



1. Python Sets
2. Acceder a valores
3. Añadir valores
4. Eliminar valores
5. Concatenar sets
6. Sets methods

Python {Sets}

Python {Sets}

Python {Sets}

Los **sets** son usados en Python para almacenar múltiples valores en una sola variable de forma **no indexada**.

Una vez creado un set, sus valores **no pueden ser modificados**.

- Los sets se representan con corchetes: `lol = {"ahri", "evelynn", "yone"}`
- Los sets son **NO** indexados.
- Los **valores** de los sets **NO** pueden ser modificados.
- Los sets **NO** aceptan valores repetidos.
- Podemos **añadir elementos** no duplicados dentro de un set ya existente.
- Los sets pueden contener elementos de distinto tipaje:

```
set1 = {"abc", 34, True, 40, "male"}
```

Acceder a valores

Python {Sets}: **acceder** a valores

Dado un set:

```
lol = {"ahri", "evelynn", "yone"}
```

Acceder	Python code
Sólo podemos acceder mediante for-each	<pre>for l in lol: print(l)</pre>
¿Un valor existe en un set?	<pre>print("evelynn" in lol)</pre>

Añadir valores

Python {Sets}: **añadir** valores

Dado un set: `lol = {"ahri", "evelynn", "yone"}`

Añadir	Python code
Un elemento	<code>lol.add("seraphine")</code>
Un {set}, una [lista], una (tupla) o un {:diccionario}	<code>lol2 = {"ezreal", "kai'sa"}</code> <code>lol.update(lol2)</code>

Eliminar valores

Python {Sets}: **eliminar** valores

Dado un set: `lol = {"ahri", "evelynn", "yone"}`

Eliminar	Python code
Un valor existente (<i>error en caso de no existir</i>)	<code>lol.remove("yone")</code>
Un valor existente (<i><u>sin</u> error en caso de no existir</i>)	<code>lol.discard("kai'sa")</code>
Un elemento aleatorio (<i>devuelve el elemento</i>)	<code>x = lol.pop()</code>
Vaciar el set	<code>lol.clear()</code>
Eliminar el set de la RAM	<code>del lol</code>

Concatenar {Sets}

Python {Sets}: **concatenar sets**

Dados dos sets:

```
lol = {"ahri", "evelynn", "yone"}
```

```
lol2 = {"seraphine", "yone", "kai'sa"}
```

Concatenar	Python code
Los valores diferentes de dos sets en uno	<code>lol3 = lol.union(lol2)</code>
Integrar un set en otro	<code>lol.update(lol2)</code>
Un elemento aleatorio <i>(devuelve el elemento)</i>	<code>x = lol.pop()</code>
Mantener solo los elementos repetidos	<code>lol.intersection_update(lol2)</code>
Crear un nuevo set con solo los repetidos	<code>lol3 = lol.intersection(lol2)</code>

{Sets} methods

Python {Sets}: methods

Method	Description
<u>add()</u>	Adds an element to the set
<u>clear()</u>	Removes all the elements from the set
<u>copy()</u>	Returns a copy of the set
<u>difference()</u>	Returns a set containing the difference between two or more sets
<u>difference_update()</u>	Removes the items in this set that are also included in another, specified set
<u>discard()</u>	Remove the specified item
<u>intersection()</u>	Returns a set, that is the intersection of two other sets
<u>intersection_update()</u>	Removes the items in this set that are not present in other, specified set(s)
<u>isdisjoint()</u>	Returns whether two sets have a intersection or not
<u>issubset()</u>	Returns whether another set contains this set or not
<u>issuperset()</u>	Returns whether this set contains another set or not
<u>pop()</u>	Removes an element from the set
<u>remove()</u>	Removes the specified element
<u>symmetric_difference()</u>	Returns a set with the symmetric differences of two sets
<u>symmetric_difference_update()</u>	inserts the symmetric differences from this set and another
<u>union()</u>	Return a set containing the union of sets
<u>update()</u>	Update the set with the union of this set and others

“Hay tres cosas extremadamente duras: el acero,
los diamantes y el conocerse a uno mismo.”

Benjamin Franklin (1706 - 1790)

