9кл

**Задача 1.1**

**Условие**

1. Создать произвольный список
2. Добавить новый элемент типа str в конец списка
3. Добавить новый элемент типа int на место с индексом
4. Добавить новый элемент типа list в конец списка
5. Добавить новый элемент типа tuple на место с индексом
6. Получить элемент по индексу
7. Удалить элемент
8. Найти число повторений элемента списка

**Решение - Интерактивный режим**

```
# 1. Создаем список
>>> lst = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

# 2. Добавляем элемент с типом str в конец списка
>>> lst.append('Строка')
#[1, 1, 1, 1, 1, 1, 1, 1, 1, 'Строка']

# 3. Вставляем элемент со значением 189 на место с индексом 4
>>> lst[4] = 189
# Результат: [1, 1, 1, 1, 189, 1, 1, 1, 1, 1, 'Строка']

# 4. Добавляем вложенный список ['a', 'b', 'a', 'hello']
lst.append(['a', 'b', 'a', 'hello'])
# Результат: [1, 1, 1, 1, 189, 1, 1, 1, 1, 1, 'Строка', ['a', 'b', 'a', 'hello']]

# 5. Вставляем кортеж со значением (1, 6, 89) на место с индексом -3 (3-й элемент с конца списка)
>>> lst[-3] = (1, 6, 89)
# Результат: [1, 1, 1, 1, 189, 1, 1, 1, 1, (1, 6, 89), 'Строка', ['a', 'b', 'a', 'hello']]

# 6. Получаем значение элемента с индексом 0
>>> lst[0]
1
# Получаем значение элемента с индексом -1(последний элемент списка)
>>> lst[-1]
['a', 'b', 'a', 'hello']

# 7. Удаляем элемент со значением 189
>>> lst.remove(189)
# Результат: [1, 1, 1, 1, 1, 1, 1, 1, 1, (1, 6, 89), 'Строка', ['a', 'b', 'a', 'hello']]

# 8. Считаем количество элементов в списке со значением 1
>>> lst.count(1)
8
```

**Задача 1.2**

**Условие**

Получите первый и последний элемент списка

**Решение - Интерактивный режим**

```
>>> lst = ['Нулевой элемент', 'One', 2, 3, 4, (5, 5, 5)]
>>> lst[0]
'Нулевой элемент'
>>> lst[-1]
(5, 5, 5)
```

9кл

## Задача 1.4

**Условие**

Проверить, есть ли в последовательности дубликаты

**Решение - Интерактивный режим**

```
# Создаем список с дубликатами lst
>>> lst = [0, 0, 1, 2, 3, 4, 5, 5, 6, 7]

# На основе списка создаем множество st
# Помним про основное свойство множеств - они не могут содержать дубликатов
# Поэтому если lst содержит дубликаты, то при создании множества на его основе дубликаты будут удалены
>>> st = set(lst)

# А значит количество элементов в списке и во множестве будет различаться
# Сравниваем количество элементов с помощью встроенного метода len()
>>> len(st) == len(lst)
False
# Длины не равны, значит в изначальном списке были дубликаты
```

## Задача 1.6

**Условие**

1. Создать множество(set)
2. Создать неизменяемое множество(frozenset)
3. Выполнить операцию объединения созданных множеств
4. Выполнить операцию пересечения созданных множеств

**Решение**

```
# 1. Создаем изменяемое множество
st = {'it', 'is', 'set', 1}

# 2. Создаем неизменяемое множество
frozen_st = frozenset({'it', 'is', 'frozen', 'set', 2})

# 3. Выполняем операцию объединения созданных множеств
# Результатом объединения будет множество, содержащее все элементы обоих множеств(без дубликатов)
union = st | frozen_st
# Результат: {'frozen', 1, 2, 'set', 'it', 'is'}

# 4. Выполняем операцию пересечения созданных множеств
# Результатом пересечения будет множество, содержащее элементы, присутствующие одновременно в обоих множествах
intersection = st & frozen_st
# Результат: {'it', 'set', 'is'}
```

9кл

## Задача 1.5

**Условие**

1. Создать произвольный словарь
2. Добавить новый элемент с ключом типа str и значением типа int
3. Добавить новый элемент с ключом типа кортеж(tuple) и значением типа список(list)
4. Получить элемент по ключу
5. Удалить элемент по ключу
6. Получить список ключей словаря

**Решение - Интерактивный режим**

```
# 1. Создаем словарь
dct = {1: 'value_1', 2: 'value_2', 3: 'value_3'}

# 2. Добавляем в словарь новый элемент с ключом 'str_key' и значением 12345
dct['str_key'] = 12345
# Содержимое словаря: {1: 'value_1', 2: 'value_2', 3: 'value_3', 'str_key': 12345}

# 3. Добавляем в словарь новый элемент с ключом ('it', 'is', 'tuple') и значением [11, 22, 'list_value', 33, {1, 2, 3}]
dct[('it', 'is', 'tuple')] = [11, 22, 'list_value', 33, {1, 2, 3}]
# Содержимое словаря: {1: 'value_1', 2: 'value_2', 3: 'value_3', 'str_key': 12345, ('it', 'is', 'tuple'): [11, 22, 'list_value', 33, {1, 2, 3}]}

# 4. Получаем элемент словаря по ключу 'str_key'
# Способ 1: Напрямую - в случае отсутствия ключа формируется исключение
item_by_key_v1 = dct['str_key']
# Значение item_by_key_v1 равно 12345
# Способ 2: Через функцию get() - в случае отсутствия ключа возвращается дефолтное значение 'No item'
item_by_key_v2 = dct.get('str_key', 'No item')
# Значение item_by_key_v2 так же равно 12345

# 5. Удаляем элемент с ключом '2' из словаря
item_deleted = dct.pop(2, 'No item')
# Содержимое словаря: {1: 'value_1', 3: 'value_3', 'str_key': 12345, ('it', 'is', 'tuple'): [11, 22, 'list_value', 33, {1, 2, 3}]}

# 6. Получаем ключи словаря
# Возвращаемое значение: dict_keys([1, 3, 'str_key', ('it', 'is', 'tuple')])
keys = dct.keys()
```

## Задача 1.3

**Условие**

Поменяйте значения переменных a и b местами

**Решение - Интерактивный режим**

```
>>> a = 100
>>> b = 'Строка'
>>> a, b = b, a
>>> a
'Строка'
>>> b
100
```