

Fundamentos de Programación

Grado en Ingeniería del Software

21/09/22

Antecedentes


- Filtrado y transformación por comprensión
- Contador
- Suma
- Máximo y mínimo por defecto
- Ordenación por defecto

Índice


- Parametro por defecto
- Paso de función como parámetro
- Funciones lambda
- Construcción de diccionarios
- Ordenaciones alternativas

**PARAMETRO POR
DEFECTO**

```
def raised_number(number, scale=1):  
    """Raise number to the power of scale."""  
  
    value = number ** scale  
    return value
```



```
print(raised_number(9, 2))  
print(raised_number(9, 1))  
print(raised_number(9))
```



81

9

9

PASO DE FUNCIÓN COMO PARÁMETRO

```
def funcion1():  
    return ', soy una función.'  
  
def funcion_con_funcion(nombre, funcion):  
    return nombre + funcion()  
  
print(funcion_con_funcion("Jorge", funcion1))
```

Jorge, soy una función.

FUNCIONES LAMBDA


```
def doble(n):  
    return n*2  
  
def triple(n):  
    return n*3  
  
print(doble(11))  
print(triple(11))
```

```
mydoubler = lambda a : a * 2  
mytripler = lambda a : a * 3  
  
print(mydoubler(11))  
print(mytripler(11))
```

```
def myfunc(n):  
    return lambda a : a * n  
  
mydoubler = myfunc(2)  
mytripler = myfunc(3)  
  
print(mydoubler(11))  
print(mytripler(11))
```

CONSTRUCCIÓN DE DICCIONARIOS

```
listado = [1, 2, 3, 3, 3, 4, 5, 6, 7, 1]

def construye_diccionario(lista):
    result = dict()

    for dato in lista:
        clave = dato
        if clave in result:
            valor_actualizado = result[dato]+1
            result[clave] = valor_actualizado
        else:
            valor_inicial = 1
            result[clave] = valor_inicial

    return result

print(construye_diccionario(listado))
#{1: 2, 2: 1, 3: 3, 4: 1, 5: 1, 6: 1, 7: 1}
```

ORDENACIONES ALTERNATIVAS

```
lst = [[1,2], [1], [2,3,4], [3], [4,2], [1,2,3,4,1], [3,6,7]]

lst.sort(key = lambda x: x[-1])
lst

[[1], [1, 2, 3, 4, 1], [1, 2], [4, 2], [3], [2, 3, 4], [3, 6, 7]]

sorted(lst, key = lambda x: len(x))

[[1], [3], [1, 2], [4, 2], [2, 3, 4], [3, 6, 7], [1, 2, 3, 4, 1]]
```

main.py > ...

```
1
2 Country_dict = {'China':982, 'Egypt':758, 'Malaysia' : 12}
3
4 new_val = max(Country_dict, key= lambda x: Country_dict[x])
5 print("maximum value from dictionary:",new_val)
6
7
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ACER\Documents\work\test2> python main.py
maximum value from dictionary: China
PS C:\Users\ACER\Documents\work\test2>
```

main.py > ...

```
1 my_new_dict = {"q": 18, "z": 10, "o": 13}
2
3 fin_max = max(my_new_dict, key=my_new_dict.get)
4 print("Maximum value:",fin_max)
5
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\ACER\Documents\work\test2> python main.py
Maximum value: q
PS C:\Users\ACER\Documents\work\test2>
```

TAREA

- Ejercicio Videojuegos