Comp Sci Curriculum

1. Syntax and Console Output
   1. Variable types
   2. Math Operators

+ - \* / \*\* %

* 1. Comments (#)
  2. Examples
     1. Hello world
     2. Print variables (round function)
     3. Print mathematical expressions

1. Strings
   1. List of Chars

"""

The string "PYTHON" has six characters,

numbered 0 to 5, as shown below:

+---+---+---+---+---+---+

| P | Y | T | H | O | N |

+---+---+---+---+---+---+

0 1 2 3 4 5

So if you wanted "Y", you could just type

"PYTHON"[1] (always start counting from 0!)

"""

fifth\_letter = "MONTY"[4]

print fifth\_letter

* 1. String Functions
     1. len
     2. lower
     3. upper
     4. str
  2. printf/concatenation of strings
  3. Examples
     1. Use all functions
     2. Print single characters of a string

1. Conditionals and Control Flow
   1. If/elif/else
   2. Whitespace/scope
      1. 4 spaces or a tab
   3. and/or/not operators
   4. Examples
      1. If ladders vs elif ladders
      2. Practice with Boolean logic
2. Loops
   1. While

while [condition]:

* 1. For

for i in range(0,100):

* 1. For each loops (with Lists)

1. Console Input
   1. Raw\_input
   2. Examples
      1. Hello world with name
2. Functions
   1. def name():
   2. return vs. print functions
   3. Examples
      1. Multiply/divide/square/power
      2. Return 3rd char of string
3. File I/O
   1. 4 modes
      1. read only (“r”)
      2. write only (“w”)
      3. read and write (“r+”)
      4. append (“a”)
4. Arduino
   1. How to use objects/classes
   2. Serial data
   3. Bluetooth for wireless communication

|  |  |
| --- | --- |
| **Command** | **Result** |
| cd [directory] | Change directory |
| ls | List everything in current directory |
| python my\_python\_file.py | Runs Python file |

***Operator Precedence***

***Basic Python Functions***

***Terminal Commands***

***Basic Syntax Structures***

|  |
| --- |
| **Assignment Statement**  *var = exp* |
| **Console Input/Output**  *var =* raw\_input( [*prompt*] )  print *exp* |
| **If statements**  if (*boolean\_exp*):  *stmt* …  [elif (*boolean\_exp*):  *stmt* …] …  [else:  *stmt* …] |
| **Loops**  while(*boolean\_exp*):  *stmt* …  for *item* in *list*:  *stmt …* |
| **Function Definition**  def *function\_name*( *parameters* ):  *stmt* … |
| **Function Call**  *function\_name*( *arguments* ) |

|  |  |
| --- | --- |
| *x*[*index*] | List/String Indexing |
| \*\* | Exponentiation |
| \*, /, % | Multiply, divide, mod(remainder) |
| +, - | Add, subtract |
| >, <, <=, >=, !=, == | Comparison |
| not, and, or | Boolean operators |

|  |  |
| --- | --- |
| **Function** | **Returns** |
| len(s) | Number of items in sequence s |
| range(x, y, step) | A list of ints from x to y with a step-size of step |
| round(x,n) | Float x rounded to n places |
| str(obj) | Str representation of obj |

|  |  |
| --- | --- |
| ***my\_arduino*.*method()*** | **Result** |
| analogWrite( *pin*, *value* ) | Gives the pin a value between 0 and 255 |
| analogRead( *pin* ) | Returns a pin’s value |
| digitalWrite( *pin*, *bool\_value* ) | Gives the pin a Boolean value (True/False) |
| digitalRead( *pin* ) | Returns a pin’s Boolean value |
| pinMode( *pin*,[*INPUT*/*OUTPUT*]) | Specifies a pin for input or output |
| delay( *milliseconds* ) | Pauses execution for a number of milliseconds |
| tone(*pin*, *frequency*, *duration*) | Gives a piezo buzzer a tone at a specified pin and an optional duration |

***String Functions***

|  |  |
| --- | --- |
| **S.method()** | **Returns (type str)** |
| lower() | Lower case copy of S |
| upper() | Upper case copy of S |

|  |  |
| --- | --- |
| **File I/O: Code** | **Result** |
| *my\_file* = open(“*file.txt*”,”[*w*/*r*/*r+*/*a*]”) | Open a file |
| *my\_file*.write(*str\_contents*) | Write string contents to a file |

|  |  |
| --- | --- |
| ***my\_util.method()*** | **Result** |
| map( *value*, *fromLow*, *fromHigh*, *toLow*, *toHigh* ) | Returns a re-mapped number of a value from one range to another |
| constrain( *value*, *low­\_bound*, *high\_bound* ) | Returns a constrained number of a value to be within a range |