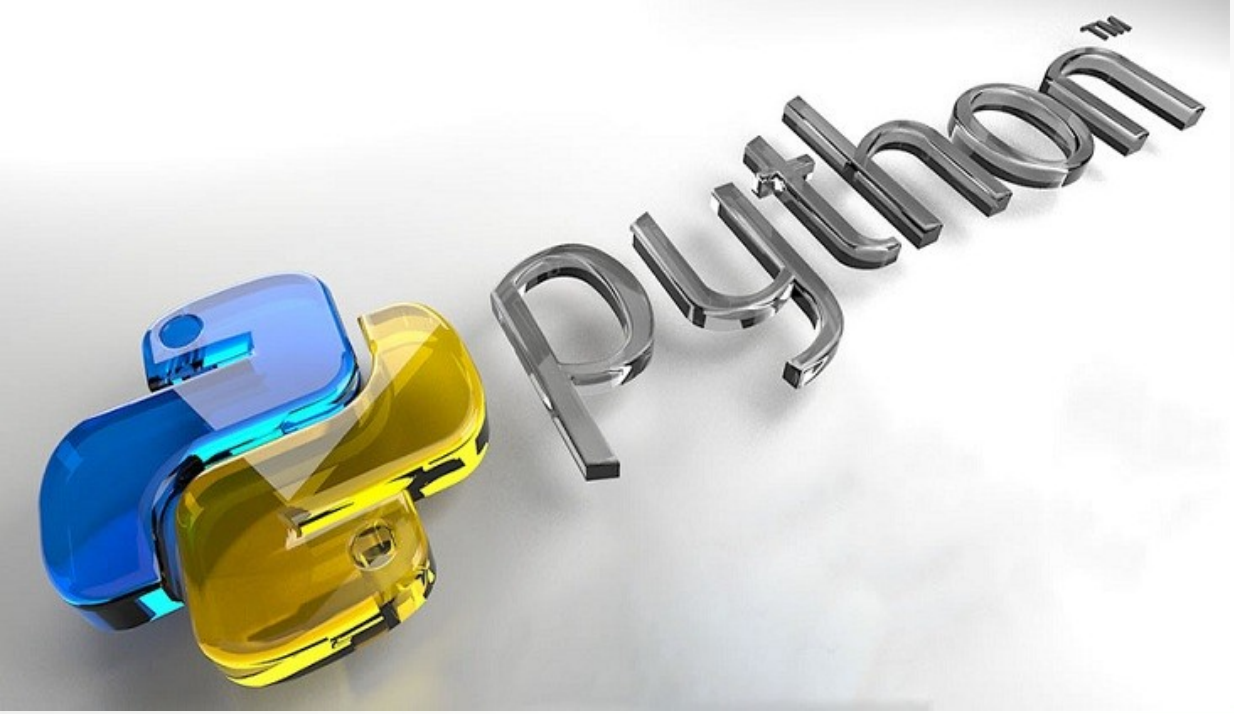
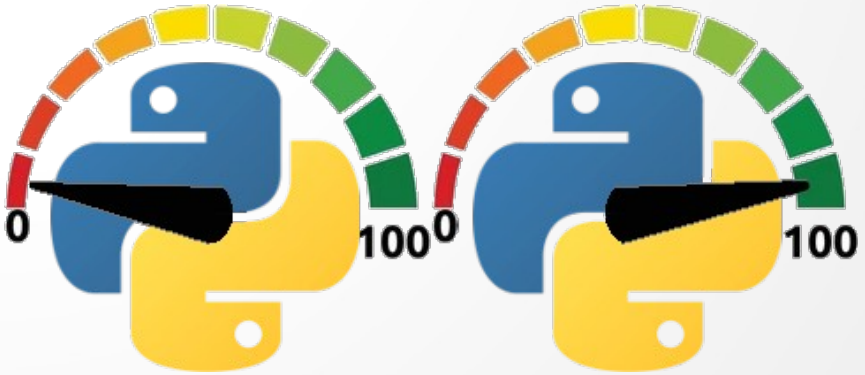


Set – a jugar con conjuntos



Las instancias de **Set** y **frozenset** ambos proporcionan las siguientes operaciones:

Operación (Equivalente)	**Resultado**
len(s)	número de elementos en el conjunto s (cardinalidad)
x in s	prueba x para la pertenencia a s
x not in s	prueba x para no membresía en s
s.issubset(t) s < t s <= t	probar si cada elemento en s está en t B ⊆ A estricto
s.issuperset(t) s > t s >= t	prueba si cada elemento en t está en s estricto

Las instancias de **Set** y **frozenset** ambos proporcionan las siguientes operaciones:

Operación (Equivalente)	**Resultado**
s.union(t) s | t	nuevo conjunto con elementos de s y t
s.intersection(t) s & t	nuevo conjunto con elementos comunes a s y t
s.difference(t) s - t	nuevo conjunto con elementos en s pero no en t
s.symmetric_difference(t) s ^ t	nuevo conjunto con elementos en s o t pero no en ambos
s.copy()	nuevo conjunto con una copia superficial de s

La siguiente tabla enumera las operaciones disponibles **Set** pero no encontradas en **frozenset**

Operación (Equivalente)	**Resultado**
s.isdisjoint(t)	devuelve True si el conjunto s no tienen ningún elemento en común con t. Dos conjuntos son disjuntos si y solo si su intersección es el conjunto vacío.
s.update(t) s |= t	devuelve el conjunto s con elementos agregados desde t
s.intersection_update(t) s & = t	return set s manteniendo solo los elementos que también se encuentran en t
s.difference_update(t) s -= t	devuelve set s después de eliminar los elementos encontrados en t
s.symmetric_difference_update(t) s ^ = t	devuelve el conjunto s con elementos de s o t pero no ambosdevuelve el conjunto s con elementos de s o t pero no ambos

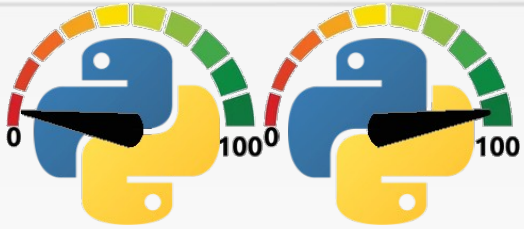
La siguiente tabla enumera las operaciones disponibles **Set** pero no encontradas en **frozenset**

Operación (Equivalente)	**Resultado**
s.add(x)	suma el elemento x al conjunto s
s.remove(x)	quitar x del conjunto s ; aumenta KeyErrorsi no está presente
s.discard(x)	elimina x del conjunto s si está presente
s.pop()	eliminar y devolver un elemento arbitrario de s ; aumenta KeyErrorsi está vacío
s.clear()	eliminar todos los elementos del conjunto s

```
# Python Set Subset
# creamos A
A = {1,2,3,4,5,6,7,8,9,10,11,12}

# creamos B
B = {4,3,7,8,11}

# creamos la lista L desde del set A
L =list(A)

# usa el issubset para chequear si  A es un subconjunto de B
print('el set A es Subset del set B?',A.issubset(B))

# usa el issubset para chequear si  B es un subconjunto de A
print('el set B es Subset del set A?',B.issubset(A))

# usa el issubset para chequear si  B es un subconjunto de la lista L
print('el set B es Subset de la lista A?',B.issubset(A))
```

La siguiente tabla enumera las operaciones disponibles **Set** pero no encontradas en **frozenset**

```
# Python Set Subset
# creamos A
A = {1,2,3,4,5,6,7,8,9,10,11,12}

# creamos B
B = {4,3,7,8,11}

# creamos C
C = {1,2,3,4,5,6,7,8,9,10,11,12}

# usa el operador  < para chequear si  B es un subconjunto de A
print('B es subconjunto de A?',B <= A)

# usa el operador  < para chequear si  A es un subconjunto de A
print('A es subconjunto de  B?',A <= B)

# usa el operador  < para chequear si  B es un subconjunto de A en forma estricta?
print('B es subconjunto estricto de A?',B < A)

# usa el operador  < para chequear si  C es un subconjunto de A en forma estricta?
print('C es subconjunto estricto de A?',C < A)
```



- https://docs.python.org/es/3.10/library/stdtypes.html#set-types-set-frozenset