

# **Listas - Parte I**

## Computer Science

**CS1100 - Introducción a Ciencia de la Computación**

# Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Comprender el concepto de listas.

# Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Comprender el concepto de listas.
- Crear listas en python.

# Listas

## Definición

- Python permite agrupar elementos.
- Uno de estos agrupamientos se le conoce como [lista](#).
- Las listas son mutables, es decir modificables, y permiten tener elementos repetidos.
- Para definir una lista sus elementos se encierran entre corchetes y se separan por comas.

```
1 numeros = [1, 2, 3]
```

```
1 palabras = ["hola", "como", "te", "llamas"]
```

# Índices

## Definición

- Los índices nos permiten seleccionar algún o algunos elementos de una lista.

```
1 numeros=[1,2,3,4,5,6]
2 print(numeros[0])
3 print(numeros[0:2])
4 print(numeros[2:])
```

# Listas

¿Cuál es el último elemento de una lista?

- Hay que tomar en cuenta que el primer elemento tiene como índice cero, lo que conlleva a que el último elemento tenga como índice  $N-1$ , donde  $N$  es el número de elementos de la lista.

```
1 lista=[1,2,3]
2 print(lista[0])
3 print(lista[2])
4 print(lista[3]) #Error
```

# Mutable e inmutables

## Lista vs String

- Una comparación entre strings y listas nos ayudará a ver la diferencia entre un objeto cuyos elementos son **inmutables** y otro cuyos elementos son **mutables**.

```
1 lista=[1,2,3]
2 lista[2]=5
3 cadena="mama"
4 print(cadena[1])
5 cadena[1]="x" #error
```

# Tipos de datos

## Dentro de una lista...

- Los elementos en una lista pueden ser del mismo tipo o de diferentes tipos.

```
1 numeros = [1, 2, 3]
2 padres  = ["mama", "papa"]
```



# Listas dentro de listas

## Dentro de una lista...

- Una lista puede contener listas.

```
1 lista1=[1,2,3,["mama","papa"],[3,2,1]]  
2 lista2=[["utopia","mediatica"],["IQ",135]]
```

# Listas vacias

## Dentro de una lista...

- Una lista puede crearse sin valores. Si a una lista se le da un valor inicial `None`, no es una lista vacía. Además, algunos elementos en una lista pueden ser variables ya existentes.

```
1 lista1 = []  
2 lista2 = [None]  
3 pepe="hola, soy pepe"  
4 lista3=[pepe, "el primer elemento es una  
    variable"]
```

# Sumando listas

## Definición

- Una lista puede ser sumada a otra lista.

```
1 padres=["mama", "papa"]
2 abuelos=["abuela", "abuelo", "hijo"]
3 familia=padres + abuelos
4 print(familia)
```

# Listas

## Agregar elementos a una lista

- Podemos usar `append` para agregar elementos a una lista.

```
1 lista=[1,2,3]
2 lista.append("hola")
3 print(lista)
```

# Listas

## Recorrer una lista

- Podemos recorrer una lista con un for, de más de una manera.

```
1 lista=[1,2,3]
2 for i in lista:
3     print(i)
4 for i in range(len(lista)):
5     print(lista[i])
```

# Ejemplo 1

## Enunciado

Escribe una función que cuenta la cantidad de vocales que tiene una palabra:

```
1 def contar_vocales (palabra):  
2     vocales = 0  
3     for c in palabra:  
4         if c in [a,e,i,o,u]:  
5             vocales = vocales + 1  
6     return vocales  
7 print(contar_vocales( hola ))
```

# Ejercicio 1

## Enunciado

El programa recibe  $N$  números, y luego imprime dos listas, la primera con los números pares recibidos, y luego otra lista con los números impares.

## Ejercicio 2

### Enunciado

El usuario nos brindará N nombres, los cuales serán guardados en una lista, luego, deberemos devolver la posición en la lista de la primera ocurrencia del nombre a buscar. En caso no encuentre al nombre devolverá el texto "Nombre no encontrado".



# Ejercicio 3

## Enunciado

Recibirás varias palabras, y crearás una lista con todas las palabras que empiezan con vocal, y otra lista con todas las palabras que empiezan con consonante. Finalmente, imprimirás ambas listas.

# Ejercicio 4

## Enunciado

Dado una lista de valores, escriba un programa que permita identificar si un elemento se repite varias veces, luego proceder a dejar solo la primera ocurrencia del elemento y eliminar el resto de apariciones.

# Evaluación

## Individual Work

■ [www.hackerrank.com/PORDEFINIR](http://www.hackerrank.com/PORDEFINIR)

## Conclusiones

- ¿Qué son las listas?
- ¿Qué quiere decir que son mutables?