



Semana 10: Funciones (II)

Programación

2020 - 2021

Ejercicio 1. Defina una función que devuelva una tupla imitando el comportamiento de la función `range(start, stop, step)`. Configure los parámetros para que se pueda ejecutar también `myrange(start, stop)` y `myrange(stop)`. Se debe comprobar que los rangos son correctos y tienen el tipo correcto (`int`) en caso contrario devolver una tupla vacía.

Ejercicio 2. Defina un programa que devuelva una lista a partir de la combinación dos listas pasadas como argumento, eliminando los duplicados: `combine(list1, list2)`.

Ejemplo:

```
list1: [1, 2, 3, 4, 5, 6]
```

```
list2: [4, 5, 6, 7, 8, 9]
```

```
Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Ejercicio 3. Defina un programa que contenga el conjunto de funciones siguientes sin usar los métodos de lista vistos en la clase de teoría. Debe realizar las comprobaciones necesarias para garantizar las operaciones.

`count(list1, x)`: devuelve el número de elementos que coinciden con `x` en `list1`.

`index(list1, x)`: devuelve la posición que ocupa `x` en `list1`. Si no existe, devolverá `None`.

`append(list1, x)`: añade el elemento `x` en la última posición de `list1`.

`find(list1, x)`: busca el elemento `x` dentro de `list1` y devuelve un valor booleano indicando si lo encontró o no.

`insert(list1, x, index)`: inserta el elemento `x` en la posición `index` de `list1`.

`remove(list1, x)`: elimina el primer valor que coincide con `x` de `list1`.

`removeAll(list1, x)`: elimina todos los elementos que coinciden con `x` de `list1`.

`clear(list1)`: elimina todos los elementos de `list1`.

`pop(list1)`: elimina el último valor de la lista y lo devuelve.

Ejercicio 4. La secuencia de Fibonacci $f(n)$ viene determinada por la siguiente fórmula:

si $n = 0$ $f(n) = 0$

si $n = 1$ $f(n) = 1$

si $n > 1$ $f(n) = f(n - 1) + f(n - 2)$

Definir una función que imprima la secuencia de Fibonacci a partir de un número dado.

Ejemplo:

```
fibonacci(7)
```

```
Output: [0, 1, 1, 2, 3, 5, 8, 13]
```

- Ejercicio 5.** Crear funciones para (a) pedir al usuario un número positivo en el rango (1, 10), (b) generar una lista aleatoria de ese número de elementos, y (c) encontrar el mínimo de esa lista. Crear un programa que use las funciones anteriores.
- Ejercicio 6.** Crear una función que reciba como parámetros una lista y un elemento y devuelva una tupla con las posiciones que ese elemento ocupa en la lista. Si no está en la lista devolverá la tupla vacía. Crear otra función que, usando la primera, reciba como parámetros dos listas y un elemento y devuelva si alguna de las apariciones del elemento coincide en la misma posición en las dos listas.

Normas de entrega

Los ejercicios se deben subir a Aula Global antes de las 7:00 de la mañana del próximo día con clase de laboratorio). Se deberá subir un fichero comprimido **zip** con un fichero para cada programa, nombrados ejercicio1.py, ejercicio2.py, etc. El nombre del fichero será “s10-iniciales-del-alumno.zip” (por ejemplo Lucía Pérez Gómez subirá un archivo llamado s10-lpg.zip).