Fundamentos de Programación

Grado en Ingeniería del Software

24/10/23

Antecedentes

- Estructura básica de un programa
- Lectura de ficheros
- Esquemas de filtrado/transformación
- Paso de parámetros
- Descomposición de problemas en funciones
- Type hinting

Índice

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

FILTRADO Y TRANSFORMACIÓN POR COMPRENSIÓN

```python
def consonants_for(sentence):
    result = []
    for x in sentence:
        result.append(x.upper())
    return result

sentence = 'we are studying list comprehensions'
print("With For Loop   : " + ''.join(consonants_for(sentence)))
```

```python
def consonants_lc(sentence):
    return [x.upper() for x in sentence]

sentence = 'we are studying list comprehensions'
print("With List Comprehension   : " + ''.join(consonants_lc(sentence)))
```

```
VOWELS = 'aeiou'

def consonants_for(sentence):
    result = []
    for x in sentence:
        if x not in VOWELS:
            result.append(x)
    return result

sentence = 'we are studying list comprehensions'
print("With For Loop    : " + ''.join(consonants_for(sentence)))
```

```
VOWELS = 'aeiou'

def consonants_lc(sentence):
    return [x for x in sentence if x not in VOWELS]

sentence = 'we are studying list comprehensions'
print("With List Comprehension    : " + ''.join(consonants_lc(sentence)))
```

CONTADOR

```python
VOWELS = 'aeiou'

def contador(sentence):
    result = 0
    for x in sentence:
        if x in VOWELS:
            result+=1
    return result

def contador2(sentence):
    result = []
    for x in sentence:
        if x in VOWELS:
            result.append(x)
    return len(result)

sentence = 'we are studying list comprehensions'
print(contador(sentence))
print(contador2(sentence))
```

SUMA

```python
lista = [600, 1, 2, -1]

def suma(datos):
    result = 0
    for x in datos:
        result += x
    return result


print(suma(lista))
print(sum (lista))
```

MÁXIMO Y MÍNIMO POR DEFECTO

```python
lista = [600, 1, 2, -1]

def max_(datos):
    result = None
    for x in datos:
        if result == None or result<x:
            result = x
    return result

def min_(datos):
    result = None
    for x in datos:
        if result == None or result>x:
            result = x
    return result

print(max(lista), max_(lista), min(lista), min_(lista))
```

ORDENACIÓN POR DEFECTO

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)

```
>>> array = [8, 2, 6, 4, 5]
>>> sorted(array)
[2, 4, 5, 6, 8]
```

TAREA

- Preparar control del jueves.
- Seguir trabajando en los ejercicios de clase avistamientos, sevici, etc. Completar resto de funciones.