Python Notes – Week of September 10th

9/10/18

**IDLE** - **I**nteractive **D**eve**l**opment **E**nvironment

The opening screen is called the Python Shell where we can enter one command at a time.

The shell waties for a command and then executes it, but once that line is used, it’s gone.

Mathematical symbols are the same as your calculators + - \* /

Number types

Integers - whole numbers, positive or negative

Decimals - in computers, we call these floating point numbers

In addition to math, the shell can return true or false based on a condition, like 2 < 6 gives True

Variable - names given to data stored in our programs

Unlike math, the names don’t have to be a single letter

When storing values into variables, store them right to left

YES age = 42

No 42 = age

Rules for Variable Names

1. Cannot be a keyword (they’ll turn a different color, purple, orange, etc…)
2. Can’t start with a number, but it may contain numbers elsewhere in the name

1num is BAD num1 is OKAY p1score is OKAY

1. Can’t contain special characters like spaces, $, periods, exclamation marks, etc...

The only exception is you may use the underscore. player\_score

Please note that Python is case sensitive, so Num and num are different variables!

A typical programmer variable naming style is called camel notation.

playerScore classGrade versus

Player\_score class\_grade

9/11/18

**“Input Exploration.py”**

To make comments, we use the # sign - computer ignores these, only for you!

print() command allows you to display text on the screen

Anything in quotes inside the print will display on the screen

print(“This prints on the screen!”)

input() command allows you to get information from the user

Set this equal to a variable to store the info into a variable

name = input()

When printing, to join two pieces of data together, we use the plus sign or comma.

We call this “concatenation”

print(“Example Text Part 1 ”, variable, “ Example Text Part 2”)

\*Note: Commas add space

print(“Example Text Part 1 ” + variable + “ Example Text Part 2”)

Data Types in Python

Integers whole numbers, positive and negative

Strings alphanumeric - both numbers and characters

“5” + “7” = “57”, not 12!

int() converts strings to integers so that you can do math with them

str() converts numbers back to strings so that you can print them

(see the program for the example)

9/13/18

**“Math Commands.py”**

In order to use most of the higher-level math commands, we need to start out programs with *from math import \**

abs(#) absolute value |-5| becomes 5, |5| says as positive 5

ceil(#) ceiling - “round” the decimal up 2.7 becomes 3, 2.1 becomes 3

floor(#) floor- “round” the decimal down 2.7 becomes 2, 2.1 becomes 2

factorial(#) factorial 5! = 1 \* 2 \* 3 \* 4 \* 5

gcd(#,#) greatest common divisor of two numbers

exp(#) e ^ #

log(#, base) logarithm of # with whatever base log ₓ (y) = z, x ᶻ =y, x = base

sqrt(#) √x

pow(base, exponent) or \*\* base ^ exponent

sort(#)

sin(#),cos(#),tan(#),asin(#),acos(#),atan(#)

pi 3.14159...

e 2.71828…

"%.2f" % rounds to hundredth digit (2f = hundredth, 1f = tenth, 3f = thousandth, etc.)

round(#, decimal place) same as "%.2f" %

float(#) represent real numbers and are written with a decimal point dividing the integer and fractional parts

chr(#) used to print characters not on keyboard using character number

**Miscellaneous ( ͡° ͜ʖ ͡°)**

How do you print quotes in your strings?

*The title of the movie is “Imitation Game”*

Use the special character combination \”

print(“the title of the movie is\”Imitation Game\””)

To print \ you put \\

\ indicates special character

Other special characters include \t for tabs and \n new blank lines

Repeating:

for n in range(x):

x = # of times repeated

while True: repeats forever

**Midpoint.py**

Example of printing variables inside print text

Python Notes – Week of September 17th

9/19/18

Pseudorandom numbers - a computer cannot generate true random numbers, as it is a machine, not a natural thing.

At the top of your code - from random import \*

Since our numbers are generated by a formula, you need what is called a “seed value” - a value to start on, typically the time.

random() gives a decimal from 0 up to but not including 1 [0,1) = 0 ≤ x < 1

int() “truncates” the decimal - chops it off, does not round!

int(3.4) = 3 int(3.9) = 3 int(3.999999999) = 3

**FORMULA FOR A RANDOM NUMBER**

int(random() \* range) + lowest number

**Example:** roll a six sided die

roll = int(random() \* 6) + 1

**Example:** Pick a random from 93-100

int grade = int(random() \* 8) + 93

range = (highest - lowest) + 1

Lists and Random String Choice:

names = ["Chuck", "Barbra", "Obama", "Won", "Dwight", "Isaac", "Harry"]

name = choice(names)

Python Notes – Week of September 24th

9/25/18 - 9/26/18

**If Statements** - command used to make decisions based on a true/false condition

The true/false conditions are formally called “Booleans”, named after George Boole (mathematician who invented algebraic logic in the 1800s)

Six symbols of comparison

< less than

> greater than

== equal to

<= less than/equal to

>= greater than/equal to

!= not equal to

% “mod”, if divisible by

if \_\_\_\_\_\_\_\_\_\_:

Anything tabbed in below the if

triggers when the condition is true

If the condition is false, it’s skipped

elif \_\_\_\_\_\_\_\_\_\_:

It’s an if statement that activates if all previous statements are false

else:

It’s a statement that activates if all previous statements are false

if \_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_:

and joins multiple conditions and forces the statement to run only when all conditions are true

if \_\_\_\_\_\_\_\_ or \_\_\_\_\_\_\_\_:

or allows for the if statement to run if only 1 of the conditions is true

if \_\_\_\_\_\_\_\_ % \_\_\_\_\_\_\_\_ == 0:

if something is divisible by something else with a remainder of 0

Python Notes – Week of October 22nd

10-22-18

**Functions in Python**

To create a function, we use the def command - which stands for define

Then, we give the function a name - it can be a single letter, but like variables, it

can be more! The same rules apply.

In math, we can create a simple function:

f(x) = x ^ 2 + 1

The name of that function is “f”. It has one input (parameter) called x. Every input

has exactly one output. The output is “returned” to the program.

In code, we write the following:

def f(x):

return x \*\* 2 + 1

To use that function, we **call** the function.

print(f(4))

Functions do not necessarily need parameters. Additionally, functions may have

more than one parameter! Functions do not necessarily need return statements.

You don’t print() to print function graphics. You simply state your function.

def triangle():

forward(100)

right(120)

forward(100)

right(120)

forward(100)

triangle()

**Python Notes - Loops**

10/26/18

**When to repeat something?**

Definite repetition - you know ahead of time the starting and ending values.

Example: roll a six sided die **ten** times

Indefinite repetition - you know the loop will stop eventually, but not exactly when.

Example: roll a six sided die **until** you get a 1.

While Loop - used for indefinite loops

while <condition>:

<code to be repeated>

<code to be repeated>

<code to be repeated>

This loop keeps going “while” the condition remains **true**.

(while/until are opposites)

For Loop - used for definite loops

for <loop control variable> in range(...):

range(#) from 0 to (# - 1)

range(#, #) from # to (# - 1)

range(#, #, #) from # to (# - 1) by #

**Python Notes - Lists and Arrays**

10/30/18

A list is a collection of items with an order

In Python, the square brackets [ ] indicate a list

Items in lists are separated by commas

One variable = one item

villain = "Magneto"

A list can hold multiple items (The backslash lets you continue a list on the next line)

xmen = ["Cyclops", "Wolverine", "Beast", "Storm",\

"Iceman", "Colossus"]

To print the entire list…

print(xmen)

To print one item from the list

print(xmen[0])

To replace an item in the list, treat it like a variable

xmen[4] = "Jean Grey"

To add an item to the end of the list, use append()

xmen.append("Deadpool")

To insert an item into the list at a certain position

xmen.insert(1, "Professor X")

To reverse a list

xmen.reverse()

To remove something by the position

del xmen[0]

“0” on list is 1st name

To remove something by name

xmen.remove("Beast")

To sort from least to greatest

xmen.sort()

To sort from greatest to least

xmen.sort(reverse = True)

To get the length of a list

len(xmen)

To get a certain section of a list

xmen[0 : 2]

Would print Cyclops and Wolverine, not include Beast

: = -

[0 : 2] = 1-3

To get the sum of a value list

sum(xmen)

To get the smallest value on a number list

min(xmen)

To get the largest value on a number list

max(xmen)

**Python Notes - Strings and Substrings**

12/12/18

String - any data which is alphanumeric - characters, words, sentences, symbols, and potentially numbers

Declare a string - set it equal to a "string literal" in quotes

str1 = "Hello world!"

Find the length of the string

len(str1)

Like lists, strings begin with character 0

So, str1[0] would pring "H"

for n in range(len(str1)):

print(str1[n])

Find the location of a character or substring in a string

str1.index("w") 6 (w is in location 6)

str1.index("o") 4 - only gives you the first match

If you try to find the index of something not in the string it will give you a crashing error

str1.index("ell") 1 - location of the e in Hello

Count the number of characters/substrings in a string

str1.count("o")

Multiply a string by a number to copy that many times

"Hello" \* 5

password = "eagles123"

"\*" \* len(password)

Concatenate - add two or more strings together (use +)

word = "exciting"

ex = "!!!!!!!!!!\t SO EXCITING"

print(word.upper() + ex.lower())

Slicing - get part of a word

To get multiple characters from a string use [# : #] where the first number is INCLUSIVE (includes it) and the second number is EXCLUSIVE (it doesn't)

word[2:6]) citi

word[2:6:2] ct (the third 2 skips every 2 letters)

word[::-1]) print the word in reverse

p = "racecar"

if p == p[::-1]:

print("Palindrome")

else:

print("Not a palindrome")