

## 15110 PYTHON REFERENCE SHEET

Arithmetic Operations:	**	*	/	//	%	+	-
Relational Operations:	==	!=	<	<=	>	>=	
Logical Operations:	and	or	not				

Variable Names: All variable names must start with a letter (lowercase recommended). The remainder of the variable name (if any) can consist of any combination of uppercase letters, lowercase letters, digits and underscores (\_). Variables are case sensitive.

Assignment Statement: `variable = expression`

Defining a function: `def functionname ( parameterlist ) :`  
`function_body`

A *parameterlist* may be empty or may include one or more variables representing data required for the function, separated by commas.

Calling a function: `functionname ( argumentlist )`

An *argumentlist* may be empty or may include one or more expressions representing data required for the function to use, separated by commas.

Importing module: `import modulename`

Using module: `modulename .functionname ( argumentlist )`

<code>print(data)</code>	prints data to screen and moves cursor to next line
<code>print(data, end=" ")</code>	prints data to screen and keeps cursor on same line
<code>print()</code>	moves cursor to next line
<code>return(data)</code>	returns data to instruction that called this function
<code>for v in range(x,y,z):</code> <code>loop_body</code>	loops for $v = x$ through $y-1$ , inclusive in steps of $z$ ( $y$ is optional, default 0. $z$ is optional, default 1.)
<code>while condition:</code> <code>loop_body</code>	loops while <i>condition</i> is True
<code>if condition1:</code> <code>instruction1_set</code>	executes <i>instruction1</i> set once if <i>condition1</i> is True
<code>elif condition2:</code> <code>instruction2_set</code>	otherwise executes <i>instruction2</i> set once if <i>condition2</i> is True. This part is optional, can be repeated.
<code>else:</code> <code>instruction3_set</code>	otherwise executes <i>instruction3</i> set once if all previous conditions tested as False. Optional.

Lists:	<code>listname = []</code>	An empty list.
	<code>listname = [ item<sub>0</sub>, item<sub>1</sub>, ..., item<sub>n-1</sub> ]</code>	A list of $n$ items, $n \geq 1$ .
	<code>listname[i]</code>	Evaluates to the $i^{\text{th}}$ element of the list

<code>len(listname)</code>	returns the number of items in the list
<code>item in listname</code>	returns True if the item is in the list, False otherwise.
<code>listname[i:j]</code>	returns a sublist of list from index $i$ to $j-1$
<code>listname = [ item ] * n</code>	creates a list with $n$ copies of the item
<code>listname.append(item)</code>	appends item to end of the list
<code>listname.remove(item)</code>	removes the first occurrence of the item in the list

`for item in listname:` performs instructions once for each item in list, no index is available  
`loop_body` (*item* can be referenced in loop body)