

Funciones built-in

Una función en programación es **todo lo que lleva paréntesis**. Usos típicos **para empezar rápido**:

```
type(x)
round()
input()
```

Convierte rápido de un tipo a otro:

```
int(x)
float(x)
str(x)
```

Crear funciones

Una función se crea cuando detectamos que hacemos lo mismo varias veces.

Sintaxis en Python básica:

```
def elevarCuadrado(a):
    print('activando funcion...')
    return a ** 2
```

```
elevarCuadrado(a)
```

Librerías

En Python tenemos **Pip para instalar librerías** fiables. Se instala una nueva desde CMD así:

```
pip install nombreLibreria
```

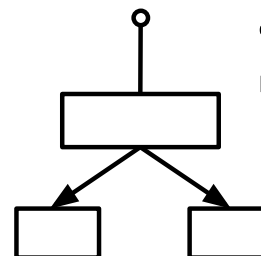
Usarlo haciendo **import nombreLibria** en el código

Estructuras de Control

Estructuras de control **condicionales**:

USO Cuando detectas que el programa puede ir por distintos caminos posibles según el caso.

```
if (a > b):
    print('¡A gana!')
elif (a < b):
    print('¡B gana!')
else:
    (¡Empate!)
```



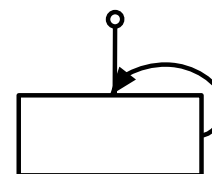
Estructuras de control **repetitivas**:

USO Cuando detectas que el programa debe **repetir pasos**.

```
while(vidaA > 0 and vidaB > 0):
    seguirCombate()
    cambiarTurno()
# 🟡 Sólo acaba si la condición es falsa.

for i in range(0, 10):
    print(i * 2)
# 🟡 Imprime i * 2 de 0 a 9. 10 no incluido.

for elem in mi_coleccion:
    print(elem)
# 🟡 Imprime cada valor dentro de la colección
```



3 principios básicos

Todo lenguaje de programación es **secuencial** y **determinista**

Estos lenguajes son o **compilados** o **dinámicos / interpretados / no compilados**

Python nos vale para hacer **scripting**:

"dime lo que necesitas que lo automatizo"

```
print('Hola Mundo')
```

Tipos de datos Primitivos o simples

Usar **type(x)** para **conocer el tipo de dato** en Python

- **int**: Representa los números naturales
- **float**: Representa los números reales
- **bool**: Son o cierto o falso: **True False**

Tipos de datos Complejos: Colecciones

Usar **len(x)** para **conocer número de elementos**

- | | |
|--|-------------------------------------|
| - listas : mutables  | <code>var = [1, 2.1, [0, 2]]</code> |
| - textos : inmutables  | <code>var = "Hola" ó 'Hola'</code> |
| - tuplas : inmutables  | <code>var = 'S', 'M', 'L'</code> |
| - diccionarios : mutables  | <code>var = {'key': 'value'}</code> |