**Python**

**BASIC**

print("AbdallaHijjawe")

print("AbdallaHijjawe") ;print("AbdallaHijjawe")

رح يطبعهن طبيعي بدل ما تحط كل وحده ع سطر

name="abdallahijjawe"

print("hello "+name)

مثال ع المتغيرات

#this is commentمثال ع التعليق=

x = "aaaabdaaaalaa\

hijjjawe"==عشان نخلي السطر يلي تحت تابع للي فوق

z=“””

abdalla

hijjawe

kaboos

“””🡺هيك رح يطبع عبدالله حجاوي كابوس طبيعي

Name=”abdalla hijjawe”

print(len(name))🡺رح يطلعلك كم حرف موجود

print(type(7+9j))🡺complex

print(str(5+5)+"this is sum")🡺تحويل الرقم لجمله

name=input("enter your name: ")🡺مثال ع عمليه الادخال

num = int(input("enter number"))🡺مثال اخر ع عمليه الادخال

names = ["abdalla",1,True,["abdalla","hijjawe"]]🡺مثال ع القوائم اللست

names.append(“kaboos”)🡺رح يضيف كابوس باخر قائمه

x=[1,2,3]

z=[4,5,6]

x.extend(z)🡺دمج زد مع اكس

print(x)🡺 [1, 2, 3, 4, 5, 6]

tuples = ("abdalla",2,True)==مثال ع التبلز

الفرق بين القوائم والتبلز انه القوائم بتسمحلك تعدل التبلز لا

PhoneBook = {"abdalla":"0779550604","omar":"07771010101"}

مثال ع المعجم

Dictionary

Phonebook.get(“abdalla”)🡺07795550604

myset = {"abdalla","hijjawe","kaboos"}🡺مثال ع الست

بتفرق عن الباقي بانها غير مرتبه كل امر طباعه رح يطلع ناتج

def test():

print("test")🡺مثال ع الفنكشين

def clac(num1,num2):

sum = num1+num2

return sum

result=clac(10,20)

print(result)🡺30

مثال اخر ع الفنكشين

if x :

print("true")

else:

print("false")

الشكل العام للاف

x=True

y=True

if x and y:

print("true")

elif x or y:

print(“not equal”)

else:

print("false")

num1 = float(input("enter num1: "))

opreter = input("enter opreter: ")

num2 = float(input("enter num2: "))

result=0

if opreter == '+':

result=num1+num2

elif opreter == '-':

result=num1-num2

elif opreter == '\*':

result=num1\*num2

else:

result=num1/num2

print("the result = " + str(result))

مثال الاله الحاسبه

i=0

while i<10:

print(i)

i+=1

مثال عن اللوب

for i in range(0,20):

print(i)

try:

num = int(input("enter number = "))

except:

print("please enter int number")

مثال ع تراي واكسبكيت والفكره اذا الكود طلع فيه ايرور رح ينفذ الاكسبيكت ةبعدين يكمل الكود طبيعي

try:

num = int(input("enter number = "))

num/0

except ZeroDivisionError:🡺هاي مخصوصه لما نقسم رقم ع زيرو

print("zerooo division")

except:🡺هاي لكل الايرور دون تحديد

print("please enter int number")

except ValueError🡺هاي اذا حطينا قيمه سترنغ بدل انتجر او العكس

except ZeroDivisionError as error:

print(error)🡺رح يطلع نص ايرور الكود الاصلي

files = open("kaboos.txt",’r’)🡺r=read,

لو كانت

r+ 🡺معناها بقرا وبكتب

files.readable()🡺هل الملف قابل للقراءه

files.close()

كود من خلاله يتم اسندعاء ملف

files.readline()🡺بقراء سطر سطر

files.readlines()🡺بقراء كل السطور

files.read()🡺بقرأ السطر طبيعي

للمزيد من اختصارات الملفات

<https://harmash.com/tutorials/python/files-handling>

FORMATING

name = "abdalla"

age = 20

country = "jordan"

print("hello %s your age %s you country %s" %(name,age,country))

مثال كيف تعمل برنت بطريقه اخرى

name = "abdalla"

age = 20

country = "jordan"

print("hello {:s} your age {:f} you country {:s}" .format(name,age,country))

شكل اخر

a=1

b="abdalla"

c=3

print(f"the num is: {a} {b} {c} ")🡺the num is: 1 abdalla 3

طريقه اخرى

**MODULES**

**In file kaboos**

def callName(name):

print(f"hello {name}")

def callAge(age):

print(f"your {age}")

**in file abdalla**

import kaboos

kaboos.callName("abdalla")

طريقه استدعاء الموديلس يلي هو ملف بايثون اخر

from kaboos import callAge

callAge(10)

مثال اخر

import random

print(dir(random))

رح يطبع جميع الفنكشين الموجوده داخل الموديل

from random import \*

print(randint(1,20))🡺الفنكشين بتعطي ررقم عشوائي بين 1 20

هيك رح تستدعي الفنكشين بدون ما تكتب اسم الموديل قبلها

**OOP**

class car:

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

print("hello world")

car()

مثال ع الكلاس

#kaboos file

class car:

def \_\_init\_\_(self,color,price,model):🡺function main

self.carColor = color

self.carPrice = price

self.carModel = model

#abdalla file

from kaboos import car

bmw = car("balck",2000,2021)

ford = car("red",2555,2020)

print(bmw.carPrice)🡺2000

class car:

def mycar(self🡺لازم تكون موجود باي فنكشين بالكلاس):

print("my car")

car().mycar()

class car:

name = "bmw"

def mycar(self):

print(f"my car {self.name🡺لازم مع سيلف بدونها ايرور}")

car().mycar()

class car:

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

self.callName()

name = "abdalla"

def callName(self):

print(f"hello {self.name}")

car()

بهاي الطريقه مجرد ما نستدعي الكلاس رح يطبع مباشره لانه بالفنكيشن الرئيسيه

#kaboos file

class car:

def \_\_init\_\_(self,color,price,model):

self.myColor = color

self.myPrice = price

self.myModel = model

self.MyCar()

def MyCar(self):

if self.myModel >= 2018:

self.myPrice +=50

print(f"new model price = {self.myPrice} ")

else:

self.myPrice -=20

print(f"old model price = {self.myPrice}")

#abdalla file

import kaboos

bmw = kaboos.car("balck",2000,2021)🡺 new model price = 2050

ford = kaboos.car("red",2555,2017)🡺 old model price = 2535

#kaboos file

class car:

def \_\_init\_\_(self,name,price,model):

self.name = name

self.model = model

self.price = price

self.setPrice()

self.getName()

self.getPrice()

self.getModel()

def getName(self):

print(f"name car: {self.name}")

def getPrice(self):

print(f"price car: {self.price}")

def getModel(self):

print(f"model car: {self.model}")

def setPrice(self):

if self.model>=2018:

self.price +=500

else:

self.price=self.price

مثال ع الغيت والسيت ميثود

@staticmethod

def sayHello():

print("hello")

car.sayHello()

@classmethod

def SayHello(cls):

print(cls.tank)

car.SayHello()

class and static

مرتبطه بالكلاس مش بالكنستركتر

Inheritance

class company(car):

print("this is company")

bmw = company()

bmw.getName()

مثال ع الوراثه

class company(car):

def \_\_init\_\_(self, name, price, model):

car.\_\_init\_\_(self,name,price,model)

print(f"this company: {self.name}")

رح يوخذ الاسم من الكار

class company(car):

def \_\_init\_\_(self, name, price, model):

super.\_\_init\_\_(name)

print(f"this company: {self.name}")

بطريقه اخرى

class jmark(car,company):

print("jmark")

وراثه اكثر من كلاس

Encapsulation

\_\_name="abdalla" 🡺private

بنقدر نوصله اذا حطها بميثود

def say\_hello(self,name):

return f”hello {self.\_\_name}”

from abc import ABCMeta, abstractmethod

class prog(metaclass=ABCMeta):

@abstractmethod

def hasOop(self):

pass

class python(prog):

def hasOop(self):

return "yes"

p=python()

print(p.hasOop())

الابستراكت كلاس لازم نعمله وراثه وما نقدر نتسدعيه طبيعي

The End