

Класс: 8

Тема урока: Списки (массивы).

Цель урока:

1) обучающий аспект: познакомить учащихся с типом данных «список» и научить использовать его при составлении программ на языке программирования Python.

2) развивающий аспект: развитие мыслительной деятельности, речи, алгоритмического стиля мышления.

3) воспитательный аспект: воспитание эмоционально-положительной направленности на практическую деятельность, интереса к информатике, личной ответственности за результаты своей работы.

Тип урока: усвоение новых знаний.

Вид урока: урок-презентация, урок-лабораторное занятие.

Ресурсы: компьютеры, проектор, среда программирования Python IDLE, презентация.

Межпредметные связи: математика.

Основные понятия: список.

Ход урока (дидактическая структура урока)

1. Организационный этап (2 мин)	<u>УУД:</u> - Личностные - Коммуникативные
Приветствие, переключка.	

2. Проверка и разбор домашнего задания (10 мин)	<u>УУД:</u> - Личностные - Познавательные - Регулятивные - Знаково-символические - Коммуникативные
--	---

Заданное на прошлом уроке домашнее задание:

Написать программы:

- 1) С помощью цикла for вывести все чётные числа от 10 до 20.
- 2) С помощью цикла for вывести 10 первых степеней двойки.
- 3) С клавиатуры вводится 7 натуральных чисел. Вывести наибольшее из них.

3. Теоретическая часть (10 мин)

УУД:

- Личностные
- Познавательные
- Знаково-символические
- Коммуникативные

Список (list) - это тип данных, представляющий собой упорядоченный набор элементов. Элементы одно списка должны иметь одинаковый тип. Например, список строк, список целых чисел. Примеры списков:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

Здесь создали список с именем **a** и элементами строкового типа.

```
b = [17, 409, 88]
```

Здесь создали список с именем **b** и элементами - целыми числами.

Список (list) - это тип данных, представляющий собой упорядоченный набор элементов.

Элементы одно списка должны иметь одинаковый тип. Например, список строк, список целых чисел.

Примеры списков:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

```
b = [17, 409, 88]
```

Каждый элемент списка имеет свой номер (индекс), и с каждым элементом списка можно отдельно работать. Нумерация элементов начинается с нуля:

0	1	2	3	4
Андрей	Вера	Даша	Коля	Юра
a[0]	a[1]	a[2]	a[3]	a[4]

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

```
print(a[2])
```

Вывод: Даша

Каждый элемент списка имеет свой номер (индекс).
Нумерация элементов начинается с нуля:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

0	1	2	3	4
Андрей	Вера	Даша	Коля	Юра
a[0]	a[1]	a[2]	a[3]	a[4]

С каждым элементом списка можно работать отдельно:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]  
print(a[2])
```

Чтобы перебрать все элементы списка используется цикл:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

```
for i in range(5):
```

```
    print("Привет,", a[i])
```

Вывод:

Привет, Андрей

Привет, Вера

Привет, Даша

Привет, Коля

Привет, Юра

Чтобы перебрать все элементы списка используется цикл:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print("Привет", a[i])
```

Создание списка

Списки можно создавать разными способами.

1) Перечислением всех элементов (этот способ был рассмотрен выше):

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

Можно создать пустой список:

```
a = []
```

2) С помощью генератора списка:

```
a = [0]*5
```

Получится список, состоящий из пяти нулей: [0, 0, 0, 0, 0].

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

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

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

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

Способы создания списков

1) Перечислением всех элементов: `b = [17, 409, 88]`

Можно создать пустой список: `a = []`

2) С помощью генератора списка: `a = [0]*5`

Получится список, состоящий из пяти нулей: [0, 0, 0, 0, 0].

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

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

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

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

3) Путём ввода элементов с клавиатуры (каждый элемент с новой строки):

```
a = [0]*5
for i in range(5):
    print("a[" + str(i) + "]= ", end = "")
    a[i] = int(input())
```

Запустите эту программу и создайте список.

4) Путём ввода элементов с клавиатуры (все элементы в одной строке через пробел). Для этого используется метод `a.split()`, возвращающий список строк, которые получатся, если исходную строку разрезать на части по пробелам:

```
s = input() # пользователь вводит строку "1 2 3"
a = s.split()
print(a)
Вывод: ['1', '2', '3']
```

3) Путём ввода элементов с клавиатуры (каждый элемент с новой строки):

```
a = [0]*5
for i in range(5):
    print("a[" + str(i) + "]= ", end = "")
    a[i] = int(input())
```

4) Путём ввода элементов с клавиатуры (все элементы в одной строке через пробел). Для этого используется метод `a.split()`, возвращающий список строк, которые получатся, если исходную строку разрезать на части по пробелам:

```
a = [0]*5
s = input() # человек вводит строку "1 2 3"
a = s.split()
print(a)
```

Вывод: ['1', '2', '3']

Вывод списка

Списки можно выводить разными способами.

1) Самый простой способ - просто дать команду вывести список:

```
b = [17, 409, 88]
print(b)
```

Вывод: [17, 409, 88]

2) Вывод каждого элемента списка по-отдельности:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
```

```
print(a[i])
```

Способы вывода списков

1) С помощью функции `print()`:

```
b = [17, 409, 88]
print(b)
```

Вывод: [17, 409, 88]

2) Вывод каждого элемента списка по-отдельности:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print(a[i])
```

3) Вывод каждого элемента списка по-отдельности в одной строке:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print(a[i], end = " ")
```

4) Вывод элементов списка без обращения к индексам элементов:

```
fruits = ["Яблоко", "Банан", "Груша"]
for x in fruits:
    print(x, end = " ")
```

3) Вывод каждого элемента списка по-отдельности в одной строке:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print(a[i], end = " ")
```

4) Вывод элементов списка без обращения к индексам элементов:

```
fruits = ["Яблоко", "Банан", "Груша"]
for x in fruits:
    print(x, end = " ")
```

Работа со списком

1) Элементы списка можно менять:

```
a = [1, 2, 3, 4, 5, 6]
for i in range(6):
```

```
if a[i] % 2 == 0:
    a[i] = 0
print(a)
```

Вывод: [1, 0, 3, 0, 5, 0]

2) Можно добавлять элементы в конец списка. Для этого используется метод

a.append(x):

```
a = [1, 2, 3]
a.append(4)
print(a)
```

Вывод: [1, 2, 3, 4]

Работа со списком

1) Элементы списка можно изменять:

```
a = [1, 2, 3, 4, 5, 6]
for i in range(6):
    if a[i] % 2 == 0:
        a[i] = 0
print(a)
```

Вывод: [1, 0, 3, 0, 5, 0]

2) Можно добавлять элементы в конец списка. Для этого используется метод **a.append(x)**:

```
a = [1, 2, 3]
a.append(4)
print(a)
```

Вывод: [1, 2, 3, 4]

3) Можно расширять список, добавляя в его конец элементы другого списка. Для этого используется метод **a.extend(b)**:

```
a = [1, 2, 3]
b = [4, 5]
a.extend(b)
print(a)
```

Вывод: [1, 2, 3, 4, 5]

4) Списки можно копировать:

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

3) Можно расширять список, добавляя в его конец элементы другого списка. Для этого используется метод `a.extend(b)`:

```
a = [1, 2, 3]
b = [4, 5]
a.extend(b)
print(a)
```

Вывод: [1, 2, 3, 4, 5]

4) Списки можно копировать:

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

5) Функция нахождения длины списка `len(a)`:

```
a = ["Яблоко", "Банан", "Груша"]
x = len(a)
print(x)
```

Вывод: 3

6) Заполнение списка случайными элементами:

```
from random import randint
x = 7
a = [0]*x
for i in range(x):
    a[i] = randint(0,100)
```

7) Перестановка элементов списка в обратном порядке. Метод `a.reverse()`:

```
a = [0, 1, 2, 3, 4, 5]
a.reverse()
print(a)
```

Вывод: [5, 4, 3, 2, 1, 0]

5) Функция нахождения длины списка `len(a)`:

```
a = ["Яблоко", "Банан", "Груша"]
x = len(a)
print(x)
```

Вывод: 3

6) Заполнение списка случайными элементами:

```
from random import randint
x = 7
a = [0]*x
for i in range(x):
    a[i] = randint(0,100)
```

7) Перестановка элементов списка в обратном порядке.
Метод `a.reverse()`:

```
a = [0, 1, 2, 3, 4, 5]
a.reverse()
print(a)
```

Вывод: [5, 4, 3, 2, 1, 0]

8) Сортировка списка. Функция `sorted(a)`:

По возрастанию:

```
animals = ["кот", "еж", "барсук", "собака", "свинья"]
animals = sorted(animals)
print(animals)
```

Вывод: ['барсук', 'еж', 'кот', 'свинья', 'собака']

По убыванию:

```
a = [5, 65, 14, 700, 8]
a = sorted(a, reverse = True)
print(a)
```

Вывод: [700, 65, 14, 8, 5]

8) Сортировка списка. Функция `sorted(a)`:

По возрастанию:

```
animals = ["кот", "еж", "собака", "барсук"]
animals = sorted(animals)
print(animals)
```

Вывод: ['барсук', 'еж', 'кот', 'собака']

По убыванию:

```
a = [5, 65, 14, 700, 8]
a = sorted(a, reverse = True)
print(a)
```

Вывод: [700, 65, 14, 8, 5]

<p>4. Работа на компьютерах (20 мин)</p>	<p><u>УУД:</u></p> <ul style="list-style-type: none"> - Личностные - Регулятивные - Познавательные - Знаково-символические - Коммуникативные
<p>Учащиеся пишут программы на компьютерах под руководством учителя.</p> <p>Задачи:</p> <ol style="list-style-type: none"> 1) Создать список своих любимых фильмов. Вывести список тремя способами: а) в строчку; б) в столбик; в) в строчку через запятую. 2) Ввести с клавиатуры список фамилий писателей. Отсортировать список в алфавитном порядке и вывести его. 3) Создать список из пяти элементов. Заполнить его случайными числами. Вывести этот список. Найти и вывести сумму его элементов. 4) Создать список из десяти элементов. Заполнить его случайными числами. Вывести этот список. Вывести наибольший элемент списка. 5) Создать список из десяти элементов. Заполнить его случайными числами. Заменить все нечетные числа нулями. Вывести исходный и получившийся списки. 6) Заполнить массив из 10 элементов случайными числами в интервале $[-100,100]$ и переставить элементы так, чтобы все положительные элементы стояли в начала массива, а все отрицательные и нули – в конце. Пример: исходный массив: 20 -90 15 -34 10 0; результат: 20 15 10 -90 -34 0. 7) Заполнить массив а случайными числами, отобразить в массив б все числа, меньшие пяти. Вывести оба массива. 	



Задачи:

- 1) Создать список своих любимых фильмов. Вывести список тремя способами: а) в строчку; б) в столбик; в) в строчку через запятую.
- 2) Ввести с клавиатуры список фамилий писателей. Отсортировать список в алфавитном порядке и вывести его.
- 3) Создать список из пяти элементов. Заполнить его случайными числами. Вывести этот список. Найти и вывести сумму его элементов.
- 4) Создать список из десяти элементов. Заполнить его случайными числами. Вывести этот список. Вывести наибольший элемент списка.



Задачи:

- 5) Создать список из десяти элементов. Заполнить его случайными числами. Заменить все нечетные числа нулями. Вывести исходный и получившийся списки.
- 6) Заполнить массив из 10 элементов случайными числами в интервале $[-100, 100]$ и переставить элементы так, чтобы все положительные элементы стояли в начала массива, а все отрицательные и нули – в конце. Пример: исходный массив: 20 -90 15 -34 10 0; результат: 20 15 10 -90 -34 0.
- 7) Заполнить массив **a** случайными числами, отобразить в массив **b** все числа, меньше пяти. Вывести оба массива.

Выставление оценок за работу на уроке.

5. Домашнее задание (3 мин)

УУД:

- Личностные
- Коммуникативные

Написать программы:

- 1) Создать список своих любимых песен. Отсортировать список в алфавитном порядке и вывести его.
- 2) Создать список из десяти элементов. Заполнить его случайными числами. Вывести этот список. Вывести наименьший элемент списка.
- 3) Заполнить массив из 5 элементов случайными числами в интервале $[-100, 100]$. Найти сумму всех отрицательных элементов массива. Если отрицательных элементов в массиве нет, вывести сообщение «отрицательных элементов нет».



Домашнее задание

Написать программы:

- 1) Создать список своих любимых песен. Отсортировать список в алфавитном порядке и вывести его.
- 2) Создать список из десяти элементов. Заполнить его случайными числами. Вывести этот список. Вывести наименьший элемент списка.
- 3) Заполнить массив из 5 элементов случайными числами в интервале $[-100, 100]$. Найти сумму всех отрицательных элементов массива. Если отрицательных элементов в массиве нет, вывести сообщение «отрицательных элементов нет».