

Python Programming

Módulo 2

Funciones de orden superior y anónimas

Funciones de orden superior

Podemos crear funciones que pueden tomar funciones como parámetros y también retornar (devolver) funciones como resultado. Una función que hace ambas cosas, o alguna de ellas, se la llama **función de orden superior**. En el ejemplo genérico de la derecha se pasa una función y una lista para que la función `superior()` trabaje.

Existen algunas funciones de orden superior incorporadas (*built-in*) en Python como `map()` y `filter()`. Te recomendamos leer la documentación oficial del lenguaje para que las conozcas y veas cuáles son sus características.

```
def sumar(x):  
    return x+100  
  
def cuadrado(x):  
    return x**2  
  
def superior(funcion,lista):  
    resultado = []  
    for n in lista:  
        resultado.append(funcion(n))  
    return resultado  
  
numeros = [2,5,10]  
  
print(superior(sumar,numeros))  
out: [102, 105, 110]  
  
print(superior(cuadrado,numeros))  
out: [4, 25, 100]
```

Funciones anónimas o lambda

Las **funciones lambda** son un tipo de funciones que típicamente se pueden escribir en una línea y cuyo código a ejecutar suele ser pequeño.

Resulta complicado explicar las diferencias porque, básicamente, la diferencia es cómo se escribe, la sintaxis. Para que tengas una idea, son funciones que queremos usar y descartar. Idealmente se usan como argumento en funciones de orden superior.

La función `cuadrado(x)` que usamos en el ejemplo anterior la podríamos escribir como:

```
lambda x : x**2
```

Y la de `sumar(x)`:

```
lambda x : x+100
```

Y con la función `superior(funcion, lista)` quedaría así:

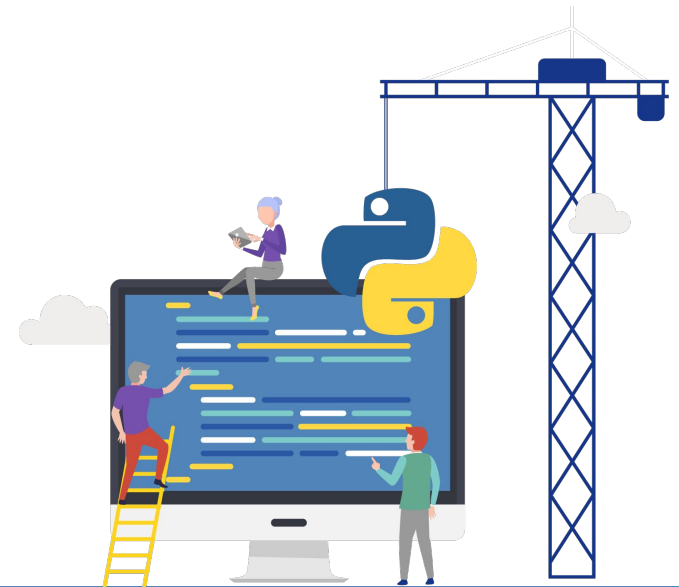
```
print(superior(lambda x : x**2 , numeros))  
out: [102, 105, 110]
```

```
print(superior(lambda x : x+100 , numeros))  
out: [4, 25, 100]
```

A las **funciones anónimas** también se les puede asignar un nombre como si fuera una variable (en realidad, como si fuera un objeto). Esto último no es muy aconsejable y no fueron pensadas para esto.

Pero si estamos en la consola interactiva haciendo pruebas, puede ser de gran utilidad para definir pequeñas subrutinas.

```
>>> cuadrado = lambda x : x**2  
>>> cuadrado(10)  
100
```



¡Sigamos trabajando!