



# Iniciación con Python

Clase 10 - "Diccionarios"

# ¡Les damos la bienvenida!



Vamos a comenzar a grabar la  
clase

## Clase 09.

### ► Bucle for

1. Control de flujo: bucles for
2. Slices de listas.

## Clase 10.

### ► Diccionarios

1. Diccionarios: uso y métodos esenciales.
2. Uso de diccionarios como medio de almacenamiento temporal de datos.

## Clase 11.

### ► Funciones

1. Funciones en Python: definición y uso.
2. Argumentos y retorno de valores.
3. Funciones que llaman a funciones.

# Pero antes...

---



¡Resolvamos los “**Ejercicios prácticos**” de la clase anterior!



# Diccionarios



# ¿Qué es un diccionario?

- Un diccionario es una estructura de datos que almacena **valores** asociados a **claves**.
- Las claves deben ser **únicas** y los valores pueden ser de cualquier tipo.
- Los diccionarios no tienen un orden específico.

## Ejempl

O:

```
# Diccionario de productos con nombre y cantidad
productos = {
    "Manzanas": 50,
    "Peras": 30,
    "Bananas": 40
}
```



# ¿Cómo acceder a un diccionario?

Los diccionarios se crean usando llaves {}. Y para acceder a un valor usamos la clave entre corchetes.

## Ejempl

o:

```
print(productos["Manzanas"])  
# Salida: 50
```





# Uso del método get()

- .get() te permite acceder a los valores sin riesgo de errores si la clave no existe.
- Si la clave no está, devuelve un valor por defecto.

## Ejemplo:

```
productos = { "Manzanas": 50, "Peras": 30, "Bananas": 40}  
  
print(productos.get("Peras", 0))    # Salida: 30  
print(productos.get("Uvas", 0))    # Salida: 0
```





# Agregar y actualizar elementos

Python permite agregar o actualizar los datos almacenados en un diccionario. Con la sintaxis del ejemplo, se modifica el valor asociado a una clave existente, o se crea una clave si no existe previamente.

## Ejempl

o:

```
# Se crea la clave "naranjas" con el valor 25
productos["Naranjas"] = 25

# Se modifica el valor asociado a la clave "Manzanas"
productos["Manzanas"] = 60
```



# Eliminar elementos

Es posible eliminar datos de un diccionario. Python posee el comando **del**, que permite realizar esta acción.

## Ejemplo:

```
productos = {  
    "Manzanas": 50,  
    "Peras": 30,  
    "Bananas": 40  
}  
print(productos)  
  
del productos["Bananas"]  
print(productos)
```

```
{'Manzanas': 50, 'Peras': 30,  
'Bananas': 40}  
  
{'Manzanas': 50, 'Peras': 30}
```



# Recorrer un diccionario

Para recorrer un diccionario podemos utilizar bucles. Hay tres métodos muy útiles:

- `.items()`: Devuelve clave-valor como tuplas.
- `.keys()`: Devuelve sólo las claves.
- `.values()`: Devuelve solo los valores.

## Ejemplo con `.items()`:

```
# Recorremos el diccionario para mostrar los productos
print("Inventario de productos:")
for producto, cantidad in productos.items():
    print("Producto:", producto, "- Cantidad en stock:", cantidad)
```



# Recorrer un diccionario

## Ejemplo con .keys():

```
for clave in productos.keys():  
    print("Producto:", clave)
```

```
Producto: Manzanas  
Producto: Peras  
Producto: Bananas
```

## Ejemplo con .values():

```
for valor in productos.values():  
    print("Cantidad:", valor)
```

```
Cantidad: 50  
Cantidad: 30  
Cantidad: 20
```



# Eliminar un elemento específico

- pop() nos permite eliminar un elemento específico del diccionario.
- Al usarlo, además de eliminar el elemento, podemos capturar el valor eliminado.

## Ejemplo con .pop():

```
cantidad_bananas = productos.pop("Bananas", "No hay bananas")  
print("Cantidad de bananas eliminada:", cantidad_bananas)  
print(productos)
```

Elimina la clave "Bananas" y devuelve su valor. Si la clave no existe, devuelve un valor predeterminado ("No hay bananas").



# Borrar un diccionario completo

Cuando queremos eliminar todo el diccionario de una vez, podemos usar el método **clear()** o la palabra clave **del**.

## Ejemplo con .clear():

```
productos.clear()  
print(productos)  
  
# Salida: {}
```

## Ejemplo con del:

```
del productos  
  
# El diccionario ya no existirá después  
# de este punto.
```



# Diccionarios para almacenar datos

Los diccionarios pueden ser usados para almacenar temporalmente ventas, productos o cualquier conjunto de datos en tus proyectos.

```
inventario = {"Manzanas": 50, "Peras": 30, "Bananas": 40}
ventas_dia = {}

# Pedimos los 3 productos / cantidad vendidas:
for _ in range(3):
    producto = input("Ingresá el producto vendido: ")
    cantidad_vendida = int(input("Ingresá la cantidad vendida: "))

    # Si el producto está en el inventario, registramos la venta
    if producto in inventario:
        # Actualizamos el diccionario de ventas
        ventas_dia[producto] = cantidad_vendida
    else:
        print("Producto no encontrado en inventario.")
```

# ¡Vamos a la práctica!





# Ejercicios prácticos



Optativos | No

## entregables

### Gestión de inventario con diccionarios

En un comercio, se necesita gestionar los productos y sus precios. Desarrollá un programa que permita:

1. Ingresar el nombre de tres productos y su precio correspondiente, guardándolos en un diccionario donde la clave es el nombre del producto y el valor es su precio.
2. Una vez ingresados, mostrará todos los productos y sus precios en pantalla.

```
Producto: Manzanas, Precio: 100  
Producto: Naranjas, Precio: 150  
Producto: Peras, Precio: 120
```



# Ejercicios prácticos

## Optativos | No entregables



### Consultar stock en inventario

El inventario de una tienda está almacenado en un diccionario, donde las claves son los nombres de los productos y los valores son las cantidades disponibles en stock. Creá un programa que:

1. Te permita ingresar el nombre de un producto.
2. Utilice el método `.get()` para buscar el stock disponible. Si el producto no existe, deberá mostrar un mensaje diciendo "Producto no encontrado".
3. Si el producto está disponible, mostrará cuántas unidades quedan en stock.



# Ejercicios prácticos



Optativos | No entregables

## Consultar stock en inventario

Diccionario inicial:

```
productos = {  
    "Manzanas": 50,  
    "Naranjas": 30,  
    "Peras": 25  
}
```

Ejemplo de salida esperada:

```
Ingresá el nombre del producto: Peras  
Stock disponible de Peras: 25
```



# ¡NUEVO CUESTIONARIO EN CAMPUS!

La resolución del cuestionario es de carácter obligatorio para poder avanzar en la cursada.