power of 2	The number of units of the base number to the power of 1	The number of units of the base number to the power of 0
b^5	b^4	b^3	b^2	b^1	b^0

In the decimal system the base number is 10

10^5	10^4	10^3	10^2	10^1	10^0
100000's	10000's	1000's	100's	10's	1's
hundred thousands	ten thousands	thousands	hundreds	tens	ones



© Jenny Eather 2014

Number System

Billions	Millions	Thousands	Ones	.	Decimals
Hundred billions					
Ten billions					
billions	Hundred millions	Ten millions	millions		
	Hundred thou sands	Ten thou sands	Thou sands		
			Hundreds		tenths
			Tens		hundredths
			Ones		thousandths
					Ten thou sandths
					Hundred thou sandths
					milonths

POLYNOMIAL EVALUATION

Symbolic

representation

(What we write)

547₁₀

Subscript used to
indicate the radix
(number base)

Digit values (coefficients)

$$= 5 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$$

Positional weights (powers of the radix)

$$= 5 \times 100 + 4 \times 10 + 7 \times 1$$

$$= 500 + 40 + 7$$

$$= 547$$

Numeric interpretation
(What we understand)

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

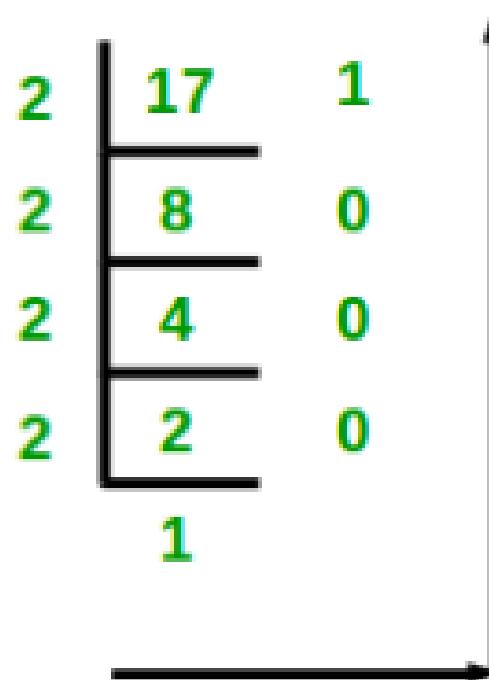


1	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---

$$\begin{aligned} & 1 \times 2^0 = 1 \times 1 = 1 \\ & 0 \times 2^1 = 0 \times 2 = 0 \\ & 0 \times 2^2 = 0 \times 4 = 0 \\ & 1 \times 2^3 = 1 \times 8 = 8 \\ & 1 \times 2^4 = 1 \times 16 = 16 \\ & 0 \times 2^5 = 0 \times 32 = 0 \\ & 1 \times 2^6 = 1 \times 64 = 64 \\ & 1 \times 2^7 = 1 \times 128 = 128 \end{aligned}$$

$$1 + 8 + 16 + 64 + 128 = 217$$

Decimal number : 17



Binary number: 10001

Decimal to Binary Conversion

Convert decimal number 57_{10} to binary	
<p>Method 1 Descending Powers of Two and Subtraction</p> <p>1. Write the power of two</p> $\begin{aligned}2^0 &= 1 \\2^1 &= 2 \\2^2 &= 4 \\2^3 &= 8 \\2^4 &= 16 \\2^5 &= 32 \\2^6 &= 64 \\2^7 &= 128 \\2^8 &= 256\end{aligned}$ <p>2. Write the number as a sum of powers of two</p> $\begin{aligned}57 &= 32 + 25 \\&= 32 + 16 + 9 \\&= 32 + 16 + 8 + 1 \\&= 1 \times 32 + 1 \times 16 + 1 \times 8 + 1 \times 1 \\&= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0 \\&= 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\end{aligned}$ <p>3. Extract the coefficients of the powers of two</p> $57_{10} = 111001_2$	<p>Method 2 Short Division by Two with Remainder</p> <p>1. Divide continuously by two until we reach 0</p> $\begin{array}{r}2:57 \text{ (remainder 1)} \\2:28 \text{ (remainder 0)} \\2:14 \text{ (remainder 0)} \\2:7 \text{ (remainder 1)} \\2:3 \text{ (remainder 1)} \\2:1 \text{ (remainder 1)}\end{array}$ <p>2. Write the remainders in reverse order (last is first)</p> $57_{10} = 111001_2$

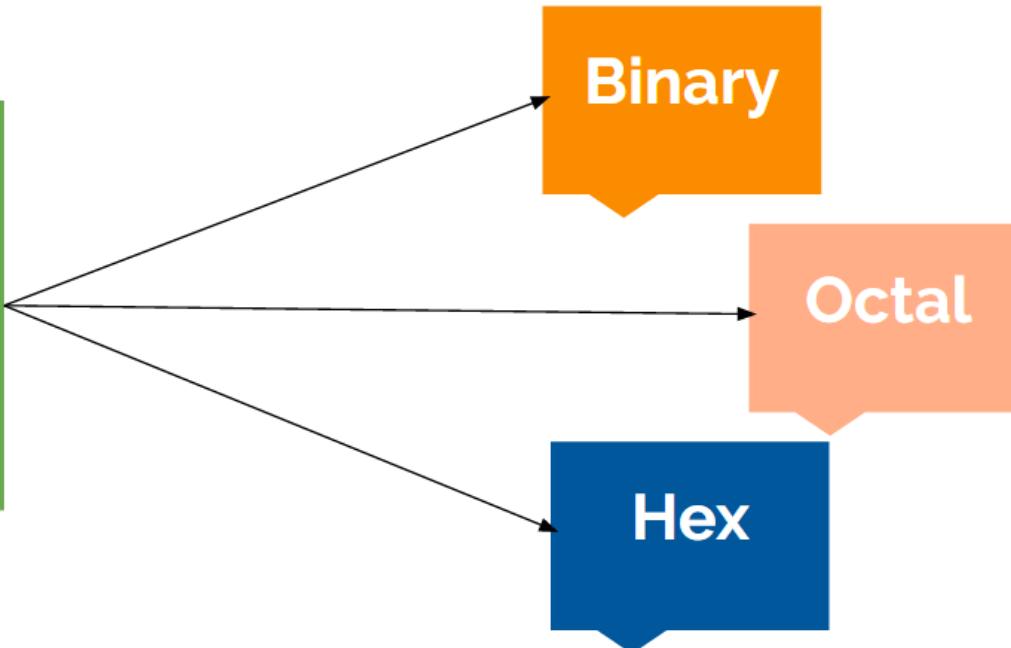
Dec	Hex	Oct	Bin
0	0	000	0000
1	1	001	0001
2	2	002	0010
3	3	003	0011
4	4	004	0100
5	5	005	0101
6	6	006	0110
7	7	007	0111
8	8	010	1000
9	9	011	1001
10	A	012	1010
11	B	013	1011
12	C	014	1100
13	D	015	1101
14	E	016	1110
15	F	017	1111



Convert integer to binary, octal, hexadecimal

TopJavaTutorial.com

Decimal

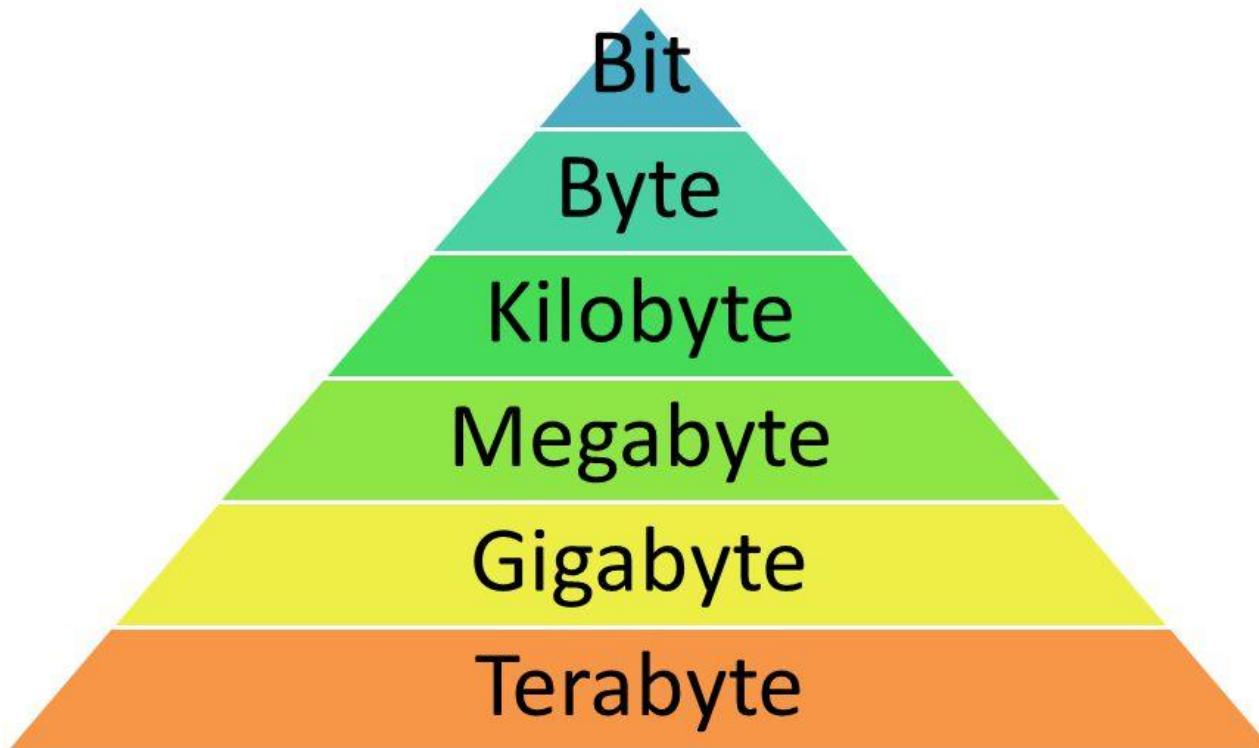


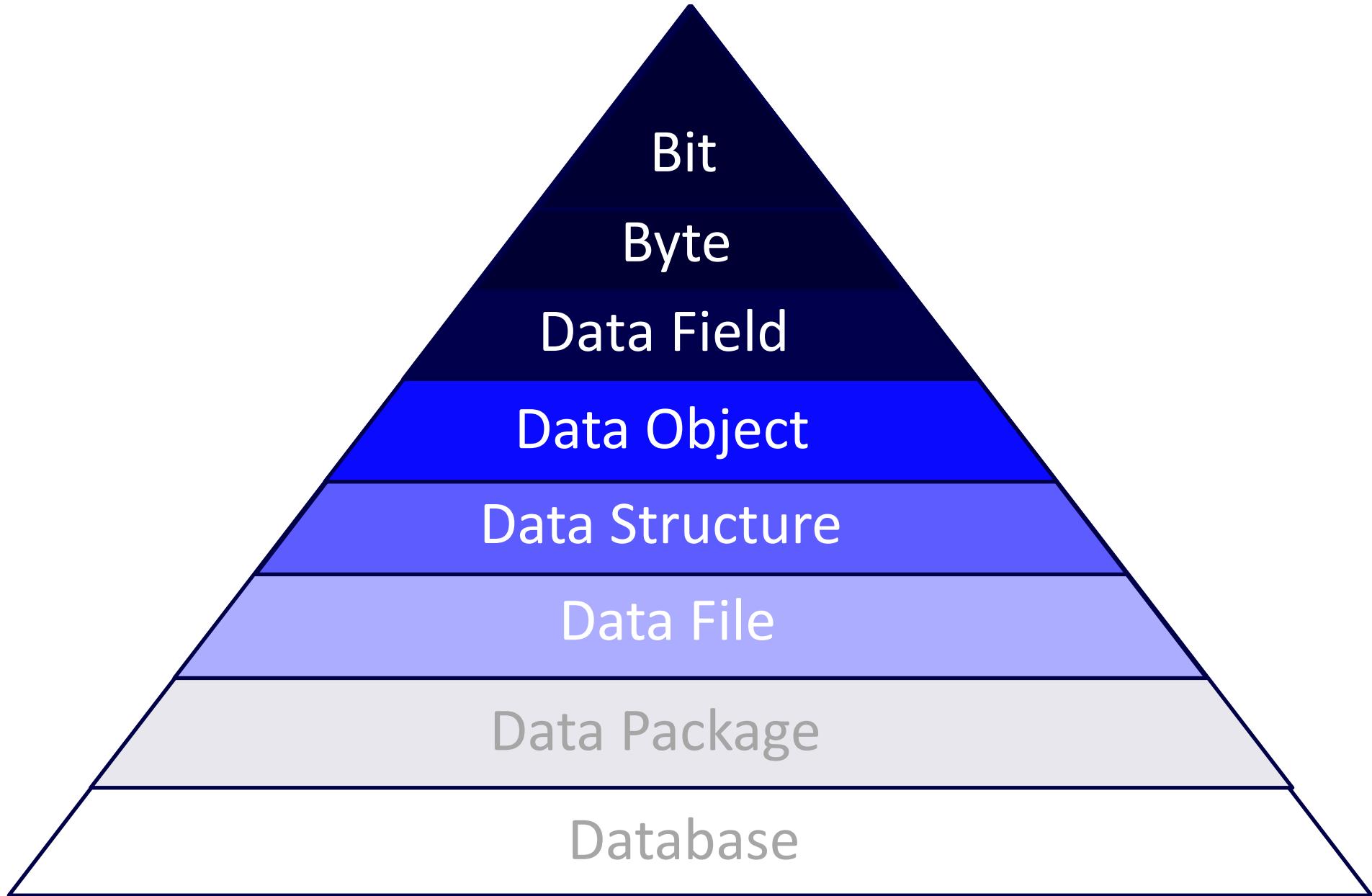


File Types

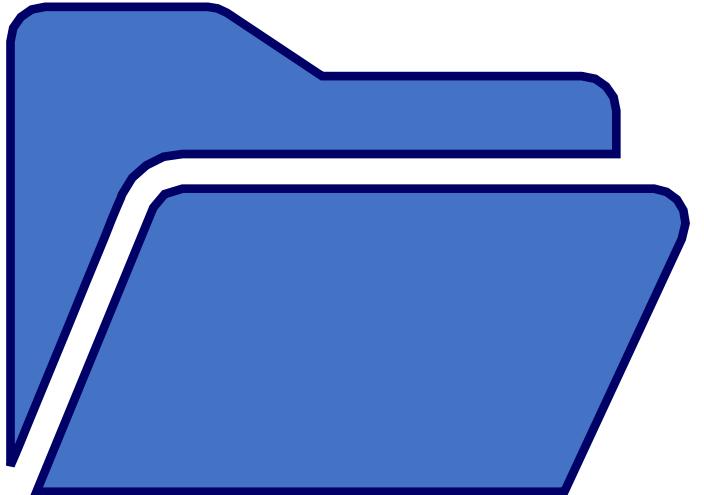
LECTURE 1

Types of Units of Storage









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:

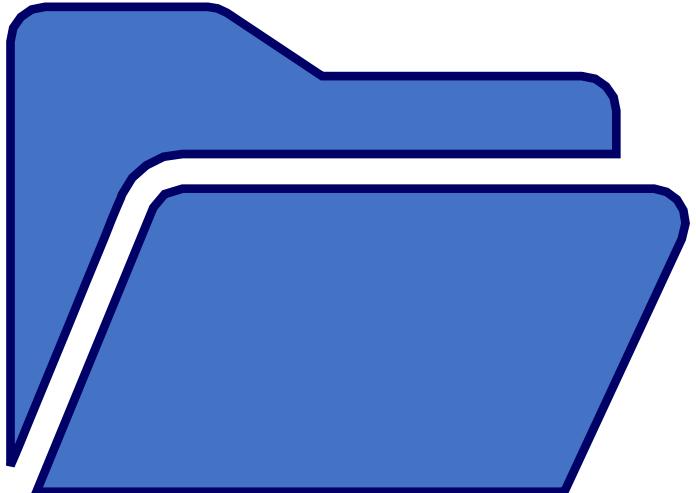
hello.txt - 5 bytes

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.



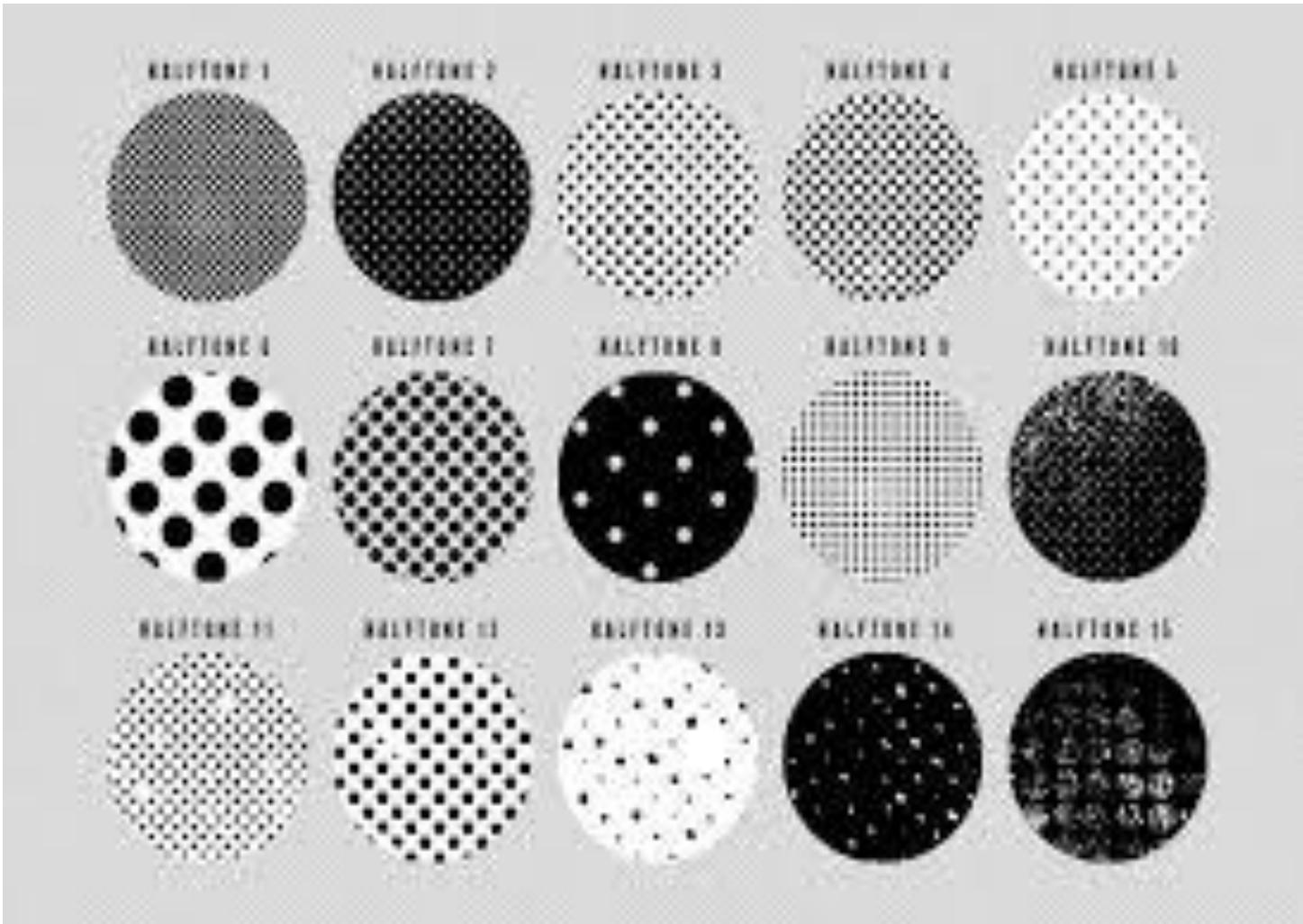
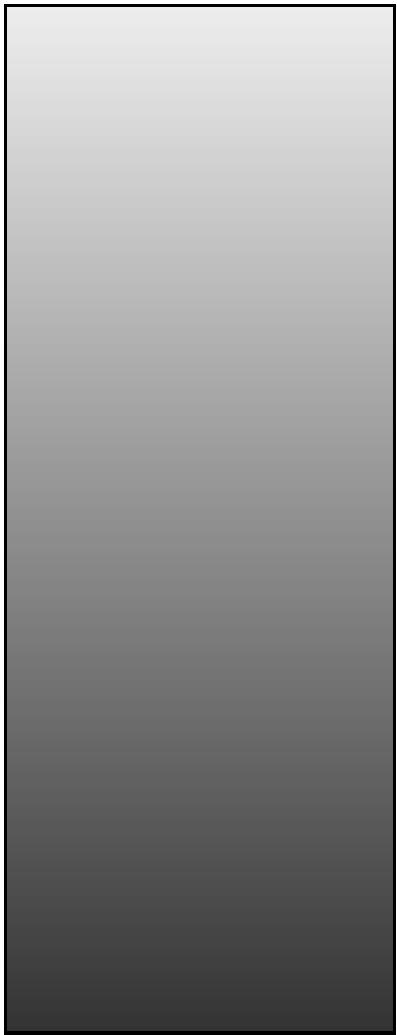
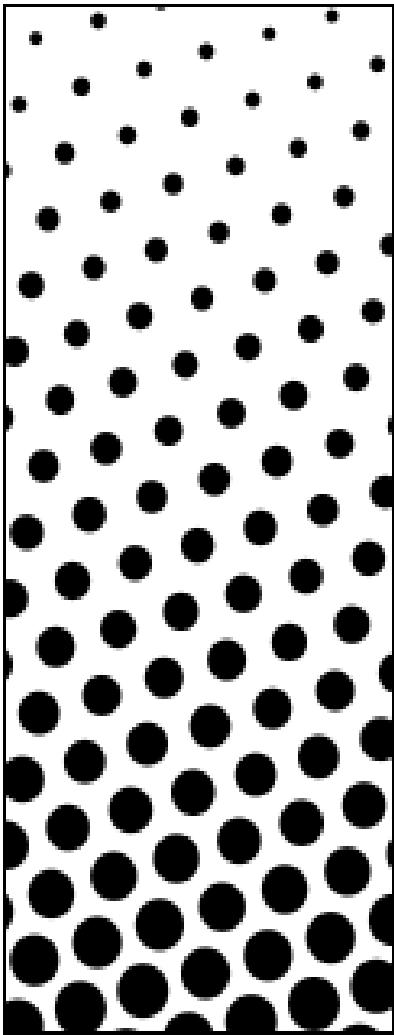
How many bytes does it take to store an image vs text?

- 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 Exercise

- 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:

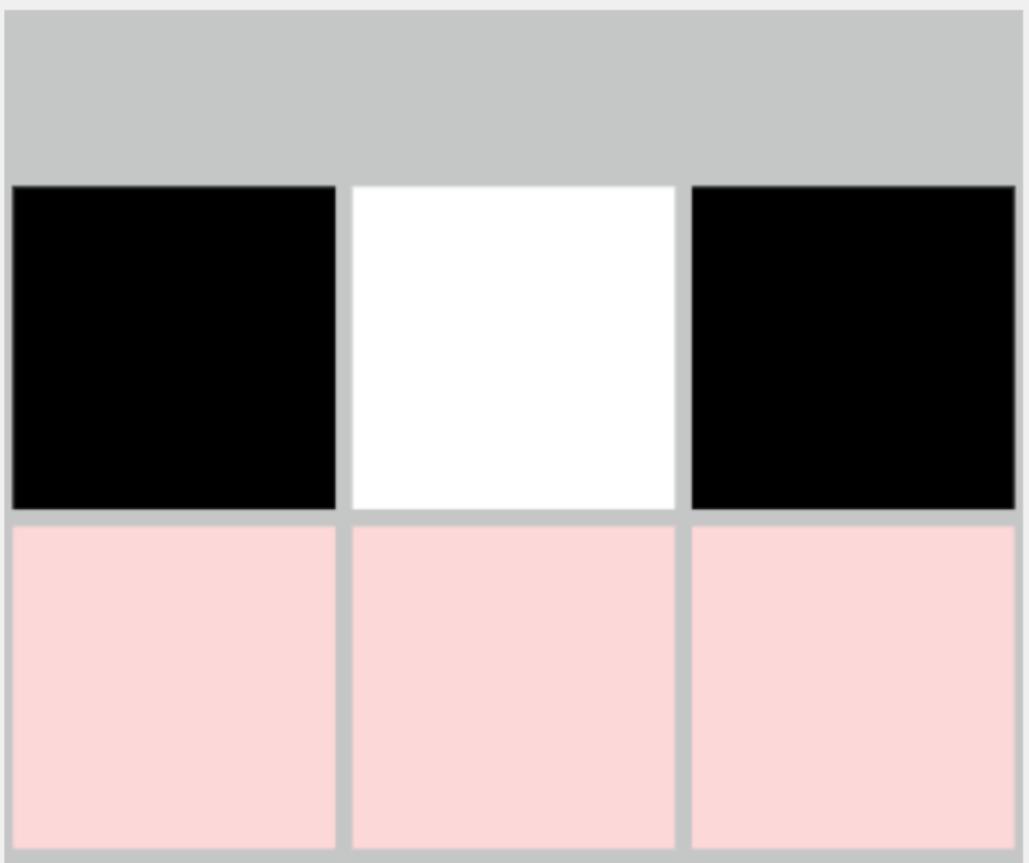
Width: 1 byte

Height: 1 byte

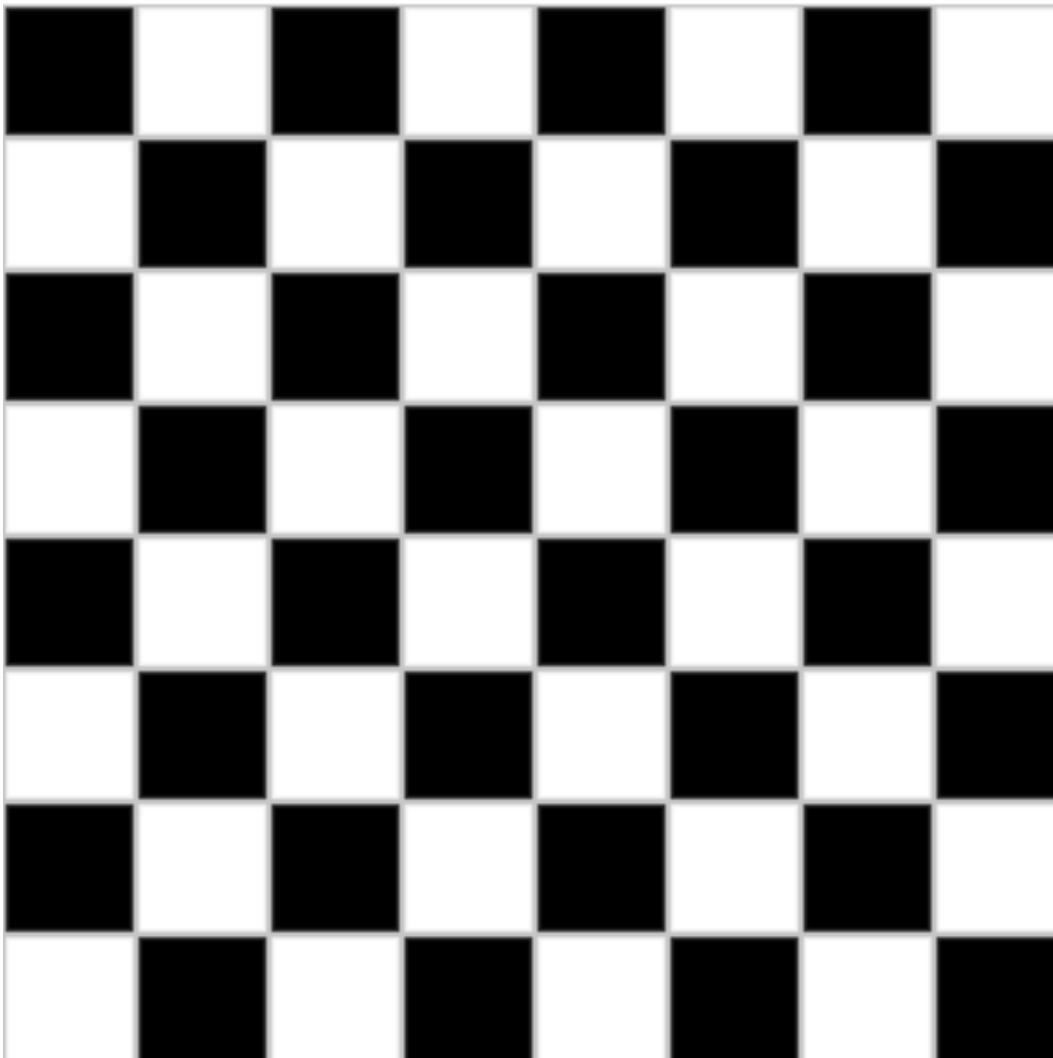
n bits of pixel data

Image width: 3

Image height: 2

Binary: Hexadecimal: 

0000 0011 ← Width
0000 0010 ← Height
0 1 0



```
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



Activity Guide

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



Activity Guide

Page 2: asks you to:

- Copy/paste a copy of your personal creation
- Copy/paste the bits that are used to encode it



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 **metadata**. In this case the metadata encodes the width and height of the image.
- We've seen forms of metadata before.
- For example: an **internet packet**. The packet contains the data that needs to be sent, but also other data like the to and from address, and packet number. [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.)

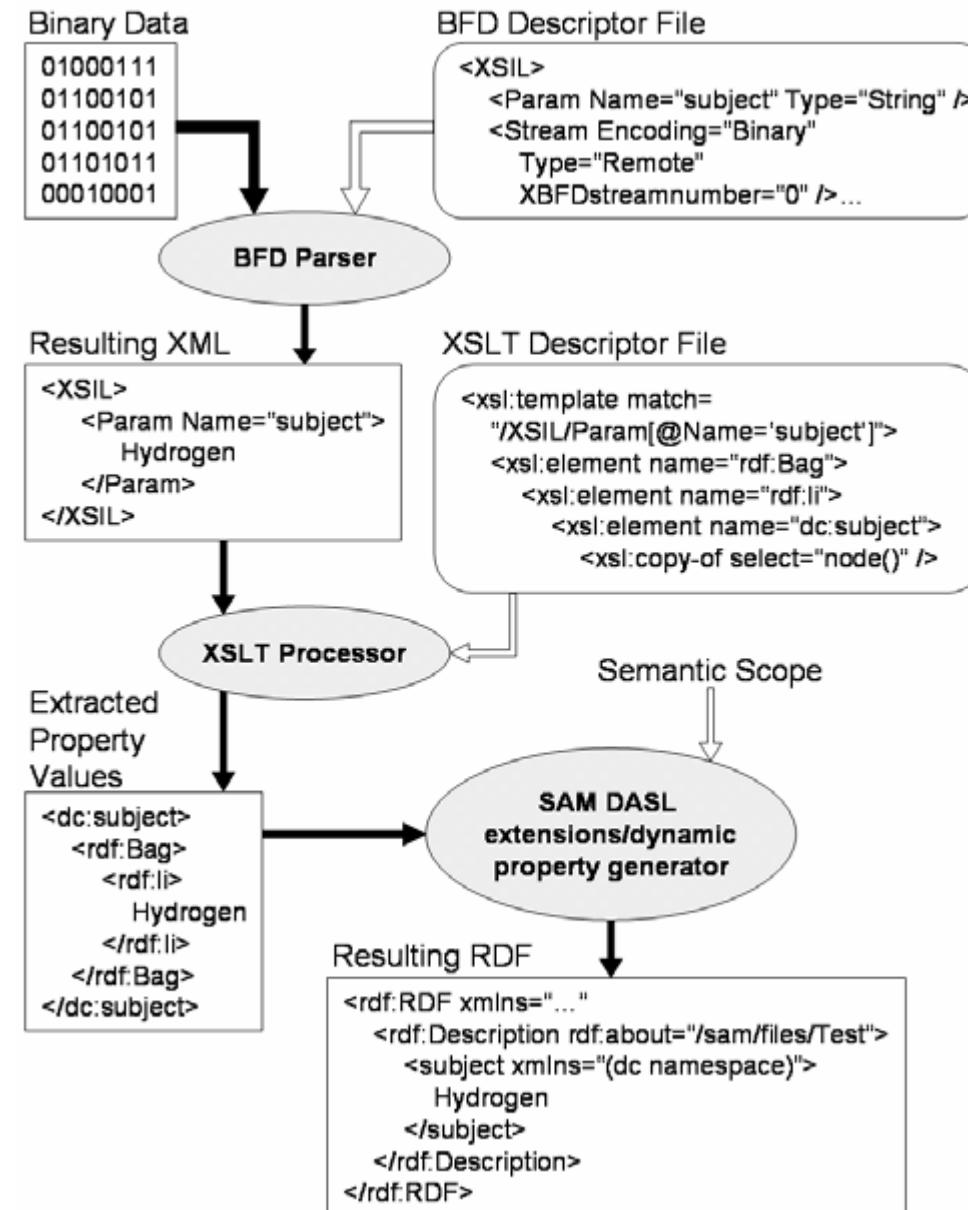
The latest NISO standards also include markup languages as a major metadata type. The latest NISO standards also include markup languages as a major metadata type. Markup languages like **HTML** and **XML** integrate metadata and flags for structural and semantic features into actual content, improving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

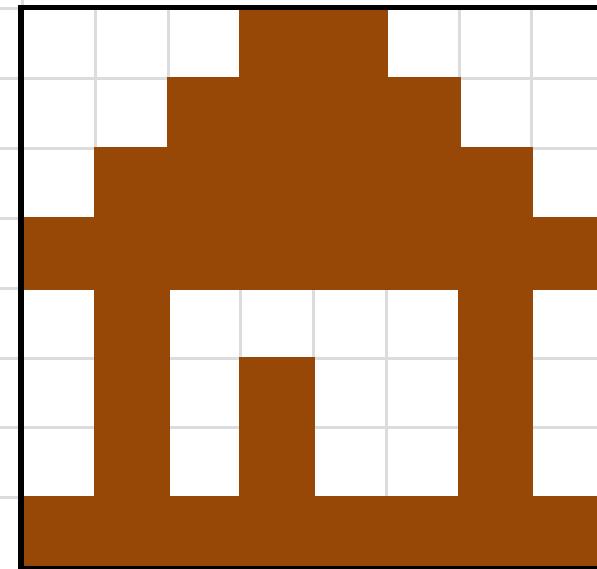
http://csunplugged.org/image-representation/#Colour_by_Numbers

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
w	w	b	b	b	b	w	w
w	b	b	b	b	b	b	w
b	b	b	b	b	b	b	b
w	b	w	w	w	w	b	w
w	b	w	b	w	w	b	w
w	b	w	b	w	w	b	w
b	b	b	b	b	b	b	b

3w2b3w

2w4b2w

1w6b1w

8b

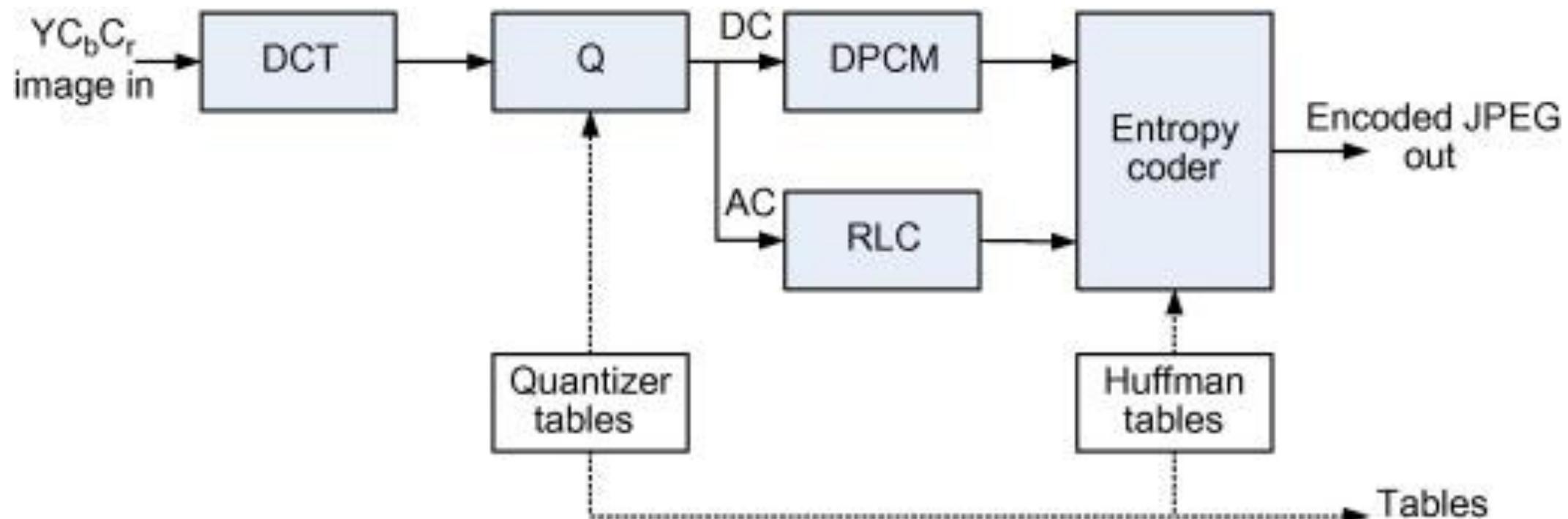
1w1b4w1b1w

1w1b1w1b2w1b1w

1w1b1w1b2w1b1w

8b







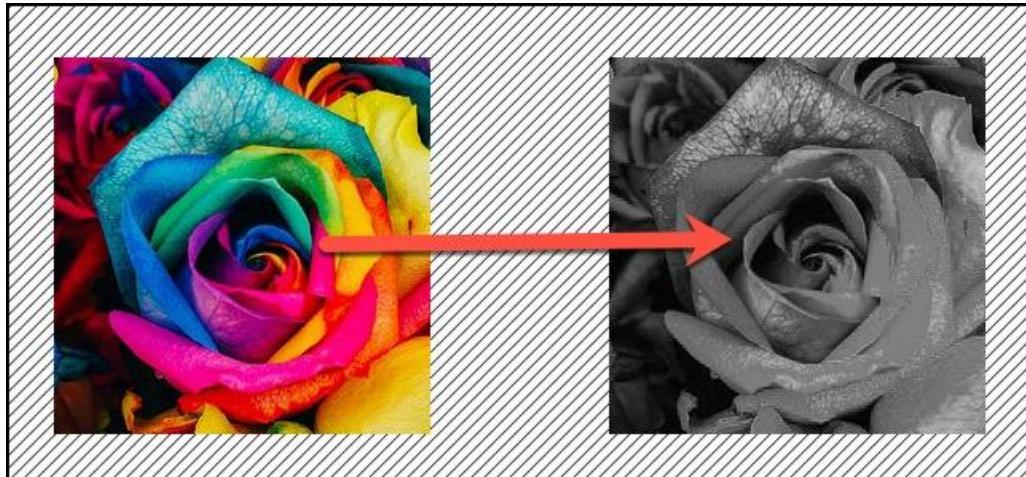
Conversion from Color Image to Black and White

Related Information

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

Online Tool:

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





Activity Guide

Magnify an Image

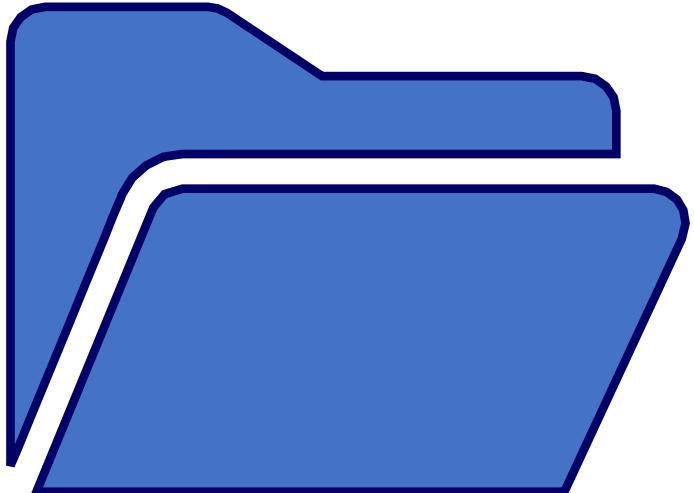
- double the size of an image on the Pixelation Tool



Homework

UNIT 1 – LESSON 3

- There is a bit to this one – you will need a sheet of graph paper.



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.



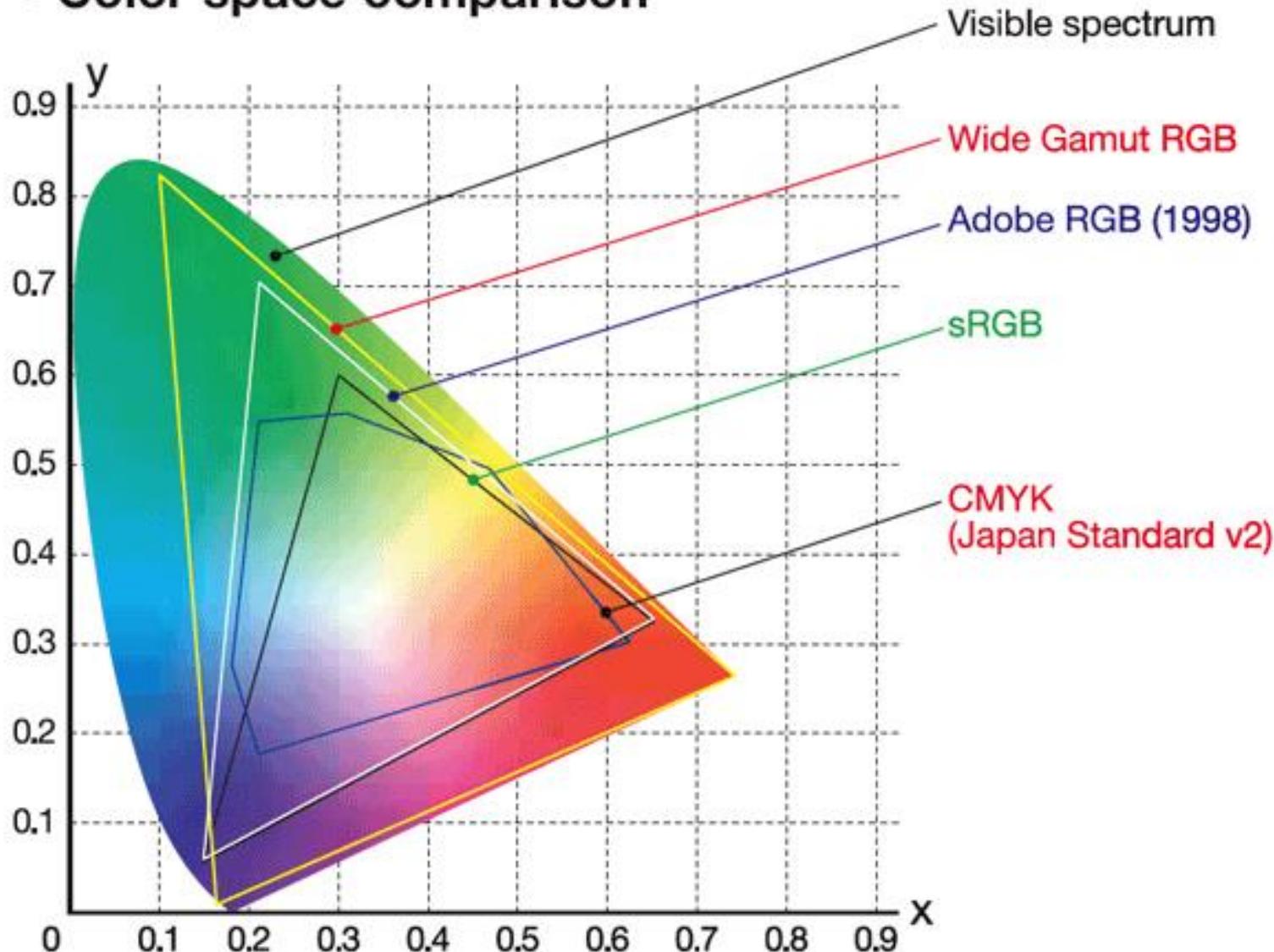
Vocabulary

- **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



- 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.



Activity Guide

- Activity Guide: [Encoding Color Images - Activity Guide](#)
 - Go into **code studio** – there are 3 tutorial videos that guide you through a few levels to get used to representing pixel data with more than one bit per pixel.
 - It works up to full 24-bit RGB color and will present hexadecimal as a convenient way to represent binary information for humans to read.



Hexadecimal Numbers:

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



Activity

- Practice with the [Hexadecimal Odometer](#) and
- Complete this [Hexadecimal Numbers Activity Guide](#)



Favicon

Vocabulary alert:

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.

Usually a small version of a company logo



Activity Guide (Optional)

Personal favicon Project

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.
- Send your favicon to others in class using the internet simulator
- Make the favicon larger and print in color.
- Class will choose a standard size.
- Print in color and we will make a “quilt” of all of them on poster board!



Tools

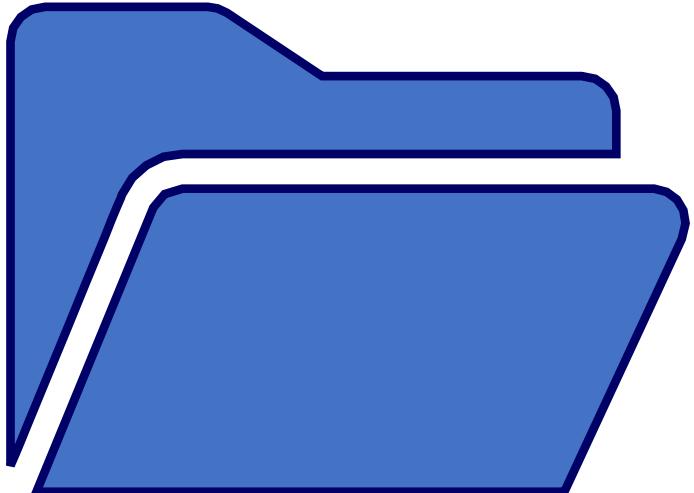
- Things to think about
- A simple design with a few basic colors is probably the best solution. How could you use more colors?
Plan ahead: Sketch your design before starting to encode the bits.
- You might want to use a tool to help you draw small images.
Suggestions:
 - Favicon Maker: <http://www.favicon.cc/>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



Rubric

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			



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** - in 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.





Sound Vs Audio

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

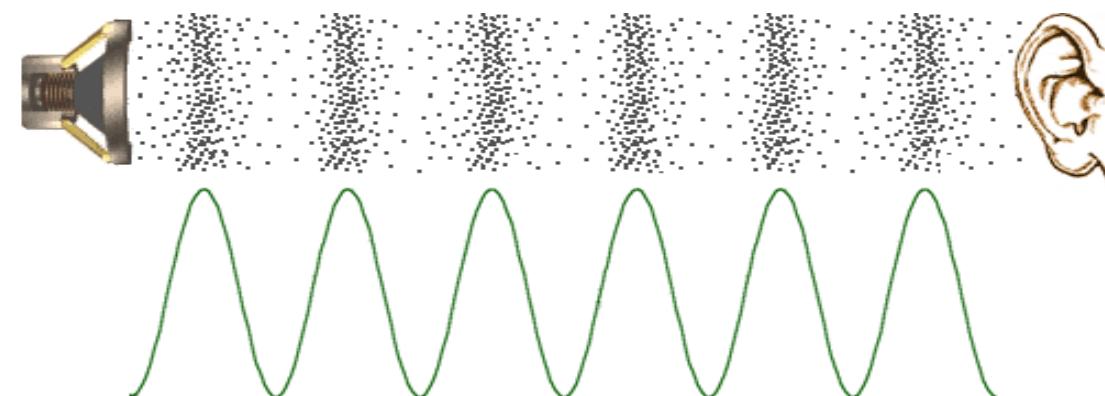
Classes of Sound:

- 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?

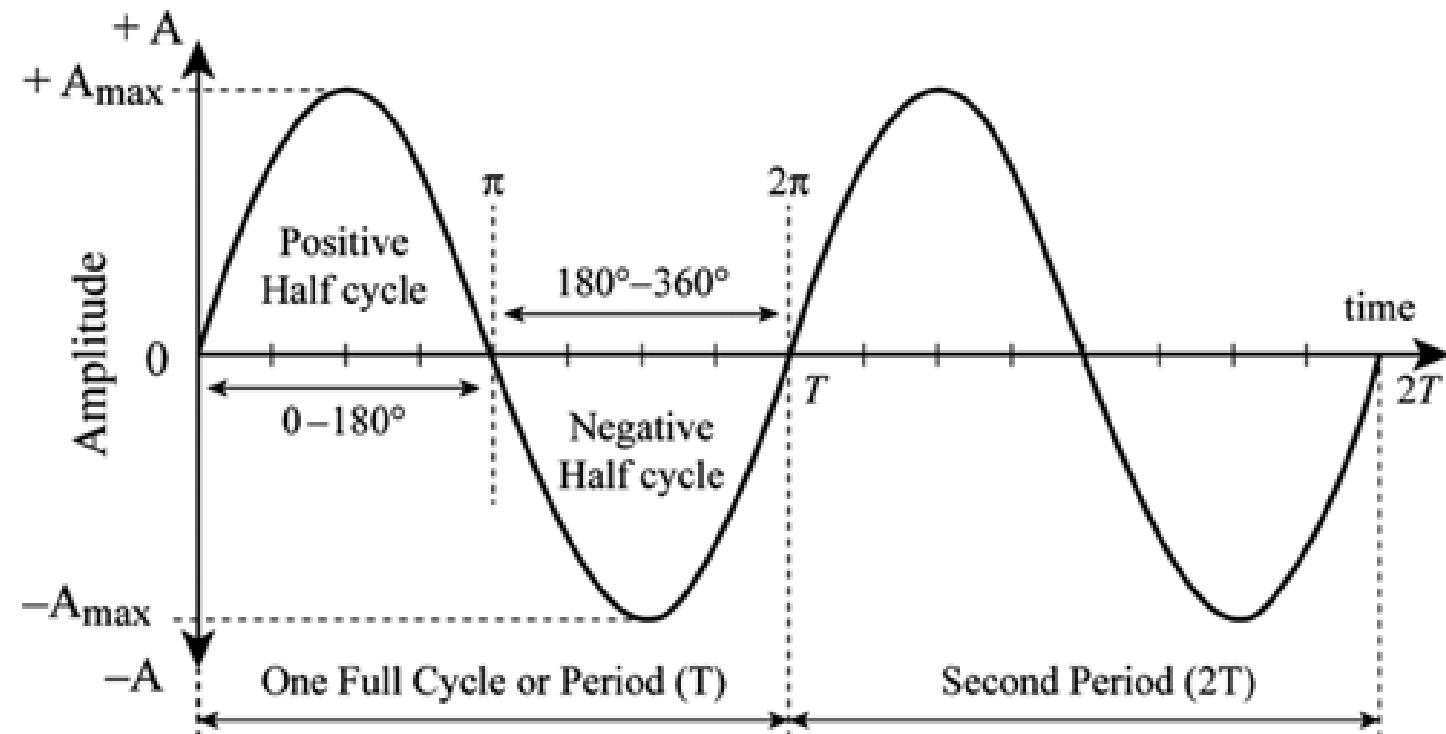
- Sound waves are variations of **pressure in a medium** such as air.
- Sound created by the **vibration** of an object, which causes the air surrounding it (medium) to vibrate.
- Vibrating air causes the human eardrum to vibrate, which the brain interprets as sound.





Properties of Sound

- frequency
- wavelength
- period
- amplitude
- speed

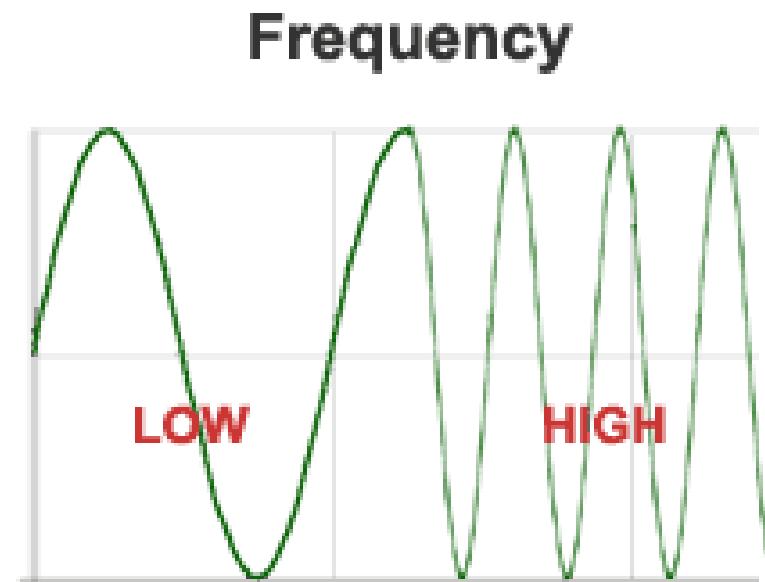




Frequency

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)

$$\text{File Size} = C * S * T * B$$



Calculate audio data size

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

- **Step 1**

$$44,100 \times 2 \text{ bytes (or 16-bits)} = 88,200 \text{ bytes}$$

- **Step 2**

$$88,200 \times 2 \text{ (for stereo)} = 176,400 \text{ bytes}$$

- **Step 3**

$$176,400 \times 30 \text{ seconds} = 5,292,000 \text{ bytes}$$



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.



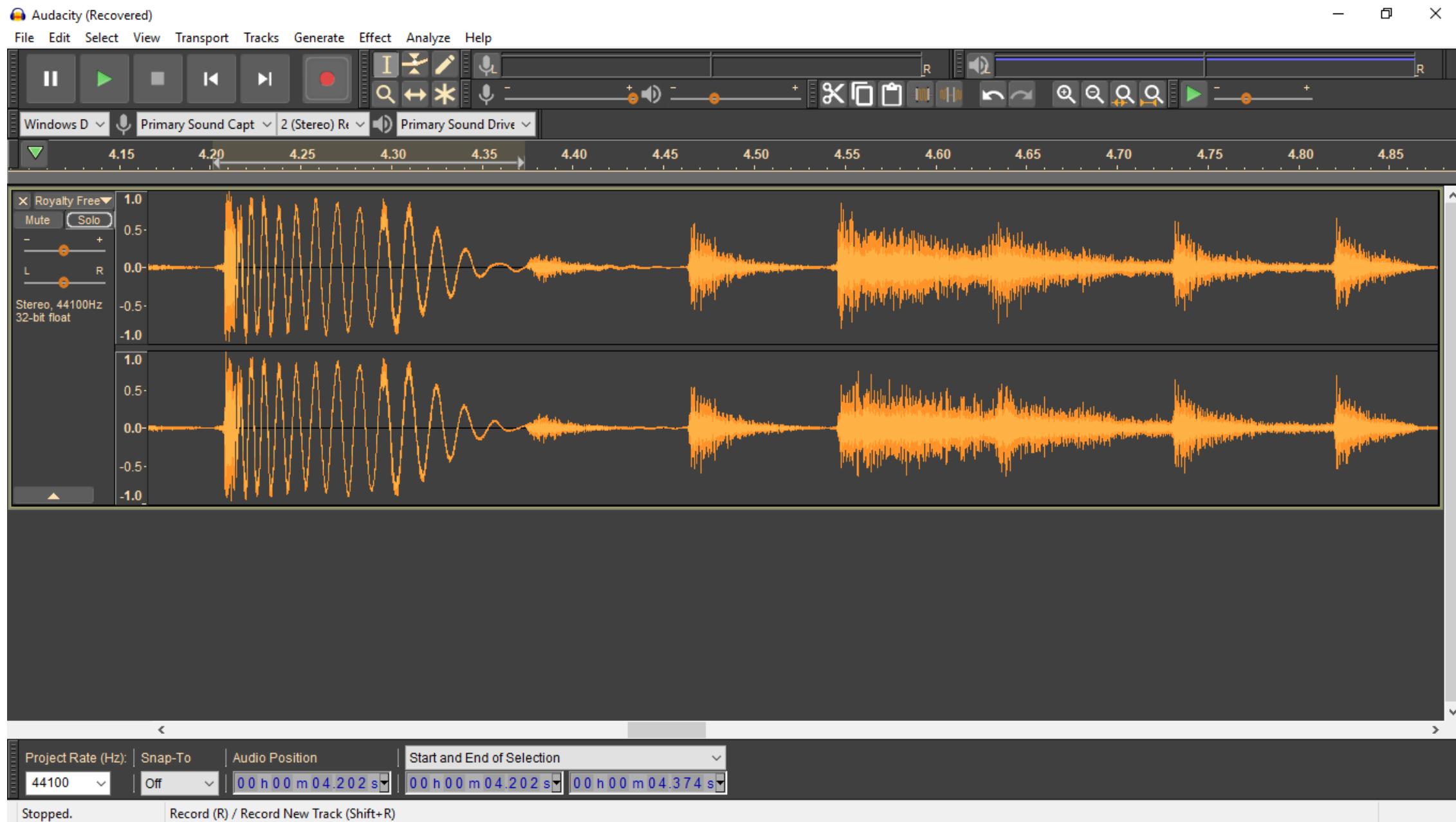
Audio file formats

<u>Extension</u>	<u>Use</u>	(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 programs offer digital equalization capabilities to modify the bass, treble or midrange frequency to make the audio sounds better.



Editing Digital Recording

Time stretching

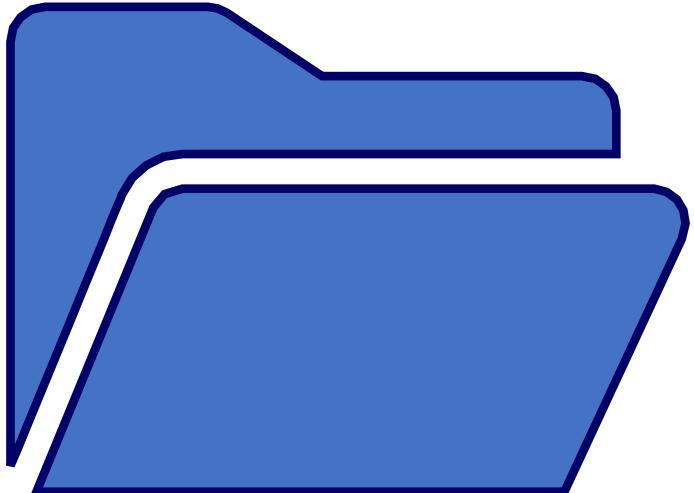
- Alter the length (in seconds) of a sound file without changing its pitch.

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](#).



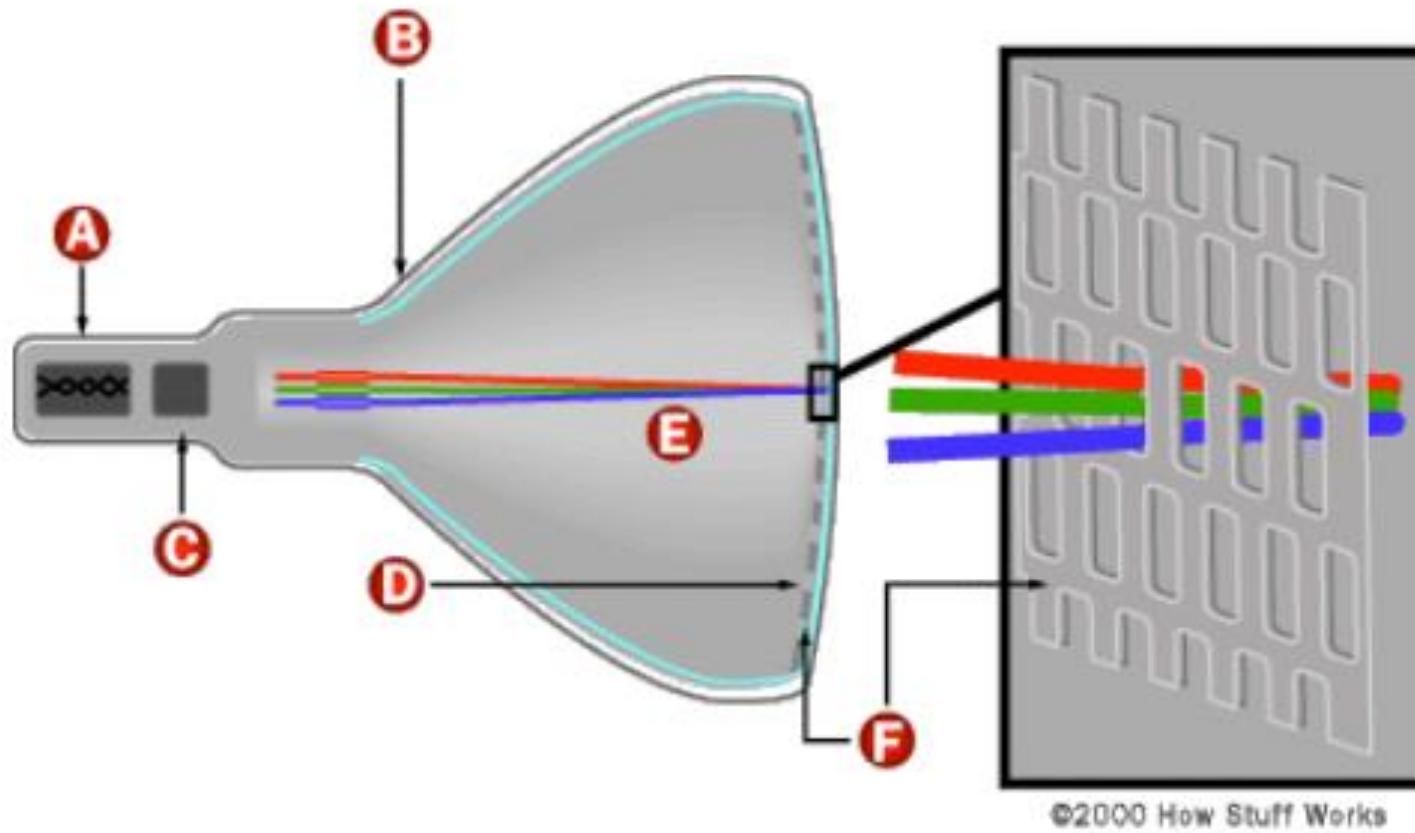
Vocabulary

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

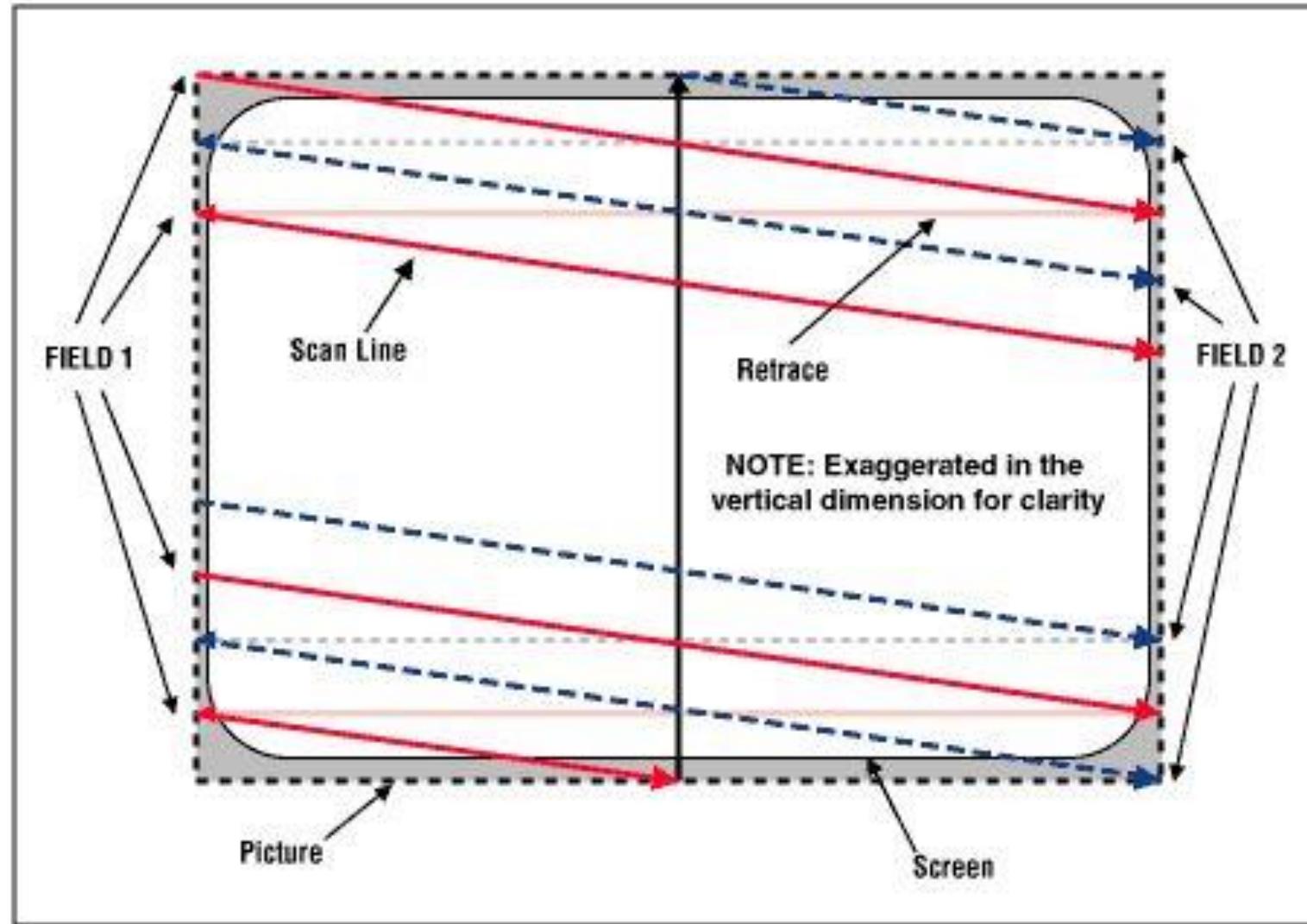
Video provides the magical elixir that keeps Vimeo's heart beating, its engine running, and its soul soaring.

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](#).

Analog TV: Inside a CRT

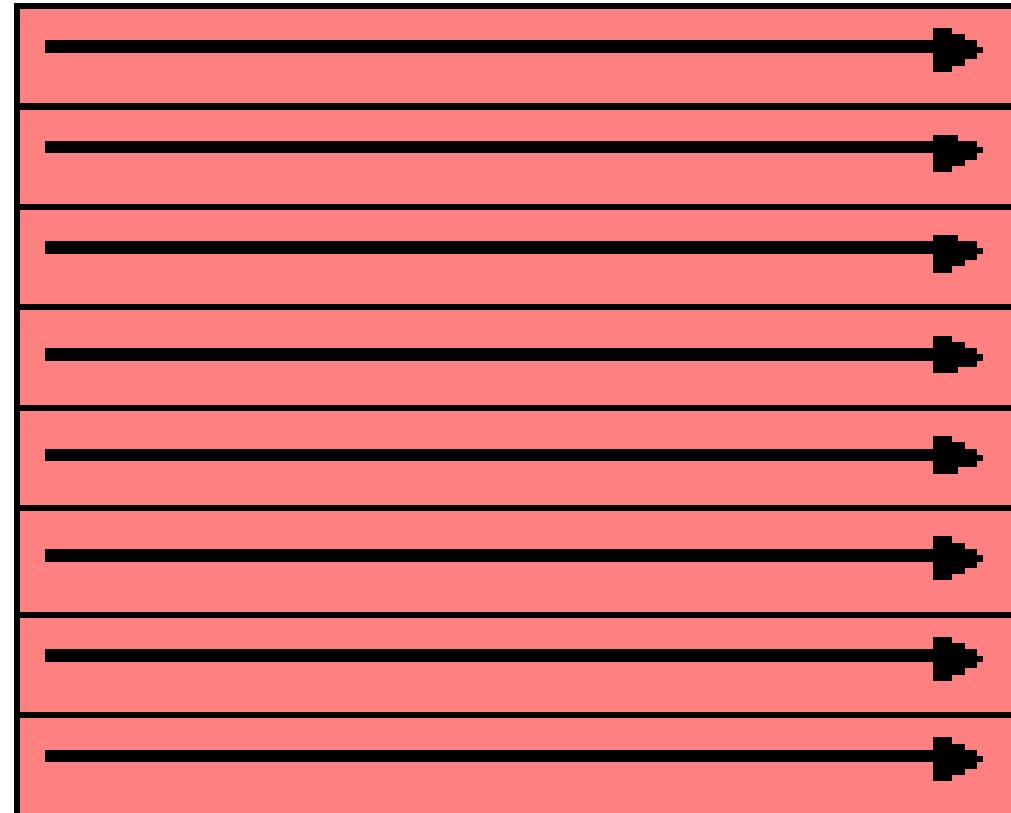


- A: Cathode
- B: Conductive Coating
- C: Anode
- D: Phosphor Coated Screen
- E: Electron Beams
- F: Shadow Mask





Interlaced



**Progressive Scan
(Non-interlaced)**

Evolution to Next Generation TVs

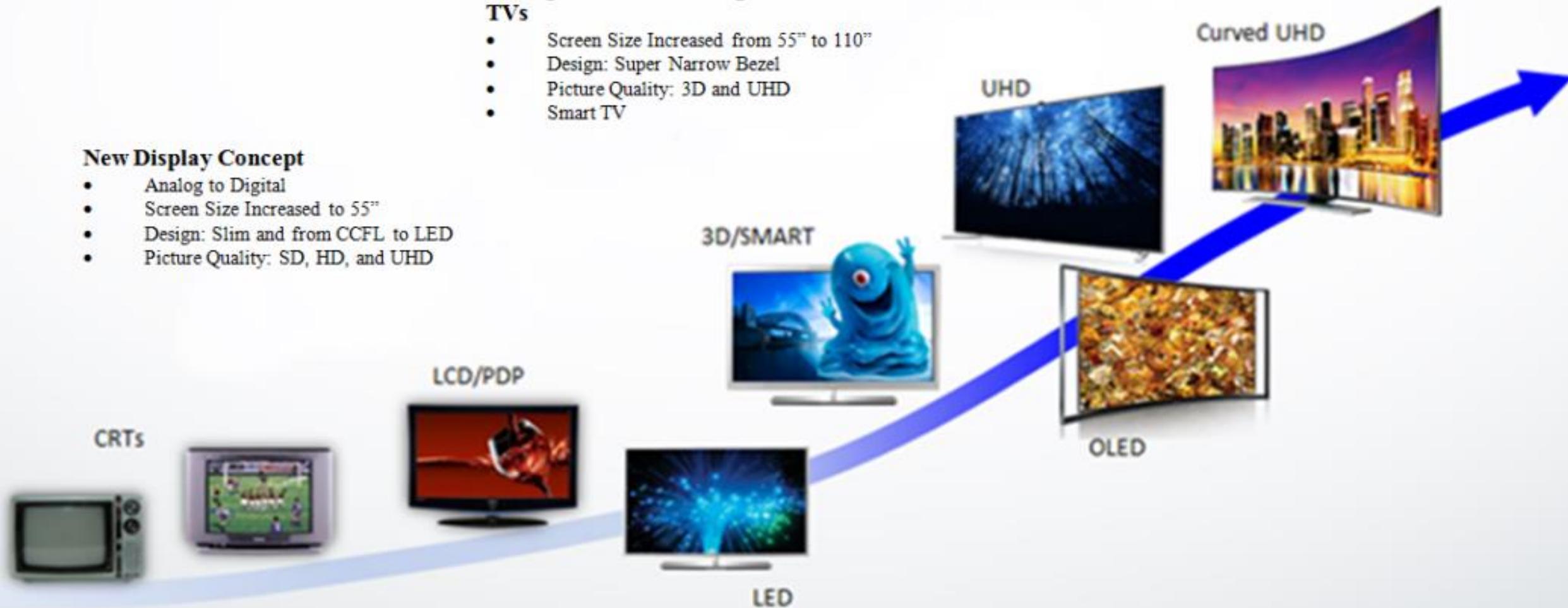
- Design: Zero Bezel, Curved TV, and Cinemascope
- Picture Quality: Glasses-Free 3D and High-Color Gamut
- Display: OLED, Transparent, Flexible, and 4K8K

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



NTSC

National Television Standard Committee

- Introduced in 1953 (in US)
- Used in US, Canada, Japan
- 30 frames per second (Actually 29.97)
 - Interlaced (even/odd field lines), so 60 fields per second.
 - Same as 60Hz AC power in these countries
- 525 lines
 - Picture only on 480 of these => 640x480 monitors
 - Rest are the vertical rescan.
- Aspect ratio is 4:3
- Needs colour calibration
 - Uses a colour burst signal at start of each line, but needs TV to be adjusted relative to this. “NTSC = Never Twice Same Colour”



PAL

Phase Alternating Line

- Introduced in 1967 (by Walter Bruch in Germany)
- PAL-I(UK), PAL-B/G (much of Europe), PAL-M (Brasil) ...
 - Differ mainly in audio subcarrier frequency.
- 25 frames per second
 - Interlaced (even/odd field lines), so 50 fields per second.
 - Same as 50Hz AC power in these countries
- 625 lines
 - Only 576 lines used for picture.
 - Rest are vertical retrace, but often carry teletext information.
- Colour phase is reversed on every alternate line.
 - Originally human eye would average to derive correct colour.
 - Now TV sets autocalibrate to derive correct colour.

SÉCAM

Séquentiel Couleur Avec Mémoire

- Introduced in 1967 (in France)
 - “System Essentially Contrary to American Method”
- Used in France, Russia, Eastern Europe...
- 625 lines, 25 fps interlaced, like PAL
- Uses FM modulation of subcarrier.
 - Red-Luminance difference on one line
 - Blue-Luminance difference on next line
 - Uses a video linestore to recombine the two signals
 - Vertical colour resolution is halved relative to NTSC and PAL.
 - Human eye is not sensitive to lack of spatial colour information.



8K

Ultra High Definition
4320 x 7680

4K

Ultra High Definition
2160 x 3840

1080p Full HD

720p HD

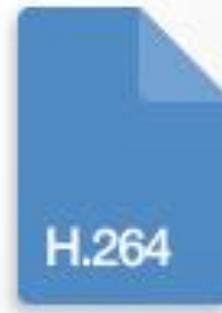
480p

1080
720
480

640 1280 1920



eLearning Channel

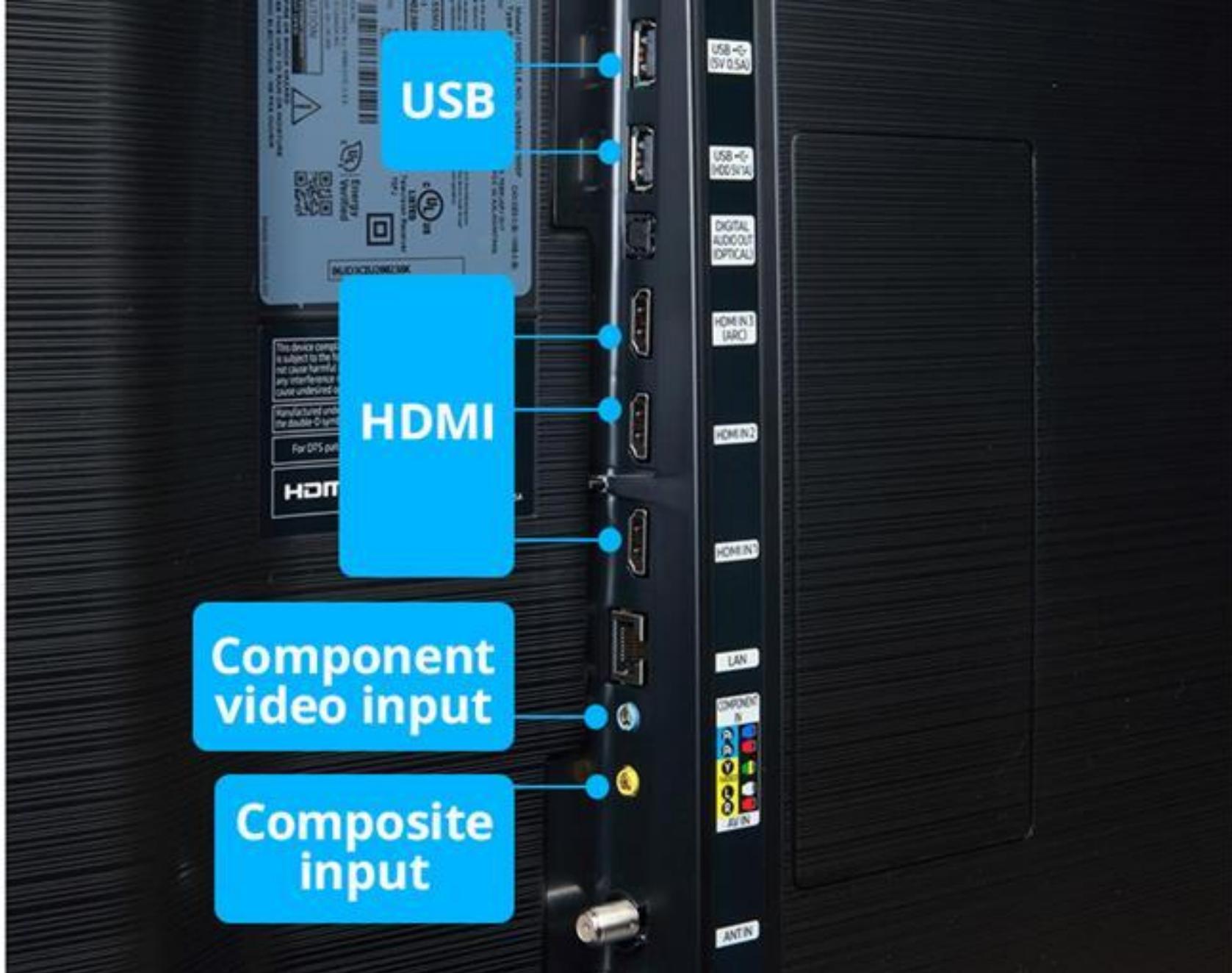


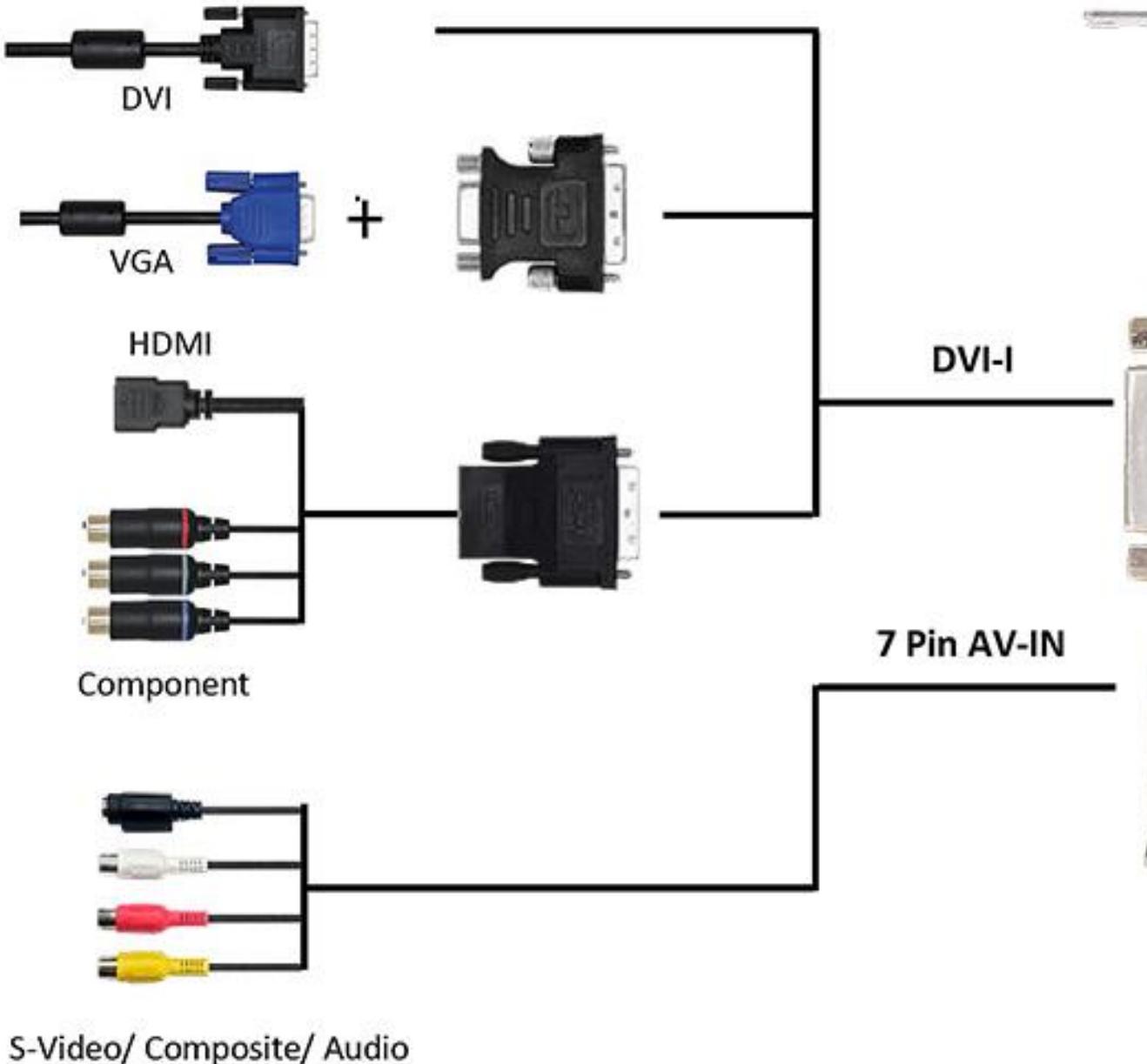
Video File Support

File Extension	Container	Video Codec	Resolution	Frame Rate (fps)	Bit Rate (Mbps)	Audio Codec
*.avi		DivX 3.11 / 4 / 5 / 6				
*.mkv						
*.ASF		MPEG4 SP/ASP				
*.wmv						
*.mp4	AVI	H.264 BP/MP/HP				
*.3gp	MKV					
*.vro	ASF					Dolby Digital
*.mpg	MP4	Motion JPEG				LPCM ADPCM
*.mpeg	3GP	Microsoft				(IMA, MS)
*.ts	MOV	MPEG-4 v3				AAC WMA Dolby
*.tp	FLV	Window Media				Digital
*.trp	VRO	Video v7,v8,v9				Plus
*.mov	VOB					MPEG(MP3) DTS
*.flv	PS	MPEG2				(Core)
*.vob	TS					
*.svi	SVAF	MPEG1				
*.m2ts					4	
*.mts		VP6	640×480			
*.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		







You
Tube

How Streaming Video & Audio Work

Basic Streaming Steps



Basic Streaming Process

1

A user visits a Web page hosted on a Web server and finds a file he'd like to see or hear.

©2007 HowStuffWorks

Web Server

2

The Web server sends a message to a streaming media server, requesting the specific file.

Media Server



Client



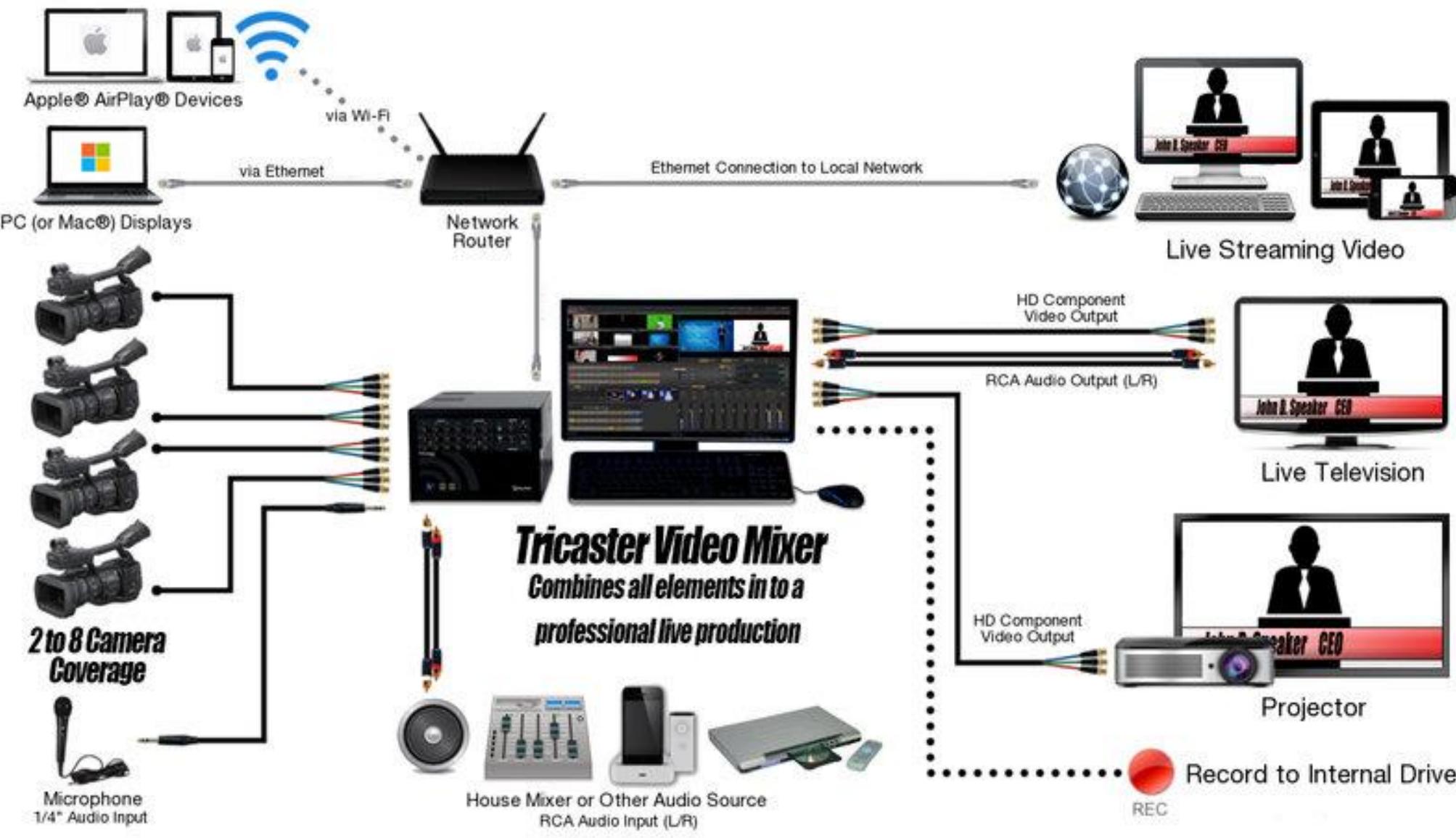
3

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.

Live Event



For workflow reference only. Example system configuration shown.

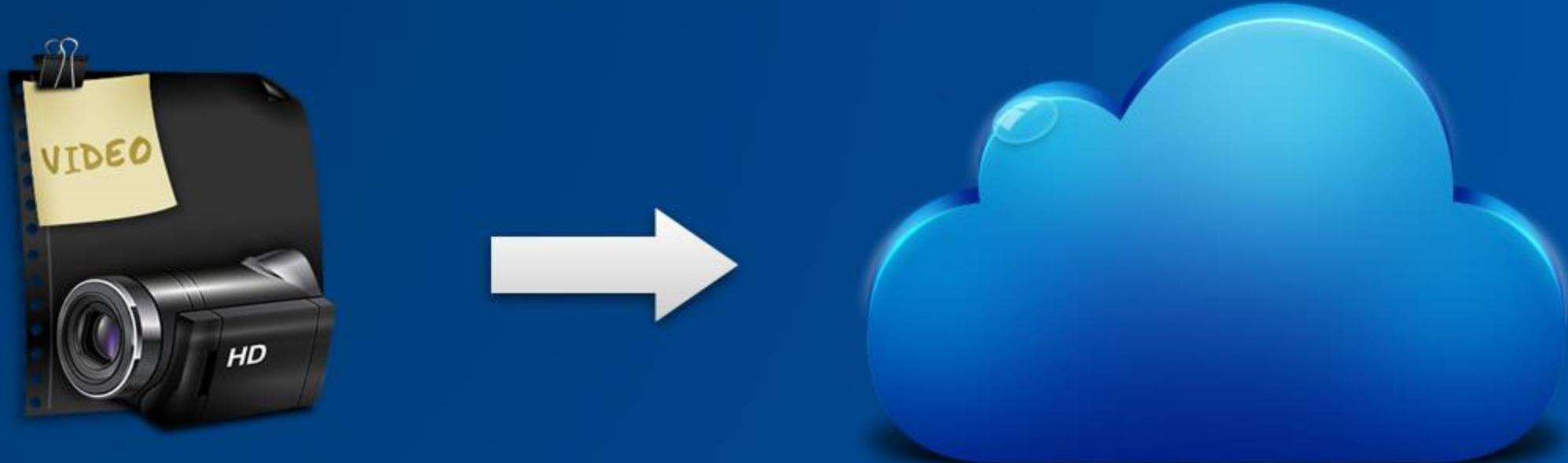


What is Digital Video Production?

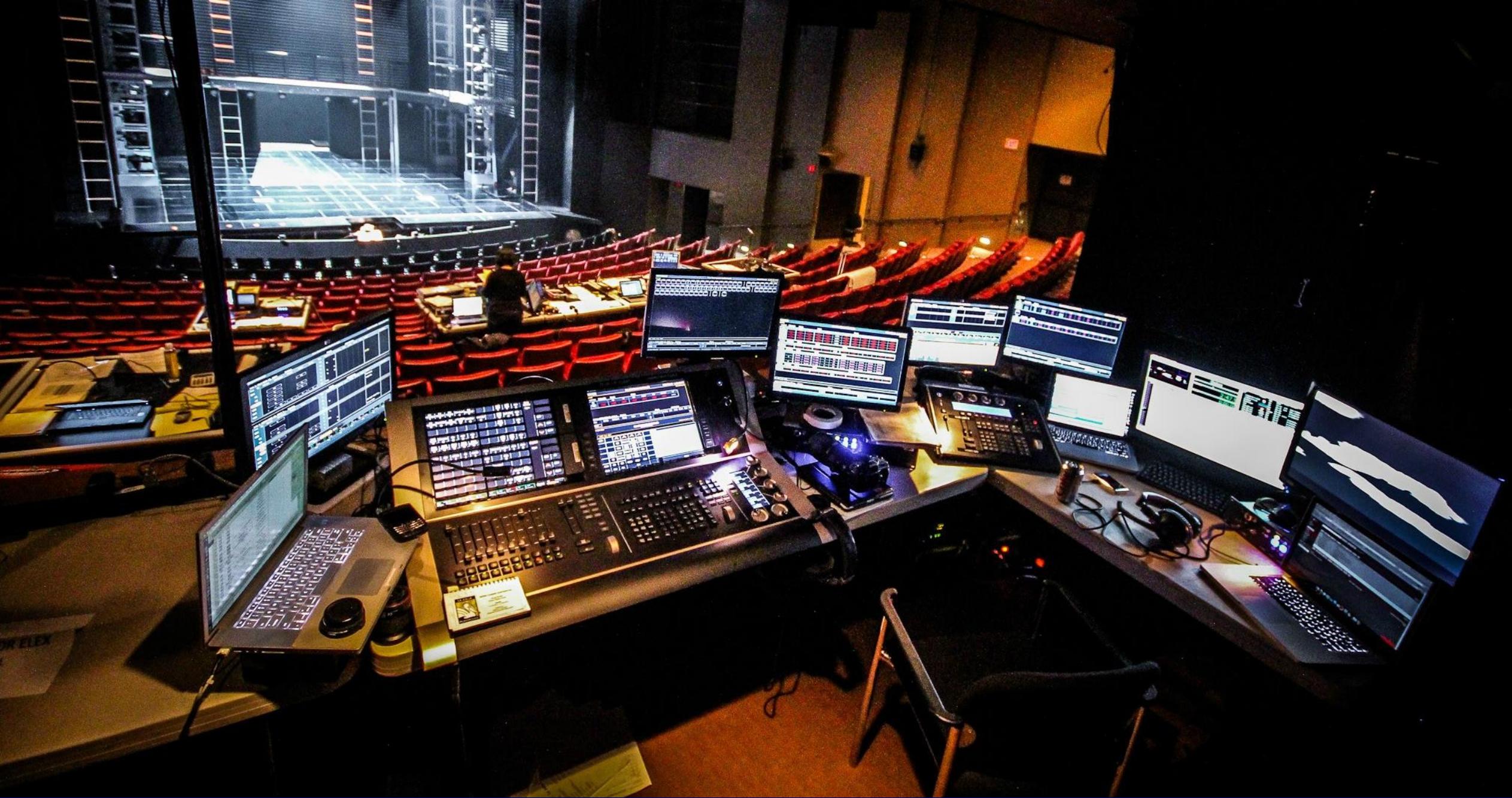
- Video Recording
- Video Editing
- Video Hosting
- Video Engineering
- Video/Audio Playback







Video Hosting?



Video Playback Position

Play / Pause

Full Screen

Stop Playback

Control Playback Settings

Mute

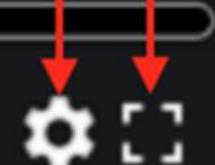
Audio Playback Level

Broadcast

Loop Video

Open Audio / Video File

Position in file / Total file time





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

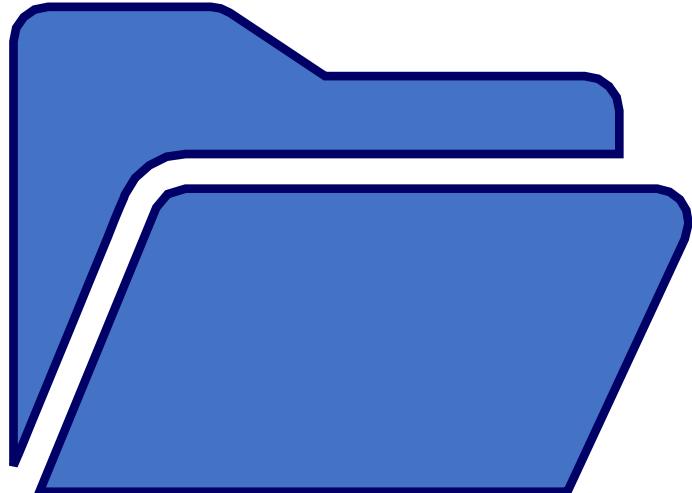
- Create tweaked video pressure profiles utilizing open-source video handling instruments, including: FFmpeg, libav and other comparable devices
- Implement calculations to distinguish scene softens up discretionary video content in an exceptionally performant way;
- Assist colleagues with planning machine 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

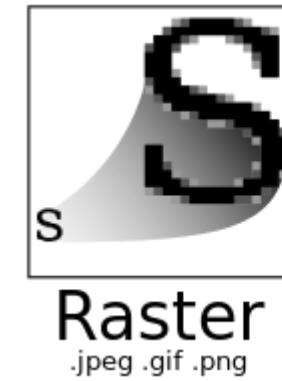
- At some point, a vector image is converted into a [raster graphics](#) image, which maps bits directly to a display space (and is sometimes called a *bitmap*). The vector image can be converted to a raster image file prior to its display so that it can be ported between systems.
- A vector file is 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.

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

car.svg X

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <!-- Created with Inkscape (http://www.inkscape.org/) -->
3 <svg
4   xmlns:dc="http://purl.org/dc/elements/1.1/"
5   xmlns:cc="http://web.resource.org/cc/"
6   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
7   xmlns:svg="http://www.w3.org/2000/svg"
8   xmlns="http://www.w3.org/2000/svg"
9   xmlns:xlink="http://www.w3.org/1999/xlink"
10  xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
11  xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape"
12  width="900"
13  height="600"
14  id="svg2"
15  sodipodi:version="0.32"
16  inkscape:version="0.44+devel"
17  version="1.0"
18  sodipodi:docbase="/home/d/ink/inkscape/share/examples"
19  sodipodi:docname="car.svgz"
20  inkscape:output_extension="org.inkscape.output.svgz.inkscape"
21  inkscape:export-filename="/home/hrum/Desktop/carrr/killxl_1.png"
22  inkscape:export-xdpi="100"
23  inkscape:export-ydpi="100"
24  sodipodi:modified="true">
25 <defs
26   <linearGradient
27     id="defs4"
28     inkscape:collect="always"
29     id="linearGradient4399">
30     <stop
31       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.