



# Основы проектной деятельности

Второе занятие



# Списки

Списки в Python – упорядоченный изменяемый набор объектов произвольных типов

Пример:

```
Num = [9, 8, 7, 6, 5, 4, 3, 2, 1]
```



# Списки

Все элементы в списка автоматически нумеруются, начиная с 0.

Пример:

```
Num = [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
print(Num[0])          ##Выведет 9
```

```
print(Num[3])          ##Выведет 6
```

```
print(Num[8])          ##Выведет 1
```



## Вывод списка

```
Num = [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
print(Num)
```

```
##Выведет [9, 8, 7, 6, 5, 4, 3, 2, 1]
```



## Умножение списков

```
a = [1, 2]
```

```
a = a*2
```

```
print(a)
```

```
##Выведет [1, 2, 1, 2]
```



## Сложение (Объединение) двух списков

```
a = ['a', 'b']
```

```
b = [1, 2]
```

```
c = a+b
```

```
print(c)          ##Выведет ['a', 'b', 1, 2]
```



## Методы работы со списками

**`list.append(x)`** – добавляет элемент **x** в конец списка. Пример:

```
a = [1, 2]
```

```
a.append(1)
```

```
print(a) ##Выведет [1, 2, 1]
```



## Методы работы со списками

**list.extend(x)** – делает то же самое, что и операция с суммой списков, то есть добавляет в конец списка **list** элементы списка **x**.

```
a = [1, 2]
```

```
b = [3, 4]
```

```
a.extend(b)
```

```
print(a) ##Выведет [1, 2, 3, 4]
```





## Методы работы со списками

**list.insert(y, x)** – вставляет элемент **x** в список **list** порядковым номером **y**.

```
a = [1, 2]
```

```
a.insert(0, 2)
```

```
print(a) ##Выведет [2, 1, 2]
```



## Методы работы со списками

**list.remove(x)** – удаляет первый элемент в списке **list**, который равен значению **x**.

```
a = [1, 2, 1]
```

```
a.remove(1)
```

```
print(a) ##Выведет [2, 1]
```



## Методы работы со списками

**list.pop(y)** – удаляет элемент списка **list** по порядковому номеру **y**. Если порядковый номер не задан, то удаляется последний элемент списка.

```
a = [1, 2, 1]
```

```
a.pop(0)
```

```
print(a) ##Выведет: [2, 1]
```



## Методы работы со списками

**list.count(x)** – возвращает количество элементов со значением **x** в списке **list**.

```
a = [1, 2, 1, 3]
```

```
b = a.count(1)
```

```
print(b) ##Выведет 2
```



## Методы работы со списками

**list.sort()** – сортировка списка. По умолчанию сортировка идет по возрастанию. Также в виде параметра в метод можно передать функцию сортировки.

```
a = [2, 1, 3]
```

```
a.sort()
```

```
print(a) ##Выведет [1, 2, 3]
```



## Методы работы со списками

**list.reverse()** – переворачивает (реверсирует список).

```
a = [2, 1, 3]
```

```
a.reverse()
```

```
print(a) ##Выведет [3, 1, 2]
```



# Методы работы со списками

**list.copy()** – копирует список.

```
a = [2, 1, 3]
```

```
newList = a.copy()
```

```
a = a*2
```

```
print(newList) ##Выведет [2, 1, 3]
```

```
print(a) ##Выведет [2, 1, 3, 2, 1, 3]
```



# Методы работы со списками

**list.clear()** – очищает список.

```
a = [2, 1, 3]
```

```
a.clear()
```

```
print(a) ##Выведет []
```





# Кортеж

Кортеж – неизменяемый список. Но можно с ними выполнять операции, например сложение, умножение на число, и некоторые другие операции.

Пример кортежа:

`a = (3, 5, 6, 1, 8)`

**## Обращаем внимание на круглые скобки, при  
создания списка используются квадратные**



## Модуль random

Random – модель представляющий функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.



## Методы random

## random.random()

`random.random()` – возвращает случайное число от 0 до 1.

Пример:

```
a = random.random()
```

```
print(a)          ##Выведет  случайное число от 0 до 1
```



## Методы random

## random.randint(A, B)

random.randint(A, B) – случайное целое число  $N$ ,  $A \leq N \leq B$ .

Пример:

```
a = random.randint(2, 10)
```

```
print(a)          ##Выведет случайное число от 2 до 10
```



## Методы random

## random.randrange(A, B, step)

в randrange() передается три аргумента, то первые два – это границы диапазона, как в случае с двумя аргументами, а третий – так называемый шаг. Если, например, функция вызывается как randrange(10, 20, 3), то "случайное" число будет выбираться из чисел 10, 13, 16, 19:

Пример:

```
a = random.randrange(10, 20, 3)
```

```
print(a)          ##Выведет  случайное число от 10 до 20
```



## Методы random

## random.choice(sequence)

random.choice(sequence) – возвращает случайный элемент непустой последовательности.

Пример:

```
l = [3, 7, 1, 6, 9, 4]
```

```
a = random.choice(l)
```

```
print(a)          ##Выведет случайный элемент из списка l
```



## Пример

с randint()

Создадим список students, в который запишем имена студентов.

```
students = [ 'Анна' , 'Федор' , 'Степан' , 'Екатерина' ]
```



## Пример

с randint()

Задача заключается в следующем, распределить студентов на 2 группы.

Создадим два пустых списка.

```
g1 = []
```

```
g2 = []
```





## Пример

c randint()

Теперь нам необходимо случайно распределить студентов.

```
for c in students:
    if random.randint(0, 1) == 0:
        g1.append(c)
    else:
        g2.append(c)

print('Первая группа: ', g1)
print('Вторая группа: ', g2)
```



## Пример

с `choice()`

Решим эту же задачу только используя другой метод `random`. Только дополнительно введем еще одно условие: в каждой группе должно быть 2 человека



# Пример

с choice()

```
students = ['Анна', 'Федор', 'Степан', 'Екатерина']

g1 = []
g2 = []

for i in range(1, 3):
    c = random.choice(students) ## Выбираем случайного студента
    g1.append(c)                ## Добавляем его в группу
    students.remove(c)         ## И чтобы он больше не участвовал в отборе,
                                ## потому что он уже попал в группу
                                ## удаляем его из списка претендентов

for i in range(1, 3):
    c = random.choice(students)
    g2.append(c)
    students.remove(c)

print('Первая группа: ', g1)
print('Вторая группа: ', g2)
```