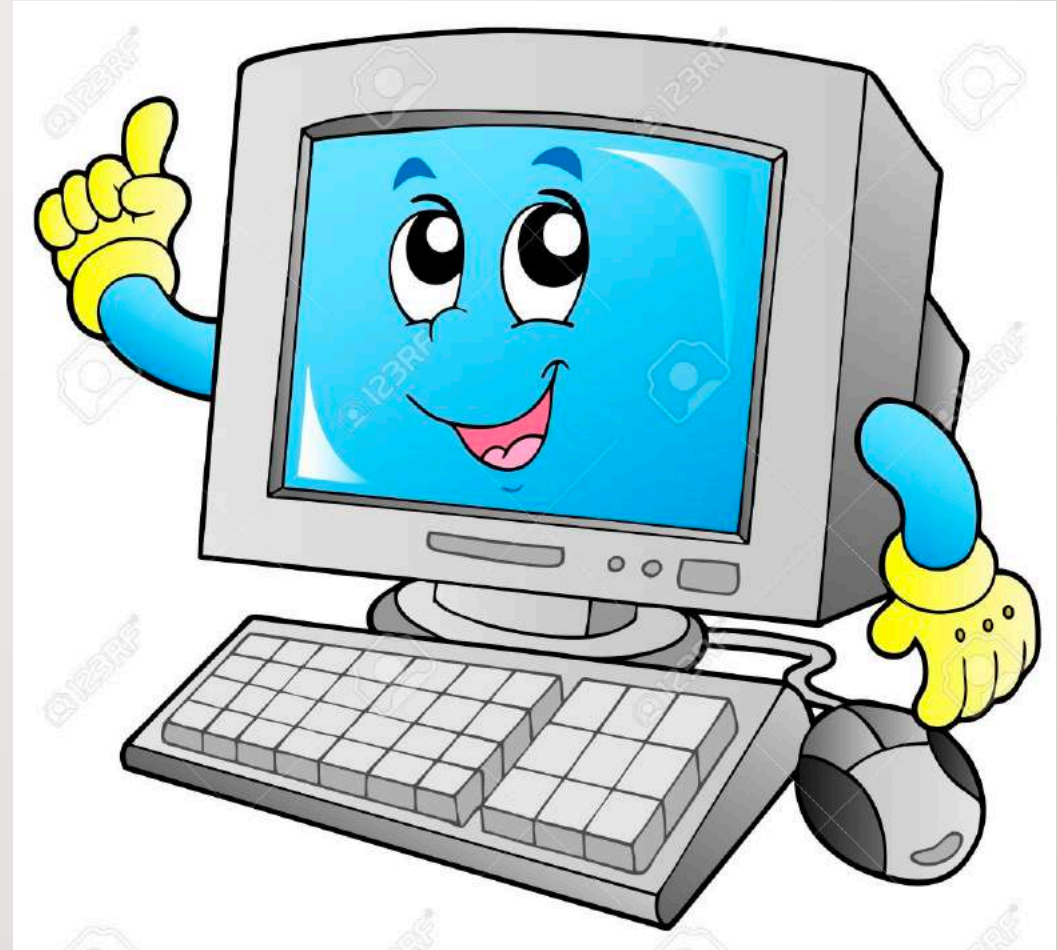


INTRODUCTION TO PROGRAMMING

WHAT IS COMPUTER?





- Hammer
- Saw
- Measuring Tapes

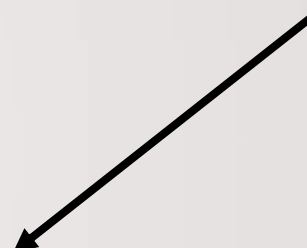
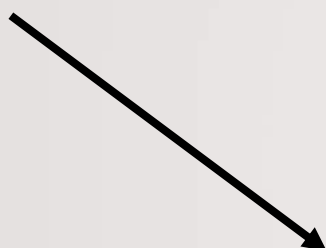


- Probes
- Scopes
- Meters

- Every profession has tools that make the job easier to do



- Computer is a tool used by so many professions that it can not be easily categorized.
- It can perform so many different jobs that is perhaps the most versatile tool ever made.



How computers can be used by so many different professions?



- The computer can do such a wide variety of tasks because it can be programmed.
- It is a machine specifically designed to follow the instructions.

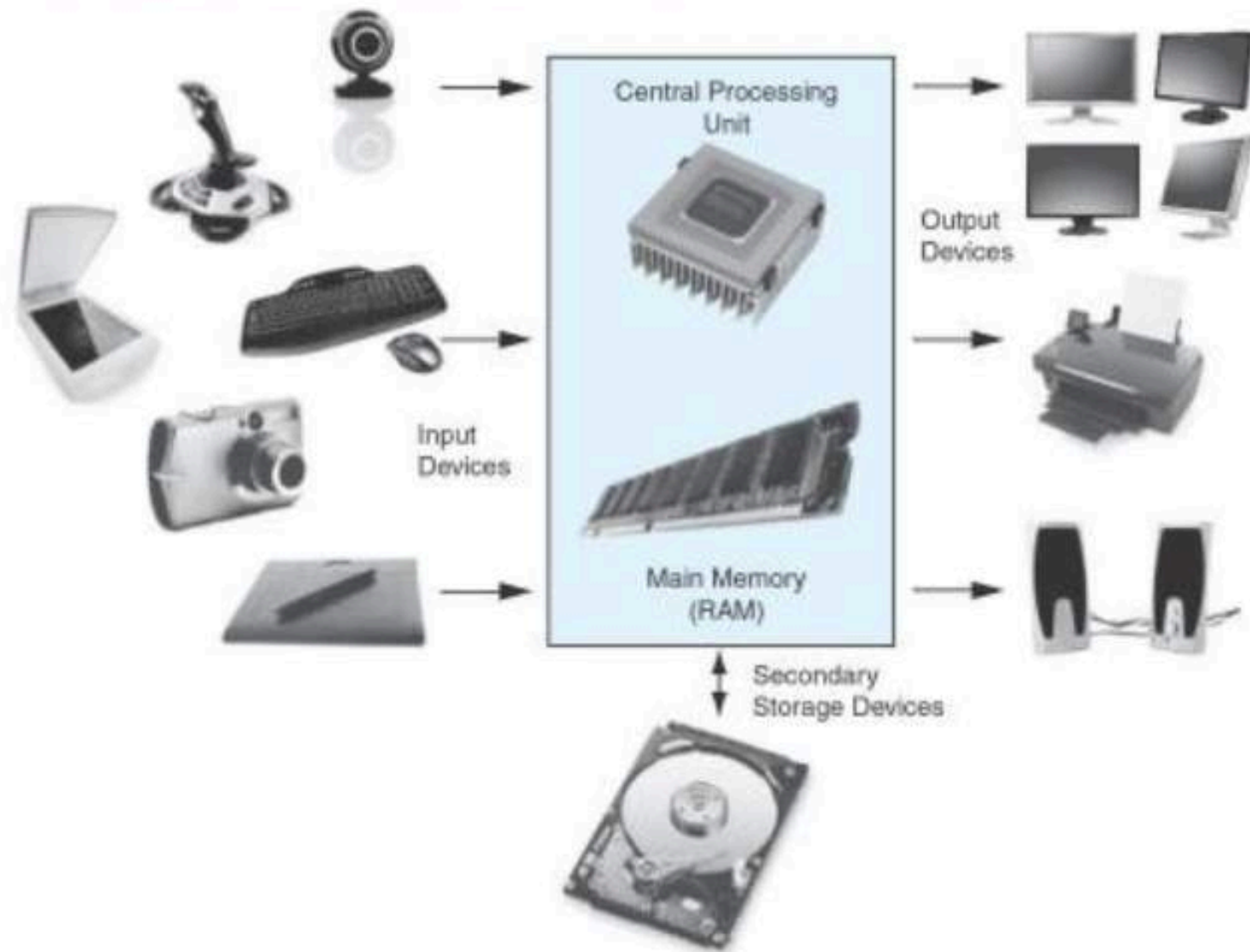
```
...element* item = el->FirstChildElement();  
GroupDesc::ElementDesc elDesc;  
  
std::string sp_name = item->Attribute("name");  
std::string spritename = item->Attribute("spritename");  
  
float x = boost::lexical_cast<float>(item->Attribute("x"));  
float y = boost::lexical_cast<float>(item->Attribute("y"));  
float offset = boost::lexical_cast<float>(item->Attribute("offset"));  
unsigned layer = 50; // default  
if (item->Attribute("layer")) {  
    layer = boost::lexical_cast<unsigned>(item->Attribute("layer"));  
}  
  
elDesc.name_ = sp_name;  
elDesc.spriteName_ = spritename;  
elDesc.x_ = x;
```

COMPUTER COMPONENTS

- All computer systems consist of similar **hardware** devices and **software** components.

HARDWARE

- Hardware refers to the physical components that a computer is made of.
- Computer is not an individual device, but a system of devices.
- A typical computer system consists of following major components:
 - The central processing unit (CPU)
 - Main memory
 - Secondary storage devices
 - Input devices
 - Output devices

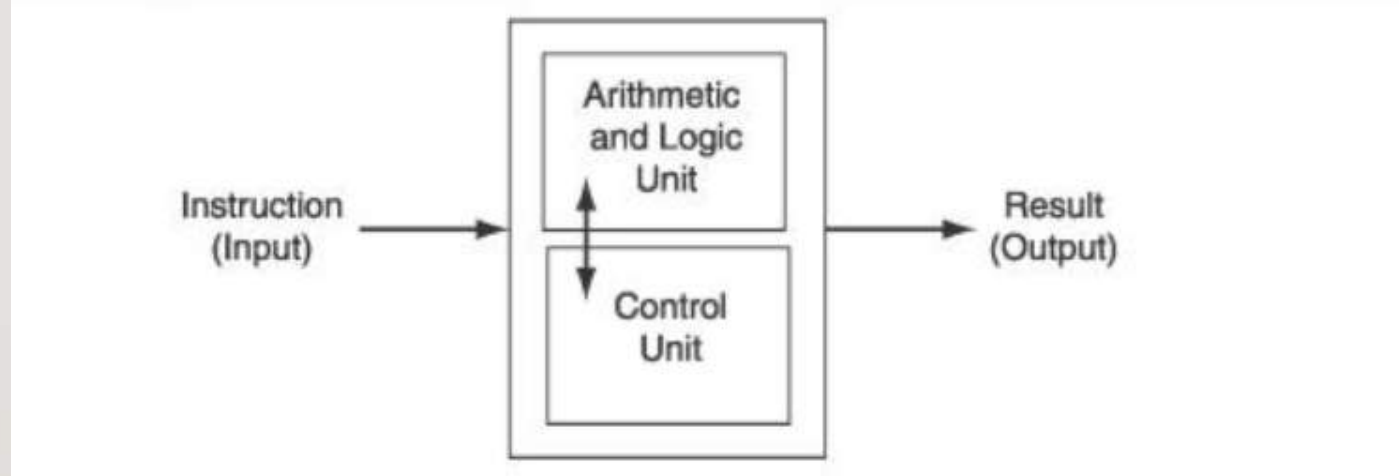


CPU



CPU

- Central Processing Unit(CPU) is the heart of a computer.
- CPU's job is to fetch instructions, follow the instructions, and produce some resulting data.

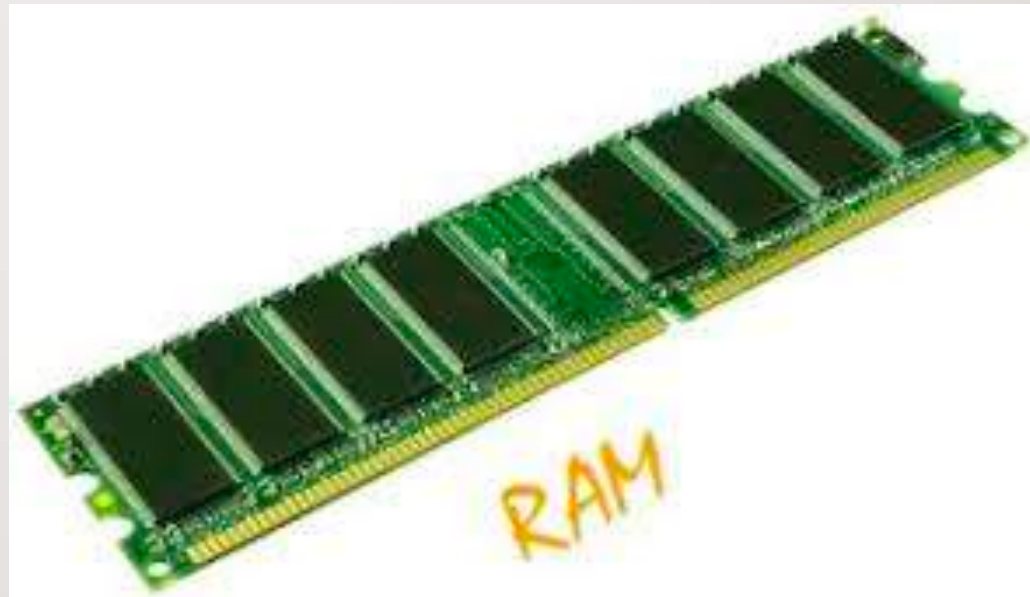


a computer is...

**a device that accepts data or input and processes it in
some way to automatically produce a result**

MAIN MEMORY

- Commonly known as random access memory, or RAM
- Main memory is a device that holds information.
- RAM holds the sequences of instructions in the programs that are running and the data those programs are using.



SECONDARY STORAGES

- Secondary storage is a type of memory that can hold data for long periods of time-even when there is no power to computer.
- Frequently used programs are stored in secondary memory and loaded into main memory as needed.
- Important data, such as word processing documents, payroll data, and inventory figures, is saved to secondary storages as well.

HDD



SSD



FLASH DRIVE



WHAT IS A PROGRAM?

- A computer program is a set of instructions a computer follows in order to perform a task.
- A programming language is a special language used to write computer programs.

```
JavaScript Source Code
var last_friday_of_month, print_last_fridays_of_month;

last_friday_of_month = function(year, month) {
  var i, last_day;
  i = 0;
  while (true) {
    last_day = new Date(year, month, i);
    if (last_day.getDay() === 5) {
      return last_day.toString();
    }
    i += 1;
  }
};

print_last_fridays_of_month = function(year) {
  var month, results;
  results = [];
  for (month = 1; month <= 12; ++month) {
    results.push(console.log(last_friday_of_month(
  )
});

year = parseInt(process.argv[2]);
return print_last_fridays_of_month(year);
})();
```

JavaScript Source Code

```
Python Source Code
def enabled(self):
    return self._ydl.params.get('cachedir') is

def store(self, section, key, data, dtype='json',
    assert dtype in ('json',))

    if not self.enabled:
        return

    fn = self._get_cache_fn(section, key, dtype)
    try:
```

Python Source Code

```
    write_json_file(cache_dir, fn)
except Exception:
    tb = traceback.format_exc()
    self._ydl.report_warning(
```

```
Swift Source Code
enum GroundType: Int {
  case Grass
  case Rock
  case Water
  case InTheAir
  case Count
}
```

Dictionary

Q Search

source code

|sɔːrs kōd|

noun *computing*

A collection of computer instructions written using some human-readable computer programming language, usually as plain text.

Source Code

```
ParticleEmitter! = nil
ParticleEmitter! = nil
ParticleEmitter! = nil
```

```
C# Source Code
enumable<DateTime> LastFridaysOfYear(int year)
{
  = 1; month <= 12; month++
  new DateTime(year, month, 1).AddMonths(1).AddDays(-1);
  e.DayOfWeek != DayOfWeek.Friday)
  {
    date = date.AddDays(-1);
  }
}
```

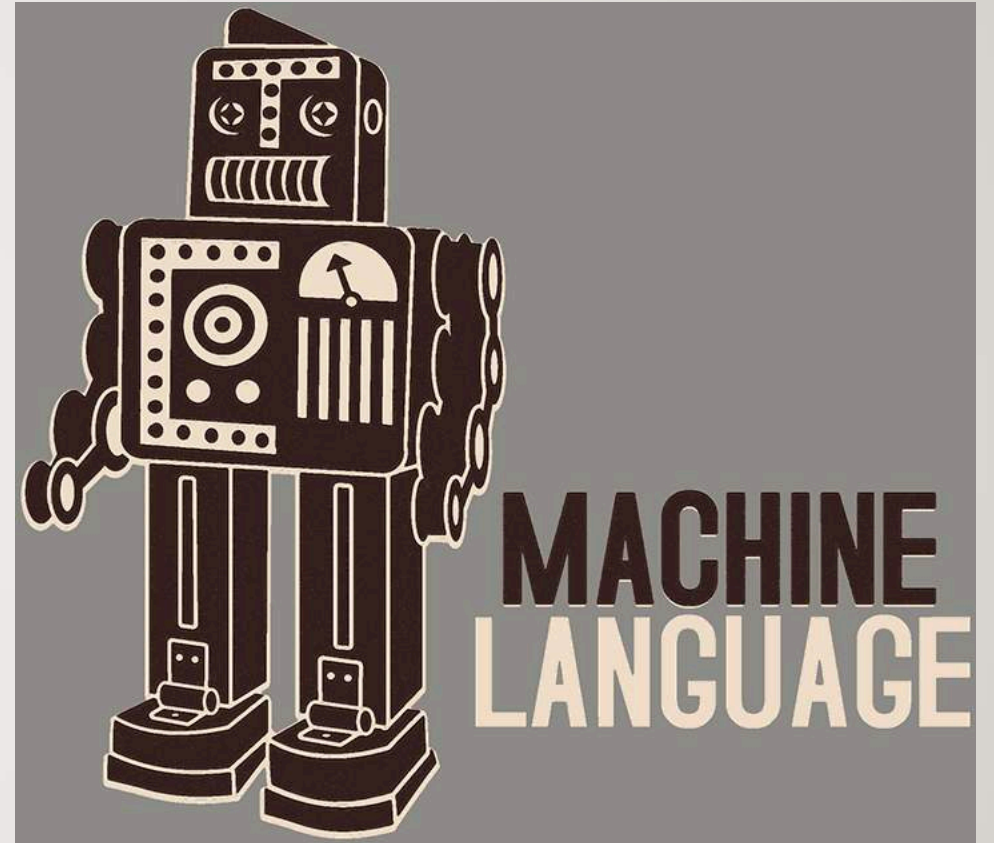
C# Source Code

```
{
  int year;
  var argument = arguments.FirstOrDefault();
```

- How computers understands the source code?
- What language computer understands?



- A computer's CPU can only process instructions that are written in **machine language**



MACHINE LANGUAGE

- Machine language is a collection of **binary digits** or **bits** that the computer reads and interprets.
- Machine language is the only language a computer is capable of understanding.

BINARY

- Binary is system of counting

1	I	6	I
2	II	7	II
3	III	8	III
4	IIII	9	IIII
5		10	



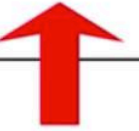


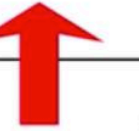
Tally marks

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22						29	30
.....									40
.....									
.....								99	100

Base ten positional

It is the system
which we use today

Two different type of counting system

	2	2	7	0	6
100,000	10,000	1,000	100	10	1
 10^5	 10^4	 10^3	 10^2	 10^1	 10^0






Base Ten:


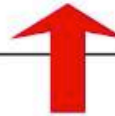
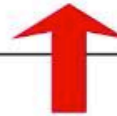
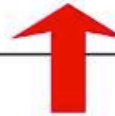


Greater than 9=New Digit

Binary:

Greater than 1=New Digit

COUNTING IN BINARY

BINARY NUMBER	AMOUNT OF THINGS
0	
1	
10	
11	
100	
101	

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

0	1	1	0	0	1
32	16	8	4	2	1

$$16 + 8 + 1 = 25$$

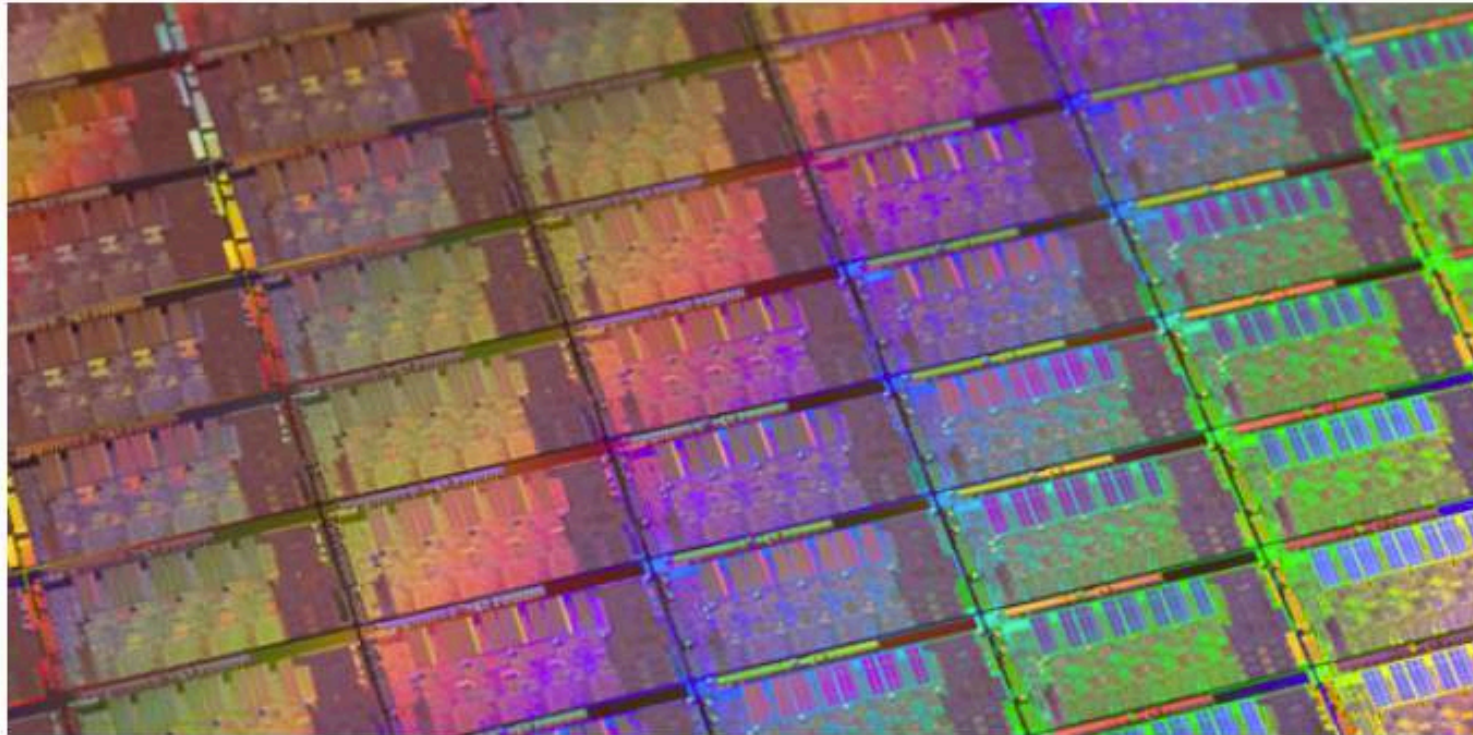
- Why computer uses binary? Why not base ten?



CPU



TRANSISTORS

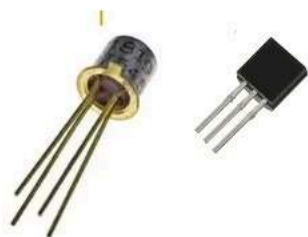




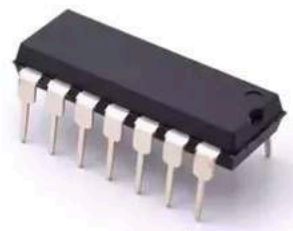
1st generation



2nd generation



**167 transistors
(74181 ALU)**



3rd generation



3510 transistors

4th generation

2,600,000,000 transistors



Binary!

1

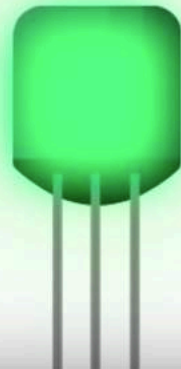
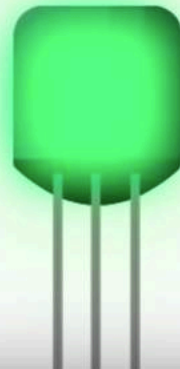
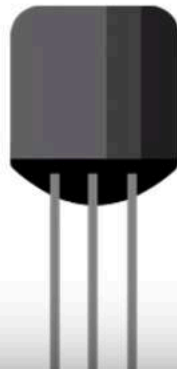
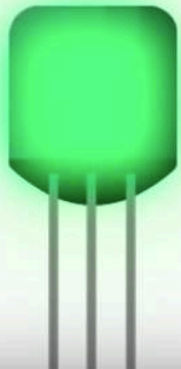
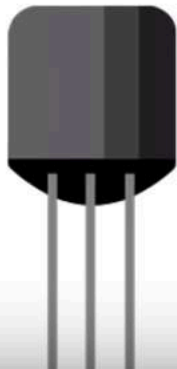
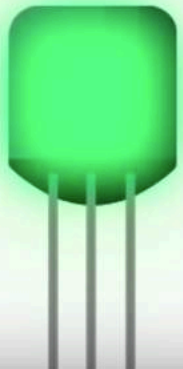
0

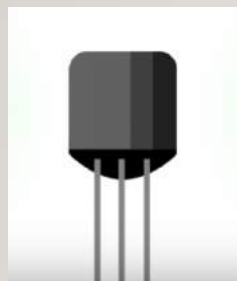
1

0

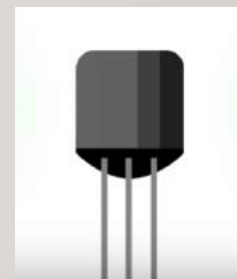
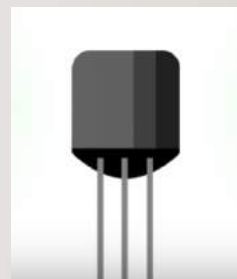
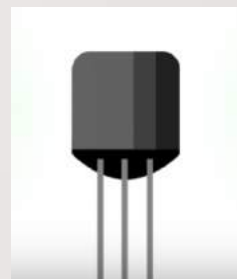
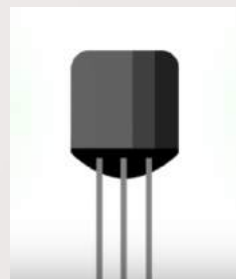
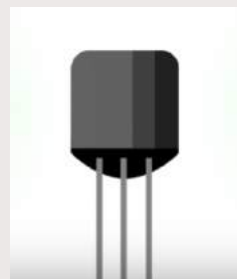
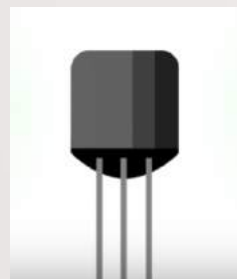
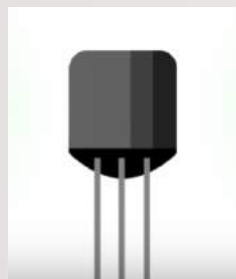
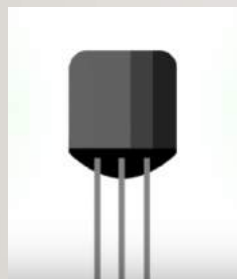
1

1

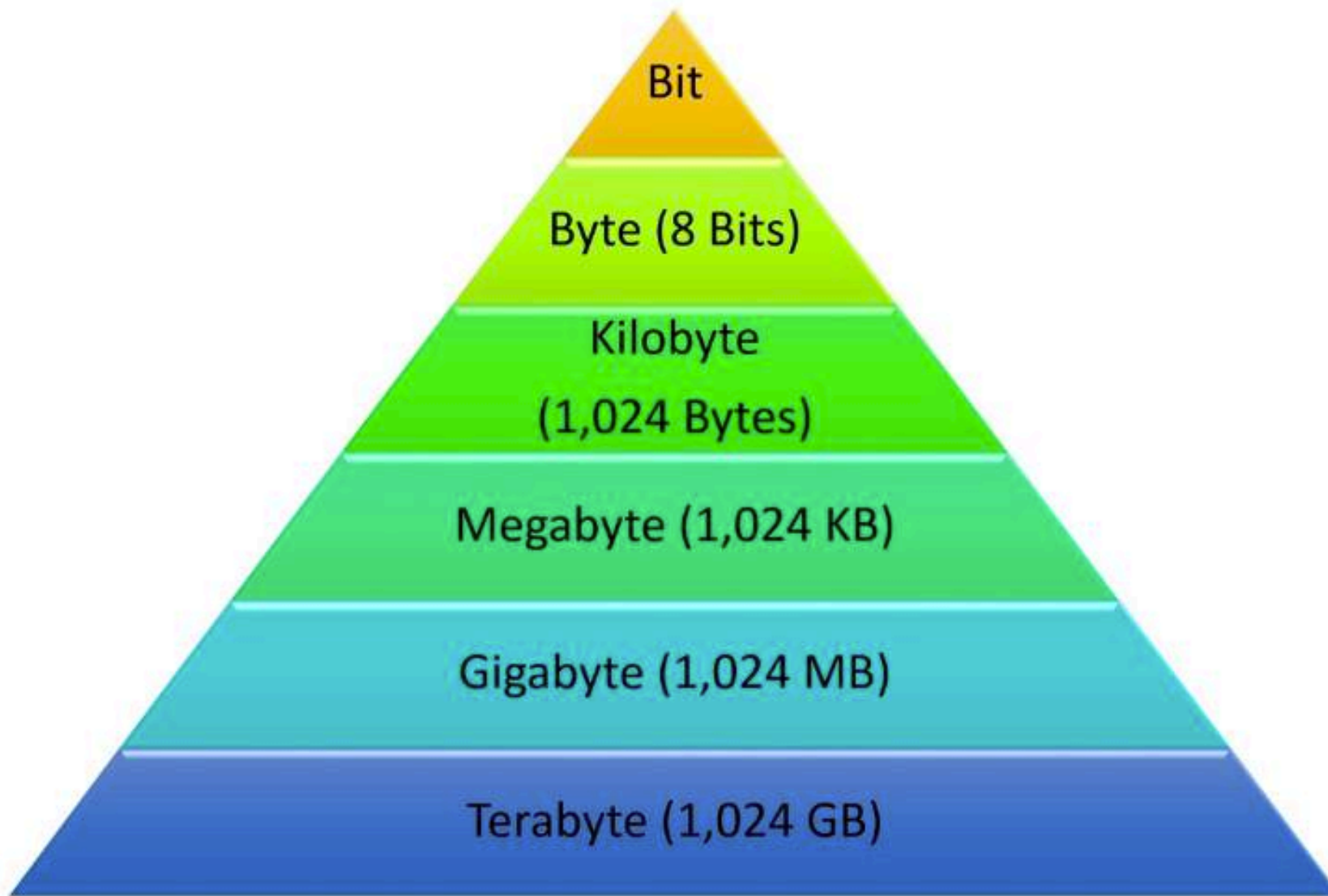




= BIT



BYTE



- How about spelling in binary?



ASCII(AMERICAN STANDARD CODE INFORMATION INTERCHANGE)

- ASCII assigns a character to each value represented by a byte of binary
- Since byte has eight digits binary to work with and eight digits of binary can represent up to 255 values ASCII have 255 letters and the symbols to choose from

Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

A

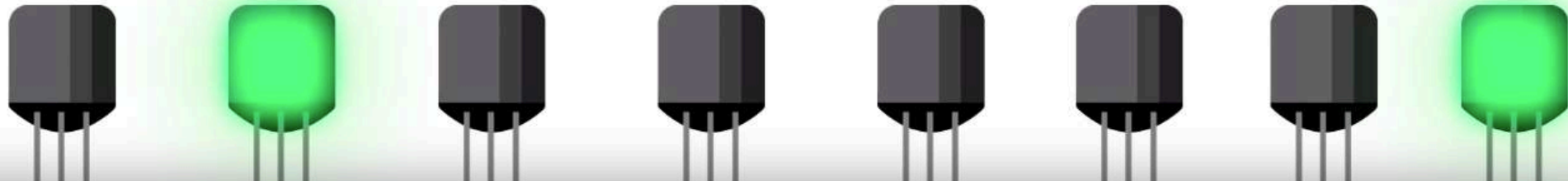
Decimal	Binary	Octal	Hex	ASCII
---------	--------	-------	-----	-------

64	01000000	100	40	@
65	01000001	101	41	A

$$64 + 1 = 65$$

128 64 32 16 8 4 2 1

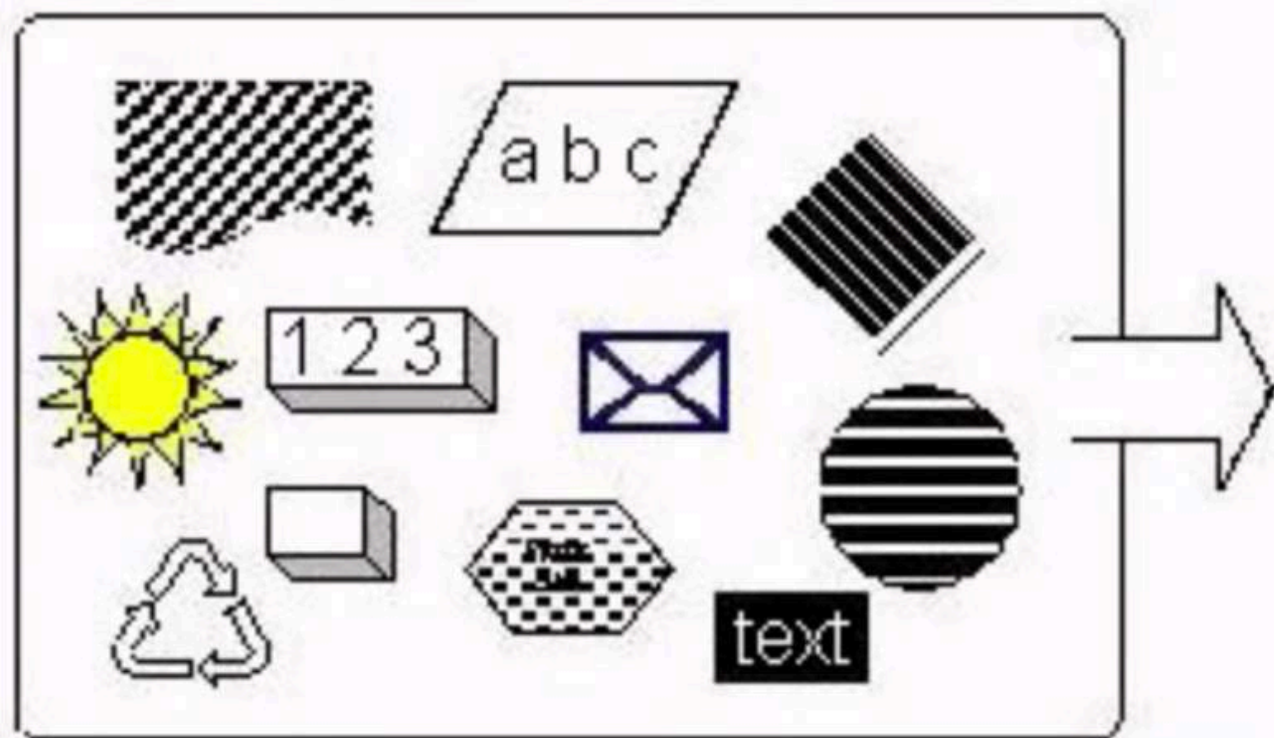
0 1 0 0 0 0 0 1




```
01001000 01100101 01101100 01101100 01101111 00100000 01010111 01101111  
01110010 01101100 01100100
```

Machine language (binary) for the text “Hello World”

Your Data



Computer Data

```
01110101011010101
10100101011010101
01010101011010101
01000101011010101
01101010101001100
00101011101100111
10101001010101010
```