Part 2 – Units -> Digits -> Number Systems -> *Data Codes*

In the beginning there was the UNIT.

UNIT = 1

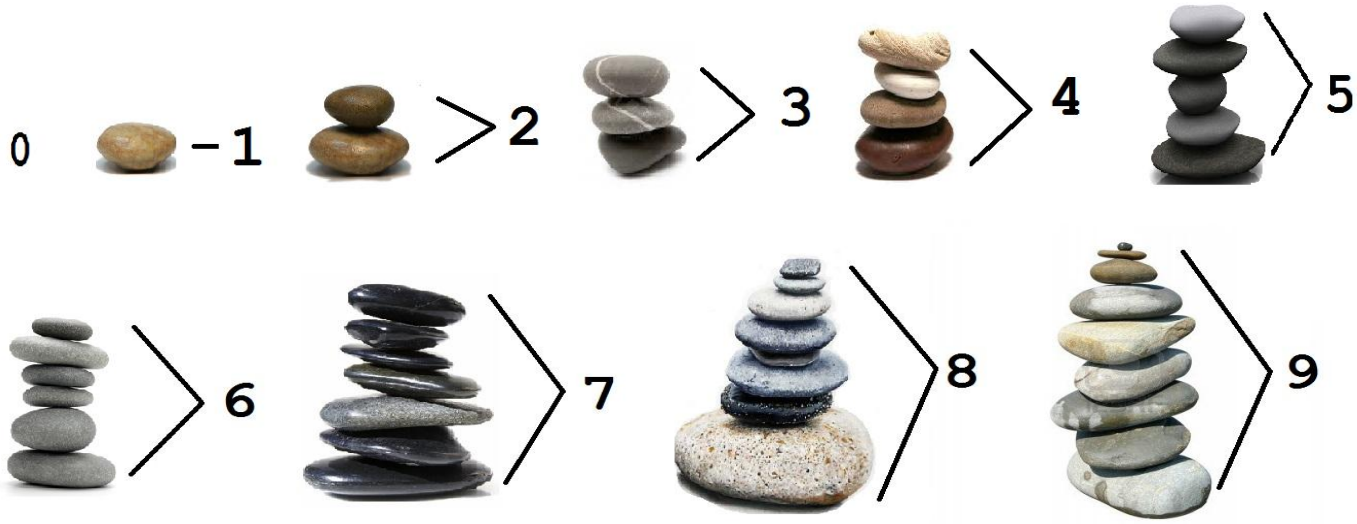
Things exist and have characteristics. An Important characteristic of an entity/thing is that it is a **UNIT**.
We can represent a UNIT as a **1**.

If there is more than one UNIT (1), how are you expected to hold all them in your head individually ? No.

Remember the crow. If they are few units in number, you could hold them in memory. If they are larger than your crow, you cannot hold them in your focus. So how do you deal with too many things/units ? Digits.

Digits -

You are quite familiar with digits: 0, 1,2,3,4,5,6,7,8,9



Each of the ten (10) digits, as seen illustrated above, represent a count of a particular number of units, as a **SINGLE** symbol. (Many to one) For years you have been using the digits 0, 1,2,3,4,5,6,7,8 and 9.

Definition - ***Decimal Digits*** – '***Single*** *symbol*' representing 0, 1, or many up to 9 units.

How do you represent an amount more than 9 ? We use a '**number system**'.

Videos – Watch ME Story of Digits – 0, 1 to 9 digits.

Part 1 - http://www.youtube.com/watch?v=gulApUKih2w

Part 2 - http://www.youtube.com/watch?v=FRpG6SKnRYA&list=PLB6E11989705F3A38

Number Systems

How do you represent a count greater than the number of … number symbols ?
Use a **Number System**.

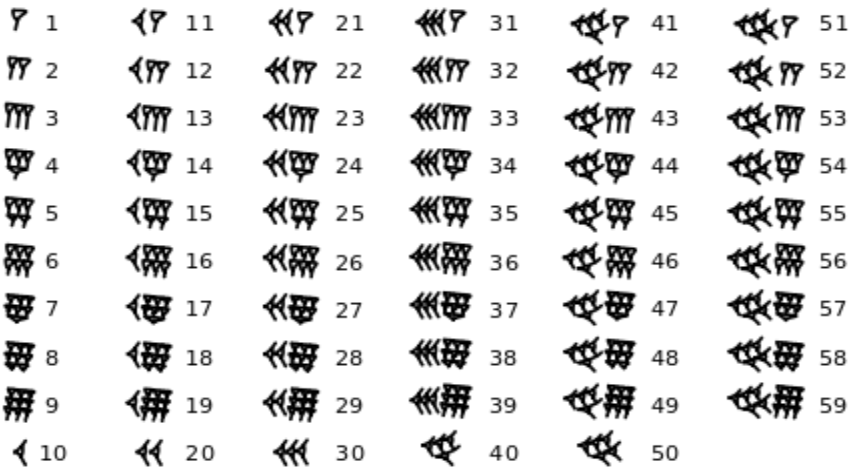
Today, the basis of all numbering system is the same.
'Modern Numbering Systems' have 3 defining characteristics:

- Maximum ***number of characters/digits used in a position***
- **Position**
- Each **Position Value** is multiplied by the Max-number-of-digits of the position value on its left.

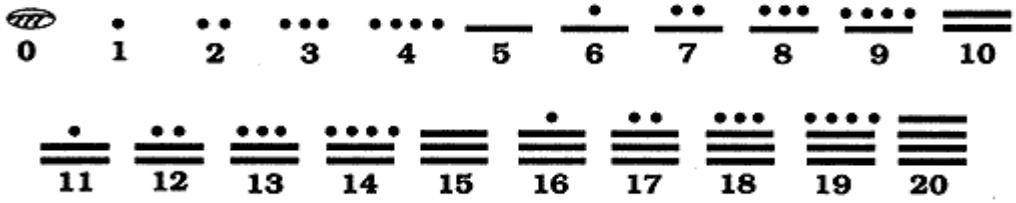
Number of Characters

The only difference between numbering systems is the ***number*** of characters in the number system.
Here are 6 different **Base** (number of characters) number Systems:

60 – Babylonian 60 different characters (Sexagesimal) :



20 – Mayan 20 different characters (Vigesimal):



16 – **Hexadecimal** 16 different characters : **0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F**

10 – **Decimal** 10 different characters : **0,1,2,3,4,5,6,7,8,9**

8 – Octal 8 different characters : **0,1,2,3,4,5,6,7**

2 – **Binary** 2 different characters : **0, 1**

Definition: **Base** – The number of digits used in number system.

Decimal Number System

Let's check out what you have been doing for years when you counted the number of things greater than 9. You used a 10 character number system, called the **Base 10** or **Decimal** Number system.

Base 10 number system – Decimal System. Your every day number system – easy to read and write.

The ***Decimal*** numbering system/code is based on using **ten** symbols/digits to uniquely represent values 0 to 9. (Dec means 10)

- The ten(10) symbols for base 10 number are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. (***Maximum*** *10 digits in Decimal system*)
- Recall, a numbering system is **positional**.
- The **position values** for the base 10 numbering system are:

  ___ ___ ___ ___ ___ ___ (The **Positions**)
  100,000's 10,000's 1,000's 100's 10's 1's (Note: a **Position Value** is a multiple of 10)

Example: Given the base 10 number: 752. This means: 7 100's, 5 10's, and 2 1's.
Example: Given the base 10 number: 65. This means: 6 10's, and 5 1's.

You have been using this modern number system for years…

Binary Number System

Let's look at another number system that only has two (2) characters in it: 0, 1.

Binary means there two (2) of something.
Example: Bicycle (2 wheels), Biannual (2 times a year), Biped (animal with 2 feet/legs)

Base 2 number system – Binary System. This system is the basis for modern digital electronics.

The **Binary** numbering system/code is based on using two (2) symbols, which you can use to uniquely represent all data.

- The two(2) symbols for a base 2 number system are: 0, 1 (**Maximum** of 2 digits)
- Recall, a numbering system is **positional**.
  The **position values** for the base 2 numbering system are:

  ___ ___ ___ ___ ___ ___ ___ ___ (The **Positions**)
  128's 64's 32's 16's 8's 4's 2's 1's (Note: a **Position Value** is a multiple of 2)

Example: Given the base 2 number: 1011.
This means: 1 8's, 0 4's, and 1 2's, and 1 '1'.
This adds up to the number 11 (8 + 2 + 1) in the base 10 number system. 1011 in base 2 … is 11 in base 10.

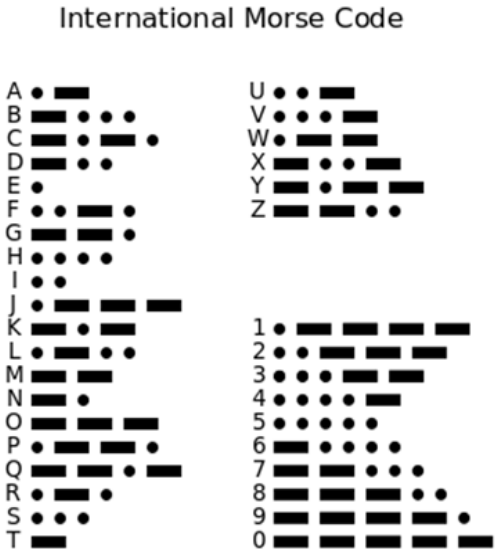
How a base 2, 8, 10 and 16 number system works:

Video – Watch Me *Number Systems*: http://www.youtube.com/watch?v=5sS7w-CMHkU
List of other Number system: https://en.wikipedia.org/wiki/List_of_numeral_systems

Early Electronic Digital Device

The Telegraph

The telegraph was used to send messages between cities. Messages were converted, letter by letter, into a Digital/Binary code. The telegraph code, called the **Morse code**, is a binary code. Binary mean that you used two (2) symbols, in unique combinations, to represent data. Below you will see how the Telegraph used Dots and Dashes (also called shorts and longs). Each letter has a unique combination of dots and dashes assigned to it.



You can also represent the code as

A 0 1
B 1 0 0 0
C 1 0 1 0 Etc. if you use binary number.

The *physical implementation* of the *binary* nature of the telegraph is the fact you used electricity to create 'longs and short' paused-clicks on the Telegraph Key device.

The telegraph consisted of copper metal lines/wires strung on poles between cities. The wires had electricity flowing over them.
You could cause the electricity to flow or not flow by pressing on a KEY, to open or close the electrical circuit.
When you close the circuit, by pressing the key, electricity would flow, and then send a pulse down the line to the next station, where their key would click audibly.
By varying the length the key presses, either long or short, you could create two(2) different lengths of audio clicks, thus sending the encoded letters to the next telegraph station.

The telegraph was the first successful electronic binary communication device. It was the email of it era.

Watch this video: http://www.youtube.com/watch?v=oo0hSZ9R_Xk

Note: You will be tested on the content of videos and web references.

The code - The sole of the modern machine

BINARY/DIGITAL *CODE* – The power of 2.

Computers do **NOT** actually store PRINTED letters, numbers or symbols, like you and I read and write them: A B C.... 1 2 3... +/?.

So, how do they do store, move, and process information ?

Computers today are mainly Digital/Binary devices. Digital/Binary refers to two(2) of something.
Commonly, these two(2) states are represented as a

'**1**' or a '**0**'

Each letter of the Alphabet, a special symbol or some commands are assigned/encoded as a unique combinations or sequence of two(2) characters 1's or 0's.

The Binary/Digital assignment/encoding is how **ALL** Information/Data/Instructions are 'represented', 'stored', 'processed' and 'moved' around inside the devices.

Digital related devices only understands the binary representation, the 1's and 0'. That is why your computing devices today are called Digital or Binary devices.

Examples: Digital Camera, Digital Video Recorder (DVD), Digital Sound, Digital Subscriber Line (DSL), Digital Computer, Digital Smart Phones, Flash Drive, Etc.

DEFINITION: **Digital/Binary Device** – A device that utilizes 1's and 0's to store, move and process data and instructions.

In the past, most devices were **Analog Devices**. Analog Devices still exist but have been largely supplanted by a Digital Design. An Analog Device uses a range of values. An analog device is like a faucet. You can go from completely off, across a range of the amount of water released, to a maximum amount of water.
Digital techniques are more frequent employed today, because it is much easier to design an electronic device to switch into one of two binary states, than to accurately reproduce a continuous range of analog values. Some devices include both Analog and digital components.

This class focuses on Digital/Binary Electronic Devices.

DEFINITION: **Analog** – Uses a range of signal values, in contrast to just two signal values like digital/binary devices.

Two coding systems commonly in use today

ASCII Code - American Standard Code for Information Interchange. This is one of first standardized binary coding schemes in use. It became commonly used in the 1980's. This code is good for the English Alphabet.

Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	␣	100 0000	100	64	40	@	110 0000	140	96	60	`
010 0001	041	33	21	!	100 0001	101	65	41	A	110 0001	141	97	61	a
010 0010	042	34	22	"	100 0010	102	66	42	B	110 0010	142	98	62	b
010 0011	043	35	23	#	100 0011	103	67	43	C	110 0011	143	99	63	c
010 0100	044	36	24	$	100 0100	104	68	44	D	110 0100	144	100	64	d
010 0101	045	37	25	%	100 0101	105	69	45	E	110 0101	145	101	65	e
010 0110	046	38	26	&	100 0110	106	70	46	F	110 0110	146	102	66	f
010 0111	047	39	27	'	100 0111	107	71	47	G	110 0111	147	103	67	g
010 1000	050	40	28	(	100 1000	110	72	48	H	110 1000	150	104	68	h
010 1001	051	41	29	)	100 1001	111	73	49	I	110 1001	151	105	69	i
010 1010	052	42	2A	*	100 1010	112	74	4A	J	110 1010	152	106	6A	j
010 1011	053	43	2B	+	100 1011	113	75	4B	K	110 1011	153	107	6B	k
010 1100	054	44	2C	,	100 1100	114	76	4C	L	110 1100	154	108	6C	l
010 1101	055	45	2D	-	100 1101	115	77	4D	M	110 1101	155	109	6D	m
010 1110	056	46	2E	.	100 1110	116	78	4E	N	110 1110	156	110	6E	n
010 1111	057	47	2F	/	100 1111	117	79	4F	O	110 1111	157	111	6F	o
011 0000	060	48	30	0	101 0000	120	80	50	P	111 0000	160	112	70	p
011 0001	061	49	31	1	101 0001	121	81	51	Q	111 0001	161	113	71	q
011 0010	062	50	32	2	101 0010	122	82	52	R	111 0010	162	114	72	r
011 0011	063	51	33	3	101 0011	123	83	53	S	111 0011	163	115	73	s
011 0100	064	52	34	4	101 0100	124	84	54	T	111 0100	164	116	74	t
011 0101	065	53	35	5	101 0101	125	85	55	U	111 0101	165	117	75	u
011 0110	066	54	36	6	101 0110	126	86	56	V	111 0110	166	118	76	v
011 0111	067	55	37	7	101 0111	127	87	57	W	111 0111	167	119	77	w
011 1000	070	56	38	8	101 1000	130	88	58	X	111 1000	170	120	78	x
011 1001	071	57	39	9	101 1001	131	89	59	Y	111 1001	171	121	79	y
011 1010	072	58	3A	:	101 1010	132	90	5A	Z	111 1010	172	122	7A	z
011 1011	073	59	3B	;	101 1011	133	91	5B	[	111 1011	173	123	7B	{
011 1100	074	60	3C	<	101 1100	134	92	5C	\	111 1100	174	124	7C	|
011 1101	075	61	3D	=	101 1101	135	93	5D	]	111 1101	175	125	7D	}
011 1110	076	62	3E	>	101 1110	136	94	5E	^	111 1110	176	126	7E	~
011 1111	077	63	3F	?	101 1111	137	95	5F	_					

The ASCII table heading are:

Binary: Base 2 value represented by 1's and 0'. *The Binary code is the actual code used inside digital devices*.

Octal: Binary value converted to the base 8 number system, *for the convenience of people*.

Decimal: Binary value converted to the base 10 number system, *for the convenience of people.*

Hexadecimal: Binary value converted to the base 16 number system, *for the convenience of people*.

Glyph: Symbol as people would write and read it, or command.

The 4 different numbering systems are explained in more detail below.

Note: You will be tested on the content of videos and web references.

See: http://en.wikipedia.org/wiki/Ascii

Read: http://edition.cnn.com/TECH/computing/9907/06/1963.idg

UNICODE - Universal Code.

The idea of the Universal code (Unicode) is that it can handle all characters found in all languages, new and old. Unicode is not just limited to Latin/English such as the ASCII Code discussed above.

Unicode Standard, the latest version of Unicode contains a repertoire of more than 110,000 characters covering 100 different alphabets.

There are Three(3) commonly used versions of Unicode. They depend on how many positions avail for 1's and 0's.

- UTF-8 – an 8 positions (bits) variable-width encoding which maximizes compatibility with ASCII.
- UTF-16 – a 16 positions (bits) variable-width encoding
- UTF-32 – a 32 positions (bits) fixed-width encoding

See: http://en.wikipedia.org/wiki/Unicode for chart.

DEFINITION: **BINARY or DIGITAL Code** - a set of two symbols or states, 0 and 1, used to represent glyphs/symbols.
DEFINITION: **CODE** - An alternate representation for letters, numbers or symbols.

The 1's and 0's

What are really the 1's and 0's ?

There are NO actual 1's or 0's inside digital devices !

The 1's and 0's are really first level concepts, used to simplify and to make it easy to think about digital devices. There are many different physical methods that can be setup/designed to store and process two (2) different states.

Using 1's and 0's is a Concept, an Abstraction. This is so you have one thing to deal with mentally, instead of many different *physical technologies* to deal with. Remember the crow.

As we look **later** at the different types of digital devices, we will discuss the many different physical designs used to create two(2) different states, which let us use Binary CODES.

Conversion between Four different coding/numbering Systems

Base 2 – uses 2 digits: 0, 1. *Base 2 is the only code actually used in digital devices.*
Base 8 – uses 8 digits: 0,1,2,3,4,5,6,7
Base 10 – uses 10 digits: 0,1,2,3,4,5,6,7,8,9
Base 16 – uses 16 digits: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Check out this online Converter between Decimal, Binary and Hexadecimal numbers:
http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html
Note: You will be tested on the content of videos and web references.

Base 16 number system – Hexadecimal – Easy to read and write – Shortest representation of a number.

The Hexadecimal numbering system/code is based on using sixteen (16) symbols to uniquely represent all data.

- The sixteen(16) symbols for base 16 number are: 0, 1, 2,3,4,5,6,7,8,9,A,B,C,D,E,F.
  A = 10, B=11, C=12, D=13, E=14, F=15.
- Recall, a numbering system is positional.
  The position values for the base 16 numbering system are:

  ___ ___ ___ ___ ___
  65,536's 4,096's 256's 16's 1's

Example: Given the base 16 number: FFFF.
This means: 15 4,096's , 15 256's, 15 16's, and 15 '1'.
This adds up to the number 65535 in the base 10 number system. 1111111111111111 in base 2.

The base 16 number system is **not** used inside digital devices. It is only use for the convenience of people to read/write.
What is easier to write/read: 1111111111111111 (base 2) or its equivalent 65,535 (base 10) ?
Note: Hexadecimal numbers are commonly used to represent the address of bytes in RAM (discussed later in detail)

Code Tables for Computer Devices

Every computer device moves, stores or processes data.
The binary coded data 'may' be different for each device.

On one device, 10010 may mean the letter A in one code.
On another device, the same binary code 10010 may mean the letter Z.

When data is moved between devices it 'may' have to be **translated/converted** into another code format that the receiving digital device can understand and use.

There is a group of special programs called '**Device Drivers'** that performs the conversion/translation task.
We will talk about *different devices*, and their associated device drivers in the upcoming topic modules.

http://en.wikipedia.org/wiki/Operating_system#Device_drivers

Code ASSIGNMENT

1) Explain how the base 8, Octal numbering systems work.
2) Convert Octal 7777 to binary, decimal and Hexadecimal.
3) What really are the 1's and 0's of digital devices ?
4) What is the history of the base 60 number system ?
5) How do you use the base 60 number system today ?
6) Who used the binary number system first ?
7) Why are digital electronic designs used more commonly then analog electronic designs ?

Write your assignment in MS Word.
Write the question, followed by the answer.
Name your file: FirstNameLastNameCodeAssignPart2.
Turn in file online to the appropriate location for grading.