Python for Programmers

## Welcome

### Welcome

### Python 2 vs 3

### Mac Setup

### Windows setup

## Numbers

### Intro

### Variables and ints

Variable do not need to be declared. Like matlab juts introduce them

### Floats

### Math

### Advanced Math

import math  # need for pi

5\*\*3  # \*\* is exponential like in Fortran

5 // 3 # integer division

5 % 3   # remainder

round(math.pi,2)

int(math.pi)

# \_ gives most recent result

2\*\_

int(pi)

float(8)

### Conversion

### Outro

## Strings

# can use single or double quotes

'hello'

"hello"

# use 3 quotes of either type for multi-line strings

'''hello

world'''

"""hello

world"""

# can use strings as comments

# + is the concatination operator

print("hello"+" "+'world!')

# \* is used for repition of a string

print(9\*"9")

# printing variables

age = 30

print("My age is {}".format(age))

print("My age is {} and pi is {}".format(age, 3.1415))

"Hello".upper()

"Hello".lower()

"Hello world".title()

len("hello")

s="Hi my name is Bob".replace("Bob","Jim")

print(s)

s[0]

s[-1]

s[-17]   # gives same as s[0]

s[16]   # gives same as s[-1]

s[-18]    # gives an error

s[17]    # gives an error

# slices. s[n:m] goes from s[n] to s[m-1]  !

s[0:2]

s[6:10]

s[3:-4]

s[:10]  # same as 0:9

s[3:]   # same as 0:len

s[3:0]  # gives "" empty string

s[-1:]  # final character

## IF statements

# booleans

True

False

sunny = False

# use indentation instead of brackets!

if sunny:

    print("Sunny!")

    print("Yes Sunny!")

else:

    print("cloudy")

print("done")

# also have elif for convenience

x = -11

if x > 0:

    print("positive!")

elif x == 0:

    print("zero")

else:

    print("negative")

# python has ability to tell if between two nums!

y = 9

if 5 < y < 10:

    print("Tween")

#  and, or, not keywords

a=6.3

if a % 2 == 0 and 7 >= a >= 5:

    print("Even and between 7 and 5")

if not False:

    print("not")

# None is liek null in Java and Javascript

if not None:

    print("not")

## PROJECT 1 – GRADE CALC

pointspossible = 100

score = 84

percentage = score / pointspossible

studentname = "Bill"

print("Percentage is {}".format(percentage))

if 0 <= percentage < .60:

lettergrade="F"

elif .60 <= percentage < .70:

lettergrade="D"

elif .70 <= percentage < .80:

lettergrade="C"

elif .80 <= percentage < .90:

lettergrade="B"

else:

lettergrade="A"

print("{} {}".format(studentname, lettergrade))

## User Input

Capture a line of text

age=input("What is your age?: \n")

print("Your age is "+age+" years old!")

type of a variable

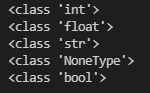
print( type(14) )

print( type(14.5) )

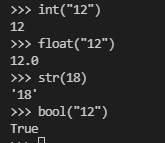
print( type(“14”) )

print( type(None) )

print( type(True) )



Type conversion is performed via int(), float(), str() etc



Exception handling

try:

    age = int(input("enter age:"))

    print("Valid age!")

except Exception:

    print("You need to provide a valid number")

## Functions

Simple function definition

def hello():

    print("Hello World")

hello()

function with paramters

def add(x1,x2):

    print(x1+x2)

add(3,4)

named parameters

def sub(x1,x2):

    print(x1-x2)

sub(4,3)

sub(x2=4,x1=3)

default parameters

def pname(name="nobody"):

    print(name)

pname("Ron")

pname()

**return value**

return causes end of function execution

def addtwo(num1, num2):

    return num1 + num2

print( addtwo(3,6) )

no return value means that the None object is returned

def hey():

    print("hey")

print( hey()   )

## Lists, Tuples, Sets

Type is **list**

games = ["Mario Bros 3", "Earthbound", "Pilotwings", "Mario Party"]

print(games)

print(type(games))

print(games[1])

games.insert(0, "Zelda")

print(games)

games[1] = 'Mario Bros 1'

print(games)

del(games[2])

print(games)

games = ["Zelda", "Mario Bros 3", "Earthbound", "Pilotwings", "Zelda", "Mario Party"]

# remove will remove the FIRST element that matches

games.remove("Zelda")

print(games)

print(len(games))

# tuples use parantheses.  both size and values are immutable!

tuple = ("Zelda", "Mario Bros 3", "Earthbound", "Pilotwings", "Zelda", "Mario Party")

print(tuple)

#to create an tuple of one element, you need to include a trailing comma

y = [ "hello" ]   # array

w = ( "hello" ) # value,, not a tuple

z = ( "hello", )  # tuple

shoes = ("Spizikes","Air Force 1","Curry 2","Melo 5")

# + operator does concatination for both arrays and tuples

# \* operator replicates

q = ["hello"] + 2\*[ "there" ]

def appendtotuple(thetuple, value):

    x = ( value, )

    t =  thetuple + x

    return t

print(appendtotuple(shoes,"loafers"))

# sets: use curly braces {}

myset= { "hello", "there "}

print(myset)

myset.discard("there")

myset.add("hi")

print(myset)

myset = set(games)

print(myset)

## Loops

## PROJECT 2: Hangman

## Dictionaries

## Classes

## Advanced Python

## PORJECT 3: Word Counter

## Random Twitter Follower

## BONUS SECTION