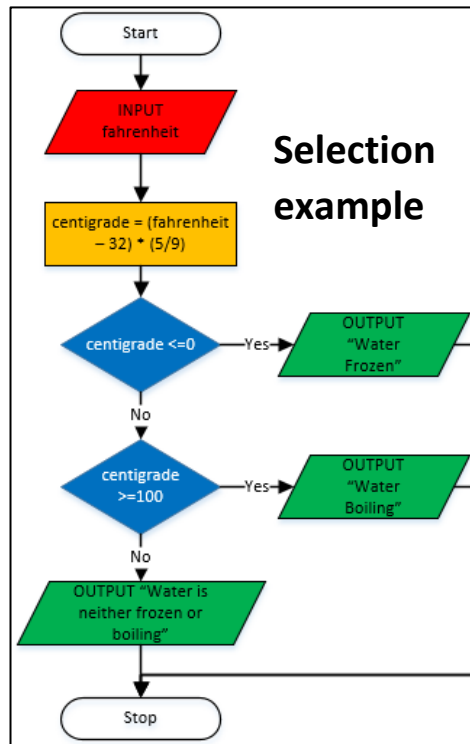



# Year 9 Computer Science – Autumn Term

## Python Programming

<b>Python</b>	A high level programming language.
<b>Programming</b>	The process of writing computer programs. Code The instructions that a program uses.
<b>Sequence</b>	Parts of the code that run in order and the pathway of the program reads and runs very line in order.
<b>Selection</b>	Selects a pathways through the code based on whether a condition is true
<b>Iteration</b>	Code is repeated (looped), either while something is true or for a number of times
<b>Algorithm</b>	A set of rules/instructions to be followed by a computer system
<b>Variable</b>	A value that is stored in the program and can change whilst the program is executed. (eg. time, speed)
<b>Arithmetic Operator</b>	Used when calculating data. + - / * % (remainder of the calculation) ^ (to the power of)
<b>Comparative Operator</b>	When comparing data, an operator is used to solve the equality such as <>, != or ==
<b>Syntax</b>	The punctuation/way that code has to be written so that the computer can understand it. Each programming language has its own syntax.
<b>Data Type</b>	This indicates how the data will be stored. The most common data types are integer, string, and float/real.
<b>String</b>	A collection of letters, numbers or characters. (eg, Hello, WR10 1XA)
<b>Integer</b>	A whole number. (eg. 1, 189)
<b>Float/Real</b>	A decimal number, not a whole number. (eg. 3.14, -26.9)
<b>Boolean</b>	1 of 2 values. (eg. True, False, Yes, No)

### Python Variable and Type Example

```

x = 10
print("My fav no is " + x)
  
```

For loop	Iteration	While loop
----------	-----------	------------

Program:	Output:	Program:	Output:
<pre> for i in [5, 4, 3, 2, 1] :     print i  print 'Blastoff!'           </pre>	<pre> 5 4 3 2 1 Blastoff!           </pre>	<pre> n = 5 while n &gt; 0 :     print n     n = n - 1 print 'Blastoff!' print n           </pre>	<pre> 5 4 3 2 1 Blastoff! 0           </pre>

Python -> English	
<code>print('hello!')</code>	Prints a value on screen (in this case, hello!)
<code>input('')</code>	Inputs a value into the computer.
<code>x=input('')</code>	Inputs a value and stores it into the variable x.
<code>x=int(input(''))</code>	Inputs a value into x, whilst also making it into an integer.
<code>print(str(x))</code>	Prints the variable x, but converts it into a string first.
<code>if name == "Fred":</code>	Decides whether the variable 'name' has a value which is equal to 'Fred'.
<code>else:</code>	The other option if the conditions for an if statement are not met (eg. name = 'Bob' when it should be Fred)
<code>elif name == "Tim"</code>	elif (short for else if) is for when the first if condition is not met, but you want to specify another option.
<code>#</code>	# is used to make comments in code – any line which starts with a # will be ignored when the program runs.

### Testing your program

**Valid** - the most obvious or common data that should work  
**Invalid** - data that should definitely fail  
**Erroneous** - data that is the wrong data type

**Example** – creating a number-guessing game, you will have some code that asks the user to choose a number in a specific range, eg "Choose a number between 1 and 10". To test this unit, you could try a whole range of inputs to see what happens:

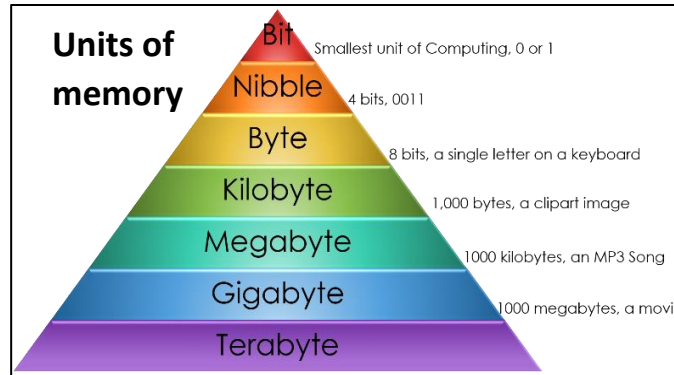
### Helpful websites:

[www.computerscience.gcse.guru](http://www.computerscience.gcse.guru)  
 w3schools website  
 learnpython.org  
<https://www.programiz.com/python-programming>

# Year 9 Computer Science – Autumn Term

## Data Representation

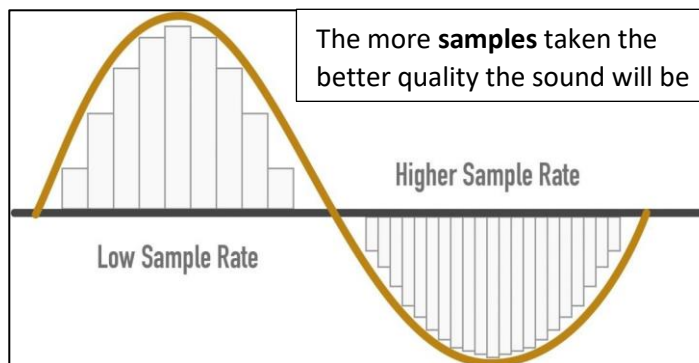
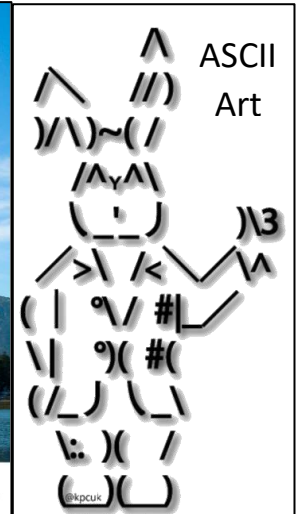
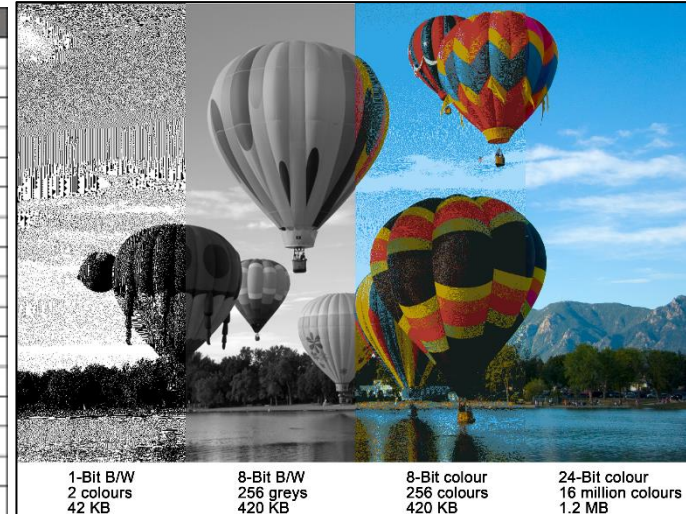
<b>Character</b>	Any single letter, number, space, punctuation mark, or symbol that can be typed on a computer.
<b>Bit</b>	Binary digit – 1 or 0
<b>Nibble</b>	4 binary digits e.g 0110 or 0001
<b>Byte</b>	8 binary digits, e.g. 01101100
<b>Binary</b>	Language used by computers to store and process data – base 2
<b>Denary/Decimal</b>	Number system used by humans 0-9 – base 10
<b>Hexadecimal</b>	Shortened version of binary. Number system using 16 characters – 0-9 and A-F – base 16
<b>Character Set</b>	Defined list of characters recognised by the computer
<b>ASCII</b>	Character set using 7 bits – covers the most common used English letters, numbers and symbols. Extended ASCII uses 8 bits.
<b>Unicode</b>	Character set using up to 32 bits – represents all possible characters across every language.
<b>Bitmap</b>	An image made up from a series of coloured dots (pixels)
<b>Vector</b>	An image made up of lines and shapes
<b>Resolution</b>	The number of pixels within a fixed area.
<b>Colour depth</b>	The number of bits used for each pixel – more bits = more colours
<b>Sample rate/frequency</b>	The number of samples taken per second.
<b>Sample size</b>	The amount of bits that are available for each sample.



### Converting binary to denary

<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
1	0	0	1	1	0	1	1
<hr style="border: 1px solid black;"/>							
<b>128 + 0 + 0 + 16 + 8 + 0 + 2 + 1</b>							
<b>= 155</b>							

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



### Helpful websites:

Youtube – Binary conversion

BBC Bitesize

<https://www.computerscience.gcse.guru>

<https://games.penjee.com/binary-bonanza/>