

Práctico 0: Introducción a Python

3 de marzo de 2023

A continuación se plantean diferentes ejercicios para realizar en Python. Previamente se debe instalar algún entorno de desarrollo que maneje este lenguaje, Anaconda- Spyder por ejemplo. Se recomienda fuertemente realizar **todos** los ejercicios, ya que en ellos, se plantean situaciones con las que se toparán a lo largo del curso.

1. Tipos de datos y listas

1. Crear una lista *lista1* a mano con las primeras tres letras del abecedario en tipo *str*, de la siguiente manera: $lista1 = ['a', 'b', 'c']$.
2. Crear una lista *lista2* a mano con los primeros tres números naturales (comenzar con el cero), en tipo *int* (entero).
3. Agregar al final de cada una de las listas el elemento siguiente, con el método *append* (la siguiente letra, o el siguiente número).
4. Agregar al final de cada lista un dato de tipo distinto a los demás, por ejemplo un número entero en *lista1*.
5. Eliminar el elemento agregado en el punto anterior de cada una de las listas con el método *pop*.
6. Imprimir en pantalla el elemento de la posición 1 de cada lista (tener en cuenta que el primer elemento está en la posición 0).
7. Definir una lista *lista2a* de la siguiente forma: $lista2a = lista2[a:b]$, donde se deben elegir valores de *a* y *b*. Estos, permiten definir el rango de valores de *lista2*. Observar cómo queda definida *lista2a* eligiendo distintos valores de *a* y *b*. Además, probar con los siguientes rangos: $[:]$, $[:3]$, $[1:]$, $[:-1]$.
Notar que de esta manera se pueden eliminar tantos elementos del principio o final como se quiera.
8. Crear una lista *lista3* vacía.
9.
 - a) Usar cuatro veces el método *append* para agregar los elementos a esta lista. Cada uno de estos, debe estar compuesto por dos valores: los que se encuentran en la misma posición en *lista1* y *lista2*. Por ejemplo, el primer elemento de *lista3* será $[a, 0]$.
 - b) Si luego de realizar esto se cambian los valores de *lista1* y *lista2* en el principio del código, ¿*lista3* se actualiza automáticamente a los nuevos valores? En caso que no, volver a escribir la líneas correspondientes para que sí lo haga.
10. Notar que *lista3* se puede ver como una matriz que tiene cuatro filas y dos columnas. Para acceder a un elemento de la misma se hace: $lista3[filas][columna]$. Usando lo anterior, imprimir en consola: “La letra a es la número 1 del abecedario”, donde cada caracter subrayado tiene que ser obtenido accediendo a un elemento de *lista3*.
11. Crear una lista *lista4* igual a *lista2*, pero estableciendo los valores en tipo *float*.
12. En consola, realizar sumas y multiplicaciones tomando valores de: *lista1*, *lista2* y *lista4*. Cubrir todas las posibles combinaciones de listas de las que se usan valores. ¿Cuál es el tipo de dato de cada uno de los resultados?

2. Numpy y matplotlib.pyplot

Numpy y *matplotlib.pyplot* corresponden a librerías de Python. La primera contiene una cantidad de funciones que resultan muy útiles, muchas de las cuales se tocarán a continuación, y la segunda, brinda un marco para la visualización de datos. Para poder usarlas, en primer lugar, se deben instalar las librerías. Para hacer esto, ir a la consola (Anaconda Prompt) e introducir los siguientes comandos por separado: `pip install numpy`, `pip install matplotlib`. Luego, en cada código que vaya a usar las librerías, es necesario importarlas, para ello, escribir al comienzo del código: `import numpy as np` y en otra línea, `import matplotlib.pyplot as plt`.

1. Hay una estructura de datos similar a las listas, pero a la que se le pueden aplicar otras funciones, los arreglos *numpy*. Se puede pasar de un formato a otro fácilmente:

- `lista2 = np.array(lista2)`
- `lista2 = list(lista2)`

Realizar estas transformaciones, y verificar que al final *lista2* se mantuvo incambiado.

2. Convertir *lista3* a arreglo *numpy*. Imprimir en pantalla las listas *lista1* y *lista2* por separado, **solamente** usando la información de *lista3*. Aplicar la siguiente información de ejemplo: `lista3[0, :]` devuelve lo que hay en la fila cero de esta lista, o visto de otra forma, todos los valores del par que compone el elemento cero de la lista.
3. ¿Qué pasó con el tipo de dato que se exhibe al extraer *lista2* de *lista3*?
4. Utilizando correctamente los rangos, usar *lista3* para devolver en consola el primer valor de los últimos dos elementos de dicha lista.
5. Probar hacer lo anterior sin convertir *lista3* a arreglo *numpy* previamente.
6. Crear un arreglo *t1* que vaya desde 0 hasta *largo* y tenga paso *h*, usando *numpy arange*, donde *largo* = 10 y *h* = 0,1, y se definen antes de crear el arreglo.
7. Crear un arreglo *t2* que vaya desde 0 hasta *largo* y tenga paso *h*, usando *numpy linspace*. ¿Qué diferencias hay entre *t1* y *t2*?
8. Crear un arreglo *sen* de igual largo que *t1*, y de frecuencia $f = 1Hz$ usando las funciones *numpy pi* y *numpy sin*, y considerando que *t1* es el tiempo. Definir *f* antes.
9. Crear una lista vacía *sen1*. Usando un bucle *for*, calcular en cada iteración el valor correspondiente de *sen1* y agregarlo a la lista, de manera que sea igual que la función seno del punto anterior en cada instante de tiempo.
10. Utilizando *t1*, y eligiendo alguno de los dos métodos anteriores, calcular vectores de otras funciones: coseno, exponencial, logaritmo, constante (investigar *numpy ones*).
11. El código que se muestra a continuación, corresponde a un ejemplo de uso de la librería *matplotlib.pyplot*, para graficar vectores.

```
1 plt.figure()
2 plt.plot(t1, funcion1, '-.')
3 plt.plot(t1, funcion2)
4 plt.legend(['Funcion 1', 'Funcion 2'])
5 plt.xlabel('Tiempo')
6 plt.ylabel('y(t)')
```

Reciclar este código para graficar las funciones matemáticas implementadas, de forma individual y conjunta. Cambiar *t1* por *t2*, ¿qué sucede?

Cambiar el valor de *h* de *t1*, ¿qué sucede al graficar?

12. Crear una lista *fib* con los primeros dos elementos de la serie Fibonacci. Hacer un bucle *for* donde en cada iteración se agregue un elemento nuevo a la lista, habiéndolo calculado usando la información de *fib*. Definir previamente la cantidad de elementos de la lista que se van a calcular.