PyhtonNote

round(0.1 **+** 0.2, 4)

0.3

t = 1.0

print(type(t))

Exponents are done with '\*\*' or the pow keyword

5\*\*2

25

Division treats things as floats in python3. '//' is used to 'floor' the division.

dist2 = 0.5\*accel\*pow(t,2)

print(6 // 5) # integer division returns the floor

print ("equa hello l (==): %s" % (x == y))

print ('not equal (!=): %s' % (x != y))

print ('greater (>): %s' % (x > y))

print ('less (<): %s' % (x < y))

# Boolean operators

# Logic operators: and, or, not

print (True or False)

print (False or True)

print (False or False)

first = True

second = True

print (not(first and second))

# Conditional Execution

if (x > 0):

print ('x is positive')

if x < 0:

pass # need to handle negative values

choice = 2

if choice == 1:

print ('opt 1')

elif choice == 2:

print ('opt 2')

elif choice == '3':

print ('opt 3')

else: # is optional but must be at the end

print ('no options')

if and else

Chained if statements with elif

import math

# Define a function -- note the way you are meant to comment a function:

def area(radius):

"""

This function computes the area of a circle when given the radius

"""

temp = math.pi \* radius\*\*2

return temp

# Call the area function

print (area(12))

# Recursion

def recurse():

"""

this is a recursive function.

"""

print ('hello')

recurse()

recurse()

## Updating values

There is no ++ or -- operator

num = 5

num**++**

num = 5

num += 5

print (num)//10

# Working with strings!

NOTE: backslashes (\) start special (escape) characters: \n = newline (\r = return) \t = tab \\ = put in a backslash

print ('hello, world!\n\n')

print ('hello, world!')

hello, world!

hello, world!

print ('hello\\world')

hello\world

first = 'Bob'

second = 'Hello'

print (first + second)

print (first + " says " + second)

print ("%s says %s" % (first, second))

Bob says Hello

print (first\*3)

BobBobBob

# Loops!

Note that the spaces (4 of them) determine what is "in" the loop. Also, a colon is needed to end the definition of the loop

n = 10

while n > 0:

print (n)

n = n - 1

print ("Blastoff!")

for i in range(5):

print (i)

## More working with strings!

fruit = 'banana'

letter = fruit[0]

print(letter)

index = 0

while index < len(fruit):

char = fruit[index]

print (char)

index = index + 1

words = 'Monty Python'

print(words[0:6])

You can also work backwards from the end of the string

print(words[-1])

print(words[-6])

print(words[-6:])

print(words[:-6])

n

P

Python

Monty

## String functions

print(words.upper())

MONTY PYTHON

## The 'in' operator

This lets you know if a string is found within another string

search = 'o'

if search in words:

print ('%s was found in %s' % (search, words))

o was found in Monty Python

fruit = 'banana'

for letter in fruit:

if letter in words:

print (letter)

n

n

## String comparison

String comparison defaults to alphabetical

word = 'toooooooo'

if word == 'banana':

print ('All right, bananas')

if word < 'banana':

print (word + ' comes before banana')

if word > 'banana':

print (word + ' comes after banana')

toooooooo comes after banana

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*