

Pasos hacia la programación con Python (I)

Introducción

A continuación vamos a realizar una serie de ejercicios de cara a repasar algunos de los fundamentos de Python. Repasaremos las partes más importantes y sobre las que más hemos hecho hincapié. En concreto:

- Declaración de variables y tipos.
- Estructuras de datos 1:
 - Listas.
 - Diccionarios.
- Estructuras de datos 2:
 - Como indexar una lista.
 - Como modificar un elemento indexado
 - Como añadir un elemento a la lista.
 - Como acceder a un valor de un diccionario por clave.
 - Como modificar un valor del diccionario
 - Como añadir elementos al diccionario.

La idea detrás es estar preparados para la siguiente parte en la que analizaremos las estructuras de control de flujo con mayor detalle e introduciremos cuestiones nuevas sobre funciones y sobre el trabajo sobre estructuras de datos en el contexto de los bucles.

Repaso

Tenéis disponible el repaso en el repositorio de GitHub `python_basics_1` podéis clonar el repositorio desde la url. Esta en el archivo `repaso.py`

ejercicio1.py

1. Declara una lista vacía llamada lista.
2. Crea un bucle for en un rango de 0 a 100 donde sea la variable `n` la que tome estos valores.
3. Dentro del bucle emplea la función `append` para llenar esta lista con todos los valores que ha ido tomando la variable `n`.

4. Haz un print del primer elemento de la lista.
5. Haz un print del ultimo elemento de la lista.
6. Haz un print de una porción de la lista del elemento 25 al 50. (Recuerda que las listas empiezan en 0 y no en 1).
7. Asigna a una variable llamada sublista los valores del print anterior.
8. Toma el primer elemento de esta sublista y cambia su valor por 1000.
9. Haz un print de lista.
10. Haz un print de sublista.
11. ¿Que a ocurrido con el primer elemento en ambos casos?

ejercicio2.py

1. Declara un diccionario vacío llamado diccionario.
2. Declara una variable llamada *count* e igualala a 0.
3. Crea un bucle for en un rango de 0 a 100 donde sea la variable n la que tome estos valores.
4. Dentro del bucle realiza las siguientes acciones:
 1. Declara una variable llamada key que debe tomar en cada iteración del bucle el valor de tipo string: "key_n" (Es decir: "key_0", "key_1", "key_2", ...).
 2. Añade al diccionario en cada iteración del bucle el item: "key_n": n (Es decir: "key_0": 0, "key_1": 1, "key_2": 2, ...)
5. Una vez llenado el diccionario haz un print del valor asignado a "key_0" y a "key_99".
6. Modifica el valor asignado a la clave "key_64" por 1000.
7. Haz un print del valor asignado a "key_64".

ejercicio_final_repaso.py

Con todo lo anterior y algunas cosas ya vistas vamos a realizar de manera individual el siguiente programa. Se trata de crear un diccionario con dos claves: "año bisiesto" y "año no bisiesto". El valor asociado a cada una de estas claves a de ser una lista que contenga en formato entero los años que son bisiestos y no respectivamente desde el año 0 al 2024.

Ejemplo:

Para ello usareis la función *es_bisiesto* que esta ya implementada en la plantilla del ejercicio. La función en sí ya devuelve el string “año bisiesto” o “año no bisiesto” según corresponda. Por ello NO debéis usar en ningún momento condicionales.

Una vez creado el diccionario quiero que mostréis de manera separada (primero un print y luego el otro) los años pares bisiestos y los no bisiestos. Para esto SI debéis usar un condicional y el operador módulo.

Finalmente quiero que calculéis la proporción de años bisiestos sobre el total de años y la proporción de años bisiestos pares sobre el total de años bisiestos. Es decir:

$n_años_bisiestos / n_años_no_bisiestos$

$n_años_bisiestos_pares / n_años_bisiestos$

Para ello deberéis usar la función *len* que da el valor del número de elementos de una lista y en el caso de los años pares almacenar en alguna lista estos años que printasteis anteriormente o utilizar un contador para contarlos.