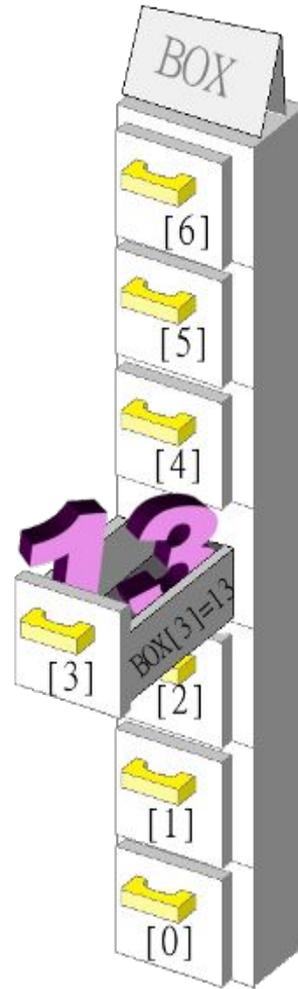


# Что общего?

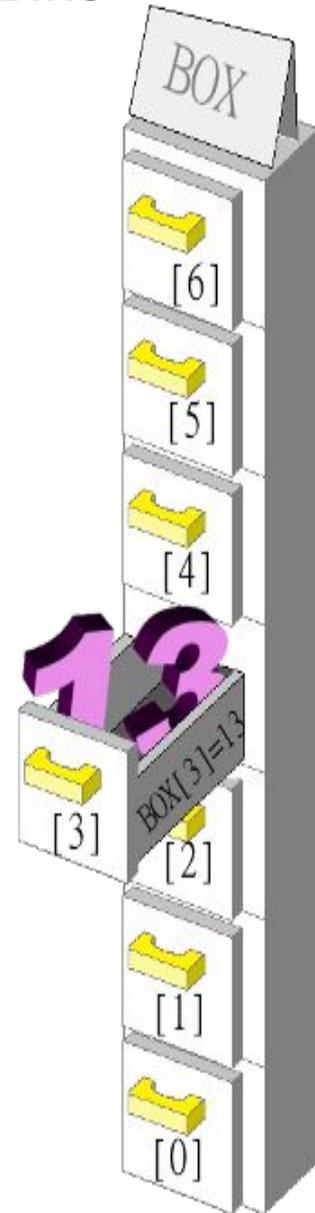


## Список класса

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

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

## **Вы узнаете:**

- что такое одномерный массив;
- какие бывают операторы для работы с одномерными массивами на языке программирования Python.

## **Вы научитесь:**

- описывать одномерные массивы на языке программирования Python;
- организовывать ввод и вывод элементов одномерного массива;
- создавать программы на языке программирования Python 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

Вспомним, что мы изучили по  
разделу программирование в  
предыдущих классах

# Python



Простой, гибкий и невероятно популярный язык, который используется практически во всех областях современной разработки.

С его помощью можно создавать веб-приложения, игры, заниматься анализом данных, автоматизировать задачи системного администрирования и многое другое

# Задание 1

Определите значение выражения в языке Python

$7//3+7\%3$

A. 4

B. 2

 C. 3

D. Вернет ошибку

## Задание 2

Какую функцию нужно использовать, чтобы текст появился на экране

\_\_\_\_\_ (“Hi”)

-  A. print
- B. input
- C. output
- D. write

# Задание 3

Каким будет результат выполнения программы?

**A=7**

**If A>5:**

**print(“пять”)**

**If A>8:**

**print(“восемь”)**

**Ответ: пять**

# Задание 4

Каким будет результат выполнения программы?

**If 1+1==2:**

**if 2\*2==8:**

**print("if")**

**else:**

**print("else")**

A. if

 B. else

C. If else

# Задание 5

Сколько чисел выведет эта программа?

```
i=3
```

```
While i>=0:
```

```
    print(i)
```

```
    i - =1
```

**Ответ: 4**

# Задание 6

Сколько чисел выведет эта программа и какое будет последнее выведенное число?

```
j=10
```

```
for i in range (j):  
    print(i)
```

**Ответ: программа выведет 10 чисел**  
**Последнее число 9**

# Списки и массивы

- Базовым типом структуры данных в Python является **список**. В других языках программирования такой структурой является **массив**.
- В Python для работы с массивом подключают модуль *array*, но чаще всего используют **список**.
- **Отличие списков от массивов** заключается в функциях, которые можно выполнять с ними.
  - Список более широкое понятие.
  - В массиве все элементы имеют одинаковый тип, а в списке типы элементов могут быть различными.

## Одномерный массив

Описание  
массива

Начальное  
значение индекса

Элементы массива  
одного типа

Имя

```
from array import array  
A = array('i', 2, 5, 10, 18, 20)
```

Список

```
A = [12, 'b', 34.6, 'derevo']
```

Список A состоит из  
элементов целого,  
вещественного,  
символьного,  
строкового типа

# Основные понятия

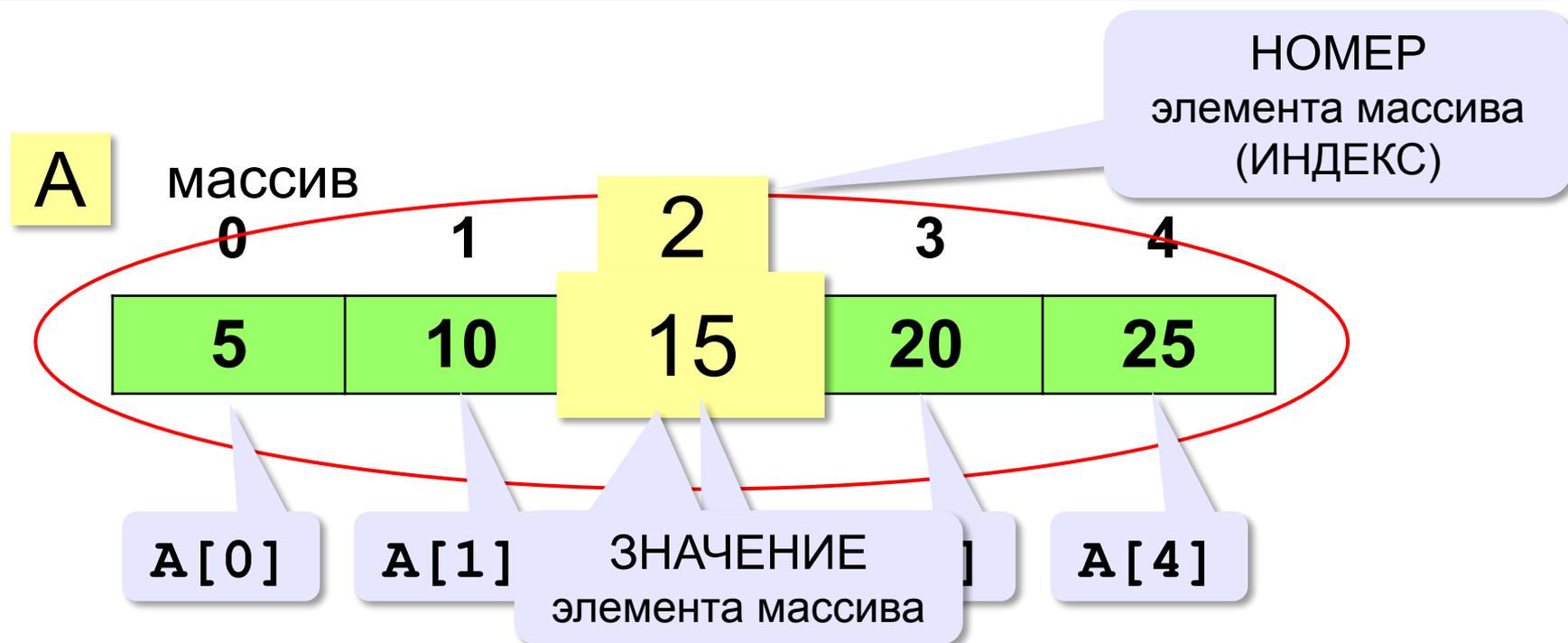
- **Список** – набор элементов произвольного типа.
- **Массив** – набор данных одного типа, записанных под определенным именем.
- **Длина (размер) массива** – количество элементов в массиве.
- **Одномерный массив** – массив, в котором для задания элемента требуется один индекс.
- **Имя** - название массива
- **Индекс** - номер элемента в массиве
- **Элемент** - каждое значение в массиве

В Python нумерация элементов начинается с 0. Индекс элемента указывается в квадратных скобках.

Пример массива *Num* из десяти целых чисел: 45, 78, 3, 9, 23, -12, 67, -15, 7, 54.

<u>Num[0]</u>	<u>Num[1]</u>	<u>Num[2]</u>	<u>Num[3]</u>	<u>Num[4]</u>	<u>Num[5]</u>	<u>Num[6]</u>	<u>Num[7]</u>	<u>Num[8]</u>	<u>Num[9]</u>
45	78	3	9	23	-12	67	-15	7	54

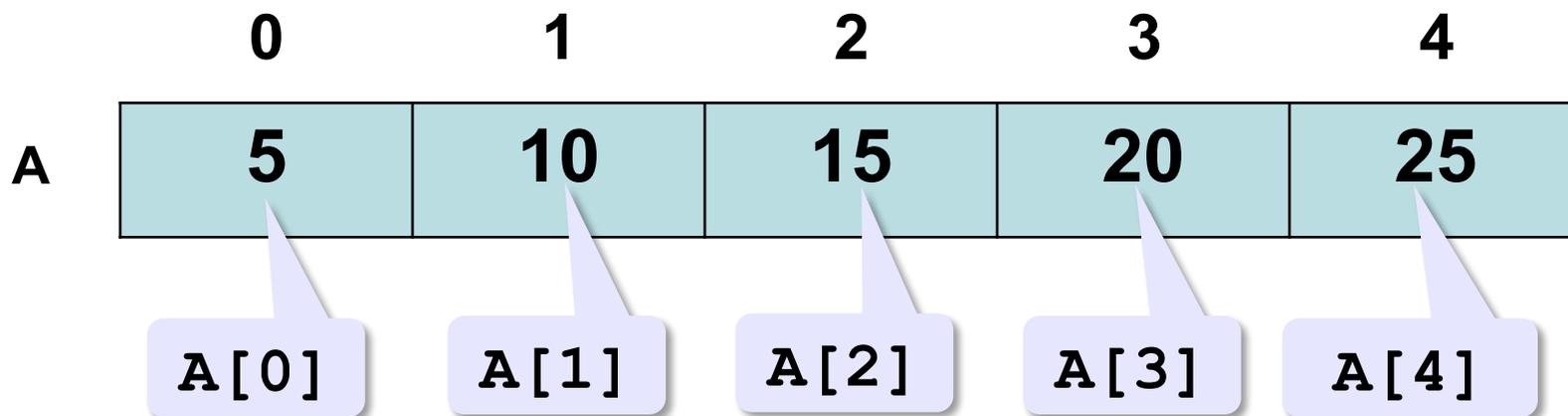
# Обращение к элементу массива



**Индекс элемента** — это значение, которое указывает на конкретный элемент массива.

**!** Нумерация с нуля!

## 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**

При объявлении массива указывается имя массива и тип его элементов. Размер массива не задается, он может быть расширен новыми элементами любое время. Для использования **массива** в Python понадобится импортирование стандартного модуля **array**.

```
import array as arr
    a = arr.array('i')
```

где `arr` – название, которое мы задаем сами для дальнейшего использования модуля `array`  
`a` – имя массива (в данном случае);  
`i` – тип элементов массива (`integer`)

Нам нужен массив, состоящий из 7 элементов.

*#объявляем массив A с указанием типа  
элементов - целые*

```
import array as arr  
A=array('i')
```

*#заполняем массив*

```
for i in range (0,7):  
    A.append(i)
```

Для ввода и вывода значения элементов массива используется цикл с параметром

*#объявляем массив A с указанием типа элементов - целые*

```
import array as arr  
A=array('i')
```

*#заполняем массив*

```
for i in range (0,7):  
    A.append(i)
```

*#выводим массив*

```
for i in range (0,7):  
    print(A[i])
```

# Создание массивов (списков) в программе

- **1-й способ.** Чтобы задать массив (список) в программе, можно указать его имя и в квадратных скобках через запятую перечислить элементы. Значения строковых элементов записываются в апострофах.

```
Num = [45, 78, 3, 9, 23, -12, 67, -15, 7, 54]
```

```
Weekday = ['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday']
```

## Создание массивов (списков) в программе

- **2-й способ.** Можно описать пустой список и с помощью цикла организовать ввод элементов с клавиатуры.

```
A=[]
```

```
A = [ input() for i in range(5) ]
```

Организуется ввод пяти элементов. Параметр цикла  $i$  определяет индекс элемента. Функция `range(5)` позволяет задать значения параметра от 0 до 4.

# Создание массивов (списков) в программе

- **3-й способ.** Заполнение массива по определенному правилу. В данном случае можно использовать генератор списка.

**Пример 1.** Массив `a` из пяти элементов заполнить десятками

`a = [10]*5`

Результат: `a = [10, 10, 10, 10, 10]`

## Создание массивов (списков) в программе

- **3-й способ.** Заполнение массива по определенному правилу. В данном случае можно использовать генератор списка.

**Пример 2.** Формирование массива `b` из чисел от 0 до 9.

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

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

# Создание массивов (списков) в программе

**Пример 3.** Формирование массива из строки.

```
str = [str * 2 for str in 'plan']
```

Результат: str = ['pp', 'll', 'aa', 'nn']

**Пример 4.** Заполнение случайными числами.

```
from random import randint
```

```
c = [randint(10,20) for x in range(5)]
```

Результат: массив содержит пять случайных чисел из диапазона от 10 до 20.

## Пример 5. Изменение значений отдельных элементов

```
Num = [45, 78, 3, 9, 23, -12, 67, -15, 7, 54]
```

```
Num[0]=98
```

```
Num[7]= - 8
```

Результат: Num = [98, 78, 3, 9, 23, -12, 67, -8, 7, 54]

Пример 6. Вывод отдельных элементов

```
Num = [45, 78, 3, 9, 23, -12, 67, -15, 7, 54]  
print(Num[2])  
print(Num[5])
```

Результат: 3 -12

Для вывода всех элементов используется  
цикл с параметром:

```
Weekday = ['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday']  
print('Дни недели')  
for i in range(7):print(Weekday[i])
```

# Работа с массивами

- Как напечатать все элементы массива  $A$  длиной  $N$ ?

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

# Вывод массива на экран

Весь массив сразу:

```
print( A )
```

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

По одному элементу:

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

В столбик

или так:

```
for x in A:  
    print( x )
```

для всех элементов в массиве A



Как вывести в строчку?

```
for x in A:  
    print( x, end=" " )
```

пробел между элементами

# Ввод с клавиатуры

```
for i in range(N):  
    A[i] = int(input())
```



Что плохо?

или так:

```
A = [int(input())  
      for i in range(N)]
```

С подсказкой для ввода:

```
for i in range(N):  
    s = "A[" + str(i) + "]= "  
    A[i] = int(input(s))
```

```
A[0] = 5  
A[1] = 12  
A[2] = 34  
A[3] = 56  
A[4] = 13
```

# Работа с массивами

- Как заполнить массив  $K$  длины  $N$  числами  $0, 1, 2, \dots$ ?

**for  $i$  in range( $N$ ):**

**$K[i] = i$**

# Массивы в Python: списки

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

Результат: `[1, 3, 4, 23, 5]`

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



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]`

Количество элементов можно определить с помощью функции `len()`.

Для добавления в конец нового элемента применяется метод `append()`.

Пример 7. Дан список цветов. Добавить в конце красный цвет. Вывести на экран список.

```
color = ['Blue', 'Green', 'Orange', 'Gray']
color.append('Red')
for i in range(len(color)): print(color[i])
```

**Списки можно соединять  
(конкатенировать) с помощью знака  
«+».**

```
m1=['a', 10, 'b']  
m2=[2.5, 'number', 65]  
m=m1+m2  
print (m)
```

Результат: m=['a', 10, 'b', 2.5,'number', 65]

В списке можно повторить элементы

```
a=[0,1,2]*3  
print (a)
```

Результат: a=[0, 1, 2, 0, 1, 2, 0, 1, 2]

# 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

# 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]

---

**Размер массива можно узнать, используя функцию len()**

```
C=[13,3,7,-5,53]*3  
print(C)  
print(len(C))
```

# Рефлексия урока

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