



# Python: Colecciones de Datos

Centro de Servicios y Gestión Empresarial  
SENA Regional Antioquia

# Conceptualización

# Colecciones de Datos

En Python, las colecciones de datos son estructuras que permiten almacenar múltiples valores en una sola variable. Python ofrece cuatro tipos principales de colecciones:

Listas (list)

↓  
Colección ordenada y modificable.

Tuplas (tuple)

↓  
Colección ordenada e inmutable.

Conjuntos (set)

↓  
Colección no ordenada y sin duplicados.

Diccionarios (dict)

↓  
Colección de pares clave: valor.

# Colecciones de Datos

En Python, el estilo recomendado para nombrar las colecciones es la convención **snake\_case** (minúsculas con guion bajo) según la **Guía de Estilo PEP 8**. Se recomienda plural cuando la colección contiene varios elementos.

## 💡 Reglas Generales

- ◊ **Listas, tuplas y conjuntos** → Plural (usuarios, productos, nombres).
- ◊ **Diccionarios** → Singular (porque representa un solo objeto con varios atributos) (usuario, producto).

# Diccionarios

# Diccionarios

Los diccionarios almacenan pares clave: valor y son mutables.

## 📌 Sintaxis

```
diccionario = {clave1: valor1, clave2: valor2, clave3: valor3, ...}
```

```
# Sintaxis Diccionario
nombre_diccionario = {
    "clave1": "valor1",
    "clave2": "valor2",
    "clave3": "valor3"
}
```

```
# Definir un diccionario
persona = {
    "nombre": "Juan",
    "edad": 30,
    "ciudad": "Bogotá"
}
```

# Diccionarios

## 🔨 Métodos Principales

Método	Descripción	Ejemplo
<b>get(clave)</b>	Obtiene el valor de clave	persona.get("nombre")
<b>keys()</b>	Devuelve todas las claves	persona.keys()
<b>values()</b>	Devuelve todos los valores	persona.values()
<b>items()</b>	Devuelve pares (clave, valor)	persona.items()
<b>update({k:v})</b>	Actualiza el diccionario	persona.update({"edad": 31})

# Diccionarios

## 🔨 Métodos Principales

Método	Descripción	Ejemplo
<code>clear()</code>	Elimina todos los elementos del diccionario.	<code>datos.clear()</code>
<code>copy()</code>	Devuelve una copia del diccionario.	<code>copia = datos.copy()</code>
<code>fromkeys(iterable, valor)</code>	Crea un diccionario con claves y un valor predeterminado.	<code>dict.fromkeys(["a", "b", "c"], 0)</code>
<code>pop(clave, valor_por_defecto)</code>	Elimina y devuelve el valor de una clave, si no existe devuelve un valor por defecto.	<code>datos.pop("edad", "No existe")</code>
<code>popitem()</code>	Elimina y devuelve el último par clave-valor insertado.	<code>datos.popitem()</code>
<code>setdefault(clave, valor)</code>	Devuelve el valor de la clave, si no existe la inserta con el valor dado.	<code>datos.setdefault("ciudad", "Bogotá")</code>
<code>len(diccionario)</code>	Devuelve el número de elementos en el diccionario.	<code>len(datos)</code>

# Métodos

# Longitud del Diccionario



Para determinar cuántos elementos tiene un diccionario, use la función **len()**:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Profesion" : "Ingeniero",  
    "Direccion" : "Medellin",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Estado": True  
}  
print(len(DatosPersonales))
```

# Tipo de colección de datos



Para determinar el tipo de colección de datos utilizado, use la función **type()**:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Profesion" : "Ingeniero",  
    "Direccion" : "Medellin",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Estado": True  
}  
print(type(DatosPersonales))
```

# Tipos de Datos



Los valores en los elementos del diccionario pueden ser de cualquier tipo de datos: cadena, int, booleanos y de lista.

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}
```

# Elementos del Diccionario



Los elementos del diccionario están ordenados, se pueden modificar y no permiten duplicados. Los elementos del diccionario se presentan en pares **clave:valor** y se puede hacer referencia a ellos mediante el nombre de la clave.

# Elementos del Diccionario II



```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Profesion" : "Ingeniero",  
    "Direccion" : "Medellin",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Estado": True  
}  
print(DatosPersonales["Direccion"])  
print(DatosPersonales["Estado"])
```

# Elementos del Diccionario III



También existe un método llamado **get()** que también permite mostrar los elementos:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
Telefono = DatosPersonales.get("Telefono")  
print(Telefono)  
# print(DatosPersonales.get("Telefono"))
```

# Obtener las claves



El método **keys()** devolverá una lista de todas las claves del diccionario. La lista de claves es una vista del diccionario, lo que significa que cualquier cambio realizado en el diccionario se reflejará en la lista de claves.

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
claves = DatosPersonales.keys()  
print(claves)  
# print(DatosPersonales.keys())
```

# Obtener los valores



El método **values()** devolverá una lista de todas los valores del diccionario. La lista de valores es una vista del diccionario, lo que significa que cualquier cambio realizado en el diccionario se reflejará en la lista.

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
valores = DatosPersonales.values()  
print(valores)  
# print(DatosPersonales.values())
```

# Obtener los items



El método **items()** devolverá cada elemento en un diccionario, como tuplas en una lista

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar", "Correr", "Futbol"],  
    "Estado": True  
}  
items = DatosPersonales.items()  
print(items)  
# print(DatosPersonales.items())
```

# Comprobar si la clave existe



Para determinar si una clave específica está presente en un diccionario, use la palabra clave **in**:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
if "Edad" in DatosPersonales:  
    print("La clave 'Edad', esta presente en el Diccionario")  
else:  
    print("La clave 'Edad', NO esta presente en el Diccionario")
```

# Cambiar valores



```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar", "Correr", "Futbol"],  
    "Estado": True  
}  
DatosPersonales["Edad"] = 30  
print(DatosPersonales)
```

# Actualizar el diccionario



El método **update()** actualizará el diccionario con los elementos del argumento dado. El argumento debe ser un diccionario o un objeto iterable con pares clave:valor.

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Telefono" : 2390483,  
    "Hobbies" : ["Nadar", "Correr", "Futbol"],  
    "Estado": True  
}  
DatosPersonales.update({"Edad": 30})  
print(DatosPersonales)
```

# Adicionar elementos



La adición de un elemento al diccionario se realiza utilizando una nueva clave de índice y asignándole un valor:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
DatosPersonales["Telefono"] = 2390483  
print(DatosPersonales)
```

# Eliminar elementos



El método **pop()** elimina el elemento con el nombre de clave especificado:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
DatosPersonales.pop("Telefono")  
print(DatosPersonales)
```

# Eliminar elementos II



El método **popitem()** elimina el último elemento insertado:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
DatosPersonales.popitem()  
print(DatosPersonales)
```

# Eliminar elementos III



La palabra clave **del** elimina el elemento con el nombre de clave especificado o el diccionario completo:

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
del DatosPersonales["Hobbies"]  
print(DatosPersonales)  
# del DatosPersonales  
# print(DatosPersonales)
```

# Limpiar el diccionario



El método **clear()** vacía el diccionario:

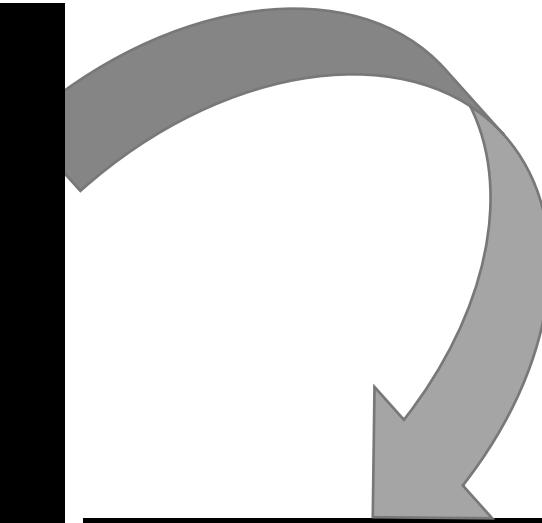
```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
DatosPersonales.clear()  
print(DatosPersonales)
```

# Ciclos con los diccionarios



Puede recorrer un diccionario utilizando un ciclo **for**. Al recorrer un diccionario, el valor devuelto son las claves del diccionario, pero también hay métodos para devolver los valores.

```
#Diccionario para el Ciclo  
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}
```



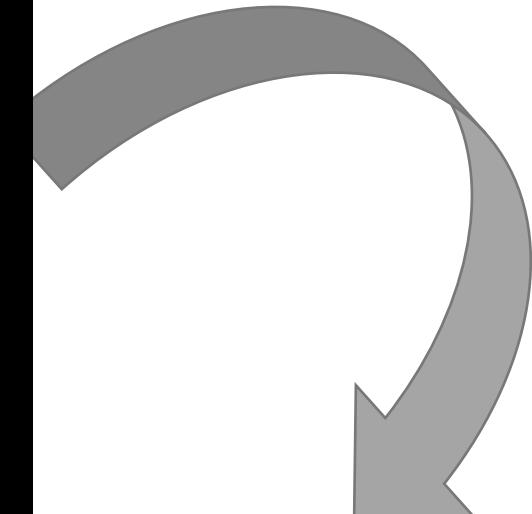
```
#Claves del Diccionario  
for x in DatosPersonales:  
    print(x)
```

# Ciclos con los diccionarios I



Puede usar el método **keys()** para devolver las claves de un diccionario:

```
#Diccionario para el Ciclo  
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}
```

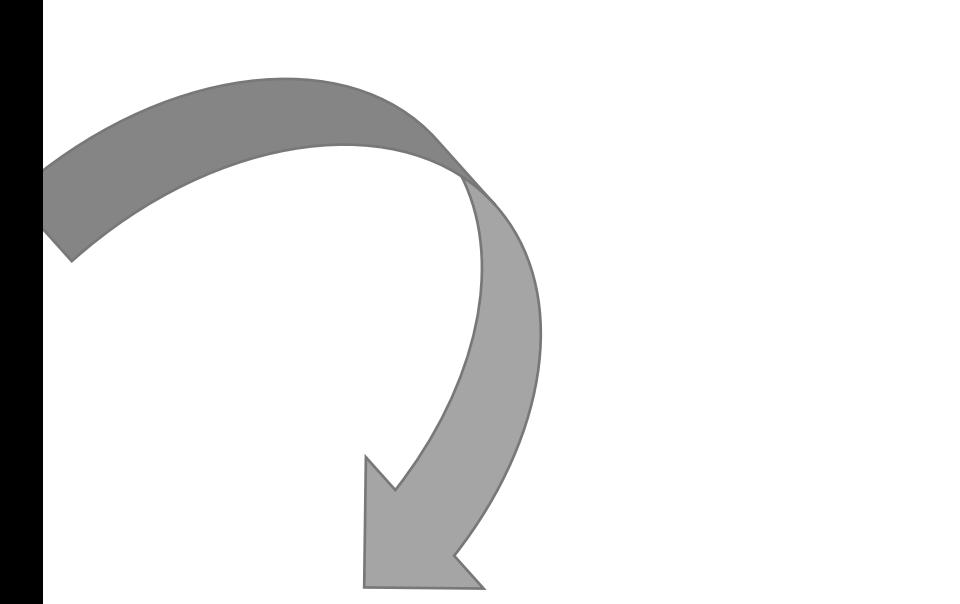


```
#Claves del Diccionario  
for x in  
    DatosPersonales.keys():  
        print(x)
```

# Ciclos con los diccionarios II

Ciclo **for** que permite mostrar los valores del diccionario uno por uno:

```
#Diccionario para el Ciclo  
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar", "Correr", "Futbol"],  
    "Estado": True  
}
```



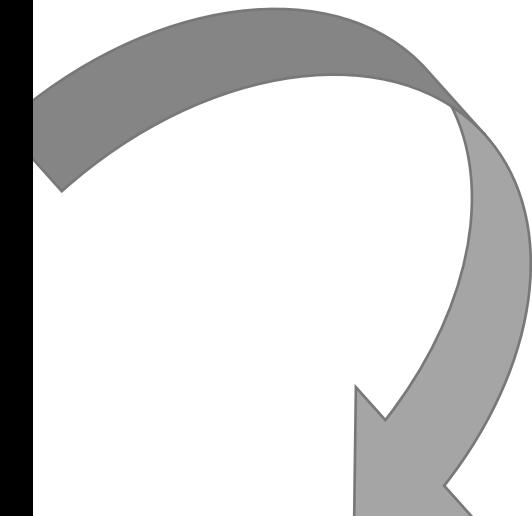
```
#Valores del Diccionario  
for x in DatosPersonales:  
    print(DatosPersonales[x])
```

# Ciclos con los diccionarios III



También puede usar el método **values()** para devolver valores de un diccionario:

```
#Diccionario para el Ciclo  
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}
```



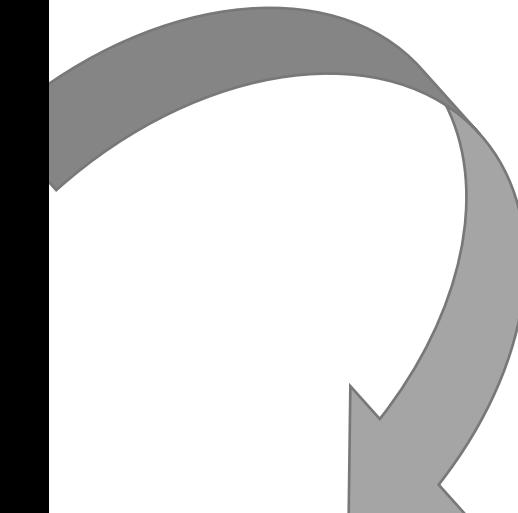
```
#Valores del Diccionario  
for x in  
    DatosPersonales.values():  
        print(x)
```

# Ciclos con los diccionarios IV



Recorra las claves y los valores usando el método **items()**:

```
#Diccionario para el Ciclo
DatosPersonales = {
    "Nombres" : "Alvaro",
    "Apellido" : "Perez",
    "Telefono" : 2390483,
    "Edad" : 50,
    "Hobbies" : ["Nadar","Correr","Futbol"],
    "Estado": True
}
```



```
#Clave:Valor del Diccionario
for x, y in
    DatosPersonales.items():
        print(x,y)
        # print(f"{x}: {y}")
```

# Copiar un diccionario



Una de las formas de copiar un diccionario es utilizando el método **copy()**.

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
diccionario=DatosPersonales.copy()  
print(diccionario)
```

# Copiar un diccionario I



Otra forma de hacer una copia es usar la función incorporada **dict()**.

```
DatosPersonales = {  
    "Nombres" : "Alvaro",  
    "Apellido" : "Perez",  
    "Telefono" : 2390483,  
    "Edad" : 50,  
    "Hobbies" : ["Nadar","Correr","Futbol"],  
    "Estado": True  
}  
nuevodic=dict(DatosPersonales)  
print(nuevodic)
```



# G R A C I A S

Presentó: Alvaro Pérez Niño

Instructor Técnico

Correo: aperezn@sena.edu.co

<http://centrodesserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)