

Pensamiento Computacional

Listas



Real Python

Características

Las listas:

- Pueden tener elementos de distintos tipos
- Son mutables – Pueden agregarse, eliminarse o cambiarse los elementos que la conforman.
- Se pueden obtener sus elementos a través de un índice. ¿?
- Veamos este video:
- <https://youtu.be/mrwSbE5MDn0>



Ejercicio:

```
>>>decenas=[10, 20, 30, 40]
```

```
>>>print (decenas[0])
```

 ¿qué valor imprime?

```
>>>print (decenas[3])
```

 ¿Qué valor imprime?

```
>>>print (decenas[-1])
```

 ¿Qué valor imprime?

```
>>>print (decenas[4])
```

 ¿Qué valor imprime?

Caracteres :

P	y	t	h	o	n
0	1	2	3	4	5
-6	-5	-4	-3	-2	-1

Índice :

Índice inverso :

¿Cómo puedo cambiar uno de los elementos?

¿Cómo puedo agregar un elemento?

¿Cuál será la ventaja de tener listas?

Realiza los ejercicios de la hoja de trabajo Listas

Indexing y Slicing

- **Indexing**

Obtener un elemento por su posición.

- **Slicing**

Obtener un segmento del string.

Mutables:

- lista[1]="algo"

```
>>> lista=['H', 'o', 'l', 'a', 'a', 't','o','d','o']
>>> lista[2]
'l'
>>> lista[-1]
's'
>>> lista[-3]
'd'
>>> lista[1:4]
['o', 'l', 'a']
>>> lista[1:4:2]
['o', 'a']
>>> lista[4:0:-1]
['a', 'a', 'l', 'o']
>>> lista[::-1]
['s', 'o', 'd', 'o', 't', 'a', 'a', 'l', 'o', 'H']
```

<https://youtu.be/mrwSbE5MDn0>

Uso de funciones sobre listas:

```
>>> decenas=[10, 20, 30, 40]
```

- `len(decenas)`
 - Regresa el número de elementos de la lista
- `max(decenas)`
 - Regresa el valor máximo de la lista
- `min(decenas)`
 - Regresa el valor mínimo de la lista
- `sum(decenas)`
 - Regresa la suma de los elementos de la lista

Para recorrer las listas usamos un ciclo for

- La forma general de un ciclo `for` es la siguiente:

`for elemento in` Lista:

acciones a realizar

`elemento` es sólo una variable que nos sirve para ir tomando en cada iteración un valor de la lista.

Ejemplo:

```
gustos=['viajar', 'comer', 'dormir', 'programar']
```

```
for cada in gustos:  
    print('A mi me gusta: ', cada)
```

¿Recuerdas Range?

`range(5)` ---- 0, 1, 2, 3, 4

`range(1,10)` ---- 1, 2, 3, 4, 5, 6, 7, 8,9

`range(2,100,2)` ---- 2, 4, 6, 8, 10, 12, 14, 16... 98

For iterando con listas

- **For en Python requiere una secuencia para poder iterar.**
- Una lista es una secuencia inmutable de objetos que pueden ser de diferente tipo
- Ejemplo de listas:
 lista1=[1, 2, "hola", 3.4]
 lista2=[[1,2], [3,4]]

```
lista=["Hugo", "Pedro", "Juan", "Daniel", "René"]  
print("Lista de alumnos:")  
for elemento in lista:  
    print(elemento)
```

Funciones y métodos de listas

Operadores que funcionan en listas:

- +
- *
- In
- not in

- Funciones aplicables a listas
 - `len(lista)` – tamaño
 - `max(lista)`
 - `min(lista)`
- Métodos comunes
 - `lista.append(obj)`
 - `lista.count(obj)`
 - `lista.extend(iterable)`
 - `lista.sort()`
 - `lista.insert(i, obj)`
 - `lista.reverse()`

A programar

THINK



CODE



Realiza los
ejercicios de la
hoja de trabajo
Listas

Ejercicio simple:

1. Crea una lista con 5 números 7
2. Modifica en la posición 3 por un 5
3. Inserta un 8 en la posición 1
4. Elimina el primer valor de la lista
5. En la posición 2, reemplázalo por un 6
6. Elimina la ultima posición de la lista
7. Ordena descendentemente la lista.

¿Cuál fue el resultado?

```
#Ejercicio de lista
```

```
#Se crea la lista
```

```
lista=[7,7,7,7,7]
```

```
#Se modifica la posición 3
```

```
lista[3]=5
```

```
#Se inserta en la posición 1 el 8
```

```
lista.insert(1,8)
```

```
#Se remueve el primer valor 7 de la lista
```

```
lista.remove(7)
```

```
#Se modifica la posición 2 por un 6
```

```
lista[2]=6
```

```
#Se elimina el último valor de la lista
```

```
lista.pop()
```

```
#Se organiza la lista Descendente
```

```
lista.sort(reverse=True)
```

```
print (lista)
```

Qué aprendimos...



Listas

Me permiten almacenar
diferentes tipos de datos



for & range

Me permite recorrer la lista



Indexing, slicing

Me permiten manipular el
contenido de la lista

Recursos recomendados

- Videos sobre listas desarrollado por Yolanda Martínez.

- <https://youtu.be/vSGLVjFbcqQ>