

Actividad III - Listas

Exactas Programa

Invierno 2019

Esta serie de actividades están orientadas para practicar los contenidos vistos en la clase de introducción, específicamente la definición y manipulación de listas.

No copies y pegues el ítem anterior, escribí desde cero cada ejercicio. La idea es que aprendas a escribir en **Python**.
Cada ejercicio que hagas, probalo en el **Python Tutor** (<http://pythontutor.com>).

Listas de números, letras, y mucho más.

Una lista es una *estructura de datos* que nos permite *guardar* elementos de forma *ordenada*. Es una manera muy cómoda de tener una serie de valores y se usa para representar una gran cantidad de situaciones que se manejan en los programas.

En **Python** se puede definir una lista dando sus elementos, por ejemplo:

```
1 A = [1,2,3,4,5]
2 B = [5,4,3,2,1]
3 C = ['a', 'b', 'c', 'd']
4 D = [True, False, True]
5 res = len(A) + len(B)
6 print(res)
7 res2=A[0]
8 print(res2)
9 A[4]=7
10 print(A)
11 if A==B:
12     print("Son iguales")
13 else:
14     print("Ufa!")
```

En la línea 1 se define la variable **A** que es una lista de números que van del 1 al 5, en tanto que en la línea 2, se define otra variable que es una lista de números ordenados de forma descendente desde el 5 al 1. Como en las listas **importa** el orden de los elementos, la lista de la línea 1 y la de la línea 2 **no** son iguales entre sí

La línea 3 muestra una definición de la variable **C** que es una lista de caracteres, en tanto que la siguiente línea define una lista de valores de verdad (textttTrue y **False**). La línea 5 ejemplifica el uso de la función **len** que toma una lista y devuelve la cantidad de elementos que posee.

Se pueden acceder a los distintos elementos que contiene una lista. Esto se puede hacer mediante el *operador* **[]** (en la jerga se lo conoce amigablemente como corchete-corchete). La línea 7 usa este operador para asignar a la variable **res2** el valor que tiene la lista **A** en su primer elemento (el elemento a acceder está identificado por su posición y en **Python** se numeran las cosas a partir de 0). Las listas también permiten modificar su contenido con este mismo operador, en la línea 9 se muestre cómo se cambia el valor del último elemento de la lista **A** (queda el valor 4 dos veces en la lista).

Así como usábamos las comparaciones entre números o caracteres, también se pueden usar entre listas como se hace en la línea 11, donde se tomará un camino u otro de acuerdo a si las listas **A** y **B** son iguales. Para que dos listas sean iguales, tienen que tener los mismos elementos en el mismo orden.

Hay una lista *especial* que es la que no tiene ningún elemento, en Python se define así: [] ('corchete que abre' seguido de 'corchete que cierra'). La lista vacía no contiene ningún elemento y su longitud (`len`) es 0 (cero).

Ejercicios para hacer

1. Completar y probar:

```
def devolver_longitud_de_un_nombre(unNombre):  
    return <completar>
```

2. Completar y probar:

```
def devolver_primer_elemento_de_la_lista(unalista):  
    return unaLista[<completar>]
```

3. Completar y probar:

```
def devolver_segundo_elemento_del_nombre(unNombre):  
    return <completar>
```

4. Completar y probar:

```
def devolver_ultimo_elemento_del_nombre(unNombre):  
    return unNombre[<completar>]
```

5. Completar y probar:

```
def devolver_la_letra_en_posicion_del_nombre(unNombre, posicion):  
    return <completar>
```

6. Completar y probar:

```
def reemplazar_ultimo_elemento_de_la_lista(unalista, unElemento):  
    unaLista[<completar>] = <completar>  
    return unaLista
```

7. Completar y probar:

```
def agregar_25_al_final_de_la_lista():  
    _lista = ["Casa", 5, "A"]  
    _elemento = <completar>  
    _lista.append(<completar>)  
    return <completar>
```

8. Completar y probar:

```
def agregar_nombre_al_final_de_la_lista(unNombre):  
    _lista = ["Casa", 5, "A"]  
    _lista.append(<completar>)  
    return <completar>
```

9. Definir una función que recibe una lista de números y devuelve esa lista pero con el primer elemento multiplicado por 2
10. Definir una función que recibe una lista de números y devuelve esa lista pero con el último elemento multiplicado por 3
11. Definir una función que recibe una lista de números y devuelve esa lista pero con el primero y el último elemento multiplicados por 2 y 3 respectivamente (todo en una única función pero usando las funciones definidas en los dos puntos anteriores)
12. Definir una función que recibe una lista y un elemento, y devuelve la lista con ese elemento agregado al final