

Funciones (lambda)

Unidad 4

Python para principiantes
dgm@uma.es



Funciones lambda

- Las denominadas funciones “lambda” son una forma rápida y concisa de trabajar con funciones en Python
- Son funciones “anónimas” (no tienen nombre) con una sintaxis más restrictiva pero más concisa que las funciones normales

Primer ejemplo directamente en el intérprete:

```
>>> (lambda x: x + 1)(2)
```

Salida: 3

¿Qué hemos hecho aquí?

Hemos definido una función que toma como argumento “x” y devuelve “x+1”. A esta función la llamamos con el valor 2, devolviendo 2+1 (3)

Funciones lambda

- Podemos dar nombre a las funciones lambda

```
>>> add_one = lambda x: x + 1
```

Y después usarlas como cualquier otra función

```
>>> add_one(2)
```

- Esto sería equivalente a haber definido la siguiente función:

```
def add_one(x):  
    return x + 1
```

Funciones lambda

- Podemos definir funciones lambda con más de un argumento

```
>>> full_name = lambda first, last: f'Full name: {first.title()} {last.title()}'
```

```
>>> full_name('guido', 'van rossum')
```

```
'Full name: Guido Van Rossum'
```

Fijaos en que no es necesario poner () para los argumentos first y last y que tampoco necesitamos return para indicar qué devuelve la función

Características de las funciones lambda

- Solo pueden contener expresiones y no sentencias en su cuerpo
- Se escriben como una sola línea.
 - Aunque el código esté dividido en varias líneas usando paréntesis o string multilineas, conceptualmente está en una sola línea
- Pueden ser llamadas inmediatamente
- No se pueden utilizar anotaciones de tipo
 - (esto es, no se pueden indicar los tipos de los argumentos de la función)

Cuándo usar o no funciones lambda

- A veces se abusa de las funciones lambda cuando su uso dificulta la lectura
- En otras ocasiones su uso puede hacer más fácil o con un estilo más “funcional” el código desarrollado
 - Las funciones lambda se usan de forma intrínseca con funciones como `map()` y `filter()`. Prueba estos ejemplos:
 - `list(map(lambda x: x.upper(), ['cat', 'dog', 'cow']))`
 - `list(filter(lambda x: 'o' in x, ['cat', 'dog', 'cow']))`
 - Se usan también en algunas funciones que reciben funciones como argumentos:

```
ids = ['id1', 'id2', 'id30', 'id3', 'id22', 'id100']
>>> print(sorted(ids)) # Ordenación lexicográfica
['id1', 'id100', 'id2', 'id22', 'id3', 'id30']
>>> sorted_ids = sorted(ids, key=lambda x: int(x[2:])) # Ordenación por números
```