

List Methods

<code>.append(<i>item</i>)</code>	добавляет один новый элемент в конец списка
<code>.extend(<i>items</i>)</code>	добавляет в конец списка все элементы из итерируемого объекта
<code>.sort()</code>	сортирует список в порядке возрастания (<i>default</i>); или убывания с аргументом <i>reverse=True</i>

Set Methods

<code>.add(<i>elem</i>)</code>	добавляет новый элемент в set
<code>.discard(<i>elem</i>)</code>	удаляет элемент из set; если отсутствует ничего не произойдет
<code>.intersection(set & other &</code>	пересечение: общие элементы в исследуемых множествах
<code>.union(<i>other</i>) set other </code>	объединение: все элементы исследуемых множеств
<code>.difference(<i>others</i>) set - other -</code>	разница

Dict Methods

<code>list(d)</code>	Возвращает список всех <i>ключей</i> в виде списка []
<code>d[key]</code>	Возвращает <i>значение</i> по <i>ключу</i>
<code>d[key] = value</code>	Добавить в словарь пару (<i>key</i> , <i>value</i>)
<code>del d[key]</code>	Удалить из словаря пару (<i>key</i> , <i>value</i>)
<code>.keys()</code>	Возвращает список <i>ключей</i> из всех пар словаря
<code>.values()</code>	Возвращает список <i>значений</i> из всех пар словаря
<code>.items()</code>	Возвращает список всех пар (<i>key</i> , <i>value</i>) в словаре
<code>.get(<i>key</i>, <i>default=None</i>)</code>	Возвращает <i>значение</i> по указанному <i>ключу</i> . Если значение не найдено - возвращает сообщение указанное вторым <i>arg</i>

f-string

Форматирование

```
name = 'Peter'
age = 23
print(f'{name} is {age} years old')
# Peter is 23 years old
```

Выражения

```
bags = 3
apples_in_bag = 12
print(f'Всего {bags * apples_in_bag} яблок')
# Всего 36 яблок
```

Использование словарей

```
user = {'name': 'John Doe', 'job': 'gardener'}
print(f"{user['name']} is a {user['job']}")
# John Doe is a gardener
```

f-string (cont)

Форматирование float

```
val = 12.3
```

```
print(f'{val:.2f}')
```

```
print(f'{val:.5f}')
```

```
# 12.30
```

```
# 12.30000
```

Процент

```
val = 1/7.0
```

```
print(f'{val}')
```

```
print(f'{val:.2%}')
```

```
# 0.1428 571 428 5714285
```

```
# 14.29%
```

Многострочные f-строки

```
name = 'John Doe'
```

```
occupation = 'gardener'
```

```
age = 34
```

```
msg = f'''name: {name}
```

```
age: {age}
```

```
occupation: {occup ati on}'''
```

```
print(msg)
```

```
# name: John Doe
```

```
# age: 34
```

```
# occupa tion: gardener
```

Разделитель тысячных

```
val = 1_200_ 400_001
```

```
print(val)
```

```
print(f'{val:_}')
```

```
print(f'{val:,}')
```

```
# 1200100001
```

```
# 1_200_ 100_001
```

```
# 1,200, 100,001
```

Больше информации на странице [Python f-string](#)