**Python**

**INDEX:**

**Page 2 Basic**

**Page 4 Files**

**Page 5 Functions and Classes**

**Page 6 Inheritance**

**Page 7 Overriding Methods**

**BASIC:**

**EXECUTE WITH Python nameProgram.py**

\*del (delete)

\*len() (length)

\*max()/min()

\*.append("") (add to the final list)

\*.count("")

\*.clear() (clear an array)

\*.keys (dictionary)

\*.values (distionary)

\*.update()

-tuplas --> tupla1=("Maths", 23, "Dogs") --> inmutables

name = imput("Enter your name")

conditions with ":" after the sentence without "(" ")"

for i in range(0,5):

\*str()

break

pass (jump the next sentences)

continue (jump to the next on a loop)

\*try:/except:

#comment

"""

Muti\_line\_comment

"""

Use \' on string

Use "\" in a contition to jump to other line and continue coding

(PE: if blue \

== \

red:

print("Hi!")

for i in range (0, 20, 2) (2 to 2 until 20)

def function(): (to define functions)

\*abs() (absolute valor of a number)

\*bool() (return TRUE or FALSE-->empty)

\*dir([])/dir("") (it gives you all predefined functions for array [] strings ""...)

\*help(array.count)

\*eval() (execute)

\*float() (transform a number into a float)

\*int() (transform a number into a integer)

\*.len() (length of array...)

\*list() (transform into a list)

**FILES:**

testFile = open("path.txt", accessMode("r","w", "a" (append), a+"(append and read)), ¿? )

testFile.read() (second time that you try to read it doesn't work)

testFile.write() --> return write len (replace content)

position = testFile.tell() (position on file)

position = testFile.seek(0, 0) (begining of file)

testFile.close()

import os

os.rename("Old.txt", "New.txt")

os.remove("Filename.txt")

newFile = open("Hola.txt", "w")

newFile.write(testFile.read()) (a copy)

**FUNCTIONS AND CLASSES:**

class Students:

def \_\_init\_\_(self, name, age, grade):

self.name = name

... (instance)

def displayName(self):

return ("Hi" + self.name)

Stu = Students("Bob",15,"17th")

Stu.displayName

\*hasattr(Stu, "age") --> TRUE (test if it has this atribute)

\*setattr(Stu, "family", 2)

\*getattr("Stu", family)-->2

\*delattr("Stu", family)

**INHERITANCE:**

class Parent:

counter = 10

def\_\_init\_\_(self):

print("Hi")

def setCounter(self, num):

Parent.counter = num

class Child(Parent):

def \_\_init\_\_(self):

print("Child")

def childFunc(self):

print("U.u")

c = Child()

c.counter

c.setCounter(10)

**OVERRIDING METHODS:**

class Parent:

def func(self):

print("2")

class Child(Parent):

def func(self):

print("1")

c = Child()

c.func() --> 1

RANDOM NUMBER GAME

import random

comGuess = random.randint(0, 100)

while True:

#Int to transform the string to int and compare int values

userGuess = int(input("Guess a number between 0-100: "))

if userGuess < 0:

print("Enter a number higher than 0")

elif userGuess > 100:

print("Enter a number lower than 100")

elif userGuess > comGuess:

print("Guess lower")

elif userGuess < comGuess:

print("Guess higher")

else:

print("Congrats, you've guessed the correct number")

break

RANDOM MODULE:

\*import random

print(random.randint(0, 100))

food = ["Rav", "Pasta", "Pizza"]

print(random.choice(food)) #from a list

random.shuffle(food) #change order

SYS MODULE:

\*import sys

#max number of char

inputStatement = sys.stdin.readline(100)

This sentence is going to be read by the function we just entered

inputStatement

sys.stdout.write("Hi")

sys.version

TIME MODULE:

import time

time.time()

#time between the beginning and the final

def number(max):

time1 = time.time()

for i in range(0, max):

print(i)

time2 = time.time()

print(str(time2-time1))

#actual date in a good format

time.asctime()

tup = (2015, 10, 15, 6, 45, 12, 6, 0, 0)

time.asctime(tup)--> take format

t = time.localtime() --> it gives you a struct

year = t[0]

day = t[2]

#...

PARSE:

str()

int()

for i in range(0,5):

print(i)

time.sleep(1)

#1 second to sleep

MODULE TURTLE:

\*import turtle

#Paint a arrow in diferent directions

t = turtle.Pen()

for i in range(0,20):

t.foward(50)

t.left(60)

t.forward(60)

t.reset()

t.up() #move without Paint

t.down() #continue painting

t.color(0,0,1)

t.begin\_fill()

for i in range(0,20):

t.foward(100)

t.left(95)

t.end\_fill()

t.circle(15)

t.setheading(0)

TURTLE FUNCTIONS

import turtle

t=turtle.Pen()

def square(side):

for i in range(0,5):

t.forward(side)

t.left(90)

square(10)

square(50)

def circle(raidus):

t.circle(raidus)

circle(10)

circle(50)

def

STRINGS:

string="Ubiquitous"

string[0:4] --> Ubiq

string[:2] --> iquitous

string[1:] --> biquitous

import re

dir(re) --> al functions on this module

string = "The night was cold"

m = re.search("night", string) --> match 4,9

print(m)

start = m.start() --> 4

end= start + 5 --> 9

print(start)

print(end)

string[start:end] --> 4:9

·SOAP:

from bs4 import BeautifulSoup

soup = BeautifulSoup("<html><p>Aassdd<strong>Hello</html>", "html.parser")

print(soup.prettify())

soup

·DRILLING DOWN

html\_doc"""<html><head><title>Hi</title></head><body><p class="gold">No</p>"""

from bs4 import BeautifulSoup

soup = beautifulSoup(html\_doc, "html.parser")

soup

soup.prettify()

soup.title

soup.body.p

array = soup.find\_all('p')

array

soup.head.title.contents

for i in soup.body.children:

print(i)

#escribe todos los hijos

for i in soup.body.descendants:

print(i)

#escribe todo lo que va viendo dentro

.String FUNCTION:

soup.head.title.string

soup.head.string

soup.head.title.parent

soup.parent

soup.head.string.parent

·SEARCHING IN SOUP

soup.find\_all(id="gold")

soup.find\_all("p", id="gold")

import re

for tag in soup.find\_all(re.compile("^b")): #contenga una b (como body)

print(tag.name)

soup.find:all(href = re.compile('elsie'))