

Python Básico

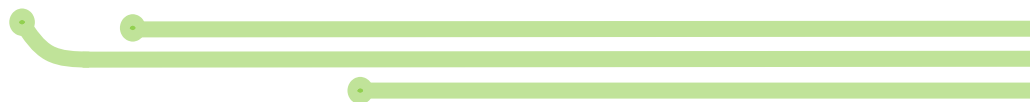


Programación
Estructurada

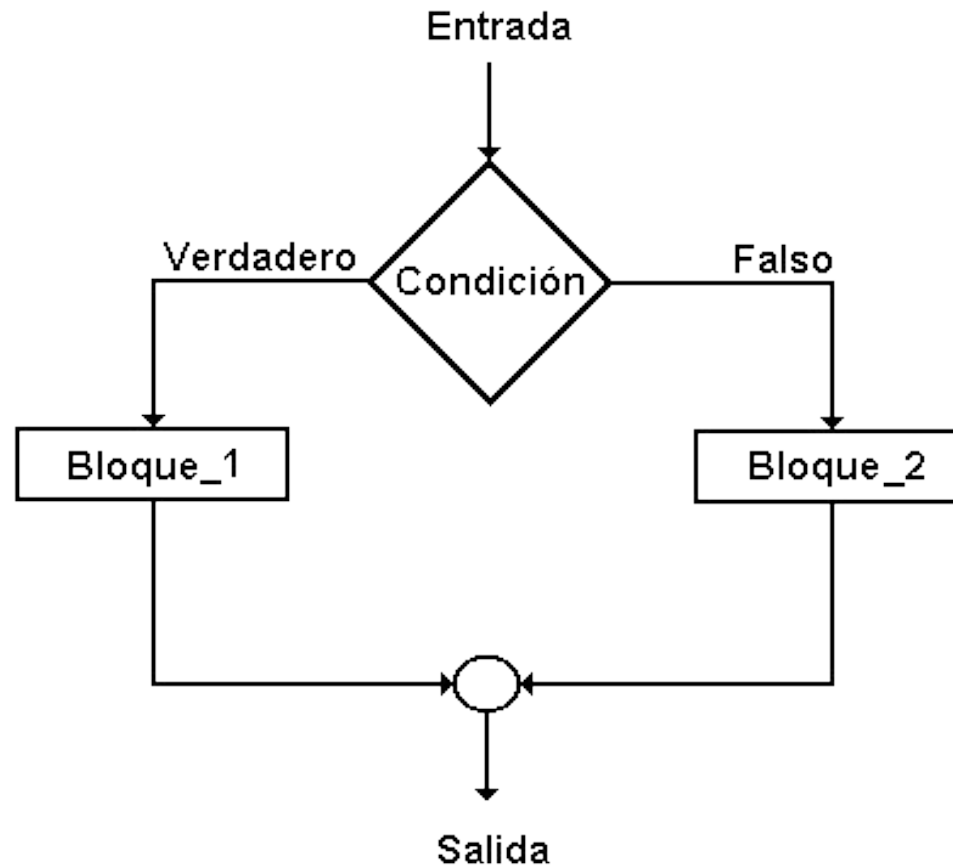
11 de enero de 2016

Estructuras de Control

- En el python como en muchos otros lenguajes del paradigma estructurado, las estructuras de control de flujo nos van a permitir controlar el comportamiento a futuro durante la ejecución del programa, estas siguen el teorema del programa estructurado o Teorema de Böhm y Jacopini (1966) que dice: “Cualquier programa de ordenador puede diseñarse e implementarse utilizando únicamente las tres construcciones estructuradas (secuencia, selección e iteración; esto es, sin sentencias goto)”



Sentencia IF



Operadores condicionales

Operador	Descripción	Ejemplo
<code>==</code>	Si los valores de dos operandos son iguales, entonces la condición sea verdadera.	<code>(A == b)</code> no es cierto.
<code>!=</code>	Si los valores de dos operandos no son iguales, entonces condición sea verdadera.	
<code><></code>	Si los valores de dos operandos no son iguales, entonces condición sea verdadera.	<code>(A <> b)</code> es verdadera. Esto es similar a <code>!=</code> Operador.
<code>></code>	Si el valor del operando de la izquierda es mayor que el valor del operando de la derecha, a continuación, condición sea verdadera.	<code>(A > b)</code> no es cierto.

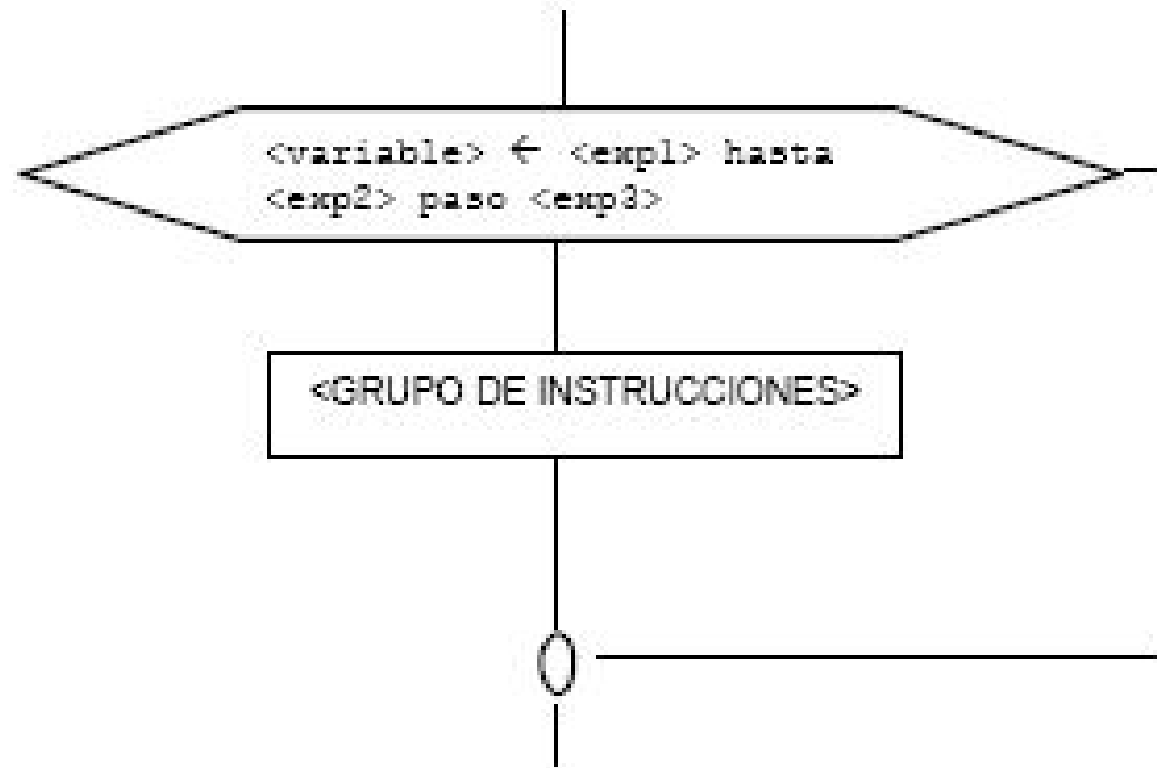
Operadores condicionales

Operador	Descripción	Ejemplo
<	Si el valor del operando de la izquierda es menor que el valor del operando de la derecha, a continuación, condición sea verdadera.	(A < b) es verdadera.
> =	Si el valor del operando de la izquierda es mayor o igual al valor del operando de la derecha, a continuación, condición sea verdadera.	(A > = b) no es cierto.
<=	Si el valor del operando de la izquierda es menor o igual al valor del operando de la derecha, a continuación, condición sea verdadera.	(A <= b) es verdadera.

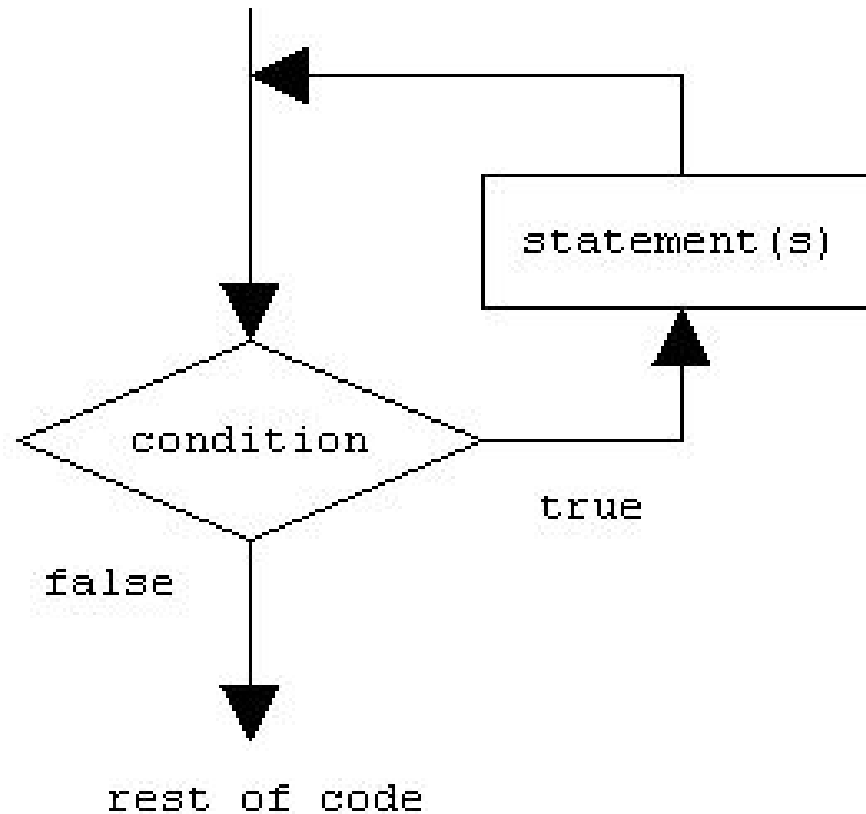
Operadores a nivel de bits

Operator	Descripción	Ejemplo
& Binary AND	operador copia un poco al resultado si existe en ambos operandos	(A y B) (0000 significa 1100)
Binary OR	Se copia un poco si es que existe en cualquier operando.	(A b) = 61 (0011 significa 1101)
^ Binary XOR	Se copia el bit si se establece en un operando, pero no ambos.	(A ^ b) = 49 (medios 0011 0001)
~ Binary Ones Complement	Es unario y tiene el efecto de bits 'flipping'.	(~ A) = -61 (significa 1100 0011 en forma de complemento a 2 debido a un número binario con signo.
<< Binary Left Shift	El valor operandos izquierdo se movió a la izquierda por el número de bits especificado por el operando de la derecha.	un << = 240 (1111 significa 0000)
>> Binary Right Shift	El valor operandos izquierdo se mueve a la derecha por el número de bits especificado por el operando de la derecha.	A >> = 15 (medios) 0000 1111

For loop

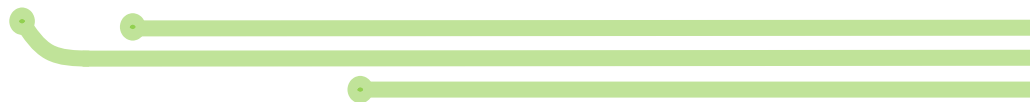


While Loop

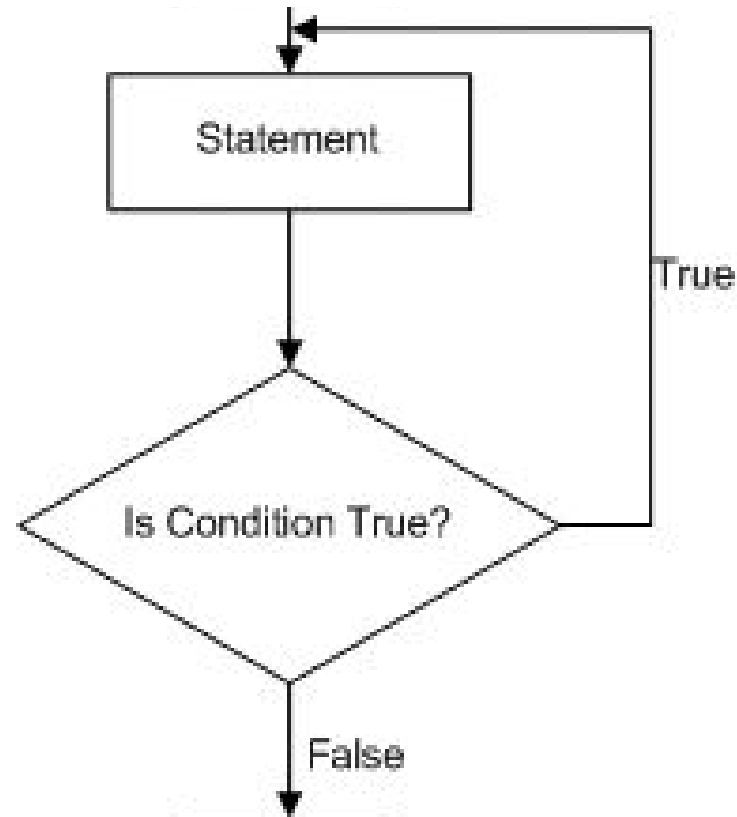


Implementaciones

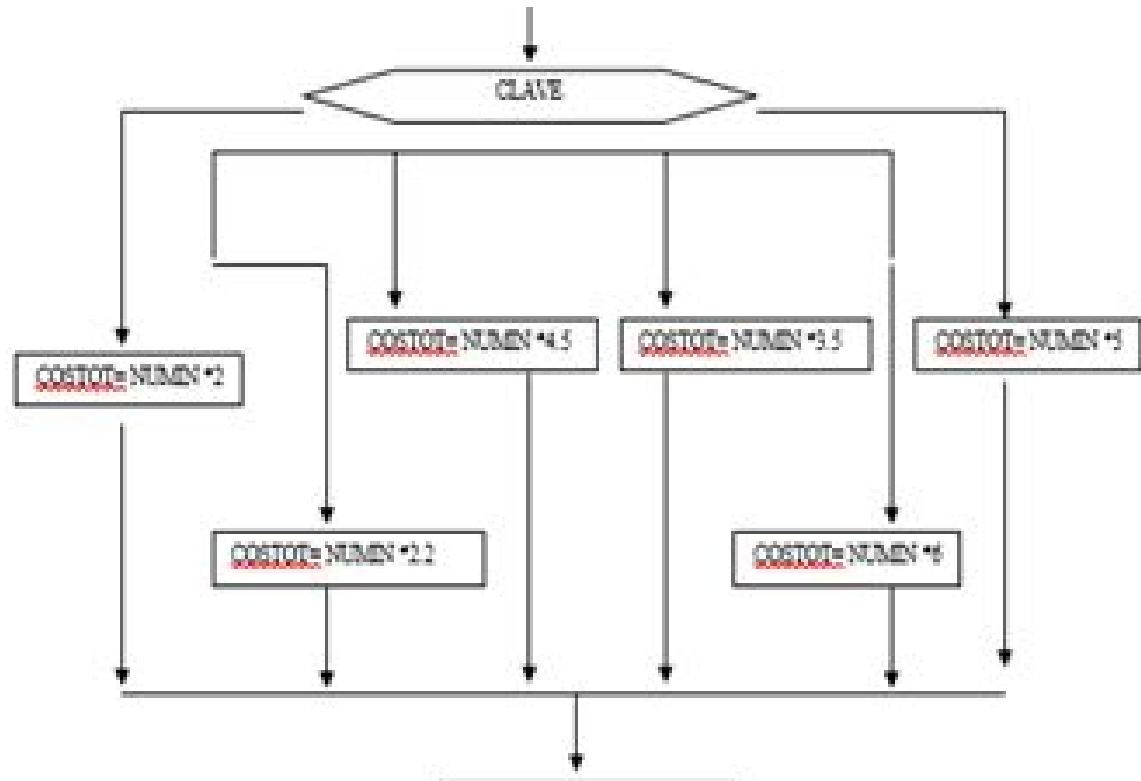
- No existen como tal en el lenguaje (no son nativas) pero se pueden implementar



Do While Loop



Switch Case



- Tarea

El índice se calcula así:

$$IMC = \frac{kg}{m^2}$$

Y tiene los siguientes resultados:

Resultado del IMC	Estado
Menos de 18.49	Infra Peso
18.50 a 24.99	Peso Normal
25 a 29.99	Sobre Peso
30 a 34.99	Obesidad Leve
35 a 39.99	Obesidad Media
40 o Mas	Obesidad Mórbida
Clasificación de la OMS de acuerdo al IMC	