

Unidad 2



Agenda

- Palabras reservadas
- Not
- Is
- help
- Bucles
- Precaución
- while
- range
- for
- break y continue
- Ejercicios

Palabras reservadas

- Hay palabras que no se pueden utilizar libremente porque tiene un significado en Python.

https://docs.python.org/3/reference/lexical_analysis.html#keywords

Not

- not es una palabra reservada que se utiliza para 'negar' un booleano
- Si **a = True** → **not a** → **False**
- Si **a = False** → **not a** → **True**

is

- Se utiliza la palabra reservada 'is' como sinónimo de '=='. Se utiliza para preguntar si “algo **es** otra cosa”

```
variable1 = 23
```

```
if variable1 is 23:
```

```
    print("Si! variable1 es 23")
```

```
-----
```

```
if variable1 == 23:
```

```
    print("Si! variable1 es 23")
```

help

```
• >>> help(int)
```

Help on class int in module builtins:

```
class int(object)
```

```
| int(x) -> integer
```

```
| int(x, base=10) -> integer
```

```
|
```

```
| Convert a number or string to an integer, or return 0 if no arguments
```

```
| are given. If x is a number, return x.__int__(). For floating point
```

```
| numbers, this truncates towards zero.
```

```
|
```

```
| If x is not a number or if base is given, then x must be a string,
```

```
| bytes, or bytearray instance representing an integer literal in the
```

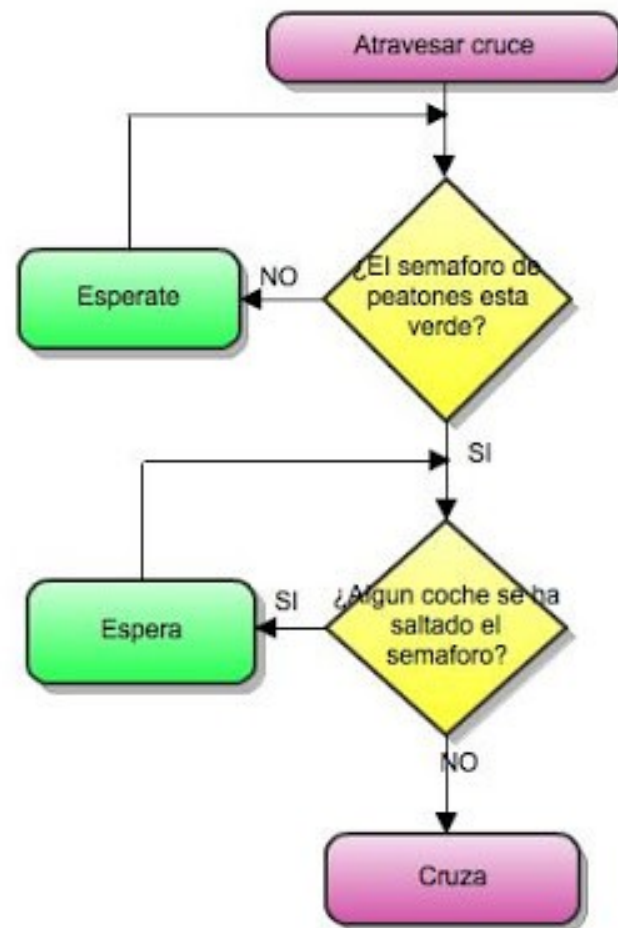
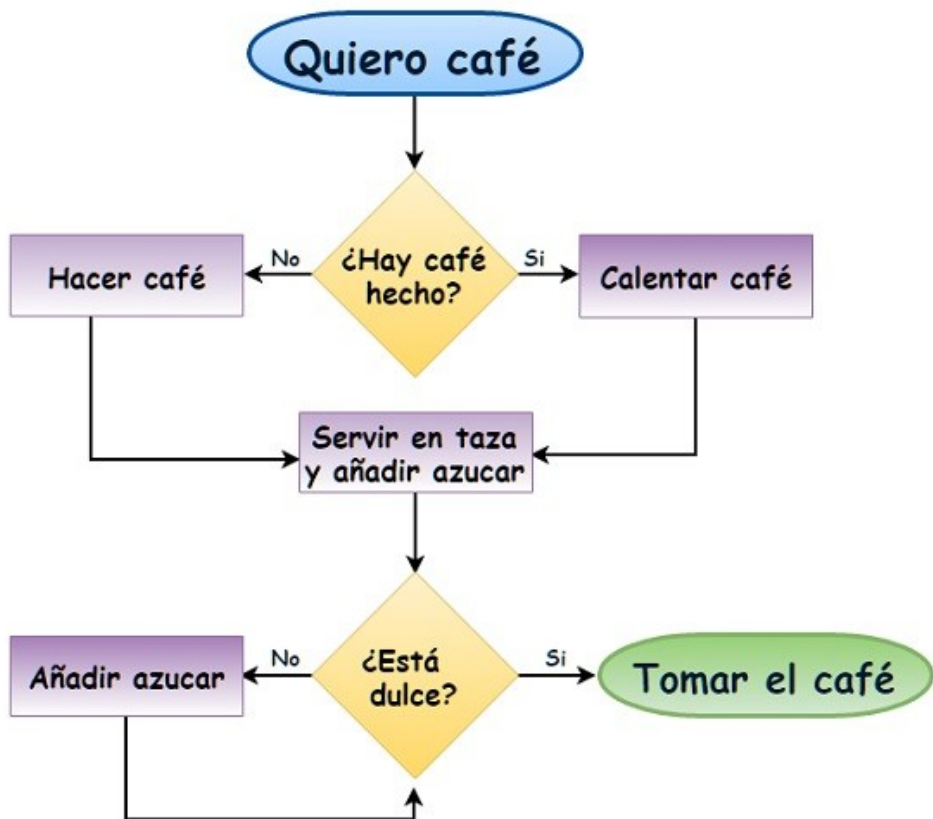
```
| given base. The literal can be preceded by '+' or '-' and be surrounded
```

```
| by whitespace. The base defaults to 10. Valid bases are 0 and 2-36.
```

```
| Base 0 means to interpret the base from the string as an integer literal.
```

```
| >>> int('0b100', base=0)
```

Bucles



Precaución – Bucles infinitos



While – bucle condicional

- Un bucle condicional repite una serie de instrucciones **mientras** se cumpla una condición.

```
x = 0
```

```
while x == 10:
```

```
    x = x + 1
```

range

- <https://docs.python.org/3/library/functions.html#func-range>
- <https://docs.python.org/3/library/stdtypes.html#range>

>>> range(Desde, Hasta, [Salto]) ← [] Significa “opcional”

>>> range(Hasta) ← Implícitamente el desde es 0

Tip: `reversed()` para dar vuelta las cosas :)

For – bucle exacto

- Bucles exactos son aquellos que de antemano sabemos cuantas veces se va a repetir el ciclo.

```
>>> for i in range(1,10):  
...     print(i)
```

Break y continue

- Son dos sentencias que nos permite frenar lo que estamos haciendo para:
 - break: salir del bucle, sea dónde sea que estemos
 - continue: frena el bucle dónde está y vuelve al principio, del bucle.



Let's play with Loops! :)

Ejercicios

- 1) Escribir en pantalla una lista del 5 al mil de números pares
- 2) Escribir en pantalla una lista del 5 al mill de números impares
- 3) Escribir la tabla del 7 en en siguiente formato: (tip: \t)

7 <Tab> x <Tab> 0 <Tab> = 0

7 <Tab> x <Tab> 1 <Tab> = 7

....

Ejercicios

- Calcular los números primos entre 0 y 1000.