UTN AVELLANEDA
Tecnicatura Universitaria en Programación

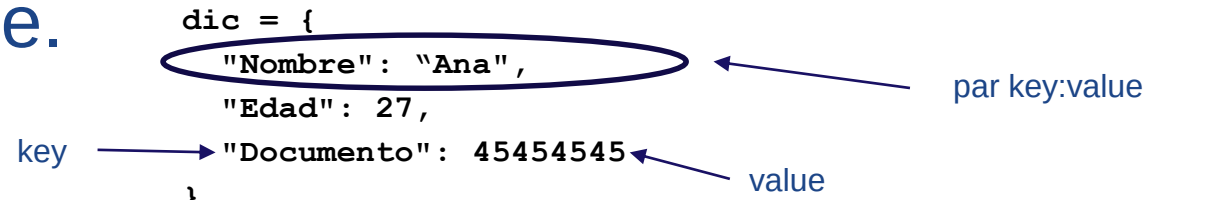
Diccionarios

Versión

Diccionarios

Un diccionario en Python es una colección de elementos, donde cada uno tiene una clave key y un valor value. Los diccionarios se pueden crear con paréntesis {} separando con una coma cada par key: value.

```
dic = {
  "Nombre": "Ana",
  "Edad": 27,
  "Documento": 45454545
}
print(dic)
#{'Nombre': 'Ana', 'Edad': 27, 'Documento': 45454545}
```



Diccionarios

Consideraciones de los diccionario en Python:

-Todos los elementos en el diccionario se encuentran encerrados en un par de corchetes {}.

-Cada elemento en un diccionario contiene una clave y un valor - es decir un par de clave-valor.

-Cada par de clave-valor es denominado como elemento (**item**).

-La ventaja de esto es que se puede acceder a todos los valores almacenados usando las claves.

Diccionarios

Propiedades de los diccionario en Python:

-Son dinámicos, pueden crecer o decrecer, se pueden añadir o eliminar elementos. Es decir que son mutables, con lo que admiten añadir, borrar y modificar sus elementos.

-Las claves deben ser únicas. A menudo se utilizan las cadenas de texto como claves, pero en realidad podría ser cualquier tipo de datos inmutable: enteros, flotantes, tuplas (entre otros).

Diccionarios

Propiedades de los diccionario en Python:

-Mantienen el orden en el que se insertan las claves.

-Son indexados, los elementos del diccionario son accesibles a través del key.

-Son anidados, un diccionario puede contener a otro diccionario en su campo value.

Diccionarios

Crear diccionario en Python:

Un diccionario en Python es una colección de elementos, donde cada uno tiene una llave key y un valor value.

```
dic = {
        "Nombre": “Ana",
        "Edad": 27,
        "Documento": 45454545
    }
print(dic)      #{'Nombre': ‘Ana', 'Edad': 27, 'Documento': 45454545}
```

Diccionarios

Acceder y modificar elementos

-Para acceder a sus elementos se usa el key entre corchetes[]

```
dic = {
    "Nombre": "Ana",
            "Edad": 27,
            "Documento": 45454545
        }
print(dic["Nombre"])   #Ana
```

Diccionarios

Acceder y modificar elementos

-O se puede acceder a través de los métodos de diccionarios items(), values() y keys()

```
dic = {    "Nombre": "Ana",
           "Edad": 27,
           "Documento": 45454545
       }
print(dic.keys())      #dict_keys(['Nombre', 'Edad', 'Documento'])
print(dic.values())    #dict_values(['Ana', 27, 45454545])
print(dic.items())     #dict_items([('Nombre', 'Ana'), ('Edad', 27), ('Documento', 45454545)])
```

Diccionarios

Acceder y modificar elementos

-Para modificar un elemento basta con usar [] con el nombre del key y asignar el valor que queremos.

```
dic = {
    "Nombre": "Ana",
            "Edad": 27,
            "Documento": 45454545
        }
dic["Edad"]=33
print(dic)   #{'Nombre': 'Ana', 'Edad': 33, 'Documento': 45454545}
```

Diccionarios

Acceder y modificar elementos

-Si el key al que accedemos no existe, se añade automáticamente.

```
dic = {
    "Nombre": "Ana",
            "Edad": 27,
            "Documento": 45454545
        }
dic["Telefono"]= "123"
print(dic)    #{'Nombre': 'Ana', 'Edad': 27, 'Documento': 45454545, 'Telefono': '123'}
```

Diccionarios

Castear un diccionario a Lista

```
dic = {
    "Nombre": "Ana",
            "Edad": 27,
            "Documento": 45454545
        }
lista = list(dic.values())
lista = list(dic.keys())
lista = list(dic.items())
```

```
print(lista)    #['Ana', 27, 45454545]
print(lista)    #['Nombre', 'Edad', 'Documento']
print(lista) #[('Nombre', 'Ana'), ('Edad', 27),
                ('Documento', 45454545)]
```

Diccionarios

Iterar diccionario

Los diccionarios se pueden iterar de manera muy similar a las listas u otras estructuras de datos.

```
dic = {    "Nombre": "Ana",
           "Edad": 27,
           "Documento": 45454545
       }
for e_dic in dic:
    print(e_dic)
```

```
#Nombre
#Edad
#Documento
```

Imprime los keys del diccionario

Diccionarios

Iterar diccionario

-Se puede imprimir también solo el value.

```
for e_dic in dic:
    print(dic[e_dic])
```

```
#Ana
#27
#45454545
```

Diccionarios

Iterar diccionario

-Se puede imprimir el key y el value a la vez.

```
for e_key, e_value in dic.items():
    print(e_key, e_value)
```

```
#Nombre Ana
#Edad 27
#Documento 45454545
```

Diccionarios

Diccionarios anidados

Los diccionarios en Python pueden contener uno dentro de otro. Para acceder se debe especificar cada key entre corchetes []

```
dic = { "Nombre": "Ana",
        "Edad": 27,
        "Documento": 45454545,
        "Domicilio":{ "calle": "Solis",
                      "numero": 345,
                      "localidad": "Avellaneda"
                      }
    }
```

```
print(dic["Nombre"])
print(dic["Domicilio"]["calle"])
```

```
#Ana
#Solis
```

Diccionarios

Diccionarios anidados

Los diccionarios también pueden contener listas[] donde a cada elemento se debe acceder por su índice/posición.

```
dic = {
            "Nombre": "Ana",
            "Edad": 27,
            "Documento": 45454545,
            "telefonos": [123,456]
    }
print(dic["telefonos"][1])          #456
```