ESTRUCTURA DE LENGUAJES

Python – parte 3

Funciones Adicionales de Python

MSc. Jimena Adriana Timaná Peña

Función: MAP

- La función map() en Python se utiliza para aplicar una función específica a cada uno de los elementos de una secuencia o un iterable (como una lista, dicionario, etc.) y devolver un iterador con los resultados.

- Es una manera concisa de transformar datos sin necesidad de escribir bucles explícitos.

- En lugar de usar un bucle for, la función map() proporciona una forma de aplicar una función a cada elemento en un iterable.

Función: MAP

Sintaxis:

map(function, Objetoiterable)

la función map toma dos parámetros.:

- El primer parámetro describe la función que se aplicará a cada elemento del iterable
- El segundo parámetro es el propio objeto iterable(lista, diccionario, tupla, etc) que se quiere recorrer. Pueden ser varios.
- Tener en cuenta que la función devuelve un objeto tipo map(que es un iterador). Por lo tanto, se podrá convertir a una lista, por ejemplo, para ver y trabajar con sus datos.

Ejercicio 1: Uso con MAP

- Dada una lista de números, calcular el cuadrado de cada uno de ellos.

Versión 1:

```
def cuadrado(x):
    return x ** 2

# Lista de números
numeros = [1, 2, 3, 4, 5]

# Usar map() para aplicar la función 'cuadrado' a cada elemento de la lista 'numeros'
resultado = map(cuadrado, numeros)

# imprimir resultado , que se obtuvo?
print(resultado)
```

Ejercicio 1: Uso con MAP

- Dada una lista de números, calcular el cuadrado de cada uno de ellos.

Versión 1:

```
# Convertir el resultado (iterador) en una lista para poder visualizarlo
print(list(resultado))
```

```
Resultado:
[1, 4, 9, 16, 25]
```

Ejercicio 1: Uso con MAP

- Dada una lista de números, calcular el cuadrado de cada uno de ellos.

Versión 2: Uso con Funciones Lambda:

También se puede utilizar map () con funciones lambda, funciones sin nombre o funciones anónimas (lambda functions). Esto es útil cuando la función es sencilla y no se necesita definirla previamente.

```
# Usar una función lambda en lugar de definir 'cuadrado'
numeros = [1, 2, 3, 4, 5]
resultado = map(lambda x: x ** 2, numeros)
print(list(resultado))
```

Ejercicio 2: Uso con MAP

- sumar elementos de dos listas.

Versión 1:

```
# Listas de números
a = [1, 2, 3]
b = [4, 5, 6]

def suma(n, m):
    return n + m
list(map(suma, a, b))
```

Versión 2:

```
list(map(lambda m, n: m + n, a, b))
```

Ejercicio 3: Uso con MAP

- sumar elementos de tres listas.

```
a = [1, 2, 3, 4, 5]
b = [2, 4, 6]
c = [1, 2, 4, 8]

list(map(lambda m, n, o: m + n + o, a, b, c))

Resultado:
[4, 8, 13]
```

Nota: Si los iterables incluidos como argumentos no tienen el mismo número de elementos, la función map() aplica la función en paralelo hasta alcanzar el final del iterable más corto.

Ejercicio 4: Uso con MAP

- Dada una lista de temperaturas en grados Celsius convertirlas a grados Fahrenheit utilizando map() y una función lambda.
- Recuerde la fórmula: *F= C* 1.8 +32*

```
# Lista de temperaturas en grados Celsius
celsius = [0, 20, 37, 100]

# Usar map() con una función lambda para convertir cada temperatura a Fahrenheit
fahrenheit = map(lambda c: c * 1.8 + 32, celsius)

# Convertir el resultado a una lista y mostrarlo
print(list(fahrenheit))
```

Función: FILTER

- La función filter() en Python se utiliza para filtrar elementos de un iterable (como por ejemplo una lista) que cumplen con una condición específica. Al igual que map(), filter() devuelve un iterador, por lo que generalmente tenemos que convertir el resultado en una lista para ver los elementos filtrados.
- Esta función filtra los elementos de una lista usando un determinado criterio.
- La función filter es similar a un bucle con un if, pero su uso es más rápido.

Ejemplo:

```
lista = [9,-3,7,-2,0,1,-12]
menor_cero = list(filter(lambda x: x < 0, lista))
print(menor_cero)
```

Resultado:

```
[-3, -2, -12]
```

Ejercicio 1: Uso con FILTER

- Dada una lista de números, indicar solo los números pares utilizando filter() y una función lambda.

Solución:

```
# Lista de números
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Usar filter() con una función lambda para filtrar los números pares
numeros_pares = filter(lambda x: x % 2 == 0, numeros)

# Convertir el resultado a una lista y mostrarlo
print(list(numeros_pares))
```

Ejercicio 2: Uso con FILTER

- Dada una lista de nombres, filtrar solo aquellos que comienzan con la letra "A".

Solución:

```
# Lista de nombres
nombres = ["Alonso", "Carlos", "Andrea", "Jimena",
"ana"]

# Usar filter() con una función lambda para filtrar
los nombres que comienzan con 'A'
nombres_con_a = filter(lambda nombre: nombre.startswith("A"),
nombres)

# Convertir el resultado a una lista y mostrarlo
print(list(nombres_con_a))
```

Función: REDUCE

- Próxima clase, averiguar en que consiste la función reduce.