

Paso de argumentos de palabra clave

Python ofrece otra manera de pasar argumentos, donde el **significado del argumento está definido por su nombre**, no su posición - a esto se le denomina **paso de argumentos con palabra clave**.

Python

```
def introduction(first_name, last_name):
    print("Hola, mi nombre es", first_name,
          last_name)

introduction(first_name = "James", last_name =
             "Bond")
introduction(last_name = "Skywalker", first_name =
             "Luke")
```

Fuente: Elaboración propia

Los valores pasados a los parámetros son precedidos por el nombre del parámetro al que se le va a pasar el valor, seguido por el símbolo de asignación (=).

Debe tenerse en cuenta ahora el nombre del parámetro de la función para utilizarlo a la hora de invocarla. Debido a que ahora se busca empatar el argumento enviado a la función con el nombre de alguno de sus parámetros, enviar el nombre incorrecto o que no exista generará un error.



Ahora, existe la posibilidad de mezclar los argumentos posicionales y de palabra clave al construir e invocar una función, solo hay una regla inquebrantable: se deben colocar primero los argumentos posicionales y después los de palabra clave.

Python

```
def adding(a, b, c):
    print(a, "+", b, "+", c, "=", a+b+c)
```

```
adding(1, 2, 3)
```

```
#1 + 2 + 3 = 6
```

```
#Ahora invocando la función utilizando un orden diferente
```

```
adding(c = 1, 2, a = 3)
```

```
#Esto daría un error, ya que no se deben mezclar los tipos de argumento, recordar la regla inquebrantable
```

```
adding(2, c = 8, b = 4)
```

```
# 2 + 4 + 8 = 14
```

```
#Otro error que se debe evitar es darle dos veces un valor a una misma variable, primero por posición y luego por palabra clave
```

```
adding(2, a= 4, b = 4)
```

```
#TypeError: adding() got multiple values for argument 'a'
```

Fuente: Elaboración propia