

Argumentos y parámetros

En la definición de una función los valores que se reciben se denominan **parámetros**, pero durante la llamada los valores que se envían se denominan **argumentos**.

Argumentos por posición

Cuando enviamos argumentos a una función, estos se reciben por orden en los parámetros definidos. Se dice por tanto que son argumentos por posición:

```
def resta(a, b):  
    return a - b  
  
resta(30, 10) # argumento 30 => posición 0 => parámetro a  
              # argumento 10 => posición 1 => parámetro b
```

20

Argumentos por nombre

Sin embargo es posible evadir el orden de los parámetros si indicamos durante la llamada que valor tiene cada parámetro a partir de su nombre:

```
resta(b=30, a=10)
```

-20

Llamada sin argumentos

Al llamar una función que tiene definidos unos parámetros, si no pasamos los argumentos correctamente provocará un error:

```
resta()
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-4-78c8f433960e> in <module>()  
----> 1 resta()  
  
TypeError: resta() missing 2 required positional arguments: 'a' and 'b'
```

Parámetros por defecto

Para solucionarlo podemos asignar unos valores por defecto nulos a los parámetros, de esa forma podríamos hacer una comprobación antes de ejecutar el código de la función:

```
def resta(a=None, b=None):  
    if a == None or b == None:  
        print("Error, debes enviar dos números a la función")  
        return # indicamos el final de la función aunque no devuelva nada  
    return a-b  
  
resta()
```

Error, debes enviar dos números a la función