

1. Tal como se discutió en clases, las *listas* constituyen un tipo de objeto muy importante en Python. Las listas son similares a las cadenas, excepto que cada elemento puede ser de un tipo diferente. La sintaxis para crear listas en Python es `[objeto1, objeto2, ..., último-objeto]`. Ver [este](#) video en Canvas para un repaso de lo visto en clases.

Por ejemplo, ejecute:

```
lista = [1, "hola", 1.0, 1-1j, True]
type(lista)
print(lista)
```

Como puede ver, la variable `lista` es un nuevo tipo de objeto: 'list'. En este caso, es una lista cuyos elementos son un entero, un string, un float, un complejo, y un booleano. Para verificar esto, imprima el valor y el tipo de cada elemento de la lista. Por ejemplo,

```
print(lista[0],type(lista[0]))
print(lista[1],type(lista[1]))
```

Este ejemplo también muestra que los índices de cada elemento de la lista son numerados de la misma manera que en un string:

```
print(lista[0:3])
print(lista[:2])
```

2. Los elementos de una lista pueden tener cualquier tipo reconocido por Python, por ejemplo, pueden ser otra lista!:

```
superlista = ["cool",lista]
print(superlista)
```

Imprima el valor y el tipo de cada elementos de esta lista. ¿Cuántos elementos tiene la lista `superlista`? (respuesta, use la función `len()`).

3. Existen diversas funciones en Python que crean listas. La función `list()` crea una lista, por ejemplo, a partir de un string. Usando el string `x` definido anteriormente, ejecute

```
y = list(x)
print(y)
print(type(y))
```

4. Otra función que crea listas útiles, esta vez de números *enteros*, es `range(inicio,fin,paso)`, que crea una lista de valores desde `inicio` (cerrado) hasta `fin` (abierto!!), con paso `paso`. Ejecute,

```
z = list(range(2,26,3))  
print(z)
```

5. ¿Qué hacen los siguientes comandos?, ¿Modifican el valor de `x` y/o `lista`?

```
x.split(" ")  
x.split("e")  
lista.append("chao")  
lista.insert(2,"cool")
```

6. En Python, tal como en todo lenguaje de programación moderno, existen comandos de **control de flujo**, que permiten escribir programas que se comporten en formas distintas dependiendo de algunas condiciones que se definan. Para familiarizarse con la lógica y la sintaxis de los comandos de control de flujo en Python vea los videos en [esta página](#) de Canvas y reproduzca los ejemplos ahí explicados.
7. Estudie los distintos casos de uso de los comandos `if`, `elif` y `else` descritos en [este pdf](#) disponible en Canvas. Realice una modificación del código de cada caso, agregando una función `input` para que se pueda ingresar el valor de `x` desde el teclado, y asegúrese que entiende la lógica de cómo funciona su código.
8. Escriba un programa que al ejecutarlo pregunte al usuario un número e imprima su valor absoluto. Recuerde que el valor absoluto (o módulo)  $|x|$  de un valor real  $x$  es definido por

$$|x| := \begin{cases} x, & \text{si } x > 0 \\ -x, & \text{si } x < 0 \end{cases} . \quad (1)$$

9. Usando lo que aprendió sobre el comando `if` y asociados, modifique el programa `test.py` que creó en la guía 08 y que resuelve la ecuación cuadrática  $ax^2 + bx + c = 0$ , para que ahora el programa informe que existen dos soluciones reales, y las imprima, si el discriminante  $b^2 - 4ac$  es positivo, o que informe que no existe solución real (si el discriminante es negativo), o bien que informe que existe sólo una solución real, y la imprima (si el discriminante es nulo).
10. Escriba un programa que ordene (e imprima) de menor a mayor, una secuencia de 3 números que son entradas a través del teclado, además que calcule e imprima el promedio de ellos.
11. Escriba un programa que permita ingresar dos números reales ( $a$  y  $b$ ) por el teclado. Si  $a$  es mayor que  $b$  el programa calcula e imprime la suma, en caso contrario el producto, y que si  $a$  es igual a  $b$  muestre la resta.
12. Escriba un programa para determinar si un entero ingresado por el teclado es o no un cuadrado perfecto.
13. Descargue el libro “*Algoritmos y Programación I: Aprendiendo a programar usando Python como herramienta*”, de la Facultad de Ingeniería de la Universidad de Buenos Aires, desde el sitio <https://algoritmos1rw.ddns.net/> (sección “Material”, archivo “Apunte”). Link directo [aquí](#). Atesórelo y estúdielo el resto de sus días.
14. Escriba un programa que resuelva el ejercicio 4.6.5 a) planteado en el libro descargado en el punto anterior (página 63).