

[Intro](#) [Intro a la Programación](#) [Tipos de Datos](#) [Flujos de Control](#)[Estructuras de datos](#) [Iteradores e Iterables](#) [Funciones](#) [Clases y OOP](#)

Contenido de la clase

[Error Handling](#) [Manejo de Archivos](#) [Repaso](#) [Henry Challenge](#)Tiempo de lectura
Clases en vivo
9 min**Grabación de la Clase 4**

Videos Part-time

Principales Objetivos de**Aprendizaje para esta Clase****Estructuras de datos**

Lista

Tupla

Diccionario

Homework**Clase en vivo de Resolución de
ejercicios.**

Grabación de la Clase 4



Principales Objetivos de Aprendizaje para esta Clase

- Conocer la diferencia entre una variable y una estructura de datos
- Manejar el tipo de dato lista en Python

Estructuras de datos

Anteriormente se mencionó que un dato representa la realidad, y se presentó el concepto de variable, que es un elemento que nos permite guardar un dato dentro de nuestro programa. Sin embargo, rápidamente vamos a llegar a la

Dejanos tu feedback! 👍



complejas, en el caso de Python contamos con listas, tuplas y diccionarios.

Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de

Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

Lista

Una estructura de datos muy importante en Python es la lista, que consiste en una serie de elementos ordenados.

Esos elementos pueden ser de distinto tipo, e incluso pueden ser de tipo lista también.

Operaciones con listas:

■ Creacion

```
mi_lista = ['Rojo', 'Azul', 'Amarillo', 'Naranja',
```



■ Imprimir

```
print(mi_lista)
```

■ Ver el tipo de dato

```
type(mi_lista)
```

Las listas, así como otras estructuras de datos que se verán en adelante, tienen varios elementos, motivo por el cual cuando se quiere acceder en específico se requiere de un **índice** que va a hacer referencia al elemento dentro de la lista:

■ Imprimir el tercer elemento de la lista (el índice comienza en cero)

```
print(mi_lista[2])
```

Dejanos tu feedback! 👍



Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de

Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

```
print(mi_lista[0:2])
```

- Al no poner primer valor, Python asume que es un 0

```
>>> print(mi_lista[:2])  
['Rojo', 'Azul']
```

- Al no poner segundo valor, Python asume que se trata de todos los elementos a partir del primero

```
>>> print(mi_lista[0:])  
['Rojo', 'Azul', 'Amarillo', 'Naranja', 'Violeta']
```

- Agregar un elemento al final de la lista (Si el elemento ya existe va a quedar duplicado)

```
mi_lista.append('Blanco')
```

- Agregar un elemento especificando el índice

```
>>> mi_lista.insert(3, 'Negro')  
>>> print(mi_lista[:])  
['Rojo', 'Azul', 'Amarillo', 'Negro', 'Naranja', 'Blanco']
```

- Concatenar una nueva lista a la lista previamente creada

Dejanos tu feedback! 👍



Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

■ Encontrar el índice de un valor específico

```
>>> print(mi_lista.index('Azul'))
1
```

■ Eliminar un elemento de la lista (Si el elemento no existe va a arrojar un error)

```
>>> mi_lista.remove('Blanco')
-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_10044\308548076.py:1: in <module>
----> 1 mi_lista.remove('Blanco')

ValueError: list.remove(x): x not in list
>>> mi_lista.remove('Negro')
```

■ Extraer y recuperar el último elemento de la lista

```
>>> ultimo = mi_lista.pop()
>>> print(ultimo)
Gris
```

■ Multiplicar la lista 3 veces

```
>>> print(['a','b','c'] * 3)
['a', 'b', 'c', 'a', 'b', 'c', 'a', 'b', 'c']
```

■ Ordenar una lista de menor a mayor

```
>>> lista= [1,4,3,6,8,2]
>>> lista.sort()
```

Dejanos tu feedback! 👍



Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de

Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

■ Ordenar una lista de mayor a menor

```
>>> lista= [1,4,3,6,8,2]
>>> lista.sort(reverse=True)
>>> print(lista)
[8,6,4,3,2,1]
```

Tupla

Son una estructura similar a las listas, la diferencia está en que no se pueden modificar una vez creadas, es decir que son **inmutables**:

■ Convertir una lista a tupla

```
mi_tupla=tuple(mi_lista)
```

■ Imprimir el índice 1 de la tupla

```
>>> print(mi_tupla[1])
Azul
```

■ Evaluar si un elemento está contenido en la tupla (Devuelve un valor booleano)

```
>>> 'Rojo' in mi_tupla
True
```

■ Evaluar las veces que está un elemento específico

```
>>> mi_tupla.count('Rojo')
1
```

Dejanos tu feedback! 👍



```
mi_tupla_unitaria = ('Blanco',)
```

Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de

Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

■ Empaquetado de tupla, tupla sin paréntesis

```
mi_tupla='Gaspar', 5, 8, 1999
```

■ Desempaquetado de tupla, se guardan los valores en orden de las variables

```
>>> nombre, dia, mes, año = mi_tupla
>>> print("Nombre: ", nombre, " - Dia:", dia, '
Nombre: Gaspar - Dia: 5 - Mes: 8 - Año: 1999
```

■ Convertir una tupla en una lista

```
mi_lista=list(mi_tupla)
```

Diccionario

Un diccionario tiene una organización de 'clave' y 'valor':

■ Crear un diccionario

```
mi_diccionario = { 'Colores Primarios': ['Rojo', 'Azul', 'Verde'],
                   'Colores secundarios': ['Naranja', 'Púrpura', 'Amarillo'],
                   'Clave3': 10,
                   'Clave4': False}
```

■ Imprimir un valor a través de su clave

```
>>> print(mi_diccionario['Colores secundarios'])
```

Dejanos tu feedback! 👍



Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

■ Agregar un valor

```
mi_diccionario['Clave5']='Otro ejemplo'
```

■ Cambiar un valor

```
mi_diccionario['Clave3']=2
```

■ Eliminar un elemento de un diccionario a través de su clave

```
del mi_diccionario['Clave4']
```

■ Utilizar una tupla como clave de un diccionario

```
mi_tupla=("Argentina", "Italia", "Inglaterra")
mi_diccionario={mi_tupla[0]:"Buenos Aires",
                 mi_tupla[1]:"Roma",
                 mi_tupla[2]:"Londres"}
```

■ Colocar una tupla dentro de un diccionario

```
mi_diccionario={'Clave1':'Valor1', 'Clave2':(1,
```

■ Colocar una lista dentro de un diccionario (Notar que la diferencia está en el paréntesis '()' y el corchete '[]')

Dejanos tu feedback! 👍



Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de Aprendizaje para esta Clase

Estructuras de datos

Lista

Tupla

Diccionario

Homework

Clase en vivo de Resolución de ejercicios.

- Colocar un diccionario dentro de un diccionario

```
mi_diccionario={'Clave1':'Valor1', 'Clave2':{'r
```



- Imprimir las claves del diccionario

```
>>> print(mi_diccionario.keys())  
dict_keys(['Clave1', 'Clave2'])
```

- Imprimir los valores del diccionario

```
>>> print(mi_diccionario.values())  
dict_values(['Valor1', {'numeros': [1, 2, 3, 4
```



- Imprimir la longitud del diccionario

```
>>> len(mi_diccionario)  
2
```

Tips: La funcionalidad ****del**** permite eliminar cualquier estructura de datos y la funcionalidad ****len**** permite obtener el tamaño de la estructura de datos

Homework

Dejanos tu feedback! 👍

Completa la tarea descrita en el archivo [README](#)



Clase en vivo de Resolución de ejercicios.

Contenido de la clase

Grabación de la Clase 4

Principales Objetivos de

Aprendizaje para esta Clase

Estructuras de datos

Lista

Los martes/jueves (segun corresponda) a las 18 hs. ARG cada dos semanas hacemos una clase de apoyo en vivo sobre este tema. El link se comparte en el canal de Slack #anuncios ese día.

Hecho con  por alumnos de Henry

Homework

Clase en vivo de Resolución de ejercicios.

Dejanos tu feedback! 