

Frozenset

Los frozenset en Python son una estructura de datos muy similar a los [set](#), con la salvedad de que son [inmutables](#), es decir, no pueden ser modificados una vez declarados.

Crear frozenset Python

Los frozensets en Python son un tipo muy parecido a los [sets](#) con la salvedad de que son inmutables, es decir, están congelados y no pueden ser modificados una vez inicializados.

```
fs = frozenset([1, 2, 3])
print(fs)           #frozenset({1, 2, 3})
print(type(fs))     #<class 'frozenset'>
```

Ejemplos frozenset

Dado que son inmutables, cualquier intento de modificación con los métodos que ya hemos visto en otros capítulos como `add()` o `clear()` dará un error, ya que los `frozenset` no disponen de esos métodos.

```
fs = frozenset([1, 2, 3])
#fs.add(4) #Error! AttributeError
#fs.clear() #Error! AttributeError
```

Los `frozenset` pueden ser útiles cuando queremos usar un `set` pero se requiere que el tipo sea inmutable. Algo no muy común, pero que podría darse, es crear un set de sets. Es decir, un set que contiene como elementos a otros sets. El siguiente código daría un `TypeError` ya que los elementos de un `set` deben ser por definición inmutables.

```
s1 = {1, 2}
s2 = {3, 4}
#s3 = {s1, s2} # Error! TypeError
```

Para resolver este problema, podemos crear un set de frozensets. Esto si es posible ya que el `frozenset` es un tipo inmutable.

```
s1 = frozenset([1, 2])
s2 = frozenset([3, 4])
s3 = {s1, s2}
print(s3) #{frozenset({3, 4}), frozenset({1, 2})}
```

Lo mismo aplica a los [diccionarios](#), ya que su `key` debe ser un tipo inmutable. Si intentamos crear un diccionario con `set` como `key`, obtendremos un `TypeError`.

```
s1 = set([1, 2])
s2 = set([3, 4])
#d = {s1: "Texto1", s2: "Texto2"} # Error! TypeError
```

Pero si podemos crear un diccionario donde sus `key` son `frozenset`, ya que son un tipo inmutable.

```
s1 = frozenset([1, 2])
s2 = frozenset([3, 4])
```

```
d = {s1: "Texto1", s2: "Texto2"}  
print(d) #{frozenset({1, 2}): 'Texto1', frozenset({3, 4}): 'Texto2'}
```

Tal vez te interese leer acerca de otras estructuras de datos similares como [los sets](#) o [las listas](#).