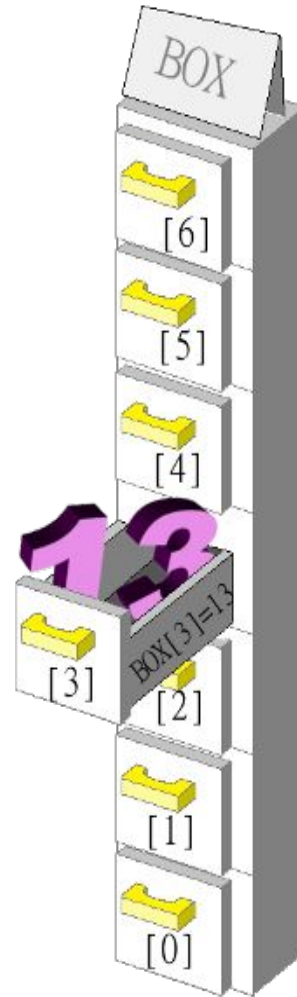


# Что общего?



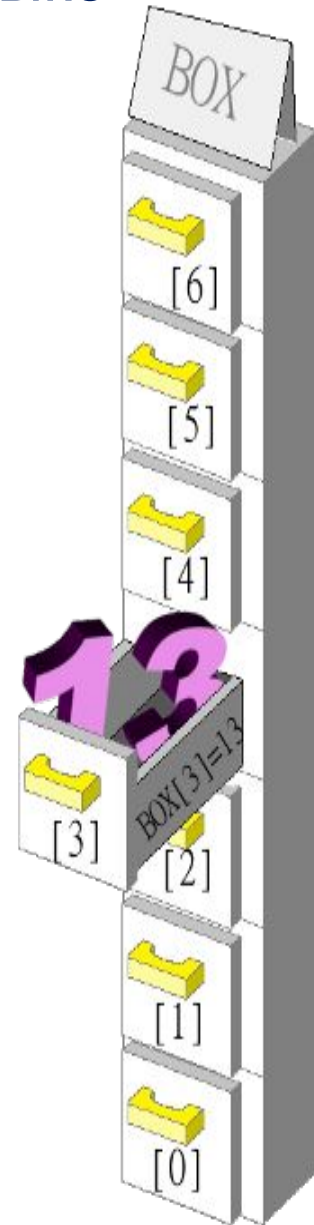
## Список класса

1. Иванов
2. Петров
3. Сидоров
4. Кошкин
5. Мышкин

A R R A Y

Раздел: Программирование алгоритмов на языке  
программирования Python

# Тема урока: Одномерный массив (9 класс)



# Цель обучения:

9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов

# Критерии оценивания

## Знание

- Дает определение понятию массив (список)

## Понимание

- Определяет элементы массива

## Применение

- Разрабатывает программный код на языке Python с использованием одномерного массива

# Что такое массив?

---

## Особенности массива:

- все элементы **одного типа**
- массив имеет **одно имя**
- все элементы расположены в памяти **рядом**

Задание: Используя особенности массива попробуйте сформулировать определение понятию массив.

**Массив** – это ...

Массивы (списки) в Питоне, как и в других языках программирования — это определенное количество элементов одного типа, которые имеют общее имя, и каждый элемент имеет свой индекс (порядковый номер).

# Характеристика массива

*Имя* - название массива

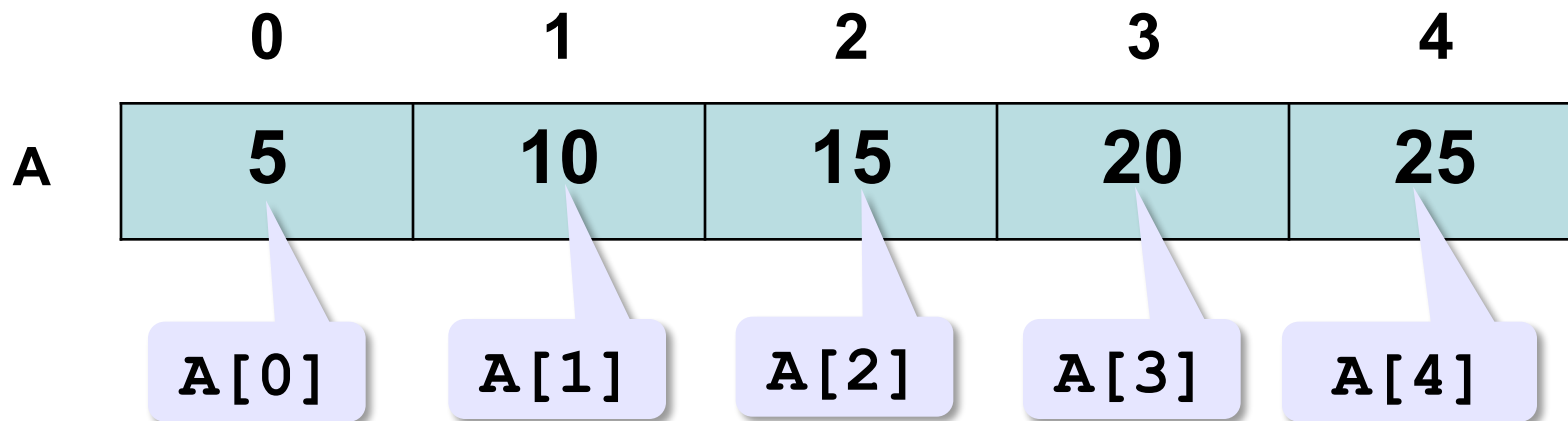
*Индекс* - номер элемента в массиве

*Элемент* - каждое значение в массиве

*Размер* - количество элементов в массиве



## Определите элементы массива



**Имя** (название массива) – **A**

**Индекс** (номера элемента массива) - **0,1,2, 3, 4**

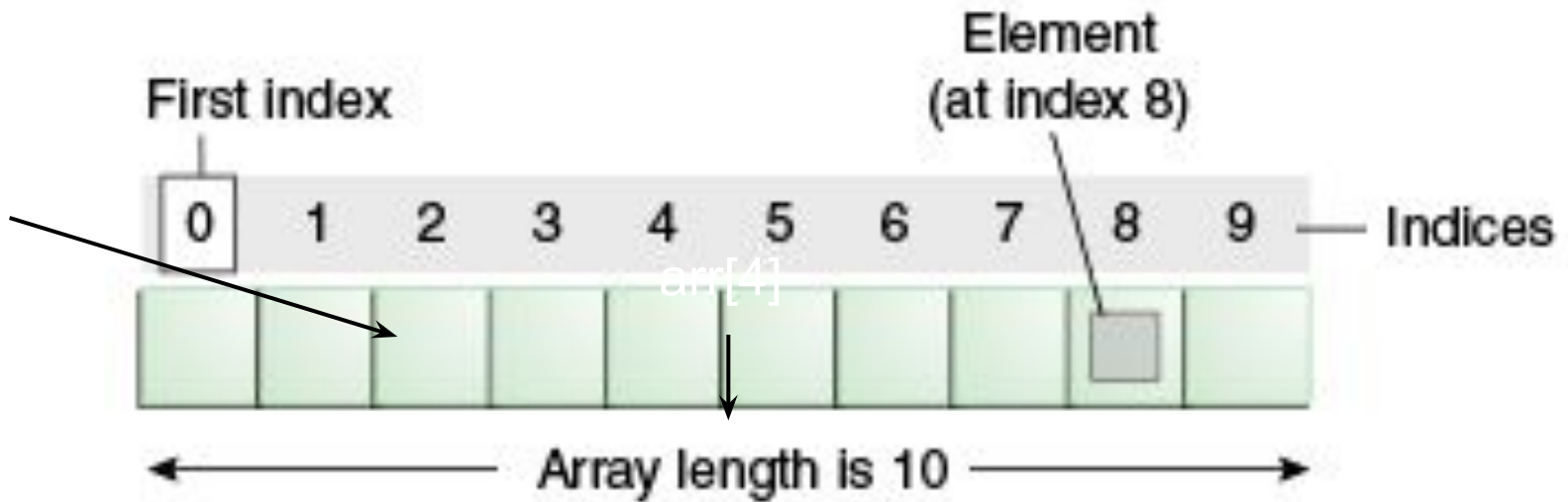
**Элемент массива** – **A[0], A[1], A[2], A[3], A[4]**

**Значение элемента массива** (каждое значение в массиве)- **5, 10, 15, 20, 25**

**Размер** (количество элементов в массиве) - **5**



- Обращение к элементам массива `arr` – операция `arr[2]`



# Работа с массивами

- Как напечатать все элементы массива `arr` длиной `N`?

```
for i in range(N):  
    print( arr[i] )
```

# Работа с массивами

- Как заполнить массив длины N числами 0,1,2...?

```
for i in range(N):  
    arr[i] = i
```

## Для работы с массивами используются *списки*

Списки являются упорядоченными последовательностями, которые состоят из различных объектов (значений, данных), заключающихся в квадратные скобки [ ] и отделяющиеся друг от друга с помощью запятой.

# Массив в Python – лист(list)

В Python массивы реализованы немного не стандартно – они могут содержать любые элементы.

## 1. Получение списка через присваивание конкретных значений

Примеры создания списков со значениями:

```
l = [25, 755, -40, 57, -41] # список целых чисел
```

```
l = [1.13, 5.34, 12.63, 4.6, 34.0, 12.8] # список из дробных чисел
```

```
l = ["Sveta", "Sergei", "Ivan", "Dasha"] # список из строк
```

```
l = [[0, 0, 0], [1, 0, 1], [1, 1, 0]] # список, состоящий из списков
```

```
l = ["Москва", "Титова", 12, 148] # смешанный список
```

Поэтому их правильно называть листами!

# Создание листов

```
arr = [13,3,7,-5,53]
```

```
len(arr) - ?
```

```
arr[2] = ?
```

```
arr = [1, 2, 3] * 3
```

```
print(arr) - ?
```

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

Как создать массив, состоящий из N нулей?

```
arr = [0]*N
```

Можно получать доступ к объектам списка по их индексам, извлекать срезы, измерять длину списка:

1. `li = ['a','b','c','d','e','f']`

`len(li)`

6

2. `li[0]`

'a'

3. `li[4]`

'e'

4. `li[0:3]`

['a', 'b', 'c']

5. `li[3:]`

['d', 'e', 'f']

## Операции с списками

Списки можно складывать с помощью знака «+»  
дублировать с помощью «\*»

**arr1 = [0,1,2]**

**arr2 = [3,4,5]**

**arr = arr1+arr2**

**arr # [0,1,2,3,4,5]**

**arr[2] = 777**

**arr # [0,1,777,3,4,5]**

**arr = [1, 3] + [4, 23] + [5]**

**arr # 1, 3, 4, 23, 5]**

**arr[3]= [[0,0],[0,1],[1,1]] \* 2**

**arr # [[0, 0], [0, 1], [1, 1], [0, 0], [0, 1], [1, 1]]**



# Массивы в Python: списки

```
A = [1, 3] + [4, 23] + [5]
```

Результат: [1, 3, 4, 23, 5]

```
A = [0] * 10
```

 Что будет?

Результат: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

```
A = list(range(10))
```

Результат: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

# Практическая работа №1

1. Создайте два любых списка и свяжите их с переменными.



2. Измените во втором списке последний объект. Выведите список



3. Соедините оба списка в один, присвоив результат новой переменной. Выведите получившийся список на экран.



# Генераторы списков

```
A = [ i for i in range(10) ]
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```



Что будет?

```
A = [ i*i for i in range(10) ]
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
from random import randint
```

```
A = [ randint(20, 100)  
      for x in range(10) ]
```

случайные  
числа

```
A = [ i for i in range(100)  
      if isPrime(i) ]
```

условие  
отбора

# Вывод массива на экран

Как список:

```
print ( A ) [1, 2, 3, 4, 5]
```

В строчку через пробел:

```
for i in range(N):  
    print ( A[i], end=" " ) 1 2 3 4 5
```

или так:

```
for x in A:  
    print ( x, end=" " ) 1 2 3 4 5
```

или так:

```
print ( *A ) ↔ print (1, 2, 3, 4, 5)
```

# Заполнение случайными числами

```
from random import randint
N = 10
A = [0]*N
for i in range(N):
    A[i] = randint(20,100)
```

или так:

```
from random import randint
N = 10
A = [ randint(20,100)
      for x in range(N) ]
```

случайные  
числа  
[20,100]

# Практическая работа №2

1. Заполните массив случайными числами в интервале  $[0,100]$  и найдите среднее арифметическое его значений.
2. Заполните массив случайными числами в интервале  $[0,100]$  и подсчитайте отдельно среднее значение всех элементов, которые  $<50$ , и среднее значение всех элементов, которые  $\geq 50$ .

# Рефлексия урока

Учащиеся вспоминают цель урока.

- Чему я научился?
- Что я могу делать лучше?
- Какой этап урока был самым полезным?  
Почему?

# Домашнее задание

Стр. 120-127 читать