**Python Reference Sheet**

**By: Jaffer Razavi**

**Turtle**

import turtle

<variable> = turtle.Turtle()

<variable>.shape(“<turtle, arrow, square, circle, triangle, or classic>”)

<variable>.right(<degrees>)

<variable>.left(<degrees >)

<variable>.forward(<distance>)

<variable>.back(<distance>)

<variable>.pencolor(<color of line>)

<variable>.fillcolor(<color of turtle>)

<variable>.begin\_fill()

<variable >.end\_fill()

<variable>.penup()

<variable>.pendown()

<variable>.goto(<x,y>)

<variable>.hideturtle()

<variable>.showturtle()

<variable>.speed(<speed>)

<variable>.write(“<string>”, font = (“<font>”,<size>))

<variable>.reset()

<variable>.width(<width>)

<variable>.circle(<radius>,<extent>,<steps>)

**Math:**

import math

math.sqrt(<number>)

math.factorial(<number>)

math.pi

**Random:**

import random

<variable> = random.randint(<start value>,<stop value>)

**Decision Structures:**

if (<condition>):

[ Code Chunk]

elif <condition>:

[Code Chunk]

else:

[Code Chunk]

(can have as many elifs as needed)

**Loops:**

**Counted Loops:**

for <variable> in range (<start value >, <stop value>, <step value>):

[code chunk]

**Conditional Loops:**

while <condition>:

[code chunk]

**Input:**

|  |  |
| --- | --- |
| int(input(“<string>”) | Gets input as an integer |
| float(input(“<string>”) | Gets input as a float |
| input(“<string>”) | Gets input as a string |

**Output:**

|  |  |
| --- | --- |
| print (“<output>”) | Prints an output |
| print (“<output>”,end= ‘ ’) | Prints a blank space and waits on that line for the next output |
| print (“<output> \n “) | New line |
| print (“<output>”,sep=‘ ’) | Separates string literals with anything inside the single quotation marks |
| ‘\t’ | Adds a tab space |
| print (format(<number>, ‘<field size>,.<number of decimals>’)) | Separates the number by an amount of spaces and adds/rounds a decimal off to a specific point |
| print (‘{<field number>:<spaces>}’.format(<output>) | Separates each field by spaces using the number after the colon in each curly bracket |
| print(round(<decimal>,<decimal spaces>)) | Rounds the decimal |
| print(<variable> [<start>:<stop>]) | Prints the letters in a string from the start to stop value. The first letter in the string has the value of 0 |

**Other Operators:**

|  |  |
| --- | --- |
| == | Equal to |
| < | Less than |
| <= | Less than or equal to |
| > | Greater than |
| >= | Greater than or equal to |
| != | Not equal to |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus/Remainder |
| \*\* | Exponentiation |
| // | Floor Division |
| and  if(<condition>) and (<condition>): | Will execute if structure only if all of the conditions are met. |
| or  if(<condition>) or (<condition>): | Will execute if structure if one of the conditions is met |
| if “<string>” in <variable> | Used to identify something in an input from a user |
| replace(<old>,<new>,<max>) | Replaces a word in a string (old) with another string (new) a set number of times (max). |
| <variable>.lower() | Makes a string all lowercase |
| <variable>.upper() | Makes a string all uppercase |