**computer science**

**bit** – smallest piece of info, 1 or 0

**file**

**computing**

**syntax error** – mistake in structure/syntax  
**white space**

**algorithm** – instructions for problem

**pseudocode** – “human speak” code

**source code** – script we write (**object code** is then what is compiled for the computer to read)

**keyword** – special words in Python

**variable** – see object reference

**assignment** – the = sign

**expression** – line of code that evaluates to a single value (arithmetic, logical, etc)

**operator** - + - / % etc

**concatenation** – adding together

**object** – chunk of memory  
**object reference** – pointer to that chunk

**delimiters** – brackets, commas, colons…  
**identifier** – the var name

**data type**  
**class** – what defines a data type (what it can do, make, etc.)  
**bool (Python type)** – True/False  
**boolean variable** – has True/False value  
**boolean expression** – evaluates to bool  
**dictionary (Python type)** – {}, key and value, unorderedm mutable,   
**float (Python type)** - decimal  
**floating-point number** – holds float  
**set (Python type)** – {}, unordered, unindexed

**list (Python type)** – [], ordered, mutable

**int (Python type)** – no decimal  
**str (Python type)** – chars

**immutable** - unchangeable

**function** – def, chunk of code

**formal parameter** - in def

**parameter –** see formal parameter

**actual parameter** - in call  
**argument** – see actual parameter  
**parameter passing** – handing off args to function call  
**positional parameter** – just list, so must be in right order

**keyword parameter** - formally stated ( x = …, y = …) in call, any order  
**main function** – calls other functions **method** – built in  
**module** – loaded in code

**scope** – the reach of the vars  
**value-returning function**

**statement** – the instruction in code **if-else**

**for loop**

**while loop**