

Curso programación python



SQLite
Studio



Alumne: Pol Botello

Números y cadenas

```
1 #Pide 3 cifras, enteras o decimales, positivas o negativas
2 X = float(input("Introduce la primera cifra: "))
3 Y = float(input("Introduce la segunda cifra: "))
4 Z = float(input("Introduce la tercera cifra: "))
5 #Calcula el máximo y el mínimo
6 print("El máximo es =", max(X, Y, Z), "y el mínimo es =", min(X, Y, Z))
7 #Media redondeada 3 decimales
8 print("La media redondeada es =", round((X+Y+Z)/3,3))
9 #Suma de los valores absolutos redondeado a 2 decimales
10 print("La suma de valores absolutos redondeada es =", round(abs(X)+abs(Y)+abs(Z),2))
```

Introduce la primera cifra: 12

Introduce la segunda cifra: -3.4

Introduce la tercera cifra: 3.4

El máximo es = 12.0 y el mínimo es = -3.4

La media redondeada es = 4.0

La suma de valores absolutos redondeada es = 18.8

```
1 palabra = "moscatell"
2 print(palabra[1:4])
3 print(palabra[:5])
4 print(palabra[-4:])
5 print(palabra[2])
6 print(palabra[:]) # lee todo
7 print(palabra[::-1]) #lee todo y la invierte por el -1
```

osc

mosca

tell

s

moscatell

lletacsom

```
1 cadena1 = input("Introduzca la primera cadena: ")
2 cadena2 = input("Introduzca la segunda cadena: ")
3 cadena3 = input("Introduzca la tercera cadena: ")
4 cadena4 = input("Introduzca la cuarta cadena: ")
5 cadena1 = cadena1.upper()
6 cadena2 = cadena2.upper()
7 cadena3 = cadena3.upper()
8 cadena4 = cadena4.upper()
9 mytable = "".maketrans("ÁÉÍÓÚÀÈÌÒÙ","AEIOUAEIOU")
10 print(cadena1.translate(mytable))
11 print(cadena2.translate(mytable))
12 print(cadena3.translate(mytable))
13 print(cadena4.translate(mytable))
```

Introduzca la primera cadena: Juan Palomo Callejón

Introduzca la segunda cadena: JUAN PALOMO CALLEJÓN

Introduzca la tercera cadena: JUAN PALOMO CALLEJON

Introduzca la cuarta cadena: Juan palomo callejòn

JUAN PALOMO CALLEJON

JUAN PALOMO CALLEJON

JUAN PALOMO CALLEJON

JUAN PALOMO CALLEJON

Conditionals

```
1 calificacion = input("Introduce el tipo de calificación: ")
2 nota = float(input("Introduce la nota obtenida: "))
3 if calificacion == "detallada":
4     if nota == 10:
5         print("Matricula de honor")
6     elif nota > 9:
7         print("Sobresaliente")
8     elif nota > 7:
9         print("Notable")
10    elif nota > 6:
11        print("Bien")
12    elif nota > 5:
13        print("Suficiente")
14    else:
15        print("Suspenso")
16 if calificacion == "breve":
17     if nota > 9:
18         print("Excelente")
19     if 4.8 < nota < 9:
20         print("Apto")
21     if nota < 4.8:
22         print("No apto")
23
```

Introduce el tipo de calificación: ba
Introduce la nota obtenida: 4

```
1 import requests #libreria
2 from IPython.display import Image, display, clear_output #importa 3 cosas de 1 libreria
3 import random #libreria random
4 img_opciones = Image("../img\\opcions.jpg", width = 140) #escoge una imagen de esa carpeta a 140 pixeles
5 img_pedra = Image("../img\\pedra.jpg", width = 140) #image = tipo de variable
6 img_paper = Image("../img\\paper.jpg", width = 140)
7 img_tisora = Image("../img\\tisora.jpg", width = 140)
8 #-----Mostra les opcions i demana jugada
9 display(img_opciones) #te muestra y te da a escoger
10 jdor = input("Escull i tecleja: pedra / paper / tisora ? ")
11 #-----Calcula i mostra jugada del ordinador (pc)
12 pc = random.choice(["pedra", "paper", "tisora"]) #pc escoge una de las 3 al azar
13 print ("Jo jugo : ", pc) #et diu el que escull y t'ensenyala imatge
14 if pc == "pedra" :
15     display(img_pedra)
16 elif pc == "paper" :
17     display(img_paper)
18 elif pc == "tisora" :
19     display(img_tisora)
20
21 #-----Calcula i mostra resultat
22 if pc == "pedra" and jdor == "paper" :
23     print("Jugador guanya")
24 elif pc == "pedra" and jdor == "tisora" :
25     print("PC guanya")
26 elif pc == "paper" and jdor == "tisora" :
27     print("Jugador guanya")
28 elif pc == "paper" and jdor == "pedra" :
29     print("PC guanya")
30 elif pc == "tisora" and jdor == "paper" :
31     print("PC guanya")
32 elif pc == "tisora" and jdor == "pedra" :
33     print("Jugador guanya")
34 elif pc == jdor:
35     print ("Empat")
36
37 # if (pc == "pedra" and jdor == "paper" ) or \
38 # (pc == "paper" and jdor == "tisora") or \
39 # (pc == "tisora" and jdor == "pedra") :
40 #     print ("Jugador gana")
41 #else:
42 #     print ("Pc gana")
43 # elif jdor == pc
44 #     print("Empate")
```



Escull i tecleja: pedra / paper / tisora ? pedra
Jo jugo : pedra



Llistes

```
1 def printau (f1, f2, f3) :
2     print (f1[0], "|", f1[1], "|", f1[2])
3     print (f2[0], "|", f2[1], "|", f2[2])
4     print (f3[0], "|", f3[1], "|", f3[2])
5
6 t = [list("J O"), list("JO"), list(" ")] #hacemos la matriz
7 printau(t[0], t[1], t[2]) #imprime la matriz separada
8 linea = int(input("J introduce linea: ")) #introducción de datos EN INT
9 columna = int(input("J introduce columna: "))
10 if t[linea - 1][columna - 1] != " ": #si la casilla añadida es diferente a blanco imprime error
11     print("Casilla ocupada")
12 else:
13
14     t[linea - 1][columna - 1] = "J" #Se resta 1 a cada dato, la maquina lee 0,1,2 y el usuario 1,2,3
15
16 printau(t[0], t[1], t[2])
```

```
J | | 0
  | J | 0
  | |
J introduce linea: 3
J introduce columna: 3
J | | 0
  | J | 0
  | | J
```

```
1 lista = ["a", "e", "i", "o", "u"]
2 print(lista[1:4])
3 print(lista[:5])
4 print(lista[-4:])
5 print(lista[2])
6 print(lista[:])
7 print(lista[::-1])
```

```
['e', 'i', 'o']
['a', 'e', 'i', 'o', 'u']
['e', 'i', 'o', 'u']
i
['a', 'e', 'i', 'o', 'u']
['u', 'o', 'i', 'e', 'a']
```

Bucles

```
1 estado = ""
2 potencia = 0
3 opcion = ""
4 print ("Bienvenido al Menú")
5 print ("1=Encender 2=Subir potencia 0=Apagar")
6 opcion = input("opcion: ")
7
8 while True :
9     opcion = input("opcion: ")
10    if opcion == "1":
11        estado = 'Encendido'
12        print (estado)
13    elif opcion == "2":
14        x = float(input("cantidad a aumentar: "))
15        potencia = potencia + x
16        print("La potencia está en ", potencia)
17    elif opcion == "0":
18        estado = 'Apagado'
19        print (estado)
20        break
```

```
1 alus = ["ana","joan", 'rosa', 'antonio']
2
3 notas = []
4 for alu in alus:
5     nota = float(input("Nota de "+ alu+ ": "))
6     notas.append(nota)
7 i = 0
8 for alu in alus:
9     print(alu, notas[i])
10    i+=1
11
12
```

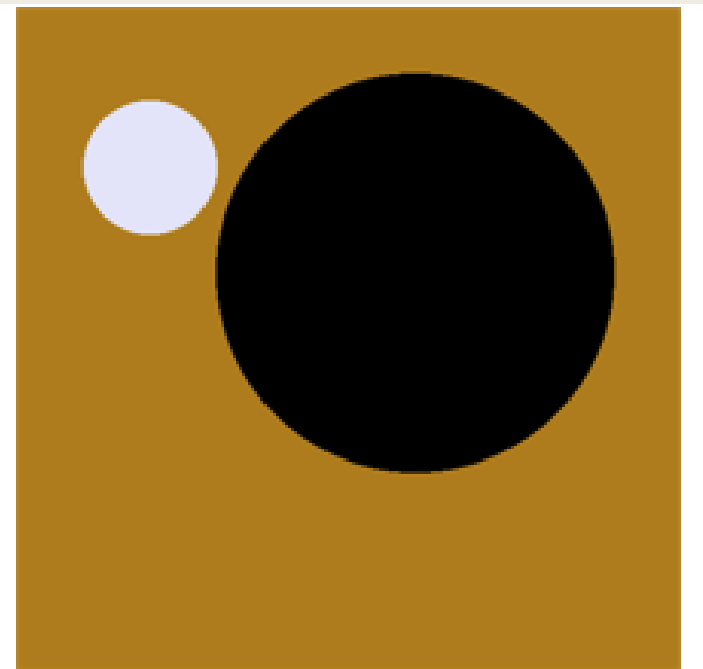
```
Nota de ana: 10
Nota de joan: 9
Nota de rosa: 7
Nota de antonio: 5
ana 10.0
joan 9.0
rosa 7.0
antonio 5.0
```

Col·leccions

```
1 #ejercicio de modificacion de cadenas
2 cadena = """POR ÚLTIMO, LA FORMA MÁS RECOMENDABLE Y PYTHONICA, PERO MÁS COMPLEJA,
3 SERÍA USAR MATPLOTLIB MEDIANTE LA INTERFAZ ORIENTADA A OBJETOS.CUANDO SE PROGRAMA CON MATPLOTLIB,
4 NO MIENTRAS SE TRABAJA INTERACTIVAMENTE, ESTA ES LA FORMA QUE
5 PERMITE TENER MÁS CONTROL SOBRE EL CÓDIGO. QUIZÁ VEAMOS ESTO EN EL FUTURO SI OS ANIMÁIS A ESCRIBIR SOBRE ELLO."""
6 texto = cadena.replace(".", " ")
7 t = texto.replace(",","")
8 t = t.lower()
9 #num de vocales
10 mytable = "".maketrans("áéíóú","aeiou")
11 t = t.translate(mytable)
12 la = t.count("a")
13 le = t.count("e")
14 li = t.count("i")
15 lo = t.count("o")
16 lu = t.count("u")
17 print(t)
18 print("a = {a} , e = {e}, i = {i}, o = {o}, u = {u}".format(a=la,e=le,i=li,o=lo,u=lu))
19 #consonante mas usada
20 m = t.count("m")
21 l = t.count("l")
22 n = t.count("n")
23 s = t.count("s")
24 r = t.count("r")
25 listac = [m, l, n, s, r]
26 mayorc = max (listac)
27 #listac = listac.sort()
28 print(listac, mayorc)
29 #crea una lista
30 lista = t.split() #importante para que la divida entre palabras y no letras
31 print(lista)
32 for x in lista:
33     if len(x) > 6 and x[0] not in ("aeiou"):
34         print(x)
35
```

por ultimo la forma mas recomendable y pythonica pero mas compleja
seria usar matplotlib mediante la interfaz orientada a objetos cuando se programa con matplotlib
no mientras se trabaja interactivamente esta es la forma que
permite tener mas control sobre el codigo quiza veamos esto en el futuro si os animais a escribir sobre ello
a = 35 , e = 33, i = 19, o = 28, u = 7
[17, 15, 15, 19, 22] 22
['por', 'ultimo', 'la', 'forma', 'mas', 'recomendable', 'y', 'pythonica', 'pero', 'mas', 'compleja', 'seria', 'usar', 'matplotl
ib', 'mediante', 'la', 'interfaz', 'orientada', 'a', 'objetos', 'cuando', 'se', 'programa', 'con', 'matplotlib', 'no', 'mientra
s', 'se', 'trabaja', 'interactivamente', 'esta', 'es', 'la', 'forma', 'que', 'permite', 'tener', 'mas', 'control', 'sobre', 'e
l', 'codigo', 'quiza', 'veamos', 'esto', 'en', 'el', 'futuro', 'si', 'os', 'animais', 'a', 'escribir', 'sobre', 'ello']
recomendable
pythonica
compleja
matplotlib
mediante
programa
matplotlib
mientras
trabaja
permite

```
1 from PIL import Image, ImageDraw
2 img = Image.new('RGB', (500, 500), 'darkgoldenrod')
3 dib = ImageDraw.Draw(img)
4 #-----dibuja una elipse y la muestra en el lienzo
5 dib.ellipse((150, 50, 449, 349), 'Black')# (x0, y0, x1, y1)
6 dib.ellipse((50, 70, 150, 170), "Lavender")
7 img.show()
8 #-----salva la imagen y su miniatura
9 img.save('./img/ikusa.jpg', "JPEG")
10 size = (128, 128)
11 img.thumbnail(size)
12 img.save('./img/ikusa-mini.jpg', "JPEG")
13 #duplas = 3
```



Funcions

```
1 def area_rectangulo(base,alt):
2     try:
3         area = base * alt
4         return(area)
5     except:
6         return(None)
7
8 area_rectangulo(m,c)
```

```
1 def sumalista_k (numeros,k = 0):
2     suma = 0
3     for num in numeros:
4         suma += num + k
5     return(suma)
6
7 print(sumalista_k([1,2,3]))
8
```

```
1 def change_char_in_string(cadena,pos,caracter):
2     try:
3         if pos < len(cadena):
4             return(cadena[0:pos]+caracter+cadena[pos+1:])
5         else:
6             return(None)
7     except:
8         return(None)
9 x = change_char_in_string("-----",1,"a")
10 print(x)
```

```
1 def input_float(mensaje): #control de errores
2     try:
3         cad = input(mensaje)
4         cad = cad.replace(",",".")
5         num = float(cad)
6     except:
7         num = 0
8         print("No es numérico")
9     return(num)
10
11 x = input_float("x: ")
12 print(x)
```


Conjunts

```
1 # test_conjuntos.py
2
3 c1 = set () # Un nuevo conjunto vacío
4 c1.add ("gato") # Agregar un solo elemento
5 c1.update (["perro", "ratón"]) # Agregar varios elementos,
6 # por extensión de la lista
7 c1 |= set (["asno", "caballo"]) # Agregar 2 elementos
8 if "gato" in c1 : # Esta en la colección ?
9     c1.remove ("gato")
10 # c1.remove ("elefante") arroja un error
11 c1.discard ("elefante") # No lanza ningún error
12 print (c1)
13
14 for item in c1: # Iteracion for each element
15     print (item)
16 print ("Item count:", len(c1)) # Longitud /tamaño/núm.elementos primero = c1[0]
17 # Error: no hay índices para conjunt
18 es_vacio = len(c1) == 0 # Prueba de vacío
19
20 c1 = {"gato", "perro"} # Inicializar el set usando llaves;
21 #c1 = {} # Esto no es un diccionario
22 c1 = set(["gato", "perro"]) # Inicializar usando una lista
23 c2 = set(["perro", "raton"])
24 c3 = c1 & c2 # Intersección
25
26
27
```

```
{'ratón', 'caballo', 'asno', 'perro'}
ratón
caballo
asno
perro
Item count: 4
```

```
1 c4 = c1 | c2 # Union
2 c5 = c1 - c3 # Diferencia de conjuntos
3 c6 = c1 ^ c2 # Diferencia simétrica
4 issubc = c1 <= c2 # Prueba de subconjunto
5 issuperc = c1 >= c2 # Prueba de superconjunto c7 =
6
7 c7 = c1.copy() # Crea una copia
8 c7.remove("gato")
9 print (c7.pop()) # Elimina un elemento arbitrario
10
11 c8 = c1.copy()
12 c8.clear() # Limpia, vacía el conjunto
13 c9 = {x for x in range(10) if x % 2}
14
15 # c comprehension; since Python 2.7
16 print (c1, c2, c3, c4, c5, c6, c7, c8, c9, issubc, issuperc)
17
```


Matplotlib

```
1 import numpy as np
2 size=10
3 print ("Retorna un número de floats entre [0.0, 1.0).")
4 print (np.random.random([size]))
5 print ("Retorna un número de floats entre low, to high, size")
6 print (np.random.uniform(2,80,10))
7 print ("Genera ejemplos randoms tomados de un array : a[, size, replace, p]")
8 print (np.random.choice([1,2,3,4,5,6],4))
9
```

Retorna un número de floats entre [0.0, 1.0).

```
[0.26440309 0.98319285 0.23528262 0.77499521 0.87640172 0.22515587
 0.31311048 0.4845564 0.6586533 0.72781503]
```

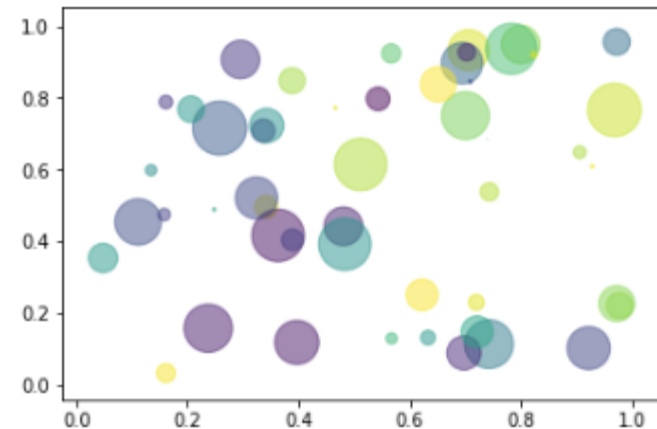
Retorna un número de floats entre low, to high, size

```
[67.1249228 10.73877755 3.07652206 78.82191778 7.73547298 63.48218002
 2.96042873 32.53500903 24.09792351 29.92115574]
```

Genera ejemplos randoms tomados de un array : a[, size, replace, p]

```
[2 4 4 3]
```

```
1 # EJEMPLO DE GRAFICO DE BURBUJAS
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Fixing random state for reproducibility
7 np.random.seed(19680801)
8
9
10 N = 50
11 x = np.random.rand(N)
12 y = np.random.rand(N)
13 colors = np.random.rand(N)
14 area = (30 * np.random.rand(N))**2 # 0 to 15 point radii
15
16 plt.scatter(x, y, s=area, c=colors, alpha=0.5)
17 plt.show()
```



Pandas

```
1 import pandas as pd
2 ventas = pd.DataFrame({"A": [41, 32, 56, 18],
3                          "B": [17, 54, 6, 78],
4                          "C": [12, 13, 16, 18] },
5                        index = ["Gen", "Feb", "Mar", "Abr"])
6 print(ventas)
7 print("SUMA")
8 print(ventas.sum(axis=1))
9 print(ventas.describe())
```

	A	B	C
Gen	41	17	12
Feb	32	54	13
Mar	56	6	16
Abr	18	78	18
SUMA			
Gen	70		
Feb	99		
Mar	78		
Abr	114		

dtype: int64

	A	B	C
count	4.000000	4.000000	4.000000
mean	36.750000	38.750000	14.750000
std	15.945219	33.260337	2.753785
min	18.000000	6.000000	12.000000
25%	28.500000	14.250000	12.750000
50%	36.500000	35.500000	14.500000
75%	44.750000	60.000000	16.500000
max	56.000000	78.000000	18.000000

```
1 #Ejercicio 1
2
3 df = df.append({"Nom": "Pol", "Dept": "MANT", "DiesV": "27"}, ignore_index = True )
4 df = df.append({"Nom": "Paco", "Dept": "ADMIN", "DiesV": "2"}, ignore_index = True )
5 df = df.append({"Nom": "Paula", "Dept": "PROD", "DiesV": "7"}, ignore_index = True )
6
7 #print(df)
8 df.sort_values(by=["Dept", "DiesV"], ascending = False)
```

	Nom	Dept	DiesV	PreuDia
6	Sara	VENDES	8	44.0
9	Paula	PROD	7	NaN
0	Sònia	PROD	32	60.0
4	Sam	PROD	30	85.0
7	Pol	MANT	27	NaN
2	David	MANT	20	90.0
5	Manel	MANT	5	90.0
8	Paco	ADMIN	2	NaN
1	Laura	ADMIN	55	80.0
3	Rosa	ADMIN	43	100.0

Control d'errors

```
1 #Ejercicio 1
2 try:
3     resultado = 10/0
4     print(resultado)
5 except ZeroDivisionError:
6     print("La division es entre 0, no tiene resultado")
7 except Exception as e:
8     print(type(e))
```

La division es entre 0, no tiene resultado

```
1 try:
2     lista = [1, 2, 3, 4, 5]
3     lista[10]
4 except IndexError:
5     print("El indice no se encuentra en la lista")
6 except Exception as e:
7     print(type(e))
```

El indice no se encuentra en la lista

```
1 try:
2     resultado = 15 + "20"
3     print(resultado)
4 except TypeError :
5     print ("No es un número válido, uno de los valores es str ")
6 except Exception as e:
7     print(type(e))
```

No es un número válido, uno de los valores es str

```
1 try:
2     nombref = "./dat/lista-negra.dat"
3     f = open(nombref,"r")
4     texto = f.read()
5     print (texto)
6 except IOError:
7     nombref = "./dat/lista-negra.dat"
8     f = open(nombref,"w+")
9     f = open(nombref,"r")
10    print("No existe el archivo : ", nombref, "Creando archivo, ejecute el programa de nuevo")
11 except Exception as e :
12    print (e)
13 else :
14    print("Contenido leído correctamente")
15    f.close()
```

Contenido leído correctamente

Fitxers

```
1 #ejercicio 5
2 fichero = open("./dat/Ejercicio 5.txt","r")
3 fichero_read = fichero.read()
4 fichero_dades = fichero_read.lower().count("dades")
5 fichero_variable = fichero_read.lower().count("variable")
6 fichero_lliure = fichero_read.lower().count("lliure")
7 fichero.close()
8 print( "dades hay:", (fichero_dades),"\n", "variable hay: ", (fichero_variable),"\n", "lliure hay: ", (fichero_lliure))
9 for palabra in ["dades","variable","lliure"]:
10     print(palabra, fichero_read.lower().count(palabra))
11 #2 formas de hacerlo
12 # fichero.seek(n) mueve el cursor a la altura que quieras, permite moverte en un fichero sin tener que cerrarlo y volverlo
13 # a abrir
```

```
dades hay: 2
variable hay: 1
lliure hay: 0
dades 2
variable 1
lliure 0
```

```
1 fichero = open("./dat/fichero.texto","r") #Leer con "r"
2 texto = fichero.readlines() #conviertes el read a variable para poder imprimirlo
3 fichero.close()
4 print(texto)
```

```
['Hola Mundo \n', '1\n', '2\n', '3\n', '4\n', '5\n', '6\n', '7\n', '8\n', '9\n', 'Hola Mundo \n', '1\n', '2\n', '3\n', '4\n', '5\n', '6\n', '7\n', '8\n', '9\n']
```

```
1 fichero = open("./dat/fichero.texto","r") #Leer con "r"
2 texto = fichero.read() #conviertes el read a variable para poder imprimirlo
3
4 print(type(texto))
5 testo = fichero.readlines()
6 print(type(testo))
7 teztto = fichero.readline()
8 print(type(teztto))
9 fichero.close()
```

```
<class 'str'>
<class 'list'>
<class 'str'>
```

Classes

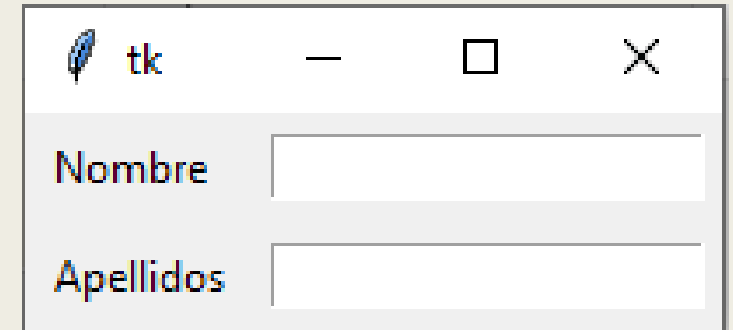
```
1 class Pelicula:
2     def __init__(self, titulo, duracion, año):
3         self.titulo = titulo
4         self.duracion = duracion
5         self.año = año
6     def __str__(self):
7         cadena = self.titulo + "\t" + str(self.duracion) + "\t" + str(self.año)
8         return(cadena)
9
10 p1 = Pelicula("matrix", 120, 1999)
11 p2 = Pelicula("Como entrenar a tu dragon", 90, 2010)
12 # print(p2)
13
14
15 class Catalogo:
16     def __init__(self, peliculas = []):
17         self.peliculas = peliculas
18     def agregar(self, pelicula):
19         if pelicula not in self.peliculas:
20             self.peliculas.append(pelicula)
21         else:
22             print("Pelicula ya añadida")
23     def eliminar(self, pelicula):
24         if pelicula in self.peliculas:
25             self.peliculas.remove(pelicula)
26         else:
27             print("Pelicula no está añadida")
28     def mostrar(self):
29         for pelicula in self.peliculas:
30             print(pelicula)
31     def vaciar(self):
32         self.peliculas = []
33
34 c = Catalogo()
35 c.agregar("Matrix")
36 c.agregar("ET")
37 c.mostrar()
38 c.eliminar("Matrix")
39 c.mostrar()
40 c.eliminar("Matrix")
```

```
1 class Galleta: #clase
2     chocolate = False #atributo o variable
3     marca = ""
4     forma = "redonda"
5     def mostrar_marca(self):
6         print("marca: ", self.marca)
7     def mostrar_forma(self):
8         print("forma: ", self.forma)
9     def chocolatear(self): #metodos
10         self.chocolate = True
11     def __str__(self): #esta funcion te permite poner print(g) y te de toda la info
12         cadena = str(self.chocolate) + self.marca + self.forma
13         return(cadena)
14 #-----
15 g = Galleta()
16 print(g.chocolate)
17 print(g.forma)
18 # ----- another galleta
19 g2 = Galleta()
20 g2.forma = "cuadrada"
21 g2.mostrar_marca()
```

Tkinter

```
1 from tkinter import *
2 root = Tk()
3
4 frame = Frame(root) #creamos un frame con nombre frame situado en root
5 frame.pack() #empaquetamos el frame, esta vacío
6 frame.config(bg="lightblue") #Le ponemos el fondo azul, da igual por lo que no tiene nada
7 frame.config(width = 480, height=320) #aquí le damos tamaño para que se vea el fondo
8
9 root.mainloop()
```

```
1 from tkinter import *
2 root = Tk()
3 frame = Frame(root)
4 entry = Entry(root)
5
6
7 label = Label(root, text="Nombre")
8 label.grid(row=0,column=0, sticky=W, padx=5, pady=5)
9
10 entry = Entry(root)
11 entry.grid(row=0,column=1, padx=5, pady=5)
12 entry.config(justify = "right", state = "normal") #el texto que escribes se justifica a la derecha
13
14 label2 = Label(root, text="Apellidos")
15 label2.grid(row=1,column=0, sticky=W, padx=5, pady=5)
16
17 entry2 = Entry(root)
18 entry2.grid(row=1,column=1, padx=5, pady=5)
19 entry2.config(justify = "center", show = "?") #el texto se justifica al centro y además salen ?
20
21 root.mainloop()
```



Tkinter

```
1 revertir = lambda cadena : cadena[::-1]
2 revertir("Hola")
```

'aloH'

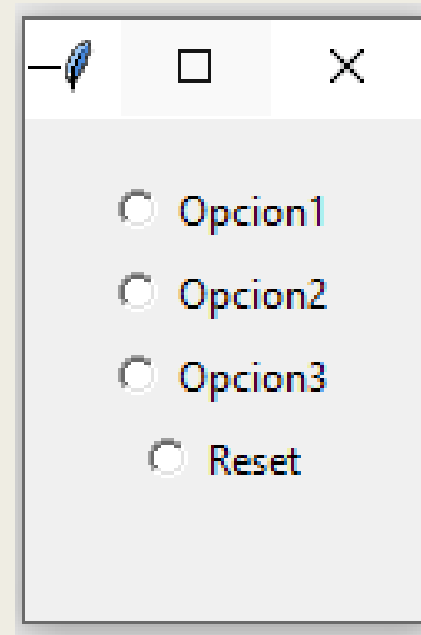
```
1 #parametros indeterminados
2 def argumentos (*args):
3     for arg in args:
4         print(arg)
5
6 argumentos(5, "Hola", [1,2,3,4], "Patata")
```

5
Hola
[1, 2, 3, 4]
Patata

```
1 def argumentos (**kwargs):
2     print(kwargs)
3
4 argumentos(n = 5, c="Hola", a =[1,2,3,4], p ="Patata")
```

{'n': 5, 'c': 'Hola', 'a': [1, 2, 3, 4], 'p': 'Patata'}

```
1 #radiobutton
2 from tkinter import *
3 def select():
4     monitor.config(text = "Opcion {}".format(opcion.get() ) )
5 def reset():
6     opcion.set(None)
7     monitor.config(text="")
8
9 root=Tk()
10 root.config(bd = 15)
11
12 opcion = IntVar()
13 Radiobutton(root, text="Opcion1", variable = opcion, value = 1, command=select).pack()
14 Radiobutton(root, text="Opcion2", variable = opcion, value = 2, command=select).pack()
15 Radiobutton(root, text="Opcion3", variable = opcion, value = 3, command=select).pack()
16 Radiobutton(root, text="Reset", variable = opcion, value = 4, command=reset).pack()
17
18 monitor = Label(root)
19 monitor.pack()
20 root.mainloop()
```



SQLite

```
1 import sqlite3
2 conexion = sqlite3.connect("Ejemplo.db")
3 cursor = conexion.cursor()
4
5 cursor.execute("SELECT * FROM usuarios")
6 print(cursor)
7 usuario = cursor.fetchall()#que coja todos los datos
8 for x in usuario:
9     print(x)
10 #con el for lo imprime por filas. Esta el dato inicial y la lista añadida antes
11
12 conexion.close()
```

<sqlite3.Cursor object at 0x000001B4B2A67CE0>

('Hector', 27, 'Hectorprofe@gmail.com')

('Mario', 51, 'mario@ejemplo.com')

('Mercedes', 38, 'mercedes@ejemplo.com')

('Juan', 19, 'juan@ejemplo.com')

SQLiteStudio (3.3.3) - [Profes (civics)]

Database Structure View Tools Help

SQLiteStudio (3.3.3) - [Profes (civics)]

Database Structure View Tools Help

Filter by name

Tenda (SQLite 3)

- Tables (1)
 - Produc...
- Views

civics (SQLite 3)

- Tables (2)
 - Cursos
 - Profes
- Views

Structure Data Constraints Indexes Triggers DDL

civics Table name: Profes ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated
1	codiprofe	CHAR (10)							NULL
2	nomprofe	VARCHAR (20)							NULL
3	cognomprofe	VARCHAR (30)							NULL
4	dniprofe	CHAR (12)							NULL

Web prácticas

```
1 #
2 # PRACTICA P01 - CONECTA CON ESTE CANAL DE NOTICIAS E IMPRIMIE LOS TITULARES DE TODAS ELLAS
3 #
4 import feedparser
5
6 NewsFeed = feedparser.parse("https://timesofindia.indiatimes.com/rssfeedstopstories.cms")
7
8 print ("hay ", len(NewsFeed.entries), " noticias")
9 entry = NewsFeed.entries[1]
10
11 print (entry.published)
12 print ("*****")
13 for i in range(0,len(NewsFeed.entries)):
14     print (NewsFeed.entries[i].summary, sep = "\n")
15 #print (entry.summary)
16 print ("-----News Link-----")
17 print (entry.link)
```

hay 18 noticias
Fri, 29 Apr 2022 09:18:49 GMT

Satellites have been helping motorists navigate their way to destinations for decades now, but on Thursday, for the first time in Asia-Pacific region, the very same constellation of satellites that offers GPS teamed up with three ISRO satellites to provide three-dimensional navigation guidance to pilots who landed their aircraft safely on to the Ajmer airport runway in a successful trial flight.

Sri Lankan president Gotabaya Rajapaksa on Friday agreed to remove brother as prime minister Mahinda Rajapaksa amid worst economic crisis in the country.

Russia used a diesel submarine in the Black Sea to strike Ukrainian military targets with Kalibr cruise missiles, the defence ministry said on Friday. Russia's defence ministry earlier today confirmed it had carried out an air strike on Kyiv during a visit by UN secretary-general Antonio Guterres. Stay with TOI for the latest developments

With 11 more students testing positive at IIT-Madras, the tally at the institute has reached 183. India has reported 3,337 new Covid-19 cases and 60 deaths in the last 24 hours, a Union health ministry bulletin released on Friday stated. The active caseload has crossed 17,000 mark with a rise of 821 such cases. The country has also registered 2,496 fresh recoveries. Stay with TOI for the latest developments

A clash broke out between two groups near Kali Devi Mandir in Punjab's Patiala on Friday. Police personnel are deployed at the spot to maintain law and order situations.

The spike in coal requirement due to the steep increase in electricity demand has forced Indian Railways to cancel around 16 mail/express and passenger trains daily in the past couple of weeks to create extra paths for rakes carrying the key ingredient to power plants located across the country. The railway ministry has notified the cancellation of nearly 670 trips of passenger trains till May 24.