

# Retorno de valores

Para comunicarse con el exterior, las funciones pueden devolver valores al proceso principal gracias a la instrucción **return**.

En el momento de devolver un valor, la ejecución de la función finalizará:

```
def test():
    return "Una cadena retornada"

test()

'Una cadena retornada'
```

Los valores devueltos se tratan como valores literales directos del tipo de dato retornado:

```
print(test())

Una cadena retornada
```

Por ejemplo no podemos sumar una cadena con un número:

```
c = test() + 10

-----
TypeError                                Traceback (most recent call last)
<ipython-input-6-0ac9d7015445> in <module>()
----> 1 c = test() + 10

TypeError: Can't convert 'int' object to str implicitly
```

También podemos devolver cualquier tipo de colección y manejarla directamente:

```
def test():
    return [1,2,3,4,5]

print(test())
print(test()[-1])
print(test()[1:4])

[1, 2, 3, 4, 5]
5
[2, 3, 4]
```

Evidentemente es posible asignar el valor retornado a una variable:

```
lista = test()
print(lista[-1])
```

```
5
```

## Retorno múltiple

Una característica interesante, es la posibilidad de devolver múltiples valores separados por comas:

```
def test():
    return "Una cadena", 20, [1,2,3]

test()
```

```
('Una cadena', 20, [1, 2, 3])
```

Estos valores se tratan en conjunto como una tupla inmutable y se pueden reasignar a distintas variables:

```
cadena, numero, lista = test()

print(cadena)
print(numero)
print(lista)
```

```
Una cadena
20
[1, 2, 3]
```