Módulo 2: Programación Ofensiva

Curso de Especialista en Ciberseguridad y Hacking Ético

Año de realización: 2021

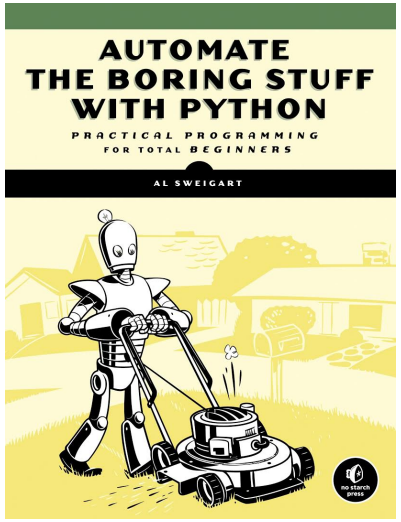
PROFESOR/A
Asier Martínez

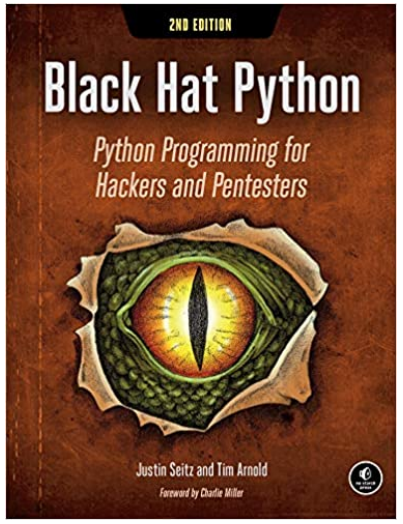
www.eoi.es

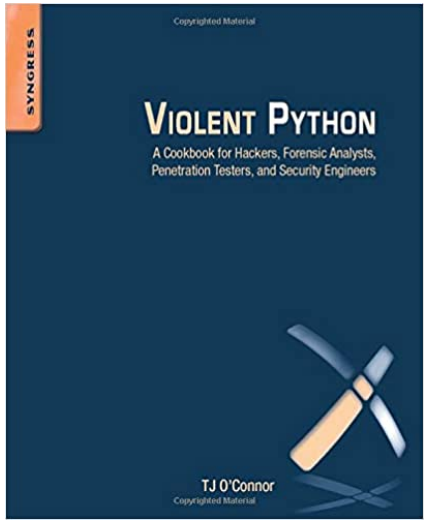
Tabla de contenidos

Documentación adicional

- Manual Oficial https://docs.python.org/es/3/tutorial/index.html
- Python para todos (Castellano) http://do1.dr-chuck.com/pythonlearn/ES_es/pythonlearn.pdf
- Tutorialspoint (inglés) https://www.tutorialspoint.com/python3/python_tutorial.pdf
- Manual de Python https://aprendeconalf.es/python/manual/







Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

INTRODUCCIÓN

Python

Características

- Es código abierto.
- Multiplataforma.
- Multiparadigma.
- Es maduro.
- Es un lenguaje interpretado.
- Rápido.
- Sencillo.
- Ordenado y limpio.
- Fuertemente tipado.

Instalación

Windows

https://www.python.org
Windows
Python 3.x.x
Windows x86-64 executable installer
Marcar “Add Python 3.x to Path”

OSX

https://www.python.org
Mac OS X
Python3.x.x
macOS 64-bit installer

Comprobar: python -v

Linux

sudo apt-get install python3
sudo yum install python3

PIP – Administrador de paquetes de Python

Instalar

pip install nombre-paquete

Desinstalar

pip uninstall nombre-paquete

Permite gestionar listas de paquetes y sus números de versión correspondientes.

pip install -r requisitos.txt

Instalar un paquete para una versión concreta de Python

pip${versión} install nombre-paquete

Actualizar pip

python -m pip install --upgrade pip

Intérprete

```
C:\Users\Python>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 6+4
10
>>> 12+17
29
>>> 2*3
6
>>> print (“Prueba")
Prueba
>>>
```

Primer programa

```
#!/usr/bin/env python


# -*- encoding: utf-8 -*-


# Este programa pregunta por tu nombre y apellidos y muestra un mensaje para saludar
print('Escribe tu nombre y 2 apellidos')
nombre = input()
print('Encantado de saludarte ' + nombre)
print('Encantado de saludarte', nombre)
print ("Encantado de saludarte, %s" % nombre)
print("Encantado de saludarte {}".format(nombre))
print (f”Encantado de saludarte {nombre}”)
```

Ejecutar scripts

- Extensión .py

```
# Este programa pregunta por tu nombre y apellidos
# y muestra un mensaje para saludar

print('Escribe tu nombre y 2 apellidos')
nombre = input()
print('Encantado de saludarte ' + nombre)
print('Encantado de saludarte', nombre)
print("Encantado de saludarte %s" % nombre)
print(f"Encantado de saludarte {nombre}")
```

```
Escribe tu nombre y 2 apellidos
Juan Rodríguez López
Encantado de saludarte Juan Rodríguez López
Encantado de saludarte Juan Rodríguez López
Encantado de saludarte Juan Rodríguez López
Encantado de saludarte Juan Rodríguez López
```

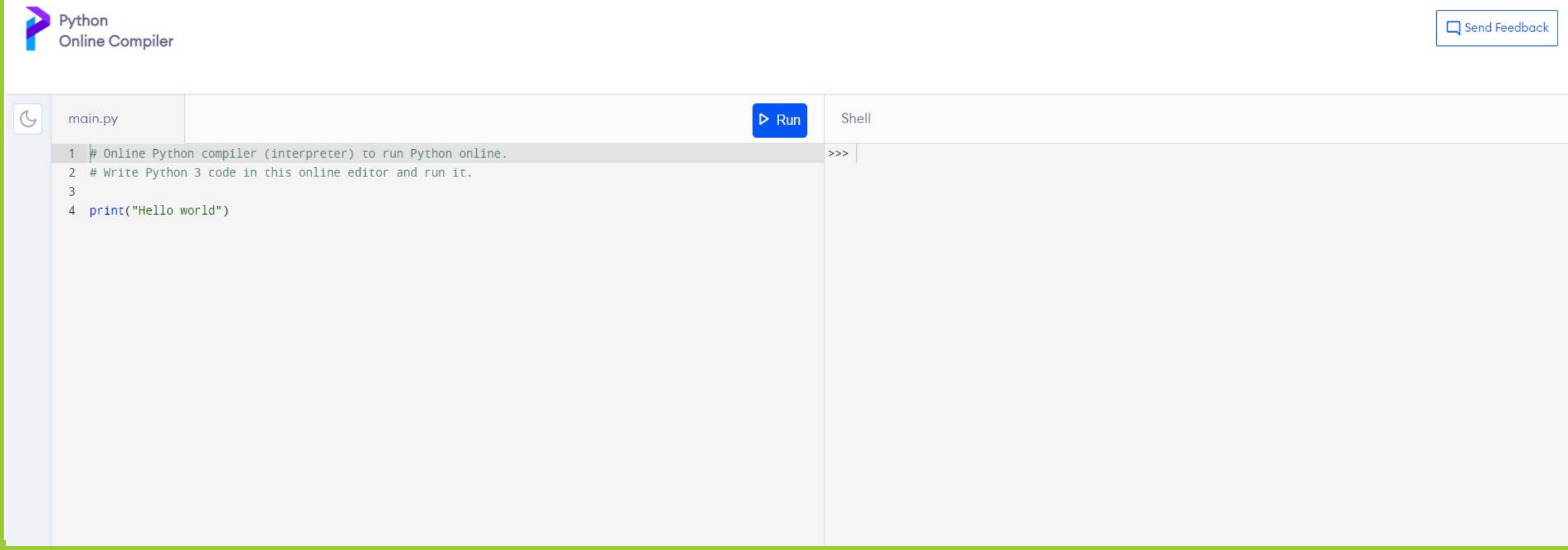
IDE - PYCHARM

https://www.jetbrains.com/es-es/pycharm/

Python Online Compiler

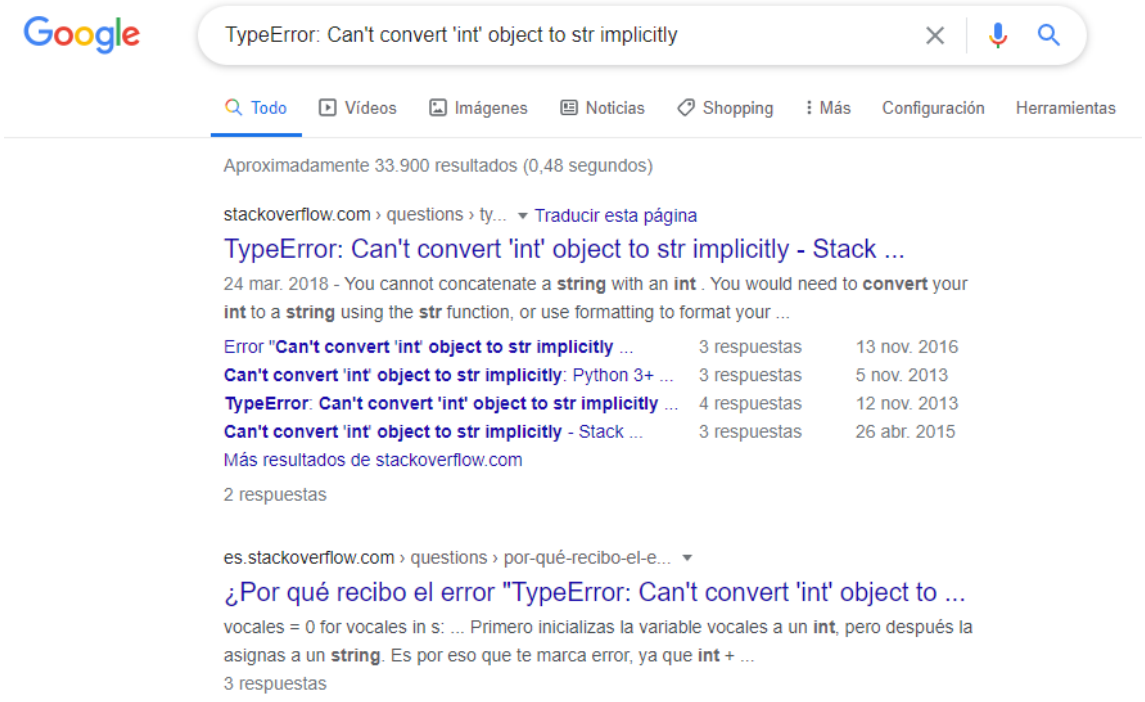
https://www.programiz.com/python-programming/online-compiler/

https://www.w3schools.com/python/trypython.asp?filename=demo_default



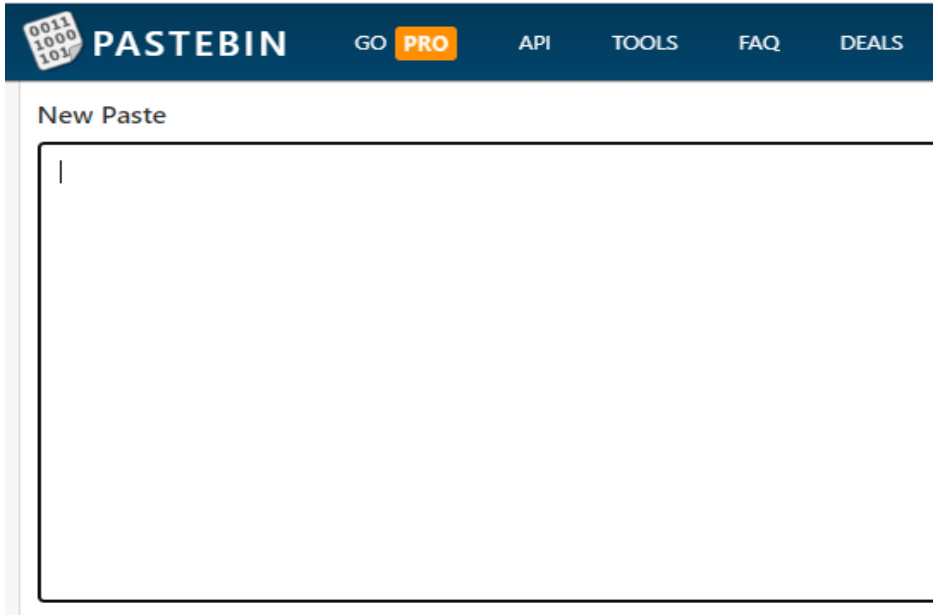
¿Cómo encontrar ayuda?

- Dudas genéricas



- Dudas concretas

https://stackoverflow.com/



Palabras reservadas

and
as
assert
break
class
continue
def
del
elif
else
except
exec
finally
for
from
global
if
import
in
is
lambda
not
or
pass
print
raise
return
try
while
with
yield

Convenciones en Python

- Los nombres de los ficheros deben escribirse en minúscula y sin guiones bajos. Ejemplo: mifichero.py
- Las llamadas a funciones se escriben en minúscula. Ejemplo: print() | type()
- Las variables se escriben en minúscula y, de estar formadas por varias palabras, éstas van unidas por guiones bajos. Ejemplo: deportes | deportes_con_raqueta
- Pon un espacio después de cada coma.
- Pon un espacio antes y después de cada operador. Ejemplo: 4 + 3 | total += 1
- Se recomienda indentar con 2 o 4 espacios, en lugar de con tabulador.
- Escribe abundantes comentarios en tu código, describiendo cada detalle, para hacer que sea lo más claro y legible posible.
- PEP 8 -- Style Guide for Python Code

Comentarios en Python

- Los comentarios en Python comienzan con una #

```
#Esto es un comentario

print("Mensaje de prueba")
```

- Comentarios Inline

```
print("Mensaje de prueba") #Esto es un comentario
```

- Bloque de comentarios

```
#Esto es un comentario
#Esto es un comentario
#Esto es un comentario
```

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

VARIABLES Y CONSTANTES

Python

Variables

Es un nombre que se refiere a un objeto que reside en la memoria.

#Nombres válidos en python:

nombre = “Pedro"

nombre_usuario = "Pedro"

_nombre_usuario = "Pedro"

nombreUsuario = "Pedro"

NOMBREUSUARIO = "Pedro"

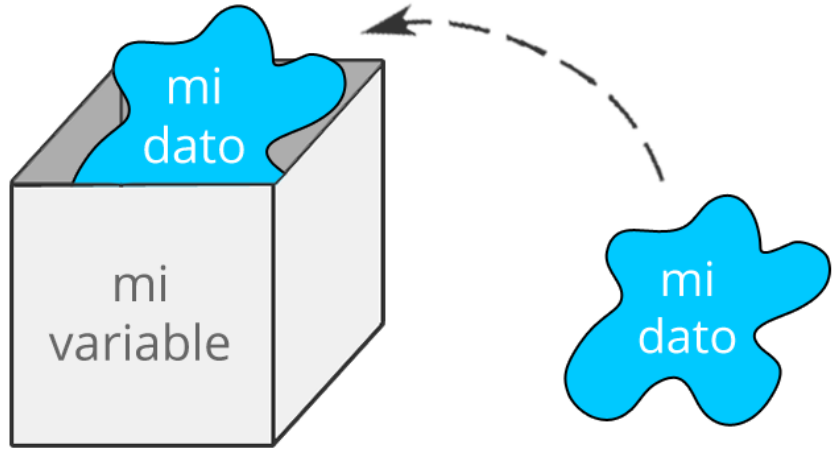
nombreusuario1 = "Pedro"

#Nombres no válidos:

2nombreusuario = "Pedro"

nombre-usuario = "Pedro"

nombre usuario = "Pedro"



Variables

- Ejemplo de asignar valor a variable

```
cantidad = 100 | cantidad = int(100)          # Asignación de un número entero.
miles   = 1000.0  | miles = float(1000.0)     # Asignación de un número de punto flotante.
nombre    = "Juan“ | str(“Juan”)       # Asignación de una cadena de texto.
```

- Ejemplo de asignar múltiples valores a múltiples variables

```
x, y, z = "Manzana", "Pera", "Naranja“
```

- Ejemplo de asignar el mismo valor a múltiples variables al mismo tiempo

```
a = b = c = 1
```

Constantes

Es un tipo de variable la cual no puede ser cambiada.

```
#Constantes
RUTA_API_SERVIDOR = "http://algun.sitio/asd"
CORREO_ADMIN = "correo@sitio.com"
```

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

OPERADORES

Python

Tipos de operadores

- Operadores aritméticos
- Operadores de asignación
- Operadores de comparación
- Operadores lógicos
- Operadores de membresía
- Operadores de identidad

Operadores aritméticos

Operador	Descripción	Ejemplo
+	Suma	3 + 2 = 5
-	Resta	4 - 7 = -3
*	Multiplicación	2 * 6 = 12
**	Exponente	2 ** 6 = 64
/	División	3.5 / 2 = 1.75
//	División entera	3.5 // 2 = 1.0
%	Módulo	7 % 2 = 1

Ejercicio con operadores aritméticos

Siendo a = 20 y b = 10, calcular: suma, resta, multiplicación, exponente, división, división entera y módulo.

a = 20

b = 10

El resultado de a + b es 30

El resultado de a - b es 10

El resultado de a x b es 200

El resultado de a ** b es 10240000000000

El resultado de a / b es 2.0

El resultado de a // b es 2

El resultado de a % b es 0

Ejercicio: Convertir Fahrenheit a Celsius

Dados dos números, mostrar la suma, resta, división y multiplicación de ambos.

Fórmula de conversión: C = (F-32)*5/9

Ejercicio: Obtener calificación

Un alumno desea saber cual será su calificación final en la asignatura de Python. Dicha calificación se compone de los siguientes porcentajes:

- 55% del promedio de sus tres calificaciones parciales.
- 30% de la calificación del examen final.
- 15% de la calificación de un trabajo final.

Ejercicio: Calcular dinero en metálico

Diseñar un algoritmo que nos diga el dinero que tenemos (en euros y céntimos) después de pedirnos cuántas monedas tenemos (de 2€, 1€, 50 céntimos, 20 céntimos o 10 céntimos).

Operadores de asignación

Operador	Descripción	Ejemplo		
=	asigna valor a una variable	r = 5	r1 = r	
+=	suma el valor a la variable	r = 5	r += 10	r = 15
-=	resta el valor a la variable	r = 5	r -= 10	r = -5
*=	multiplica el valor a la variable	r = 5	r *= 10	r = 50
/=	divide el valor a la variable	r = 5	r /= 10	r = 0
**=	calcula el exponente del valor de la variable	r = 5	r **= 10	r = 9765625
//=	calcula la división entera del valor de la variable	r = 5	r //= 10	r = 0
%=	devuelve el resto de la división del valor de la variable	r = 5	r %= 10	r = 5

Ejercicio con operadores de asignación

Siendo a = 20 y b = 10, utilizar todos los operadores de asignación.

a = 20

b = 10

a += b 30

a -= b 10

a *= b 200

a /= b 2.0

a **= b 10240000000000

a //= b 2

a %= b 0

Operadores de comparación

Operador	Descripción	Ejemplo
==	¿son iguales a y b?	5 == 3 False
!=	¿son distintos a y b?	5 != 3 True
<	¿es a menor que b?	5 < 3 False
>	¿es a mayor que b?	5 > 3 True
<=	¿es a menor o igual que b?	5 <= 5 True
>=	¿es a mayor o igual que b?	5 >= 3 True

Operadores lógicos / booleanos

Operador	Descripción	Ejemplo
and	Devuelve True si ambas afirmaciones son verdaderas.	x < 3 and x < 20
or	Devuelve True si una de las afirmaciones es verdadera.	x < 3 or x > 5
not	Invertir el resultado, devuelve False si el resultado es verdadero.	not(x < 10 and x < 20)

Operadores de membresía

Operador	Descripción	Ejemplo
in	Devuelve True si una secuencia con el valor especificado está presente en el objeto.	‘coche’ in vehiculos
not in	Devuelve True si una secuencia con el valor especificado no está presente en el objeto	‘coche’ not in vehiculos

Operadores de identidad

Operador	Descripción	Ejemplo
is	Devuelve True si ambas variables son el mismo objeto.	x is y
is not	Devuelve True si ambas variables no son el mismo objeto.	x is not y

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

TIPOS DE DATOS

Python

Tipo números

int

x = 6
y = 12345678910
z = -7654321

float

x = 5.50
y = -40.35

complex

x = 2+3j
y = 3j

Los números se pueden declarar en decimal, binario, octal y hexadecimal:

- 344 en decimal.
- 0b101011000 en binary.
- 0o530 en octal.
- 0x158 en hexadecimal.

Declarar y convertir números:

x = int(20)
x = float(20.5)
x = complex(1j)

Tipo números – Funciones matemáticas

Función	Valor de retorno (descripción)
abs (x)	Devuelve el valor absoluto, tal como ABS (-10) devuelve 10
cmp (x, y)	Si x <y devuelve -1 si x == y devuelve 0 si x> y devuelve 1
exp (x)	Devuelve e elevado a la potencia de x (e x), como math.exp (1) devuelve 2.718281828459045
piso (x)	Devuelve el número entero redondeado, como Math.floor (4,9) devuelve 4
log (x)	Como Math.log (Math.E) devuelve 1,0, Math.log (100,10) devuelve 2.0
log10 (x)	Devuelve el logaritmo en base 10 de x, tales math.log10 (100) devuelve 2.0
max (x1, x2, ...)	El rendimiento máximo para un parámetro dado, la secuencia de parámetros.
min (x1, x2, ...)	Devuelve el valor mínimo para un determinado parámetro, la secuencia de parámetros.
round (x [n])	Devuelve el valor redondeado de float x, como el valor de n dado representa el redondeo al número de cifras decimales.

Tipo números – Funciones de números al azar

Función	Descripción
random ()	Para obtener un número float entre 0 y 1.
randrange ([inicio,] dejar [, incremento])	Para obtener un número aleatorio dentro de la colección base incrementos de rango especificado, por defecto de base a 1
randint()	Para obtener un número entero entre 2 números.
choice()	
shuffle (LST)	Todos los elementos de una secuencia en orden aleatorio

Ejercicio: Calcular horas y minutos

Realiza un programa que reciba una cantidad de minutos y muestre por pantalla a cuantas horas y minutos corresponde. Por ejemplo: 1000 minutos son 16 horas y 40 minutos.

Ejercicio: Calcular precio final

Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuanto deberá pagar finalmente por su compra.

Ejercicio: Calcular nota final

Escribir un algoritmo para calcular la nota final de un estudiante, considerando que: por cada respuesta correcta 5 puntos, por una incorrecta -1 y por respuestas en blanco 0. Imprime el resultado obtenido por el estudiante.

Tipo cadenas de caracteres

Los literales de cadena en python están rodeados por comillas simples o comillas dobles.

texto = “Mensaje de prueba”

mensaje = “”” Esto es un mensaje muy largo.
Por eso he preferido dividirlo en varias líneas
para que sea legible.
“””

P	Y	T	H	O	N
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

Función	Descripción
len()	Devuelve la longitud de una cadena.
strip()	Elimina cualquier espacio en blanco desde el principio o el final.
lower()	Devuelve la cadena en minúsculas.
upper()	Devuelve la cadena en mayúsculas.
tittle()	Convierte en mayúscula la primera letra de cada palabra.
replace()	Reemplaza una cadena con otra cadena.
split()	Divide la cadena en subcadenas si encuentra instancias del separado.
find()	Busca una cadena y devuelve su posición en el caso de que exista.

Operaciones con cadenas de caracteres

- El operador suma para realizar concatenación de cadenas de caracteres:

```
a, b = "uno", "dos"
a + b
'unodos’
a + “ ” + b
‘uno dos’
```

- El operador multiplicación repite la cadena de caracteres N veces:

```
c = "tres"
c * 3
'trestrestres'
```

Formateo de cadenas de caracteres

- El operador % entonces substituye la frase ‘%tipodato’ con cero o mas elementos del tipo de datos especificado:

```
tipo_calculo = "raíz cuadrada de dos"
valor = 2**0.5
print "el resultado de %s es %f" % (tipo_calculo, valor)
el resultado de raíz cuadrada de dos es 1.414214
```

%c = str, simple carácter.
%s = str, cadena de carácter.
%d = int, enteros.
%f = float, coma flotante.
%o = octal.
%x = hexadecimal.

- Para obtener el valor con 8 dígitos después de la coma:

```
tipo_calculo = "raíz cuadrada de dos"
valor = 2**0.5
print "el resultado de %s es %.8f" % (tipo_calculo, valor)
el resultado de raíz cuadrada de dos es 1.41421356
```

```
print("%d %f %s" % (2.5,2.5,2.5))
2 2.500000 2.5
```

f-strings

```
print(f”Hola {nombre}”)
2 2.500000 2.5
```

Caracteres de escape

Secuencia Escape	Significado
\newline	Ignorado
\\	Backslash (\)
\'	Comillas simple (')
\"	Comillas doble (")
\a	Bell ASCII (BEL)
\b	Backspace ASCII (BS)
\f	Formfeed ASCII (FF)
\n	Linefeed ASCII (LF)
\N{name}	Carácter llamado *name* en base de datos Unicode (Solo Unicode)
\r	Carriage Return ASCII (CR)
\t	Tabulación Horizontal ASCII (TAB)

Ejercicio: Mostrar iniciales de una persona

Pedir el nombre y los dos apellidos de una persona y mostrar las iniciales en mayúsculas.

Tipo booleanos

TRUE o FALSE

print(10 > 9) True

print(10 == 9) False

print(10 < 9) False

```
>>> bool(25)
True
>>> bool(-9.5)
True
>>> bool("abc")
True
>>> bool((1, 2, 3))
True
>>> bool([27, "octubre", 1997])
True
>>> bool({27, "octubre", 1997})
True
```

Tipo listas

Una lista en Python es una estructura de datos formada por una secuencia de elementos que son **mutables**, es decir, que pueden ser modificados y pueden estar repetidos.

Ejemplo:

countries = ["Mexico", "Colombia", "Uruguay“, “Panama”]

countries = [["Mexico", 40], ["Colombia", 20], ["Uruguay“, 60], [“Panama”, 10]]

Método	Descripción
append()	Agrega un elemento al final de la lista.
clear()	Elimina todos los elementos de la lista.
copy()	Devuelve una copia de la lista.
count()	Devuelve el número de elementos con el valor especificado.
extend()	Agregue los elementos de una lista (o cualquier otro iterable) al final de la lista actual.
index()	Devuelve el índice del primer elemento con el valor especificado.
insert()	Agrega un elemento en la posición especificada
pop()	Elimina el elemento en la posición especificada.
remove()	Elimina el elemento con el valor especificado.
reverse()	Invierte el orden de la lista.
sort()	Invierte el orden de la lista.

Tipo tuplas

Una tupla en Python es una estructura de datos formada por una secuencia de elementos que son **inmutables**, es decir, que no pueden ser modificados.

Ejemplo:

countries = ("Mexico", "Colombia", "Uruguay“, “Panama”)

countries = (["Mexico", 40], ["Colombia", 20], ["Uruguay“, 60], [“Panama”, 10])

Método	Descripción
count()	Devuelve el número de ocurrencias de un valor en la tuple.
index()	Busca un valor en la tuple y devuelve su posición.

Tipo conjuntos

Un conjunto es una colección no ordenada de objetos únicos.

Ejemplo:

countries = {"Mexico", "Colombia", "Uruguay“, “Panama”}

Método	Descripción
add()	Añade un elemento al conjunto.
clear()	Elimina todos los elementos del conjunto.
copy()	Devuelve una copia del conjunto.
difference()	Devuelve un conjunto con los elementos distintos de dos o más conjuntos.
difference_update()	Elimina los elementos existentes en otro conjunto.
discard()	Elimina el elemento especificado.
pop()	Elimina el último elemento del conjunto.
remove()	Removes the specified element
update()	Actualiza el conjunto con la union de otro/s conjunto/s.

Tipo diccionarios

Los diccionarios son una colección **desordenada** de datos **indexados**, que pueden ser **modificados**.

Ejemplo:

```
diccionario = {
    'nombre' : 'Carlos',
    'edad' : 22,
    'cursos': ['Python','Django','JavaScript']
    }

print(diccionario)
```

Método	Descripción
clear()	Elimina todos los elementos.
copy()	Devuelve una copia del diccionario.
fromkeys()	Returns a dictionary with the specified keys and value
get()	Devuelve el valor de la clave indicada.
items()	Devuelve una lista de valores: clave – valor.
keys()	Devuelve una lista con las claves.
pop()	Elimina el elemtno con la clave indicada.
popitem()	Elimina el ultimo registro insertado.
values()	Devuelve todos los valores del diccionario.

Tipo diccionarios: diccionarios anidados

Ejemplo:

```
peliculas_dc = {
   5 : {
      "titulo": "Batman vs Superman:
Dawn of Justice",
      "lanzamiento": 2016,
      "nota": 6.5
   },
   6 : {
      "titulo": "Man of steel",
      "lanzamiento": 2013,
      "nota": 6
   }
}
```

Método	Descripción
clear()	Elimina todos los elementos.
copy()	Devuelve una copia del diccionario.
fromkeys()	Returns a dictionary with the specified keys and value
get()	Devuelve el valor de la clave indicada.
items()	Devuelve una lista de valores: clave – valor.
keys()	Devuelve una lista con las claves.
pop()	Elimina el elemtno con la clave indicada.
popitem()	Elimina el ultimo registro insertado.
values()	Devuelve todos los valores del diccionario.

Conversión de tipos

Función	Descripción
int(x [,base])	Convierte x a integer. base especifica la base si x es una string.
long(x [,base])	Convierte x a long integer. base la base si x es una string.
float(x)	Convierte x a un número de coma flotante.
complex(real [,imag])	Crea un número complejo.
str(x)	Convierte un objeto x a una representación tipo cadena.
repr(x)	Convierte un objeto x a una expression tipo cadena.
eval(str)	Evalúa un string y devuelve un objeto.
tuple(s)	Convierte s a una tupla.
list(s)	Convierte s a una lista.

Conversión de tipos

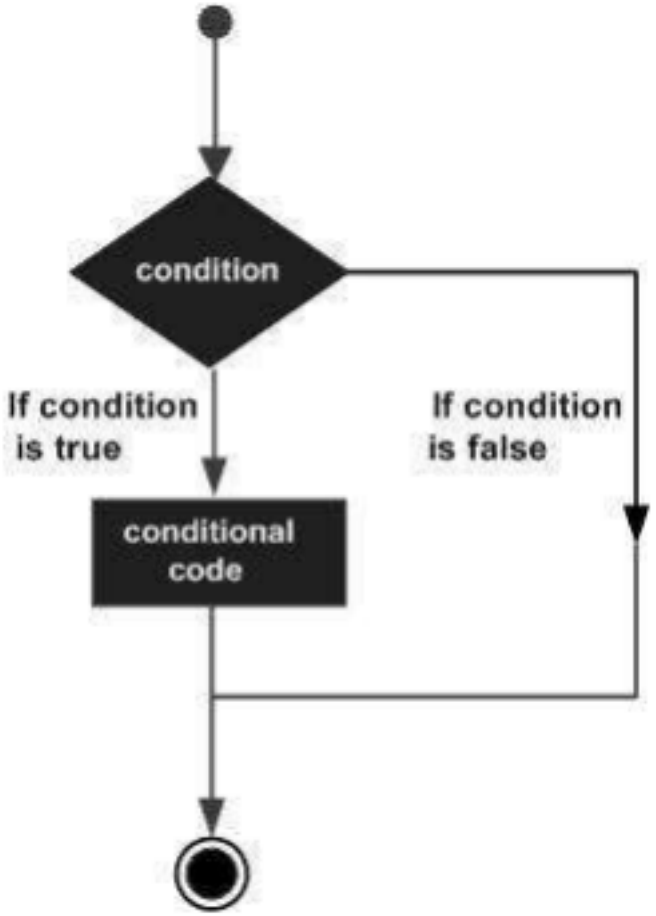
Función	Descripción
set(s)	Convierte s a un set.
dict(d)	Crea un diccionario. d debe ser una secuencia de (key,value) tuplas.
frozenset(s)	Convierte s a un frozenset.
chr(x)	Convierte un integer a un carácter.
unichr(x)	Convierte un integer a un carácter Unicode.
ord(x)	Convierte un integer a su correspondiente valor integer.
hex(x)	Convierte un integer a una cadena hexadecimal.
oct(x)	Convierte un integer a una cadena octal.

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

ESTRUCTURAS DE CONTROL

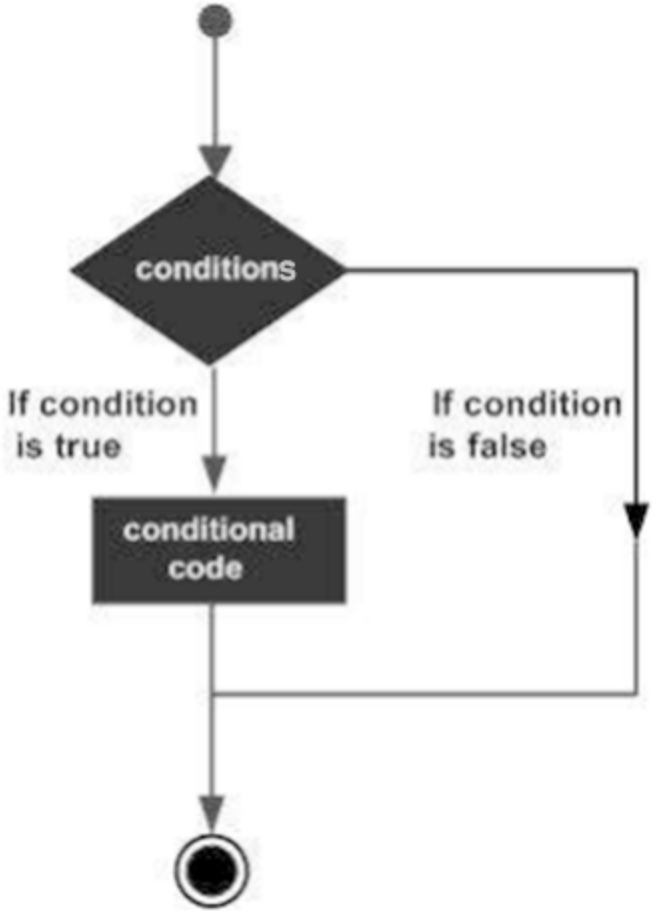
Python

Condicional if



Sintaxis	Ejemplo
if condición: bloque de código	if edad >= 18: print(‘Eres un adulto.’)

Condicional if (varias condiciones)



Sintaxis	Ejemplo
if condición1 and condición2: bloque de código	if edad > 10 and edad < 18: print(‘Eres un adolescente.’)
if condición1 or condición2: bloque de código(s)	if edad < 18 or edad < 65: print(‘Tienes un descuento.’)

Ejercicio con operadores de comparación

Siendo a = 10 y b = 20, comparar utilizando todos los operadores de comparación:

a = 20

b = 10

a y b son distintos

a y b son distintos

a no es menor que b

a es mayor que b

a no es menor o igual que b

a es mayor o igual que b

Ejercicio con operadores de membresía

Siendo a = 20, b = 10 y nums = [1, 2, 3, 4, 5] comprobar si a y b están en nums, y si a/b está en nums.

a = 20

b = 10

nums = [1, 2, 3, 4, 5]

a no está en nums

b no está en nums

El resultado de a / b sí está en nums

Ejercicio con operadores de identidad

Siendo a = 20 y b = 10 utilizar los operadores de identidad.

a = 20

b = 10

a y b no son iguales

a es distinto de b

Ejercicio: calcular impuestos (if)

Preguntar por pantalla el salario anual y si el salario es superior a 30.000€ calcular el porcentaje de impuestos a pagar siendo este de un 30%. Si el salario no es superior a 30.000€ calcular el porcentaje de impuestos a pagar siendo este de un 10%.

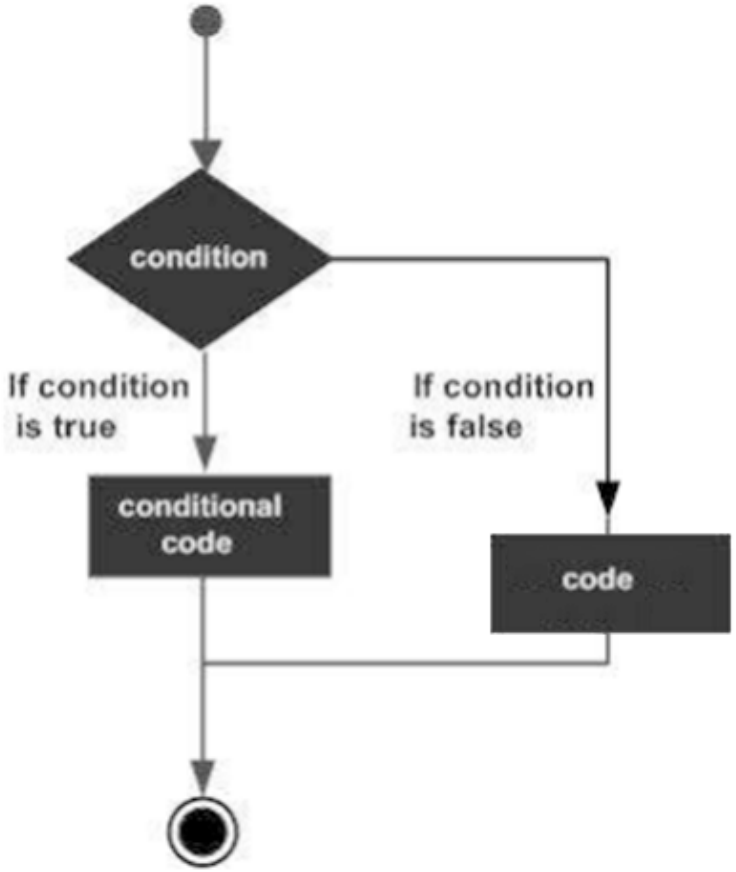
Ejemplo de ejecución del programa:

Introduce tu salario anual: 40000

Impuestos: 12000.0

Salario real: 28000.0

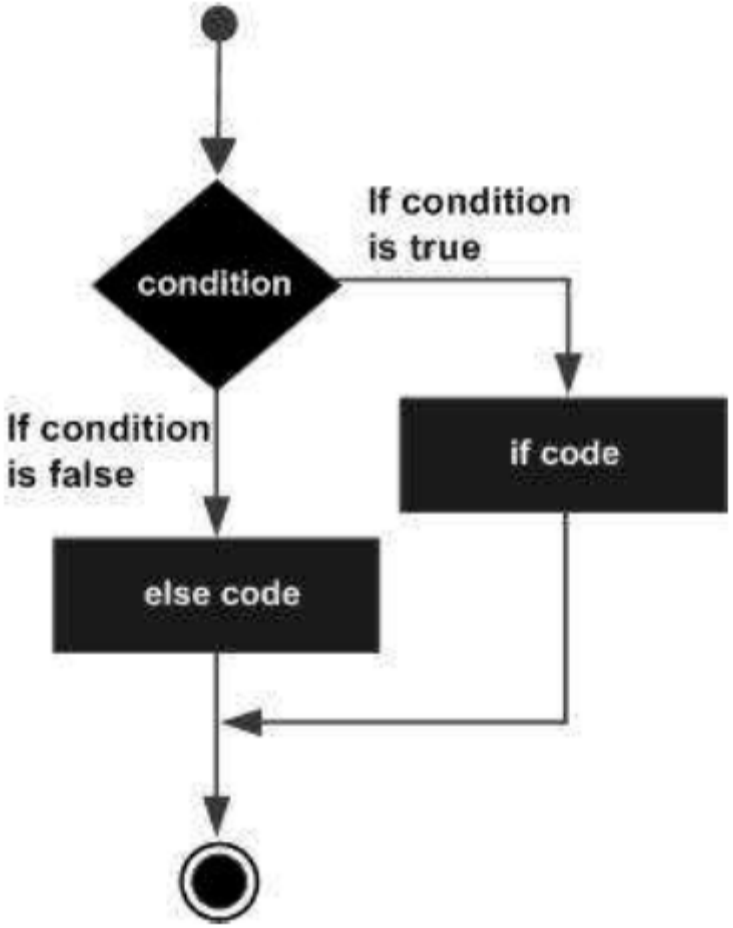
Condicional if…else



Sintaxis	Ejemplo

if condición:
  bloque de código
else:
  bloque de código
	if edad > 17:
	  print(‘Eres un adulto.’)
	else:
	  print(‘No eres un adulto.’)

Condicional if…elif…else



Sintaxis	Ejemplo

if condición:
  bloque de código
elif condición:
  bloque de código
else:
  bloque de código
	if edad>17:
	  print(‘Eres un adulto.’)
	elif edad > 10:
	  print(‘Eres un adolescente.’)
	else:
	  print(‘Eres un niño.’)

Ejercicio: Día de la semana

Realiza un programa que pida el día de la semana (del 1 al 7) y escriba el día correspondiente. Si introducimos otro número nos da un error.

Ejercicio: Número de días del mes

Escribe un programa que pida un número entero entre uno y doce e imprima el número de días que tiene el mes correspondiente.

Ejercicio: Nota de un alumno

```
Proceso notas
        Definir nota como float;
        Escribir "Dime tu nota:";
        Leer nota;
        Segun nota Hacer
                de 0 a <5: Escribir "Suspenso";
                5: Escribir "Suficiente";
                6,7: Escribir "Bien";
                8: Escribir "Notable";
                9,10: Escribir "Sobresaliente";
                De Otro Modo:
                        Escribir "Nota incorrecta";
        FinSegun
FinProceso
```

Ejercicio: Viaje de estudios

El director de una escuela está organizando un viaje de estudios, y requiere determinar cuánto debe cobrar a cada alumno y cuánto debe pagar a la compañía de viajes por el servicio. La forma de cobrar es la siguiente: si son 100 alumnos o más, el costo por cada alumno es de 65 euros; de 50 a 99 alumnos, el costo es de 70 euros, de 30 a 49, de 95 euros, y si son menos de 30, el costo de la renta del autobús es de 4000 euros, sin importar el número de alumnos. Realice un algoritmo que permita determinar el pago a la compañía de autobuses y lo que debe pagar cada alumno por el viaje.

Ejercicio: Compañía telefónica

La política de cobro de una compañía telefónica es: cuando se realiza una llamada, el cobro es por el tiempo que ésta dura, de tal forma que los primeros cinco minutos cuestan 1 euro, los siguientes tres, 80 céntimos, los siguientes dos minutos, 70 céntimos, y a partir del décimo minuto, 50 céntimos. Además, se carga un impuesto de 3 % cuando es domingo, y si es otro día, en turno de mañana, 15 %, y en turno de tarde, 10 %. Realice un algoritmo para determinar cuánto debe pagar por cada concepto una persona que realiza una llamada.

Ejercicio: juego de dados

Escriba un programa que simule un juego en el que dos jugadores (Usuario1 y Usuario2) tiran un dado. El que saque el valor más alto, gana. Si la puntuación coincide, empatan.

Genera los números aleatorios mediante la función: random.randrange

Ejercicio: calcular impuestos

Preguntar por pantalla el salario anual y si el salario es superior a 30.000€ calcular el porcentaje de impuestos a pagar siendo este de un 30%. Si el salario es superior a 20.000€ pero inferior a 30.000€ calcular el porcentaje de impuestos a pagar siendo este de un 20%. Si el salario es inferior a 20.000€ calcular el porcentaje de impuestos a pagar siendo este de un 10%.

Ejercicio: Calcular precio de la entrada

Escribir un programa para una empresa que tiene salas de juegos para todas las edades y quiere calcular de forma automática el precio que debe cobrar a sus clientes por entrar.

El programa debe preguntar al usuario la edad del cliente y mostrar el precio de la entrada.

Si el cliente es menor de 4 años puede entrar gratis, si tiene entre 4 y 18 años debe pagar 5€ y si es mayor de 18 años, 10€.

Ejercicio: Juguetería

Una juguetería tiene mucho éxito en dos de sus productos: payasos y muñecas. Suele hacer venta por correo y la empresa de logística les cobra por peso de cada paquete así que deben calcular el peso de los payasos y muñecas que saldrán en cada paquete a demanda. Cada payaso pesa 112 g y cada muñeca 75 g. Escribir un programa que lea el número de payasos y muñecas vendidos en el último pedido y calcule el peso total del paquete que será enviado.

Ejercicio: Calcular balance

Imagina que acabas de abrir una nueva cuenta de ahorros que te ofrece el 4% de interés al año. Estos ahorros debido a intereses, que no se cobran hasta finales de año, se te añaden al balance final de tu cuenta de ahorros. Escribir un programa que comience leyendo la cantidad de dinero depositada en la cuenta de ahorros, introducida por el usuario. Después el programa debe calcular y mostrar por pantalla la cantidad de ahorros tras el primer, segundo y tercer años. Redondear cada cantidad a dos decimales.

Utilizar la función round().

Ejercicio: Calcular bonificación anual

En una determinada empresa, sus empleados son evaluados al final de cada año. Los puntos que pueden obtener en la evaluación comienzan en 0.0 y pueden ir aumentando, traduciéndose en mejores beneficios. Los puntos que pueden conseguir los empleados pueden ser 0.0, 0.4, 0.6 o más, pero no valores intermedios entre las cifras mencionadas. A continuación se muestra una tabla con los niveles correspondientes a cada puntuación. La cantidad de dinero conseguida en cada nivel es de 2.400€ multiplicada por la puntuación del nivel.

Nivel	Puntuación
Inaceptable	0.0
Aceptable	0.4
Meritorio	0.6

Escribir un programa que lea la puntuación del usuario e indique su nivel de rendimiento, así como la cantidad de dinero que recibirá el usuario.

Ejercicio: Pizzería

La pizzería Bella Napoli ofrece pizzas vegetarianas y no vegetarianas a sus clientes. Los ingredientes para cada tipo de pizza aparecen a continuación.

- Ingredientes vegetarianos: Pimiento y Tofu.
- Ingredientes no vegetarianos: Peperoni, Jamón y Salmón.

Escribir un programa que pregunte al usuario si quiere una pizza vegetariana o no, y en función de su respuesta le muestre un menú con los ingredientes disponibles para que elija.

Solo se puede elegir un ingrediente además de la mozzarella y el tomate que están en todas la pizzas. Al final se debe mostrar por pantalla si la pizza elegida es vegetariana o no y todos los ingredientes que lleva.

Ejercicio: Generar contraseña

Escribe un generador de contraseñas aleatorias. Cada contraseña tiene que tener un tamaño de 8 caracteres.

Utiliza la función random.sample.

Ejercicio:

Escribe un programa que a partir de una lista con 10 elementos numéricos, elimine los duplicados y muestre los elementos únicos por pantalla, indicando previamente la cantidad de elementos de la lista y la cantidad una vez que se hayan eliminado los duplicados. Así mismo, se debe mostrar el elemento mayor y menor de la lista, y mostrar el valor de la suma de todos los elementos y el valor medio. Finalmente, en el caso de que el valor medio sea mayor que 30 se mostrará un mensaje, en el caso de que sea mayor 20 se mostrará otro mensaje, en el caso de que sea mayor que 10 se mostrará otro mensaje y finalmente si es menor o igual que 10 se mostrará otro mensaje.

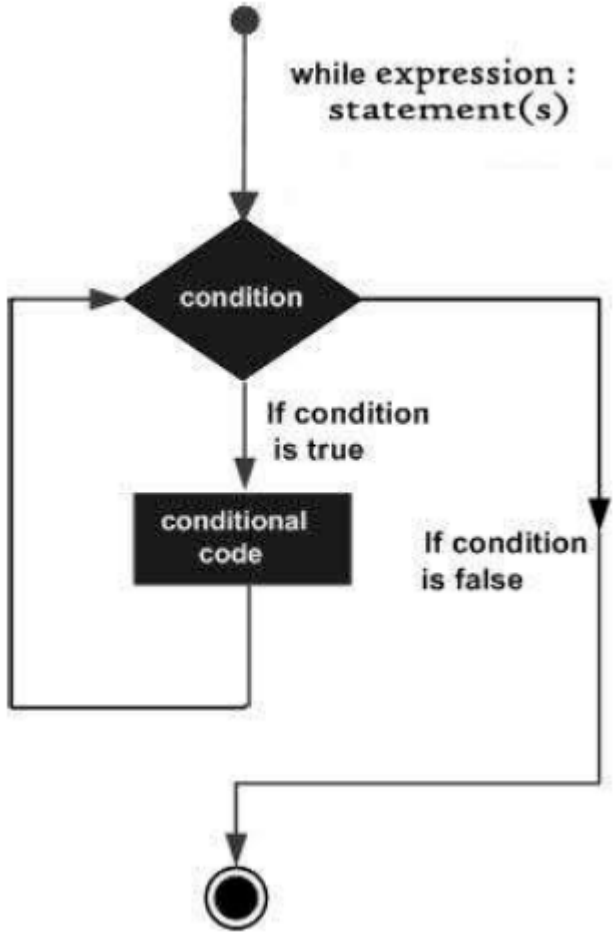
```
La lista inicial tiene 20 elementos.
Una vez eliminados los elementos duplicados la lista tiene 14 elementos.
El número mayor de la lista es 89
El número menor de la lista es 1
La suma de los elementos de la lista es: 277
La media de los elementos de la lista es: 13.85
La media es mayor que 10.
```

Ejercicio:

Escribe un programa que almacene los módulos que se imparten en un curso en una lista, solicite por pantalla 2 nuevos módulos # a añadir, los muestre por pantalla ordenados, solicite por pantalla 1 módulo a eliminar, contabilice el número de módulos del curso y los muestre por pantalla, y en el caso de que el módulo de Python esté incluido, muestre un mensaje que diga: “Que la fuerza te acompañe”.

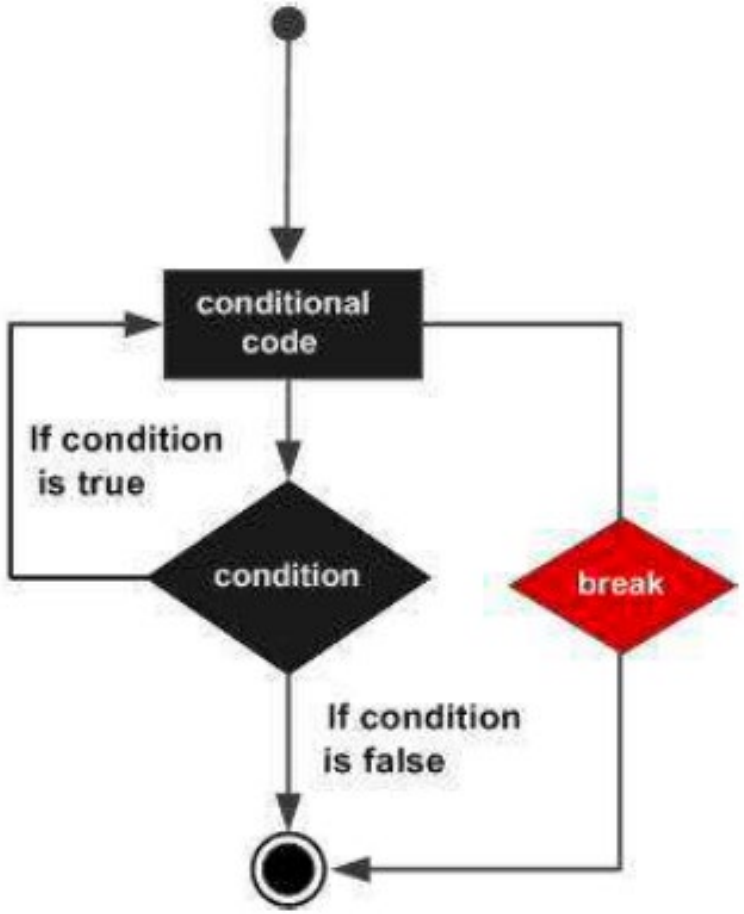
```
        CURSO DE ESPECIALISTA EN CIBERSEGURIDAD

Las asignaturas del curso son ['Introducción a la ciberseguridad', 'Python', 'Ba
stionado de sistemas y gestión de la seguridad', 'Ciberinteligencia, cibervigila
ncia y ciberterrorismo', 'Seguridad de sistemas', 'Seguridad gestionada,
threathuntig y pentest', 'Análisis de datos', 'Marco normativo y regulación']

Introduce el primer módulo a añadir en la lista.
análisis forense
Introduce el segundo módulo a añadir en la lista.
reversing

['Análisis de datos', 'Análisis forense', 'Bastionado de sistemas y gestión de
la seguridad', 'Ciberinteligencia, cibervigilancia y ciberterrorismo',
'Introducción a la ciberseguridad', 'Marco normativo y regulación', 'Python',
'Reversing', 'Seguridad de sistemas', 'Seguridad gestionada, threathuntig y
pentest']

Introduce el número del módulo que no quieras cursar.

1.  Análisis de datos
2.  Análisis forense
3.  Bastionado de sistemas y gestión de la seguridad
4.  Ciberinteligencia, cibervigilancia y ciberterrorismo
5.  Introducción a la ciberseguridad
6.  Marco normativo y regulación
7.  Python
8.  Reversing
9.  Seguridad de sistemas
10.  Seguridad gestionada, threathuntig y pentest

Módulo: 6
Se procede a eliminar el módulo de Marco normativo y regulación.
La lista tiene 9 elementos.
Python es uno de los módulos que se van a impartir: "Que la fuerza te acompañe".
```

Bucle while



Sintaxis	Ejemplo
while expression: statement(s)	cant = 0 while (cant < 9): print ('Cant es:', cant) cant = cant + 1 print ('Fin del programa.')

break



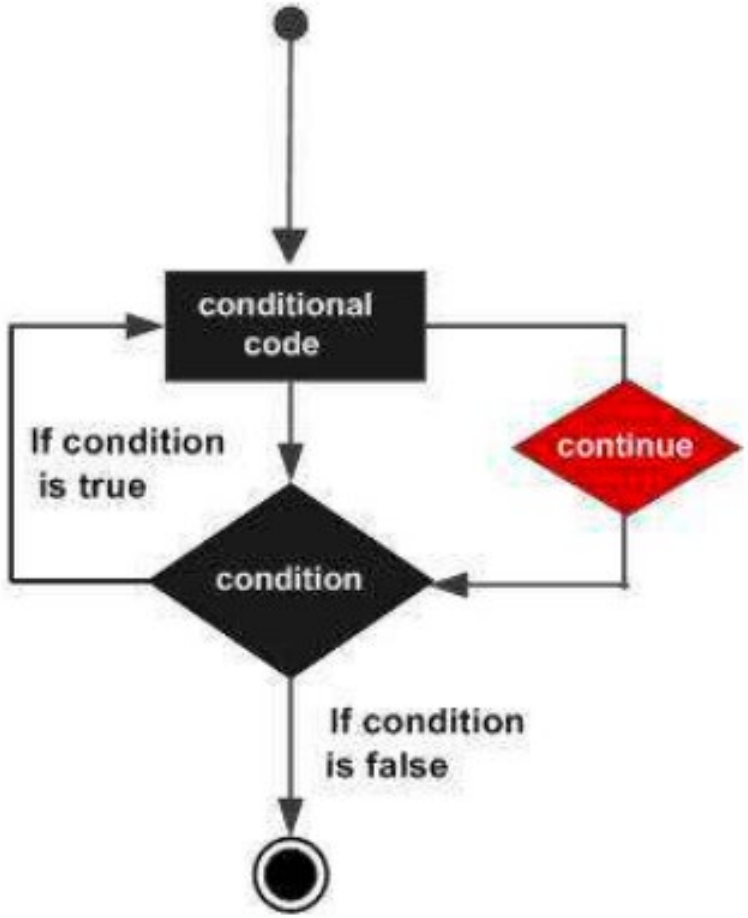
Sintaxis	Ejemplo

break	cant = 0
	while (cant < 9):
	  print (‘Cant es:’, cant)
	  cant = cant + 1
	  if cant == 5:
	  break
	print (‘Fin del programa.’)

Ejercicio:

Realizar un ejemplo de menú, donde podemos escoger entre 5 opciones hasta que seleccionamos la opción de “5. Salir”. Al elegir cada una de las opciones debe mostrar un mensaje.

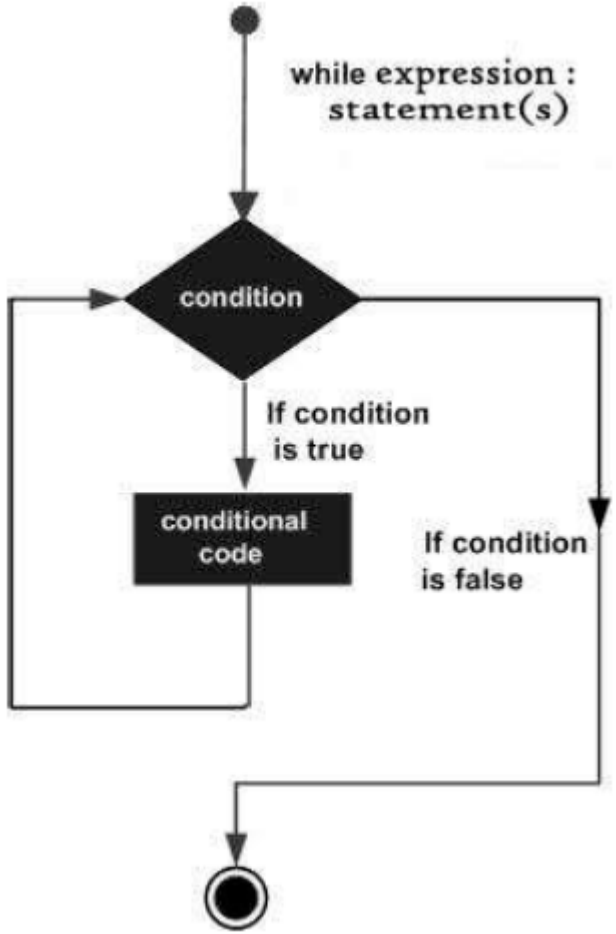
continue



Sintaxis	Ejemplo

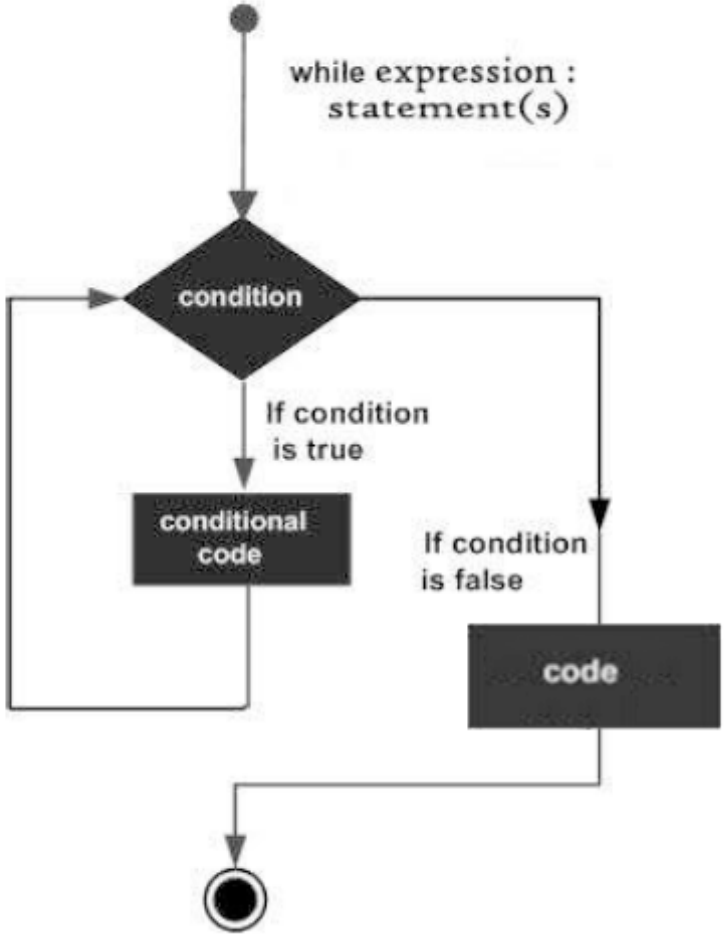
continue	cant = 0
	while (cant < 9):
	  print (‘Cant es:’, cant)
	  cant = cant + 1
	  if cant == 9:
	  cant = 0
	  continue
	print (‘Fin del programa.’)

Bucle while infinito



Sintaxis	Ejemplo
while expression: statement(s)	valor = 1 while valor == 1 : num = int(input(“Introduce un número:")) print (“Has introducido: ", num) print (“Fin del programa.”)

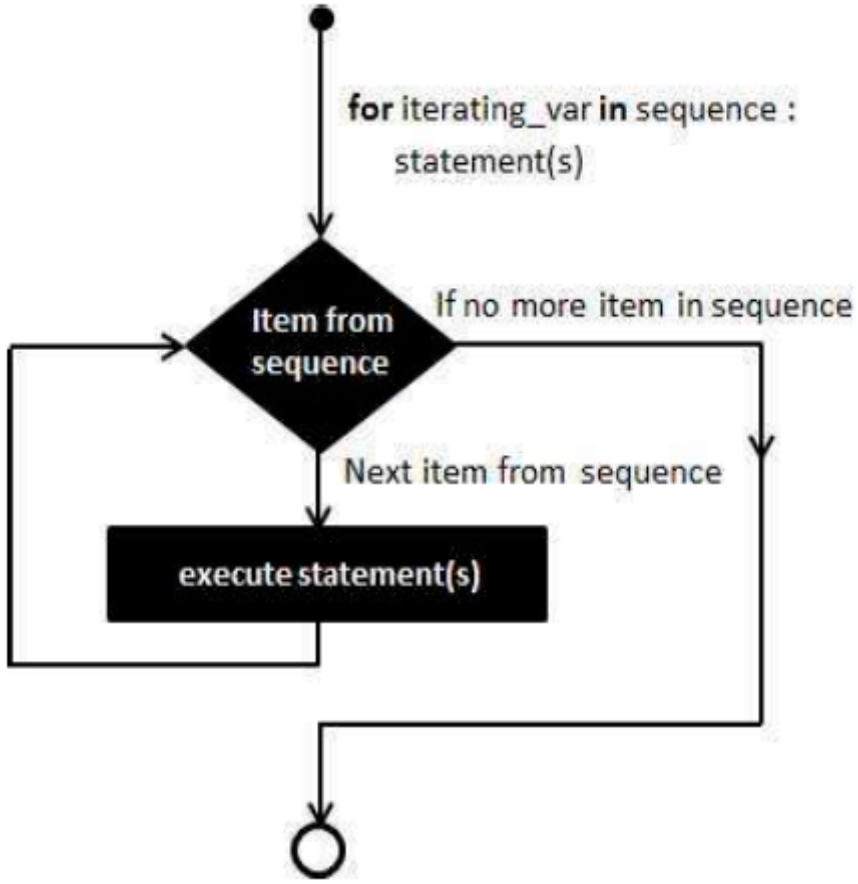
Bucle while…else



Sintaxis	Ejemplo

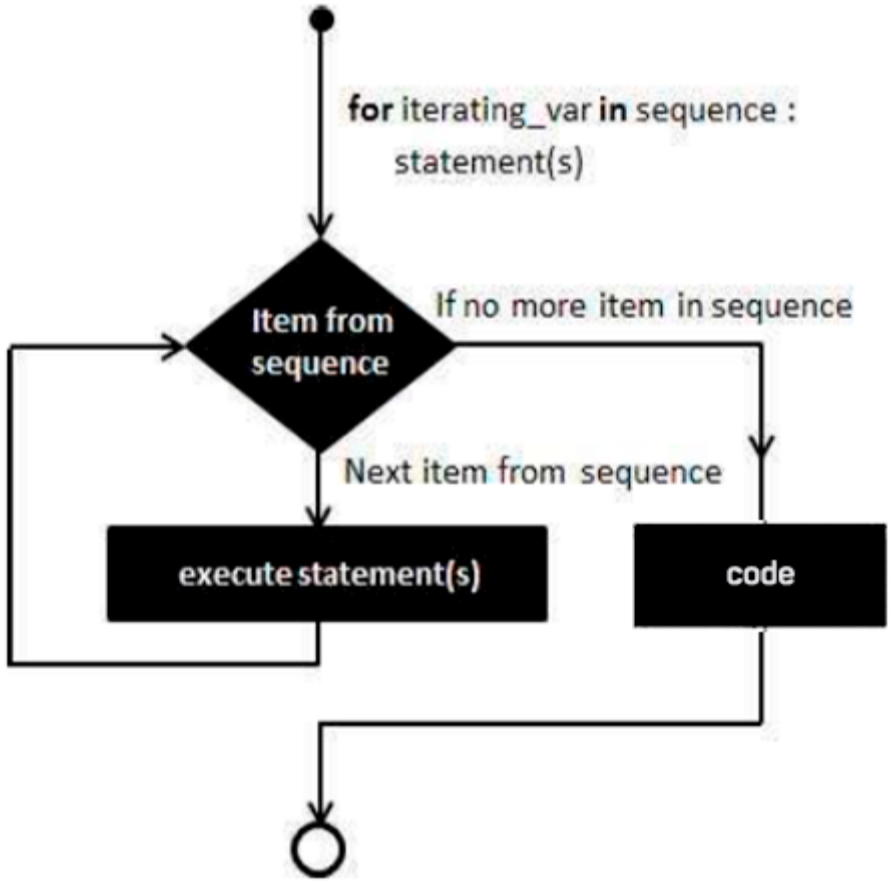
while expression:
  statement(s)
else:
  statements(s)
	cant = 0
	while (cant < 9):
	  print (‘Cant es:’, cant)
	  cant = cant + 1
	else:
	  print (‘Fin del programa.’)

Bucle for



Sintaxis	Ejemplo
for iterating_var in sequence: statement(s)	for num in range(10): # recorre del 0 al 9 print(num)

Bucle for…else



Sintaxis	Ejemplo

for iterating_var in sequence:
  bloque de código
else:
  bloque de código
	for num in range(10): #
	recorre del 0 al 9
	  print(num)
	else:
	  print (“Fin del bucle for”)

Instrucción pass

Sintaxis	Ejemplo

pass	for letra in 'Python’:
	  if letra == 'h’:
	  pass
	  print ('Bloque pass’)
	  print ('Letra actual :', letra)
	print ("Fin")

Ejercicio: Calcular el factorial de un número

Crea una aplicación que pida un número y calcule su factorial (El factorial de un número es el producto de todos los enteros entre 1 y el propio número y se representa por el número seguido de un signo de exclamación. Por ejemplo 5! = 1x2x3x4x5=120).

Ejercicio: Mostrar años cumplidos

Escribir un programa que pregunte al usuario su edad y muestre por pantalla todos los años que ha cumplido (desde 1 hasta su edad).

Utilizar la función: range

Ejercicio: Mostrar impares hasta número elegido

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número separados por comas.

Ejercicio: Sumar primeros N números

Escribe un programa que muestre la sumatoria de todos los números entre el 0 y un número introducido por el usuario por pantalla.

Ejercicio: Contar vocales totales

Escribe un programa que, dada una frase por el usuario, muestre la cantidad total de vocales (tanto mayúsculas como minúsculas) que contiene.

Ejercicio: Contar cantidad cada vocal

Escribe un programa que pida al usuario una frase y muestre por pantalla el número de veces que contiene cada vocal.

Ejercicio: Calcular media

Escribe un programa que pregunte por una muestra de números, separados por comas, los guarde en una tupla y muestre por pantalla su media. Posteriormente, que solicite otro número por pantalla, lo añada a la tupla y vuelva a calcular la media.

Ejercicio: Números de la lotería

Escribe un programa que pregunte al usuario los números ganadores de la lotería primitiva (6), los almacene en una lista y los muestre por pantalla ordenados de menor a mayor.

Ejercicio: Mostrar elementos lista

Pide por teclado los números de una lista de uno en uno hasta que se introduzca la S de salir. Introducir en la lista únicamente aquellos números que no hayan sido previamente introducidos. En el caso de que haya sido previamente introducido, el programa debe indicar que el número ya está en la lista, indicando en qué posición.

Ejercicio: Login

Escribí un programa que solicite ingresar un nombre de usuario y una contraseña. Si el nombre es “admin” y la contraseña es “123456”, mostrar en pantalla “Usuario y contraseña correctos, aunque deberías utilizar una contraseña más segura.”. Si el nombre o la contraseña no coinciden, mostrar “Acceso denegado” y volver a preguntar por el nombre de usuario y la contraseña. Realizar el programa con una única condición.

Ejercicio: Tabla de multiplicar de un número.

Realizar una algoritmo que muestre la tabla de multiplicar de un número introducido por teclado.

Ejercicio: Tabla de multiplicar de los números 1 a 10.

Realizar una algoritmo que muestre la tabla de multiplicar de todos los números del 1 al 10.

Ejercicio: Cifrado César

Realiza un programa que solicite un texto por pantalla y lo cifre utilizando el cifrado César.

Ejercicio: Cifrado César

Realiza un programa que solicite un texto por pantalla y lo descifre utilizando el cifrado César.

Descifrar los siguientes mensajes:

QS HINIMW XMXIVI GSQ GEFIDE – Desp. 4
IS MUMTPÑW UP IÑCI – Desp. 8

Ejercicio: Adivinar un número entre 0 y 100

Crea una aplicación que permita adivinar un número. La aplicación genera un número aleatorio del 1 al 100. A continuación va pidiendo números y va respondiendo si el número a adivinar es mayor o menor que el introducido, además de los intentos que te quedan (tienes 10 intentos para acertarlo). El programa termina cuando se acierta el número (además te dice en cuantos intentos lo has acertado), si se llega al limite de intentos te muestra el número que había generado.

Ejercicio: piedra, papel o tijera

Escribe un programa que simule el juego piedra, papel, tijera para dos jugadores, tú y el ordenador.

Las reglas del juego son las siguientes:

- El ordenador elige opción aleatoria mediante la función: random.randrange
- El usuario elige opción a partir de un menú con 4 opciones a elegir: 1. Piedra, 2. Papel, 3. Tijera, 4. Salir.

- El jugador que ha sacado Piedra gana al jugador que ha sacado Tijera.
- El jugador que ha sacado Tijera gana al jugador que ha sacado Papel.
- El jugador que ha sacado Papel gana al jugador que ha sacado Piedra.

Ejercicio: Cantidad de pares e impares

Escribe un programa que pregunte cuántos números se van a introducir, pida esos números, y diga al final cuántos han sido pares y cuántos impares.

Ejercicio de la caja de una frutería

Vamos a crear un programa en python donde vamos a declarar un diccionario para guardar los precios de las distintas frutas. El programa pedirá el nombre de la fruta y la cantidad que se ha vendido y nos mostrará el precio final de la fruta a partir de los datos guardados en el diccionario. Si la fruta no existe nos dará un error. Tras cada consulta el programa nos preguntará si queremos hacer otra consulta.

Ejercicio con notas de alumnos

Codifica un programa en python que nos permita guardar los nombres de los alumnos de una clase y las notas que han obtenido. Cada alumno puede tener distinta cantidad de notas. Guarda la información en un diccionario cuya claves serán los nombres de los alumnos y los valores serán listas con las notas de cada alumno.

El programa pedirá el número de alumnos que vamos a introducir, pedirá su nombre e irá pidiendo sus notas hasta que introduzcamos un número negativo. Al final el programa nos mostrará la lista de alumnos y la nota media obtenida por cada uno de ellos. Nota: si se introduce el nombre de un alumno que ya existe el programa nos dará un error.

Ejercicio: Calcular salario y gasto total.

Una empresa les paga a sus empleados con base en las horas trabajadas en la semana. Realiza un algoritmo para determinar el sueldo semanal de N trabajadores y, además, calcule cuánto pagó la empresa por los N empleados.

Ejercicio: Entrenar multiplicaciones

Realiza un programa que pregunte aleatoriamente una multiplicación de dos números entre 1 y 9. El programa debe indicar si la respuesta ha sido correcta o no (en caso que la respuesta sea incorrecta el programa debe indicar cuál es la correcta). El programa preguntará 5 multiplicaciones, y al finalizar mostrará el número de aciertos * 2 para obtener la nota.

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

MANEJO DE ERRORES

Python

Manejando excepciones: try…except

```
while True:
    try:
        x = int(input("Introduce un número:"))
        break
    except ValueError:
        print ("Debes introducir un número")
```

Manejando excepciones: try…except

```
try:
    num = 5
    print (10/int(num))
except ValueError:
    print("No se puede convertir a entero")
except ZeroDivisionError:
    print("No se puede dividir por cero")
```

Manejando excepciones: try…except…finally

```
try:
    num = 5
    print (10/int(num))
except ValueError:
    print("No se puede convertir a entero")
except ZeroDivisionError:
    print("No se puede dividir por cero")
except Exception as mensaje_de_error:
    print(str(mensaje_de_error))
finally:
    print("Fin del programa")
```

Ejemplo: División de 2 números

Escribe un programa que solicite por pantalla 2 números y realice su división. Maneja los errores que se te ocurran que pueden producirse como por ejemplo al introducir un 0 en el segundo número.

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

FUNCIONES

Python

Funciones

```
def cuadrado(num):
    return num**2


num=int(input("Dime un número:"))
total = cuadrado(num)
print(f"El resultado de {num}² es {total}")
```

Ejercicio: Validar email

Solicitar al usuario que ingrese su dirección email. Imprimir un mensaje indicando si la dirección es válida o no, valiéndose de una función para decidirlo. Una dirección se considerará válida si contiene el símbolo "@".

Ejercicio: Calcular cuantía de la factura

Escribir una función que calcule el total de una factura tras aplicarle el IVA. La función debe recibir la cantidad sin IVA y el porcentaje de IVA a aplicar, y devolver el total de la factura. Si se invoca la función sin pasarle el porcentaje de IVA, deberá aplicar un 21%.

Ejercicio: Calcular temperatura media

Crear una función que calcule la temperatura media de un día a partir de la temperatura máxima y mínima. Crear un programa principal, que utilizando la función anterior, vaya pidiendo la temperatura máxima y mínima de cada día y vaya mostrando la media. El programa pedirá el número de días que se van a introducir.

Ejercicio: Obtener número máximo y mínimo

Crea una función “obtenerMaxMin” que recibe una lista con valores numéricos y devuelve el valor máximo y el mínimo. Crea un programa que genere 10 números aleatorios entre 1 y 100 y muestre el máximo y el mínimo, utilizando la función anterior.

Ejercicio: Login en el sistema

Crear una función llamada “Login”, que recibe un nombre de usuario y una contraseña y te devuelve Verdadero si el nombre de usuario es “admin” y la contraseña es “123456”. Además recibe el número de intentos que se ha intentado hacer login y si no se ha podido hacer login incremente este valor.

Crear un programa principal donde se pida un nombre de usuario y una contraseña y se intente hacer login, solamente tenemos tres oportunidades para intentarlo.

Ejercicio: Validar DNI

El DNI (Documento Nacional de Identidad) en España está formada por 8 números y una letra. La letra nos sirve para verificar que el número es correcto, por lo tanto la letra se calcula a partir del número. Busca información de cómo se realiza el calculo y crea una función CalcularLetra que recibe un número y devuelva la letra que le corresponde.

La función anterior la podemos utilizar para crear una nueva función ValidarDNI que recibe un DNI (cadena de caracteres con 8 números y una letra) que valida el DNI, es decir comprueba si la letra del DNI ces igual a la letra calculada a partir del número.

Realiza un programa principal que lea un DNI y valide que es correcto (se debe comprobar también que tiene 9 caracteres).

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

PAQUETES Y MÓDULOS

Python

Paquetes y módulos

`module.py`

```
def hi():
    print("Hello world!")
```

`my_script.py`

```
import module
module.hi()
```

en un intérprete

```
>>> from module import hi
>>> hi()
# Hello world!
```

`package` carpetas

- `__init__.py`
- `dog.py`
- `hi.py`

`__init__.py`

```
from package.dog import woof
from package.hi import hi
```

`dog.py`

```
def woof():
    print("WOOF!!!")
```

`hi.py`

```
def hi():
    print("Hello world!")
```

Módulo Datetime: Fechas

```
import datetime
import locale

# Establezco como localización geográfica España
locale.setlocale(locale.LC_ALL, 'esp')

# Obtengo la fecha y hora actual.
fecha = datetime.datetime.now()

print("Fecha y Hora:", fecha) # Muestra fecha/hora
print("Fecha y Hora UTC:", fecha.utcnow())  # Muestra fecha/hora UTC

print("Año:", fecha.year)
print("Mes:", fecha.month)
print("Día:", fecha.day)
print("Hora:", fecha.hour)
print("Minuto:", fecha.minute)
print("Segundo:", fecha.second)
```

Módulo Datetime: Fechas

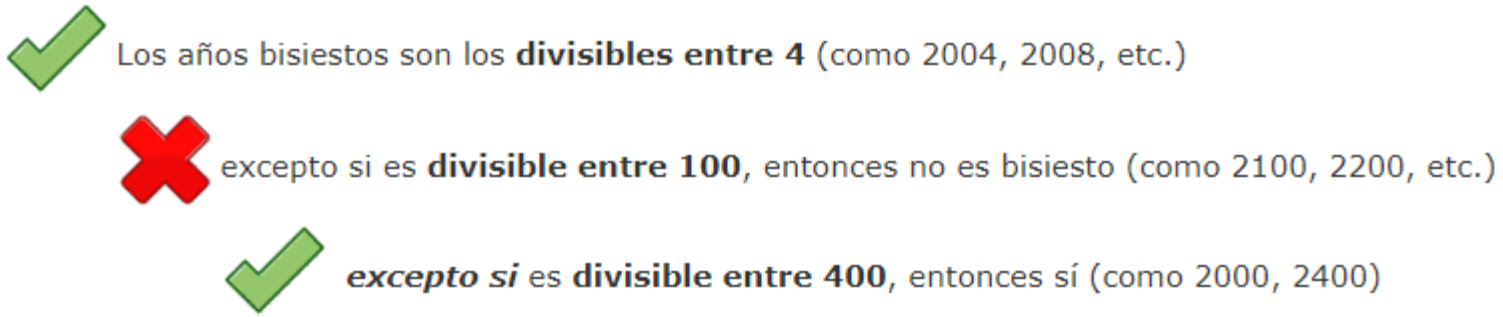
Variable	Descripción	Ejemplo
%a	día de la semana, versión corta	do.
%A	día de la semana, versión larga	domingo
%w	día de la semana como número de 0 a 6, 0 es domingo	0
%d	día del mes de 0 - 31	19
%b	nombre del mes, versión corta	jul.
%B	nombre del mes, versión larga	julio
%m	número del mes, de 1 a 12	7
%y	año, versión corta	20
%Y	año, versión larga	2020
%H	Hora 00-23	9
%I	Hora 00-12	9
%M	Minuto 00-59	23
%S	Segundo 00-59	32
%f	Micrsegundo 000000-999999	772892
%j	Día del año 001-366	201
%U	Semana del año, Domingo primer día de la semana, 00-53	29
%W	Semana del año, Lunes como primer día de la semana, 00-53	28
%c	Versión local de fecha y hora	19/07/2020 9:23
%x	Versión local de fecha y hora	19/07/2020
%X	Versión local de hora	9:23:32

Ejercicio:

Crea un programa que pregunte por pantalla un año y diga si fue bisiesto.

Un año es bisiesto si:

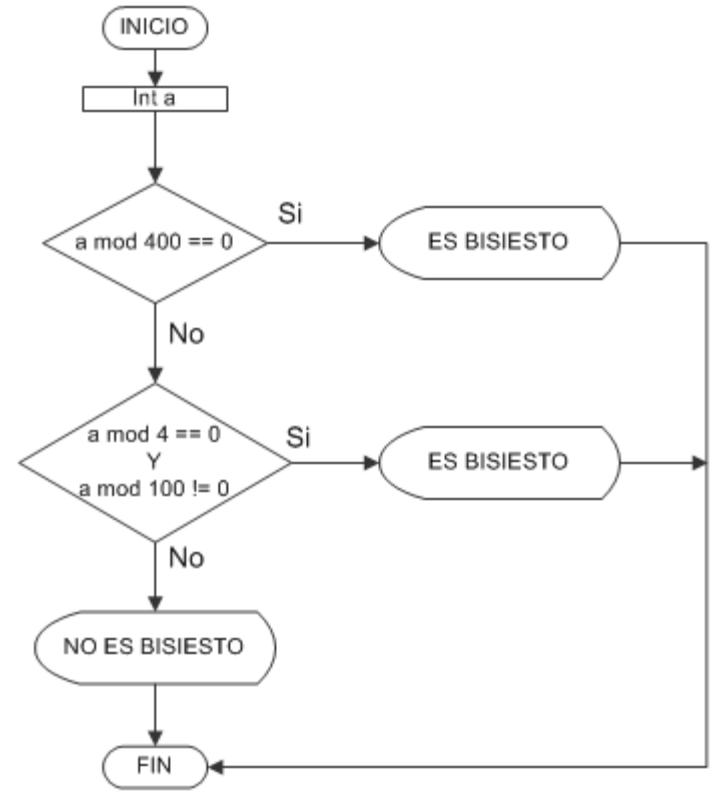
- Es divisible entre 4.
- No es divisible entre 100.
- Es divisible entre 400.



Los años bisiestos son los **divisibles entre 4** (como 2004, 2008, etc.)

excepto si es **divisible entre 100**, entonces no es bisiesto (como 2100, 2200, etc.)

excepto si es **divisible entre 400**, entonces sí (como 2000, 2400)



Módulo Requests

```
from urllib import request
    from urllib.error import URLError
    try:
        file = request.urlopen(url)
        content = file.read()
        print (content)
    except URLError:
        return('¡La url ' + url + ' no existe!')
```

Módulo webbrowser

```
import webbrowser

webbrowser.open('https://www.google.es')
```

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

FICHEROS

Python

Trabajar con ficheros

open() devuelve un objeto file, y comúnmente se utiliza con dos argumentos: open(nombre del archivo, modo)

- r: Abrir fichero para lectura. El puntero se posiciona al principio del fichero
- r+: Abrir fichero para lectura y escritura. El puntero se posiciona al principio del fichero
- w: Trunca a cero la longitud o crea un fichero de texto para escritura. El puntero se posiciona al principio del fichero
- w+: Abrir fichero para lectura y escritura. Si el fichero no existe, se crea, de lo contrario se trunca. El puntero se posiciona al principio del fichero
- a: Abrir fichero para lectura. Se creará el fichero si no existe. El puntero se posiciona al final del fichero.
- a+: Abrir fichero para lectura y escritura. Se creará el fichero si no existe. El puntero se posiciona al final del fichero.

Trabajar con ficheros

Se puede especificar si el fichero se abre en modo texto o en modo binario.

- t: Texto – Por defecto.
- b – Binario

```
f = open("fichero.txt")
f = open("fichero.txt", "rt")

with open('fichero.txt', 'r') as f:
```

Ejercicio: Guardar tabla de multiplicar de 1 número

Escribir una función que pida un número entero entre 1 y 10 y guarde en un fichero con el nombre tabla-n.txt la tabla de multiplicar de ese número, donde n es el número introducido.

Ejercicio:

Escribir una función que pida un número entero entre 1 y 10, lea el fichero tabla-n.txt con la tabla de multiplicar de ese número, donde n es el número introducido, y la muestre por pantalla. Si el fichero no existe debe mostrar un mensaje por pantalla informando de ello.

Ejercicio:

Escribir una función que pida dos números n y m entre 1 y 10, lea el fichero tabla-n.txt con la tabla de multiplicar de ese número, y muestre por pantalla la línea m del fichero. Si el fichero no existe debe mostrar un mensaje por pantalla informando de ello.

Ejercicio: Fichero y módulo Requests

Escribir un programa que acceda a un fichero de internet mediante su url y muestre por pantalla el número de palabras que contiene.

URL: https://www.gutenberg.org/cache/epub/2000/pg2000.txt

Trabajar con ficheros

```
# Elimino un fichero.
os.remove("fichero.txt")

# Comprobar si un fichero existe.
if os.path.exists("fichero.txt"):
    os.remove("fichero.txt")
else:
    print("El fichero no existe.")

# Elimino un directorio.
os.rmdir("nombre_directorio") # Sólo se pueden borrar directorios vacíos.
```

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

DEPURANDO EL CÓDIGO

Python

Curso Especialista en Ciberseguridad

M.3611.048.010

20 de julio 2020 15:45-17:45

Python

Asier Martínez

Unión Europea
Fondo Social Europeo
Iniciativa de Empleo Juvenil
El FSE invierte en tu futuro

Sistema Nacional de Garantía Juvenil

polo de contenidos digitales

Ayuntamiento de Málaga