

# Python Básico

Estructuras de control

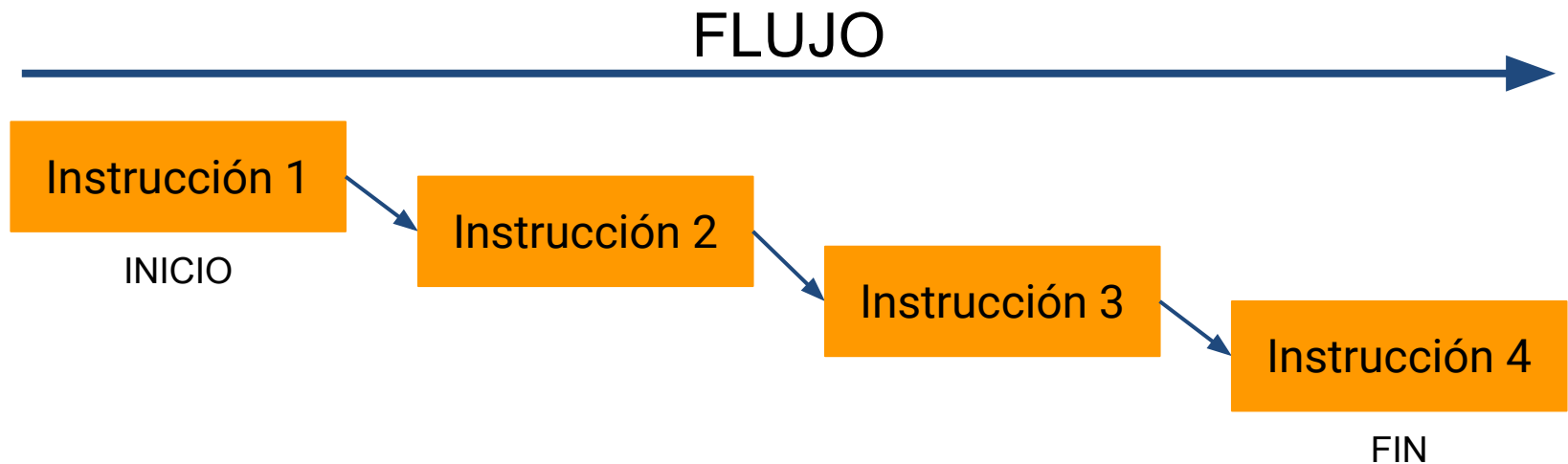


Marzo 2019



# ¿Qué es el flujo?

- Sucesión de instrucciones de un programa.



# Operadores Condicionales

Operador	Descripción	Ejemplo
==	Si los valores de dos operandos son iguales, la condición es verdadera.	A == B
!= <>	Si los valores de dos operandos son diferentes, la condición es verdadera.	A != B A <> B
>	Si el valor del operando de la izquierda es mayor que el valor del operando de la derecha, la condición es verdadera.	A > B



# Operadores Condicionales

Operador	Descripción	Ejemplo
<	Si el valor del operando de la derecha es mayor que el valor del operando de la izquierda, la condición es verdadera.	$A < B$
>=	Si el valor del operando de la izquierda es mayor o igual que el valor del operando de la derecha, la condición es verdadera.	$A >= B$
<=	Si el valor del operando de la derecha es mayor o igual que el valor del operando de la izquierda, la condición es verdadera.	$A <= B$



# Operadores Lógicos

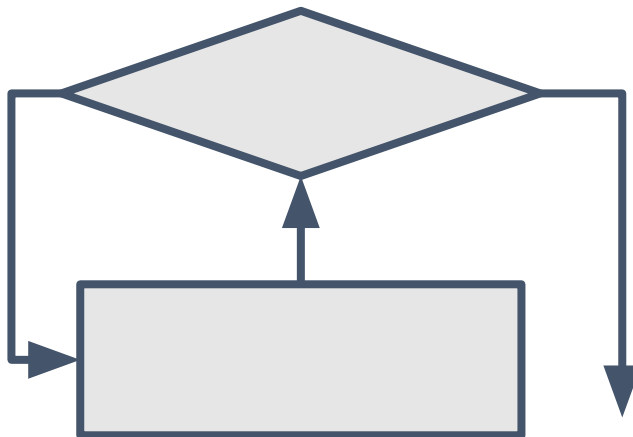
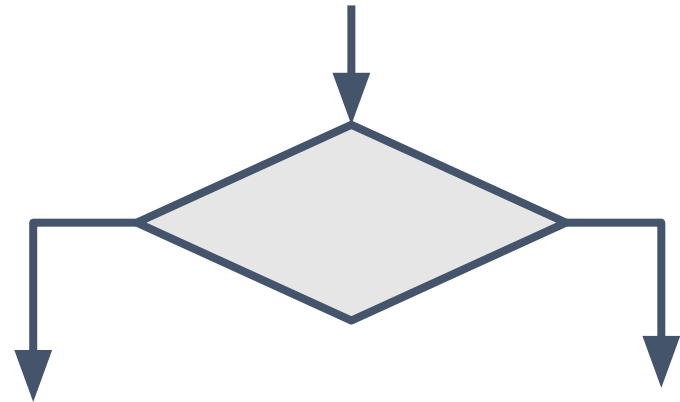
- Se utilizan para combinar valores de verdad (verdadero y falso) y obtener nuevos valores que determinen el flujo de control de un algoritmo o programa.

Operador	Descripción	Ejemplos
and	¿Se cumple A y también se cumple B?	c = True and False c = True and True
or	¿Se cumple A o se cumple B?	c = True or False c = True or True
not	A cambia al otro valor posible	c = not True c = not False



# Sentencias de Control

**Condicionales:** para elegir entre distintas posibilidades.

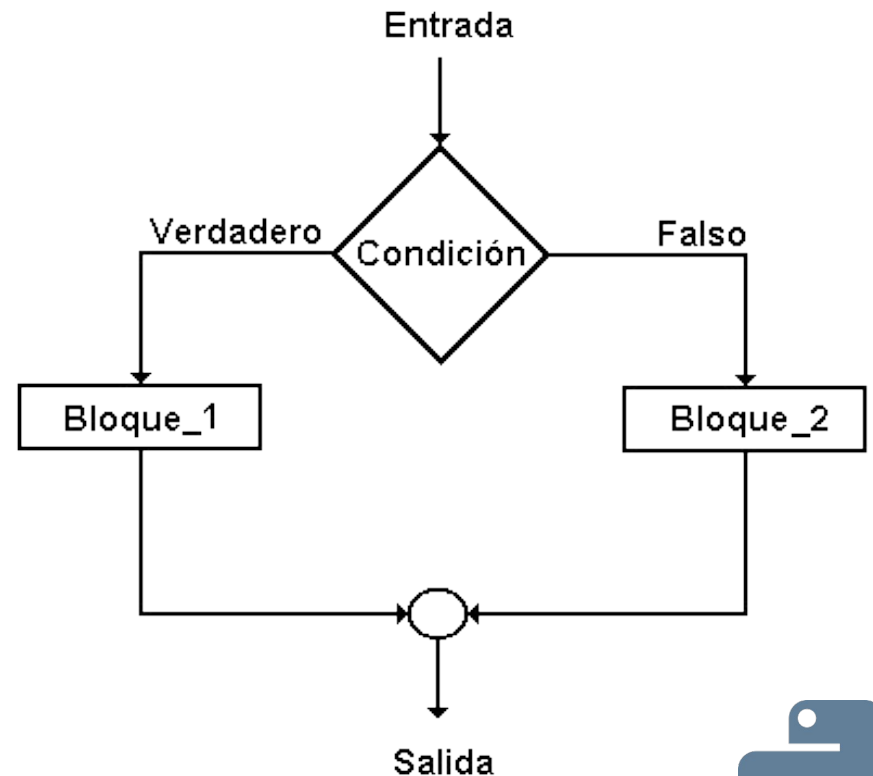


**Iterativas:** Para repetir un bloque de instrucciones.



# Condicionales

- Permiten dividir la ejecución de un programa en diferentes caminos de acuerdo a una condición dada.



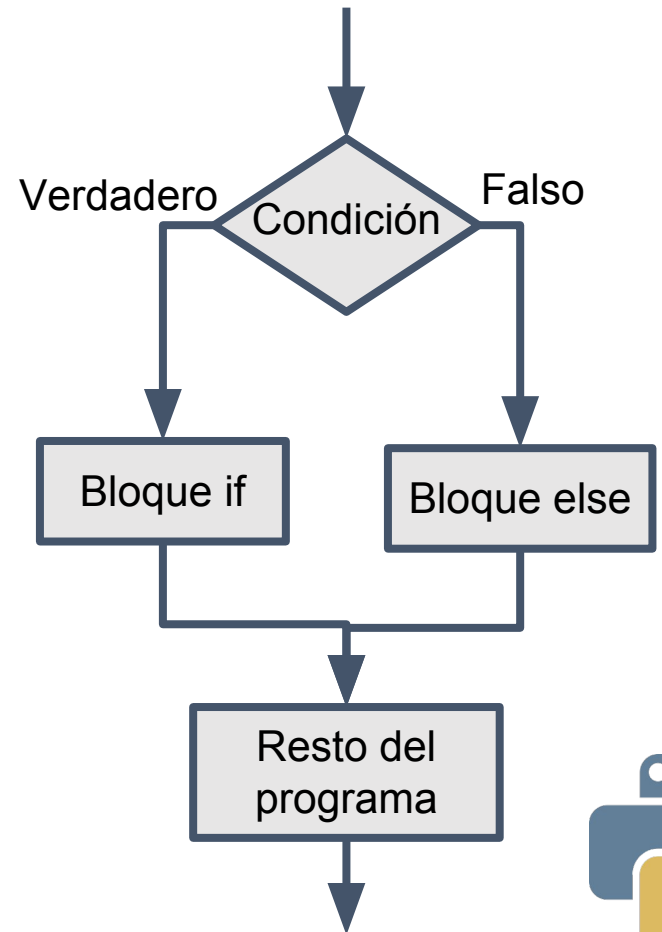
# Condicionales

## Sentencia if (si)

- Se ejecuta su bloque de código siempre que la expresión que comprueba devuelva True.

## else (caso contrario)

- Se encadena al final de un if para ejecutar el caso contrario, es decir, cuando la condición es False.





# Ejercicio

Realizar un programa que determine si un número dado es par o impar.

¿Cómo sé si un número es par?

- Sea **N** el número a determinar si es par o impar.  
Dividimos **N** entre 2. Si el residuo es 0, entonces el número es par, en caso contrario (residuo distinto de 0) el número es impar.

\*Pista: El residuo de una división se obtiene con el operador '%’.

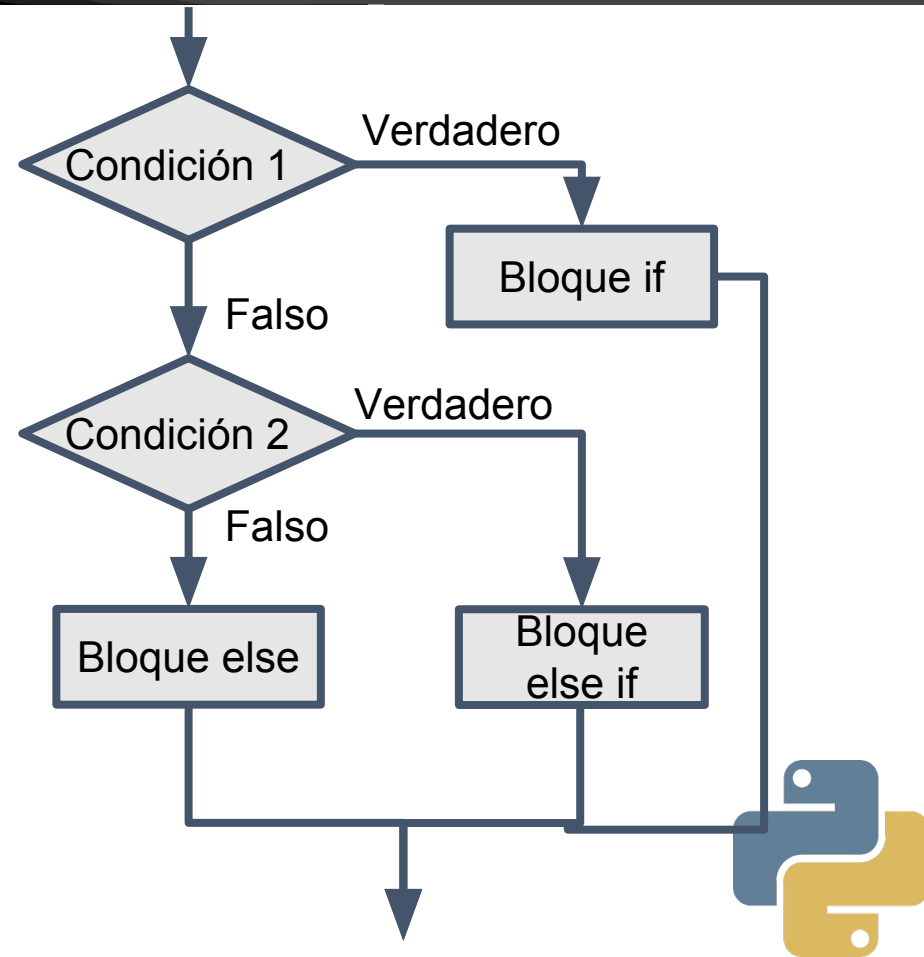
Ejemplo:  $3 \% 2 = 1$



# Condicionales

## Sentencia elif (si no)

- Sirve para comprobar múltiples condiciones. Únicamente ejecuta el código de aquella cuya comprobación regresa True.



# Ejercicio

Realizar un programa que determine el índice de masa corporal dado un peso y una estatura e imprima su condición de acuerdo a la siguiente tabla:

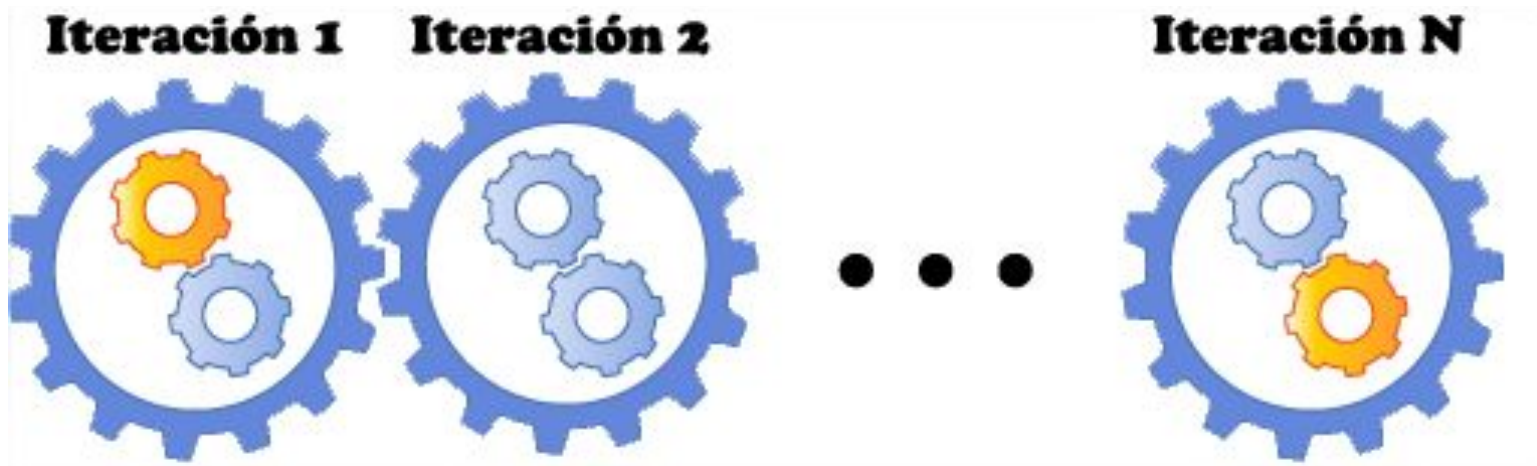
IMC	Condición
Menor a 18.5	Desnutrición
Entre 18.5 y 24.9	Peso saludable
Entre 25 y 29.9	Sobrepeso
30 o mayor	Obesidad

$$\text{IMC} = \text{peso} / \text{estatura}^2$$



# Iterativas

Iterar es el acto de repetir un proceso con la intención de alcanzar una meta deseada.

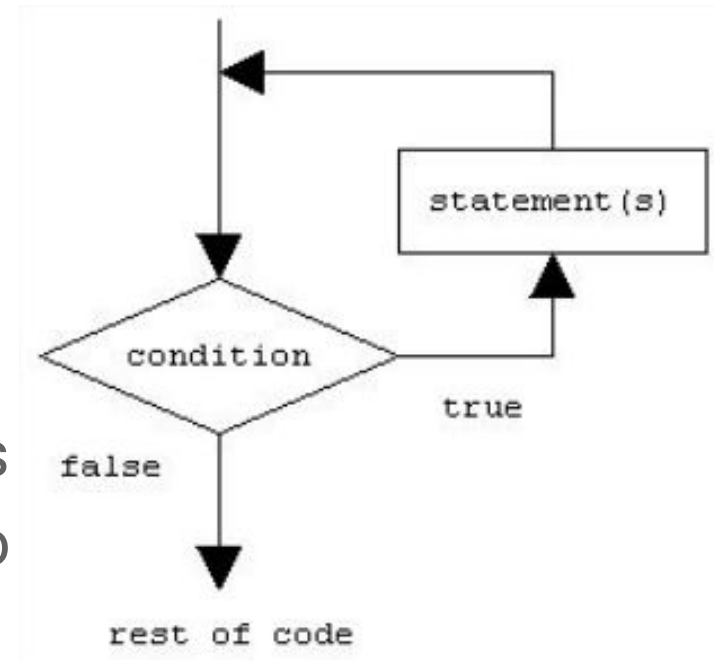


# Iterativas

## Ciclo while (mientras)

- Su propósito es repetir un bloque de código mientras una condición sea verdadera.

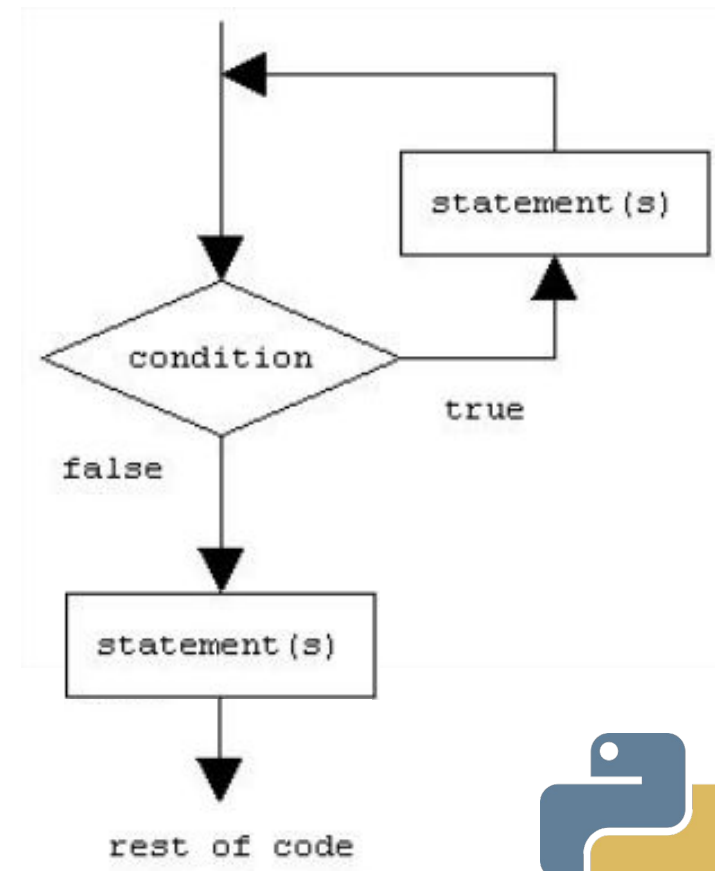
**¡OJO!** En algún momento debemos cambiar la condición si no queremos entrar a un ciclo infinito.



# Iterativas

## else en un ciclo while

- Se encadena al while para ejecutar un bloque de código una vez que la condición devuelve False, lo cual sucede normalmente al finalizar la última iteración.



# Ejercicio

Realizar un programa que solicite un número al usuario mientras el número ingresado sea menor a 10.



# Iterativas

## Sentencia break

- Sirve para "romper" o interrumpir la ejecución normal del ciclo while en determinado momento. No se ejecutará el else, ya que éste sólo funciona tras finalizar la última iteración.





# Iterativas

## Sentencia continue

- Sirve para "saltarse" la iteración actual sin romper el ciclo.



# Ejercicio

Realizar un programa que incremente de uno en uno el valor de un número desde 0 hasta 20 y, utilizando la sentencia `continue`, imprima dicho valor únicamente cuando corresponda a un número par. El programa debe implementarse utilizando un ciclo infinito que se rompa con un `break`.



# Iterativas

## Ciclo for

- Recorre los elementos de una secuencia, por ejemplo, listas o tuplas, y ejecuta un bloque de instrucciones por cada uno de ellos.



Nota: El diccionario no es una secuencia como las listas o las tuplas, para recorrerlo se necesita conocer sus llaves.



# Función range()

range() crea una lista de números enteros en sucesión aritmética. Puede recibir uno, dos, y hasta tres argumentos numéricos enteros.

- Primer argumento: valor inicial.
- Segundo argumento: valor final.
- Tercer argumento: incremento.



# Ejercicio

Realizar un programa que calcule el factorial de un número utilizando un ciclo for e imprima el resultado en pantalla.

¿Qué es un factorial?

- El factorial de un entero positivo  $n$  ( $n!$ ) se define en principio como el producto de todos los números enteros positivos desde 1 hasta  $n$ .

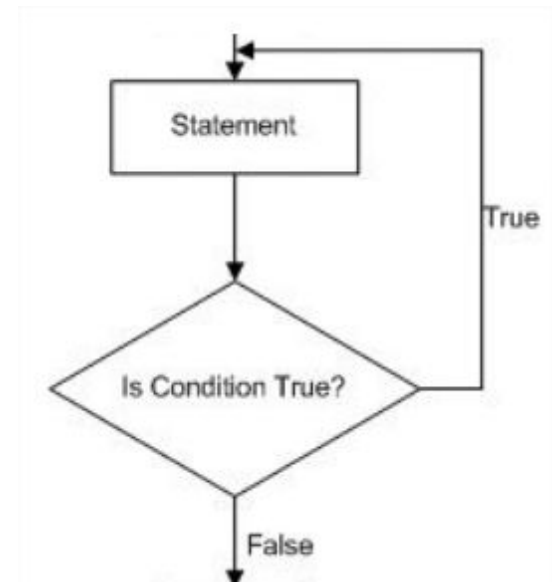
Ejemplo:  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$



# Implementaciones

## Ciclo do while

- Su propósito es ejecutar un bloque de código una vez y posteriormente repetir su ejecución mientras se cumpla cierta condición expresada en la cláusula while.



# Implementaciones

## Switch Case

- Evalúa una variable y ejecuta el bloque de instrucciones correspondiente a dicha evaluación.

