



Основные типы данных в Python. Массивы



- Массив – набор данных **одного типа**, имеющих одно и то же имя, и различающихся по номеру(индексу).
- Индекс элемента – порядковый номер элемента массива $a_1 \ a_2 \ a_3 \ \dots \ a_n$



Массивы

```
graph TD; A[Массивы] --> B[Статические  
е  
(кортеж)]; A --> C[Динамические  
кие  
(список)];
```

Статически
е
(кортеж)

Динамичес
кие
(список)



Статический тип

- **Место** для всех данных **отводится сразу**, во время программы не изменяется.
- Используется при объявлении переменных.

Динамический тип

- **Количество элементов заранее неизвестно**, меняется во время исполнения программы.
- Память отводится и освобождается самой программой.

СПИСКИ

- Списки — это изменяемые последовательности.



- **Список** в Python — это упорядоченный набор объектов, в список могут одновременно входить объекты разных типов (числа, строки и другие структуры, в частности, он может содержать другие списки и кортежи). Объекты, входящие в список – это элементы списка.


```
>>> numbers = [1, 2, 3, 4, 5]
```



Создание пустого списка

- `numbers = []`
- С помощью функции `list()`:
`numbers = list()`

Функция — это фрагмент кода, который выполняет какую-то задачу.



Для обращения к элементам списка надо использовать индексы, которые представляют **номер** элемента в списке. Индексы начинаются с нуля. То есть второй элемент будет иметь индекс 1.

Для обращения к элементам с конца можно использовать отрицательные индексы, начиная с -1. То есть у последнего элемента будет индекс -1, у предпоследнего -2 и так далее.

```
numbers = [1, 2, 3, 4, 5]
print(numbers[0]) # 1
print(numbers[2]) # 3
print(numbers[-3]) # 3
```




Способы заполнения списка

- С заранее заданным набором данных:

```
numbers = [1, 2, 3, 4, 5]
```

- Также, список можно заполнить вручную с помощью цикла, метода `append` и команды `input`

```
mas = []
```

```
for i in range(20): //ввод 20 элементов в цикле
```

```
    mas.append(int(input())) // добавление  
целого числа, введенного с клавиатуры, в конец списка
```


Способы заполнения списка

- **Случайными числами:**

Функция **random** позволяет генерировать случайные числа. Прежде чем использовать **random**, необходимо подключить ее с помощью инструкции: **import random** .
`random.randint(a, b)` — возвращает псевдослучайное целое число в отрезке от a до b

```
import random                //подключение модуля
случайных чисел random mas=[] //
объявление пустого списка
for i in range(0, 10):
    mas.append(random.randint(0, 100))

//заполнение списка 10-ю случайными числами в диапазоне от 0
до 100
print(mas)                    // вывод списка
```

Способы заполнения списка

- Создание списка случайных чисел с помощью генератора списка

```
import random
```

```
mas = [random.randint(0, 100) for i in range(0, 10)]
```

Метод `randint(0, 100)` генерирует случайное целое число в диапазоне от 0 до 100.

```
mas = [i for i in range(1,15)]
```

range(end): создается набор чисел от 0 до числа `end`

range(start, end): создается набор чисел от числа `start` до числа `end`

range(start, end, step): создается набор чисел от числа `start` до числа `end` с шагом `step`