

Tema 4 PYTHON Estructuras-de-con...



Anónimo



Program for Data Science II: Python



2º Bachelor's Degree in Marketing + Degree in Data Science



**Esic Escuela de Negocios (Campus Madrid)
ESIC Universidad**



¡Pruébalo ahora!

Resume tus apuntes y prepara un cuestionario para evaluarte con **Gemini**



Preparando un cuestionario con tus apuntes...



MÉTELE ZOOM

HONOR 400
con AI Super Zoom

Índice:

ESTRUCTURAS DE CONTROL	2
Estructuras condicionales:	2
Sentencia if	2
if condición:	2
instrucciones si se cumple la condición	2
Combinación de condiciones	2
Instrucción else	3
if condición:	3
instrucciones si se cumple la condición 1	3
else:	3
instrucciones si no se cumplen ninguna de las condiciones	3
Instrucción elif	3
if condición 1:	3
instrucciones si se cumple la condición 1	3
elif condición 2:	3
instrucciones si se cumple la condición 2 y no la 1	3
elif condición N:	3
instrucciones si se cumple la condición N y no todas las anteriores	3
else:	3
rama por defecto: instrucciones si no se cumplen ninguna de las condiciones	3
Estructuras iterativas:	3
Bucle for	3
for elemento in iterable:	3
operaciones para elemento realizadas en cada iteración	3
La función range	3
Range(valor inicial, valor final, incremento)	3
Bucles anidados	4
Iterando con dos variables	4
BUCLE WHILE	4
while condición:	4
operaciones mientras se cumple la condición	4
Terminando el bucle	4
SENTECIAS CONTINUE Y BREAK	4

Tu compi tiene los apuntes...
tú tienes el HONOR 400.
Métele zoom, haz la foto y listo.



Métele zoom a
lo que de verdad
importa

HONOR

WUOLAH

ESTRUCTURAS DE CONTROL

El flujo de un programa es el orden en el que se van ejecutando las instrucciones de este. Estas instrucciones se ejecutan de arriba abajo, y de izquierda a derecha, según el orden en que aparezcan las mismas.

Sin embargo, van a existir ocasiones en las que necesitemos romper este flujo secuencial. Estas pueden ser de diverso tipo, puede que necesitemos tomar una decisión entre una o varias opciones posibles, o a lo mejor necesitamos realizar algunas instrucciones varias veces. Por tanto, necesitamos instrucciones que nos permitan interrumpir el flujo secuencial del programa, y pasar a ejecutar otros bloques de código.

Podemos identificar dos tipos de estructuras de control:

- Condicionales: Nos permiten determinar el flujo que sigue el programa en función del resultado de la evaluación de ciertas condiciones.
- Iterativas: Para realizar un conjunto de instrucciones de forma repetitiva.

Estructuras condicionales:

Las estructuras condicionales nos permiten evaluar condiciones en Python y modificar el flujo del programa, en función del resultado de la evaluación de la condición.

Sentencia if

La sentencia if, evalúa una condición y ejecuta un bloque de instrucciones en el caso de que la condición sea verdadera.

Su estructura general es la siguiente:

if condición:

instrucciones si se cumple la condición

Para evaluar estas condiciones, disponemos de los siguientes operadores:

Operador	Significado
==	Igualdad
!=	Desigualdad
>	Mayor que
<	Menos que
>=	Mayor o igual que
<=	Menor o igual que
in	Está en la lista

Combinación de condiciones

La comprobación a realizar puede combinar varias condiciones, empleando los operadores lógicos: and, or y not.

Instrucción else

La instrucción “else” provee un camino adicional en el caso de que la condición se evalúe como falsa. Es decir, si es falsa, pasamos a ejecutar otro bloque de instrucciones.

La sintaxis de la estructura completa sería:

if condición:

instrucciones si se cumple la condición 1

else:

instrucciones si no se cumplen ninguna de las condiciones

También podemos emplearlo para saber si una lista está vacía. Si empleamos el nombre de una lista como condición, Python devolverá falso si no tiene elementos.

Instrucción elif

En ocasiones, cuando no se cumpla la condición, queremos comprobar otra cosa y así, varias veces seguidas, concatenando varias comprobaciones. En estos casos empleamos la instrucción elif, que puede verse como una conjunción de las instrucciones else y if.

La sintaxis de la estructura sería:

if condición 1:

instrucciones si se cumple la condición 1

elif condición 2:

instrucciones si se cumple la condición 2 y no la 1

elif condición N:

instrucciones si se cumple la condición N y no todas las anteriores

else:

rama por defecto: instrucciones si no se cumplen ninguna de las condiciones

Estructuras iterativas:

Las estructuras iterativas permiten ejecutar de forma repetida un conjunto de operaciones un número de veces determinado. Fundamentalmente vamos a trabajar con dos estructuras iterativas: for y while.

Bucle for

El bucle for lo emplearemos para recorrer una estructura iterable, como puede lista, tupla, diccionario o rango de números, y para cada elemento se ejecuta un bloque de código.

La sintaxis es:

for elemento in iterable:

operaciones para elemento realizadas en cada iteración

La función range

Genera secuencias de valores desde un valor inicial hasta uno final, indicando el incremento de un valor con respecto al anterior.

La sintaxis es:

Range(valor inicial, valor final, incremento)

Si no se indica incremento se toma por defecto el valor 1. Y el valor final no está incluido en el rango. Y si utilizamos un único argumento, toma como valor inicial 0 y como incremento 1.

Para las estructuras iterativas, como las listas, podemos emplear los dos puntos para indicar un rango a recorrer de dicha lista. Por ejemplo, si queremos recorrer los tres primeros elementos



LOS GRANDES REYES TE MIRAN DESDE
LAS ESTRELLAS... ¡Y TE DARÁN SUERTE!

Disney
EL REY LEÓN
TEATRO LOPE DE VEGA, MADRID

de la lista, utilizaríamos: [:3] (recuerda que los índices empiezan en cero y que el extremo superior no está incluido).

Bucles anidados

Podemos incluir bucles dentro de bucles

Iterando con dos variables

En ocasiones nos va a resultar muy útil ser capaz de iterar con dos variables, es decir, recuperar pares de valores. El ejemplo clásico de este uso sería iterar los elementos de un diccionario (nos los devuelve la función `items()`), donde recobraríamos la clave y el valor a la vez.

BUCLE WHILE

La instrucción while también nos permite construir estructuras iterativas. En este caso, se ejecutan las instrucciones del bloque mientras se cumpla la condición del while.

La sintaxis de esta instrucción es:

while condición:

operaciones mientras se cumple la condición

Terminando el bucle

Es fundamental definir correctamente la condición del bucle while. Esta debe de dejar de cumplirse en algún momento de la ejecución del programa, de otra forma estaríamos corriendo un bucle infinito y el programa no terminaría nunca.

Lo más inmediato es colocar en el bucle alguna instrucción que modifique la condición del bucle, asegurándonos de que se llegue a una situación en la que la condición será falsa y se saldrá del bucle.

También podemos dar la opción al usuario de que termine el bucle. Para ello será necesario que este introduzca un dato durante cada interacción. Cuando este dato coincida con el mensaje para salir, terminaríamos el bucle.

También se puede emplear una variable booleana para terminar un bucle, lo que se denomina una bandera. Ej:

```
terminar=False
while not terminar:
    nombre=input("introduce un nombre\n")
    if (nombre=="salir"):
        terminar=True
else:
    print("Hola ",nombre)
```

SENTENCIAS CONTINUE Y BREAK

Dentro de una estructura iterativa existen algunas sentencias útiles para su manejo:

- continue: pasa a la siguiente iteración del bucle sin completar las instrucciones restantes del bloque de código del bucle.
- break: termina el bucle sin completar las iteraciones restantes.

