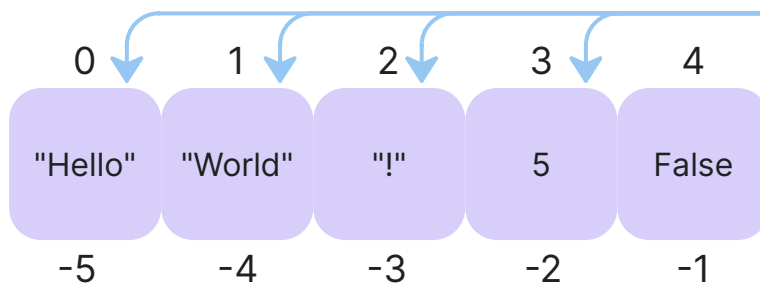


tuple — массив-образованная коллекция, которая НЕ умеет менять свою величину

переменная



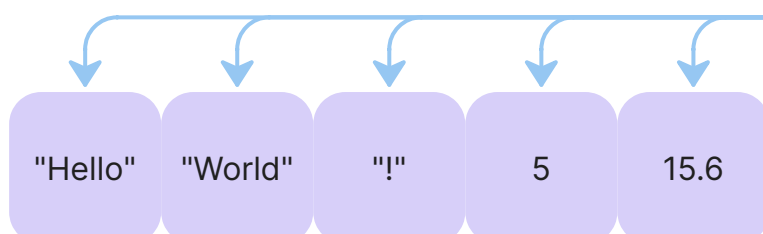
Обычно используют для работы с гетерогенными данными

Гетерогенные данные — разнородные данные представляющие разные типы

```
1 tuple_a = tuple(); tuple_a = ()
2 tuple_a = (1, 2, 3) # наш тупляра
3 tuple_a = 1, 2, 3 # наш тупляра
4
5 generator_ = (x for x in range(3)) # генератор, не
# списков или чего-либо. Это функция, которая
# возвращает по значению за каждый вызов функции
# next()
6
7 # получение данных
8 print(tuple_a[0]) # > 1
9
10 # индексирование — доступ к переменной. По сути,
# выражение "tuple_a[0]" — это обращение к переменной
11
12 # срезы — индекс который указывает на часть
# коллекции tuple_a[0:2:1], где первый элемент —
# откуда начинаем включительно, второй — где
# заканчиваем строго, третий — с каким каждым (логика
# "каждый n-ный", например "каждый первый, второй и
# тд")
13
14 tuple_a[1::-1] # > [3, 2]
15
16 # tuple не умеет в append!
17 # Если вы хотите менять размер коллекции — берите
# list
18 list_a.append(4)
19 list_a # > [1, 2, 3, 4]
20
21 reverse(list_a) # не модифицирует сам лист, а
# возвращает его перевернутую копию — [4, 3, 2, 1]
22 print(list_a) # > [1, 2, 3, 4]
23
24 tuple_a.reversed() # модифицирует лист, но
# возвращает None
25 print(tuple_a) # > [3, 2, 1]
26
27
28
```

set — массив-образованная коллекция, неиндексируемая, хранит только уникальные значения

переменная



```
1 set = set();
2 seta = {1, "Hello"}
3
4 uniques = list(set([1, 1, 2, 2, 2, 4]))
5 uniques
6 # > {1, 2, 4}
7
8 seta.add(2) # добавляет значение
9 seta
10 # > {1, "Hello", 2}
11
12 setb = {1, 2, 3, 4}
13
14 # Провайдит теорию множеств в питон
15 # Можно проверять отношения множеств между
    коллекциями
16
17 seta.intersection(setb)
18 seta & setb
19 # > {1, 2}
```

frozenset — массив-образованная коллекция, неиндексируемая, хранит только уникальные значения.

Это неизменяемая коллекция и может быть использована в качестве ключа в dict