**SETS**

**SETS:**

**{} →** Sets are written with curly brackets.

Los conjuntos son **colecciones DESORDENADAS** de **elementos ÚNICOS.** Al igual que las listas, también **son mutables.** Por lo que **su método DEDUPLICA**

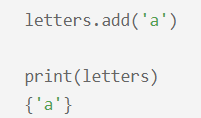
girls\_name = **set(**[“ana”, “ana”, “paula”]**)**

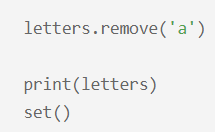
girls\_name = {“ana”, “paula”}

\* Nota: Los elementos del conjunto no se pueden **cambiar**, pero **se puede eliminar elementos** y **agregar elementos nuevos**.

\* Los elementos del conjunto **pueden ser de cualquier tipo de datos**:

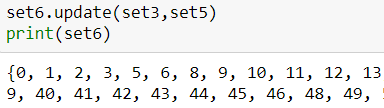


* Podemos crearlos de las siguientes maneras:
  + **Crear el set directamente** con sus **elementos**:
  + A partir de una lista o un tuple **con el comando set()**:
  + Podemos **definir un conjunto vacío** de la siguiente manera:
* Luego **agregamos** al conjunto el elemento usando el **add()** comando.
* Podemos **eliminar** de nuestro conjunto usando **remove()** or **discard()**



*\* podemos usar* ***pop()*** *(removes a random item from the set)*

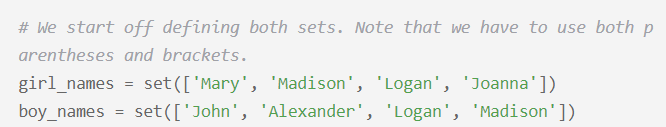
* Podemos **actualizar** un set a partir de otros iterables (tuples, lists, dictionaries etc.) con **update()** comando.

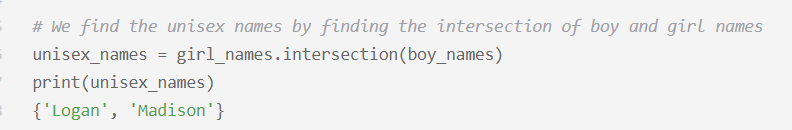


**METHODS MAS USADOS (abajo están todos los methods con sus enlaces)**

* //set en el que quieres buscar//**.intersection(**//lista o set a buscar//**)** → nos **da** **los elementos comunes** a ambos sets
* //set en el que quieres buscar//**.union(**//lista o set a buscar//**)** → nos **une todos los elementos deduplicados** de ambos sets
* //set en el que quieres buscar//**.difference(**//lista o set a buscar//**)** → nos **da los** **elementos diferentes** a ambos sets
* //set en el que quieres buscar//**symmetric\_difference\_update(**//lista o set a buscar//**)** → **almacena** los **elementos que NO se repiten entre sets** (estos los descarta

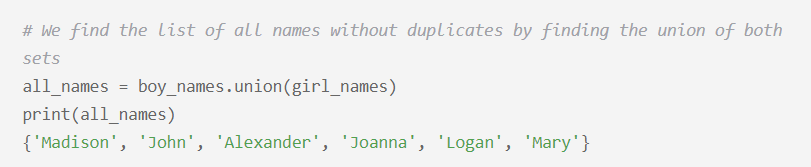
EJEMPLOS VISUALES:

****

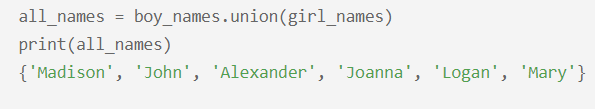
1. **.intersection()**

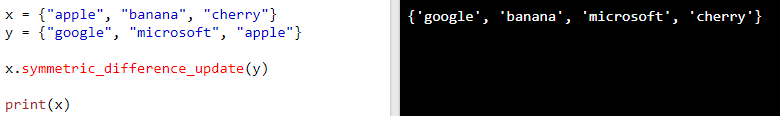
→ como un buscarv, dices donde quieres buscar, le pones el .intersection

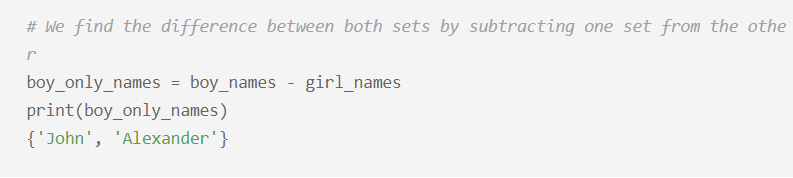
1. **.union**()



1. **.difference()**



1. **.symmetric\_difference\_update()**

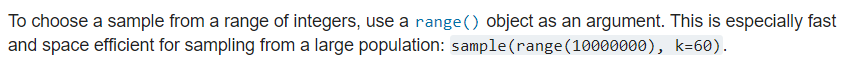
\* Los sets se pueden restar SUSTRAYENDO LOS ELEMENTOS COMUNES

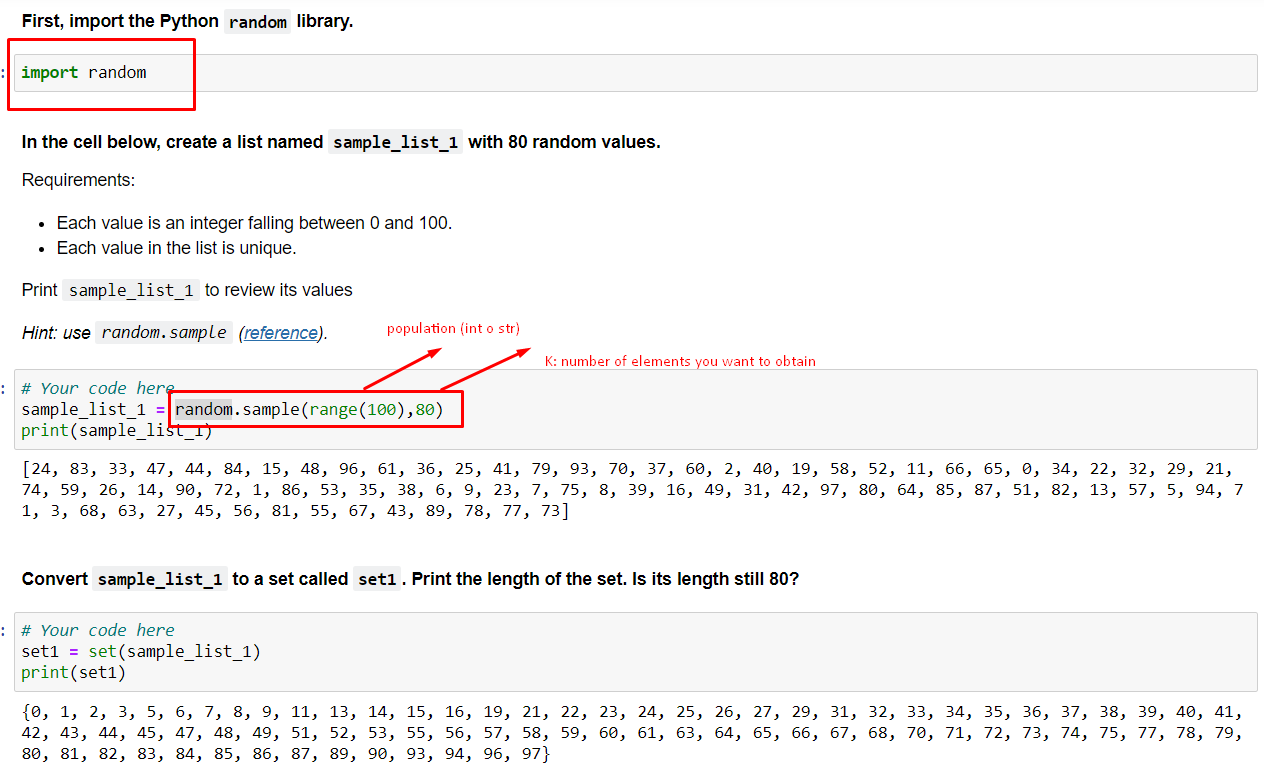
**ALL METHODS**

| **Method** | **Description** |
| --- | --- |
| [add()](https://www.w3schools.com/python/ref_set_add.asp) | Adds an element to the set |
| [clear()](https://www.w3schools.com/python/ref_set_clear.asp) | Removes all the elements from the set |
| [copy()](https://www.w3schools.com/python/ref_set_copy.asp) | Returns a copy of the set |
| [difference()](https://www.w3schools.com/python/ref_set_difference.asp) | Returns a set containing the difference between two or more sets |
| [difference\_update()](https://www.w3schools.com/python/ref_set_difference_update.asp) | Removes the items in this set that are also included in another, specified set |
| [discard()](https://www.w3schools.com/python/ref_set_discard.asp) | Remove the specified item |
| [intersection()](https://www.w3schools.com/python/ref_set_intersection.asp) | Returns a set, that is the intersection of two other sets |
| [intersection\_update()](https://www.w3schools.com/python/ref_set_intersection_update.asp) | Removes the items in this set that are not present in other, specified set(s) |
| [isdisjoint()](https://www.w3schools.com/python/ref_set_isdisjoint.asp) | Returns whether two sets have a intersection or not |
| [issubset()](https://www.w3schools.com/python/ref_set_issubset.asp) | Returns whether another set contains this set or not |
| [issuperset()](https://www.w3schools.com/python/ref_set_issuperset.asp) | Returns whether this set contains another set or not |
| [pop()](https://www.w3schools.com/python/ref_set_pop.asp) | Removes an element from the set |
| [remove()](https://www.w3schools.com/python/ref_set_remove.asp) | Removes the specified element |
| [symmetric\_difference()](https://www.w3schools.com/python/ref_set_symmetric_difference.asp) | Returns a set with the symmetric differences of two sets |
| [symmetric\_difference\_update()](https://www.w3schools.com/python/ref_set_symmetric_difference_update.asp) | inserts the symmetric differences from this set and another |
| [union()](https://www.w3schools.com/python/ref_set_union.asp) | Return a set containing the union of sets |
| [update()](https://www.w3schools.com/python/ref_set_update.asp) | Update the set with the union of this set and others |

Usar la **librería random** para crear sets







También se puede con “str” como los de una lista

