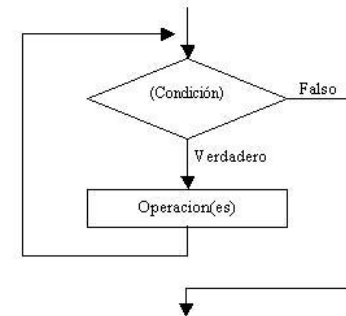
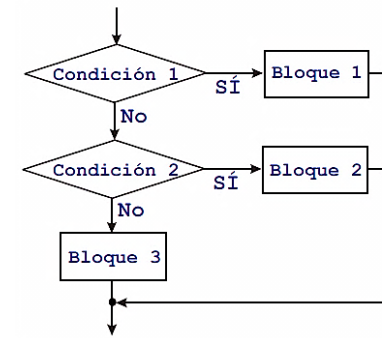
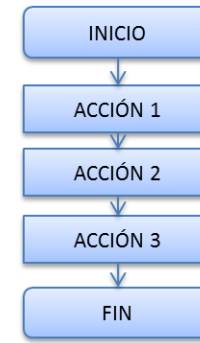




Programación estructurada



Programación estructurada



Estructura de decisión simple:

if *expresión*:

Bloque de
instrucciones

Estructura de decisión doble:

if *expresión*:

Bloque de
instrucciones 1

else:

Bloque de
instrucciones 2

Estructura de decisión múltiple:

if *expresión 1*:

Bloque de
instrucciones 1

elif *expresión 2* :

Bloque de
instrucciones 2

elif *expresión 3* :

Bloque de
instrucciones 3

•
•
•
•
•

else:

Bloque de
instrucciones N

Aquellas que al evaluarse dan como resultado un valor booleano: **True** o **False**

operadores relacionales:

mayor: >

menor: <

mayor o igual: >=

menor o igual: <=

comparacion: ==

diferente: !=

operadores lógicos:

Y: and

O: or

NO: not

while expresion:

Bloque de
instrucciones

Mientras la
expresión sea
verdadera se
ejecutará el bloque
de instrucciones

Función *range()*: genera un rango de valores enteros

```
range(start, end, step)
```

Pero puede funcionar con 1 o dos argumentos

- `range(10)` genera el rango de valores: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- `range(1, 10)` genera el rango de valores: 1, 2, 3, 4, 5, 6, 7, 8, 9
- `range(1, 10, 2)` genera el rango de valores: 1, 3, 5, 7, 9
- `range(1, 10, 4)` genera el rango de valores: 1, 5, 9
- `range(7, 1, -1)` genera el rango de valores: 7, 6, 5, 4, 3, 2
- `range(8, 1, -2)` genera el rango de valores: 8, 6, 4, 2

for x in range(1,16):

Bloque de
instrucciones

Para $x \rightarrow 1$:

Ejecuta bloque de instrucciones

Para $x \rightarrow 2$:

Ejecuta bloque de instrucciones

Para $x \rightarrow 3$:

Ejecuta bloque de instrucciones

•
•
•
•
•

Para $x \rightarrow 15$:

Ejecuta bloque de instrucciones

La estructura *try – except*:

try:

Bloque de
instrucciones

except nombre 1:

Bloque de instrucciones 1

except nombre 2:

Bloque de instrucciones 2

•
•
•
•

except nombre N:

Bloque de instrucciones N

Nota: No todas las excepciones
pueden ser capturada por el
try

Python cuenta con un conjunto de **módulos o librerías** para usar funciones que no pertenecen al estándar de Python.

Por ejemplo, si queremos usar funciones para generar números aleatorios debemos importar el módulo **random**

Algunas funciones del módulo *random*:

randint : Para generar un número entero aleatorio en un intervalo $[a, b]$

uniform : Para generar un número flotante aleatorio en un intervalo $[a, b>$