| **Keyword/symbol** | **Meaning/Explanation** |
| --- | --- |
| print() | The print function. Used to output text to the console. |
| ' | Single quote. Used to encapsulate a single line string. |
| " | Double quote. Used to encapsulate a single line string. |
| ''' | Triple quote. Used to encapsulate a multi-line string. |
| # | Hash/Octothorpe/Pound sign. Used to denote a comment. |
| % | Modulo sign / formatter. Used as modulo in mathematical operations (returns remainder after division).  Also used as a formatter when using format strings (only used in python 2 lessons). |
| \* | Asterisk. Used for multiplication during mathematical operations.  Also used when using \*args in functions where you are not sure how many arguments you are passing to the function. Specifically, the \* allows you to accept an arbitrary number of arguments. |
| - | Hyphen. Used as a minus sign when performing subtraction in mathematical operations. |
| / | Forward slash. Used as a division sign when performing mathematical operations. |
| + | Plus sign. Used as an addition sign during mathematical operations.  Also used in loops as an iterator.  Also used in shorthand such as += meaning 'equal to plus' e.g. a += b which can be written a = a+b |
| = | Equals sign. Used to create variables or to set one value to equal another. can be paried with other operators like +. |
| > | Greater than sign. Returns True if value on the left of the sign is greater than that on the right, else returns False. |
| < | Less than sign. Returns True is value on the left is less than that on the right, else returns False. |
| >= | Greater than or equal to sign. Returns True if value on the left of the sign is greater than or equal to that on the right, else returns False. |
| <= | Less than or equal to sign. Returns True is value on the left is less than or equal to that on the right, else returns False. |
| === | Comparison operator. Returns True if the value on the left is exactly the same as the value on the right. Else returns False.  Can be used to compare strings, boolean operators, integers etc. Concerned with 'equality' and not 'sameness'. Concomitantly, distinct from is which checks if the values are literally refering to the same instance of a value. |
| f"" | f string. Python 3.6 syntax for handing format strings. Allows the inclusion of variables via the {} modifiers. |
| {} | Curly brackets (?). Used in positional formatting when using format strings. |
| .format() | Python 3 replacement for the % used in format strings. Has form '{} {}'.format('one', 'two'). |
| end=' ' | Python 3 syntax for ending a line with a space rather than a linebreak. Python 2 would use , to achieve the same effect. |
| join() | The method join() returns a string in which the elements passed to the string have been joined by a str operator. Example [**here**](https://www.tutorialspoint.com/python/string_join.htm). |
| input() | Python 3 version of raw\_input(). Receives an input from user via the keyboard. Not to be confused with input()available in Python 2 which tries to eval the input. |
| from | Part of an import statement used to import external modules into a script. To fully explain this keyword read [**here**](https://docs.python.org/3/reference/simple_stmts.html#import), [**here**](https://docs.python.org/3/reference/import.html#importsystem)and [**here**](http://stackoverflow.com/a/21547572) |
| import | Similar to from but is the 'meat' of the import statement. To fully explain this keyword read [**here**](https://docs.python.org/3/reference/simple_stmts.html#import), [**here**](https://docs.python.org/3/reference/import.html#importsystem) and [**here**](http://stackoverflow.com/a/21547572). |
| argv | Accepts a list of command line arguments passed to it when running a script. Must be imported from the sys module. Example [**here**](https://www.tutorialspoint.com/python/python_command_line_arguments.htm) but you can also import using the form from sys import argv which will allow you to simply call the keyword argv and not the full path. Read more on importand from above for more detail on importing modules. |
| open() | Opens a file and returns the corresponding [**file object**](https://docs.python.org/3/glossary.html#term-file-object). For further details and arguments see [**here**](https://docs.python.org/3/library/functions.html#open). |
| read() | Reads the contents of a [**file object**](https://docs.python.org/3/glossary.html#term-file-object). See [**here**](https://docs.python.org/3/tutorial/inputoutput.html#methods-of-file-objects) for more info. |
| write() | Writes the contents of a string to a file. See [**here**](https://www.tutorialspoint.com/python/file_write.htm) and [**here**](https://docs.python.org/3/tutorial/inputoutput.html#methods-of-file-objects) for more info (scroll down slightly to see examples). |
| close() | Closes an open file object. Important to call this method and not leave it to the interpreter or you could risk unusual behaviour (though check the interpreter doesn't do it for you automatically or that too could ause an derror - no one said this was easy!) More info [**here**](https://www.tutorialspoint.com/python/file_close.htm). |
| os.path | Module containing useful functions regarding pathnames. For more info see [**here**](https://docs.python.org/3/library/os.path.html). |
| exists | Method/Function that is part of the os.path module which returns True if path given refers to an existing path or an open file descriptor. |
| ; | Semi-colon. Allows for '[**compound statements**](https://docs.python.org/3/reference/compound_stmts.html)' but that explanation there is a bit crazy so [**this stackoverflow**](http://stackoverflow.com/questions/8236380/why-is-semicolon-allowed-in-this-python-snippet) post should give greater insight. Basically allows you to collapse statements onto one line, not that that's the best thing to do most of the time. |
| def | Keyword meaning 'function definition' and used to create user defined function object. Put simply, this allows you to create your own functions (note functions are sometimes called methods). More info [**here**](https://docs.python.org/3/reference/compound_stmts.html#function-definitions). |
| \*args | Allows a function to accept an arbitrary argument list. Used for when you don't know how many arguments are going to be passed to a function. More info [**here**](https://docs.python.org/3/tutorial/controlflow.html#arbitrary-argument-lists) and [**here**](http://stackoverflow.com/a/3394898). |
| seek() | The method seek() sets the file's current position at the offset. Default position is 0 so starts at the start of the file. More info [**here**](https://docs.python.org/3/tutorial/inputoutput.html#methods-of-file-objects) (scroll down) and [**here**](https://www.tutorialspoint.com/python/file_seek.htm). |
| readline() | Reads a single line from a file. A newline character is left at the end of the string so will automatically read the next line until the end of the file each time it's called. More info [**here**](https://docs.python.org/3/tutorial/inputoutput.html#methods-of-file-objects)and [**here**](https://www.tutorialspoint.com/python/file_readline.htm). |

This further table details the escape sequences we encountered in the course so far.

| **Escape example** | **What it does/escapes** |
| --- | --- |
| \\ | Backslash () |
| \' | Single-quote (') |
| \" | Double-quote (") |
| \a | ASCII bell (BEL) |
| \b | ASCII backspace (BS) |
| \f | ASCII formfeed (FF) |
| \n | ASCII linefeed (LF) |
| \N{name} | Character named name in the Unicode database (Unicode only) |
| \r | Carriage Return (CR) |
| \t | Horizontal Tab (TAB) |
| \uxxxx | Character with 16-bit hex value xxxx (Unicode only) |
| \Uxxxxxxxx | Character with 32-bit hex value xxxxxxxx (Unicode only) |
| \v | ASCII vertical tab (VT) |
| \ooo | Character with octal value ooo |
| \xhh | Character with hex value hh |