

Conjuntos y diccionarios

Programación de Inteligencia Artificial



Junta de Andalucía

FPA

FORMACIÓN
PROFESIONAL
ANDALUZA



Fondo Social Europeo

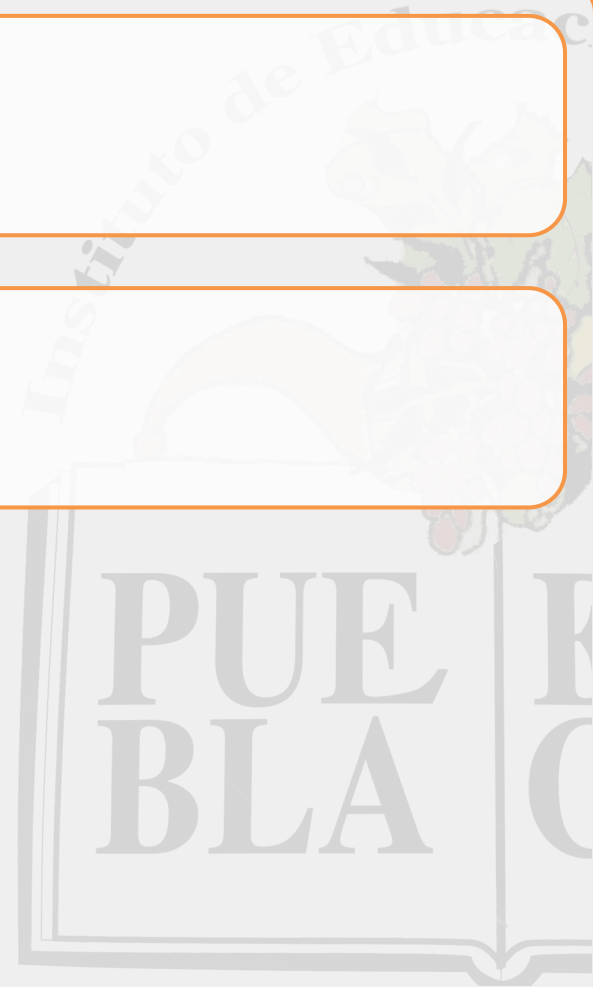


Conjuntos:

- Introducción.
- Operaciones.
- Métodos.

Diccionarios:

- Introducción.
- Operaciones.
- Métodos.



Introducción

Operaciones

Métodos

- Colección no ordenada de objetos únicos.
- Existen 2 tipos: conjuntos modificables o *set* y no modificables *frozenset*.
- Los elementos de un conjunto van entre llaves {}.

```
conjunto = {'Almería', 'Granada', 8, (10,2), 1, 8}  
print(conjunto)
```

- Un conjunto no puede incluir objetos mutables (por ejemplo: una lista).

```
# La siguiente línea producirá un error ya que no puede contener  
# la lista ['Almería', 'Granada']  
c2 = {4, 7, True, ['Almería', 'Granada']}
```

Introducción

Operaciones

Métodos

- No se puede generar un nuevo conjunto a partir de otros dos mediante la suma $+$.
- En cambio, si se puede obtener un nuevo conjunto con los elementos de un primer conjunto que no aparece en otro. Esto se hace mediante el guion $-$.
- Se pueden realizar operaciones lógicas como AND $\&$, OR $|$ o XOR \wedge .
- Podemos comprobar si un objeto está en un conjunto (o no) mediante *in* y *not in*.

```
conjunto = {'Almería', 'Granada', 8, (10,2), 1, 8}
print(conjunto)
```

```
conjunto1 = {4, 5, 2, 0}
print(conjunto1)
```

```
conjunto2 = set({2, 8, 1})
print(conjunto2)
```

```
cFinal = conjunto1 + conjunto2 # Producirá un error
cFinal = conjunto1 - conjunto2; print(cFinal)
cFinal = conjunto1 & conjunto2; print(cFinal)
cFinal = conjunto1 | conjunto2; print(cFinal)
cFinal = conjunto1 ^ conjunto2; print(cFinal)
```

```
print(5 in conjunto1)
```

Introducción

Operaciones

Métodos

- `add(valor)`: permite añadir un elemento al conjunto.
- `clear()`: elimina todos los objetos del conjunto.
- `copy()`: crea una copia del conjunto.
- `pop()`: saca el primer elemento del conjunto.
- `remove(valor)`: elimina un elemento del conjunto.

```
conjunto = {4, 5, 2, 0}
conjunto.add(20); print(conjunto)
conjunto.clear(); print(conjunto)

conjunto = {4, 5, 2, 0}
c2 = conjunto.copy(); print(c2)
print(conjunto.pop()); print(conjunto)
conjunto.remove(2); print(conjunto)
```

Introducción

Operaciones

Métodos

- Colección no ordenada.
- Está compuesta por dos elementos: clave (que es una cadena de caracteres y única) y el valor.
- Los elementos de un diccionario van entre llaves {}.

```
diccionario = {"nombre": "Almudena", "apellidos": "Grandes", "obras": 6}
```

-

```
diccionario["nacimiento"] = "Madrid"; print(diccionario)  
diccionario["obras"] = 11; print(diccionario)
```

Introducción

Operaciones

Métodos

- Para añadir un nuevo elemento al diccionario, se pone la clave entre corchetes y se da un nuevo valor mediante la asignación. Este proceso también se realiza para la modificación de un elemento.
- Podemos comprobar si una clave está en el diccionario (o no) mediante *in* y *not in*.
- `del(diccionario["clave"])`: elimina una entrada del diccionario.
- `len(diccionario)`: indica cuantos registros hay en el diccionario.

```
diccionario = {"nombre": "Almudena", "apellidos": "Grandes", "obras": 11}
print("nombre" in diccionario)
del(diccionario["obras"]); print(diccionario)
print(len(diccionario))
```

Introducción

Operaciones

Métodos

- `clear()`: elimina todos los objetos del diccionario.
- `copy()`: crea una copia del diccionario.
- `get("clave")`: devuelve el valor para la clave pasada por parámetro. Se le puede añadir un segundo parámetro para devolver un valor en caso de que no exista. También se puede obtener el valor de una clave mediante el uso de corchetes, pero si no existe se producirá un error.
- `items()`: devuelve una lista con la tupla clave-valor.
- `keys()`: devuelve una lista con las claves.
- `pop("clave")`: devuelve el valor de la clave pasada por parámetro y lo saca de la lista.
- `values()`: devuelve una lista con los valores almacenados en el diccionario.
- `update(otroDiccionario)`: actualiza el diccionario con las claves y valores del pasado por parámetro.

```
dic1 = {"nombre": "Almudena", "apellidos": "Grandes", "obras": 11}
dic2 = dic1.copy(); print(dic2)
dic2.clear(); print(dic2)
print(dic1.get("nombre")); print(dic1["nombre"])
print(dic1.get("nacimiento")); print(dic1.get("nacimiento", "no existe el campo"))
print(dic1.items()); print(dic1.keys()); print(dic1.values())
print(dic1.pop("nombre")); print(dic1)
dic2 = {"apellidos": "Concostrina", "residencia": "Almería"}
dic1.update(dic2); print(dic1)
```




Fondo Social Europeo