**todo**:

make all code text italic

**How this works:**

-This should only relate to Python 3.X

-re-organizing should be done later in categories

-Section, or how to titles are font size 11, bold

-descriptions are font size 10

-code is font size 9

-examples titles(ex1:) are font size 10, underlined

- “!Check!” are font size 11, underlined and meaning this should be verified/finished/assured.

**Types:** !Check!

int: holds signed integers of non-limited length.

2, 100, -754, 0x46, -0x352

long: long numbers

0xDEFAAEl, -0x19323L, -2849*939L, 0122L*

float- holds floating precision numbers and it’s accurate up to 15 decimal places.

5.7, 32.3+e18, -90.2, 70.2-E12

complex- holds complex numbers.

9.322e-36j, .876j, 3e+26J

string- bunch of text.

“hi”, ‘world’ - single or double quotes

ex:

*print(‘hi ’+’world’)*

**Variables:**

# variables give python data A repeatable name

# variable names in Python can only contain. letters, numbers or underscores. The first letter cannot be a number.

#simplest variables:

var1 = ‘data’

var2 = 0

# can define multiple variables in one line:

var1,var2,var3 = ‘hi’,3,4.2

# variables can change(nullable). ex:

x,y=0,1

x=y

x is now 1 opposed to initial value 0.

ex1:

x = ‘hi’

y = ’tj’

print(x,y)

ex2:

num1,num2=2,4

answer = num1+num2

print(answer)

**Operators and signs:**

= - assignment: holds same value

+ - add, or string concatenate(“hi ” + “world”)

// - quotient: (20 // 6 = 3) 6 goes into 20 three times

**String character extentions:**

\ - escape character: for separating quotations marks from string statement quotation marks.

ex.1:

print(“He says: \“TJ’s phone\””)

**Comments to add self written info for developer reference:**

ex1:

# this is a comment line

print(“some code”)

# Use that number/hash symbol to add info/

# documentation. Starting A line with this does not run # any code in that line.

print(“more code”)

**Wait for some time, then continue code:**

ex1:

from time import sleep  
sleep(0.1) # Time in seconds

ex2:

import time  
time.sleep(5) # Int or Float Value.

ex3:

import time   
while True:  
 print("This prints once a minute.")  
 time.sleep(60) # Delay for 1 minute (60 seconds).

ex4:

# This gets time to wait using localtimes’ return value.

# Useful if you want to wait for any localtime.

time.sleep(time.localtime(time.time())[5])!Check!

**Notes:**

* Dividing Integers or float numbers(5/2), always produces float decimal numbers.
* A