

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ЛИЦЕЙ №8

Методическая разработка  
«Изучаем язык программирования Python»  
8 класс

*учитель информатики*

***Битковский Лучия Ивановна***

г. о. Солнечногорск

2023 г.

## Аннотация

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения и его знание позволит учащимся с лёгкостью выучить любой другой язык программирования.

**Тема урока:** Знакомство с языком программирования Python. Ввод. Вывод. Оператор присваивания. Математические операции.

**Цель урока:**

1) **обучающий аспект:** познакомить с языком программирования Python, научить пользоваться функциями ввода и вывода, ввести понятие переменной и оператора присваивания, ознакомить с математическими операциями.

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:** программирование, язык программирования, функция, переменная, оператор присваивания.

**Ход урока (дидактическая структура урока)**

1. Организационный этап (2 мин)	<u>УУД:</u> - Личностные - Коммуникативные
Приветствие, переключка. Объявление, чем мы будем заниматься сегодня на уроке:?	

2. Вводная часть (5 мин)	<u>УУД:</u> - Личностные - Познавательные - Коммуникативные
<p>Сегодня мы начнём большую новую тему. Программирование на языке Python. Ответьте на вопрос, что такое программирование?</p> <p>Программирование – это создание компьютерных программ. Все программы: игры, антивирусы, текстовые редакторы на компьютере были написаны программистами. Мы с вами, конечно, не сможем создать такую большую программу как антивирус или редактор Microsoft Office Word, но маленькие игры сделать попытаемся.</p> <p>Компьютерные программы пишут на специальных языках программирования. Язык программирования – это язык, понятный компьютеру. В настоящее время языков программирования очень много. Кто может назвать какие-либо языки программирования?</p> <div data-bbox="338 1312 1144 1912" style="border: 1px solid black; padding: 20px; margin: 20px auto; width: fit-content;"> <div data-bbox="410 1379 1056 1563" style="background-color: #e1f5fe; padding: 10px; margin-bottom: 10px; text-align: center;"> <p>Программирование – это создание компьютерных программ.</p> </div> <div data-bbox="410 1610 1056 1787" style="background-color: #e1f5fe; padding: 10px; text-align: center;"> <p>Язык программирования – это язык, понятный компьютеру.</p> </div> </div> <p>Самыми востребованными языками сейчас являются Java, JavaScript, C#, C, C++, Python, PHP, SQL, Ruby.</p>	

Самые популярные языки программирования:

- Java
- JavaScript
- C#
- C
- C++
- **Python**
- PHP
- SQL
- Visual Basic .NET
- Ruby

Мы будем изучать программирование на языке Python. Это современный язык, он постоянно развивается, дорабатывается. Этот язык используется в таких проектах, как Google, YouTube, Instagram, Яндекс, Facebook и других. Он легок и прост в использовании.

Python – это интерпретируемый язык программирования с динамической типизацией данных, поддержкой объектно-ориентированного программирования для создания программ самого разнообразного назначения.

Python используется в таких проектах как:



3. Практическая работа на компьютерах (20 мин)

УУД:

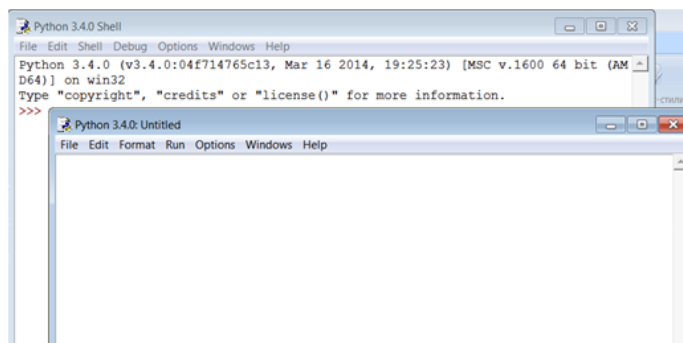
- Личностные
- Познавательные
- Знаково-символические
- Коммуникативные

Программы пишутся в специальных средах программирования. Откроем среду программирования Питона:

Пуск → Python 3.4 → IDLE (Python GUI) → File → New File

Открыть среду программирования Python:

**Пуск => Python 3.4 => IDLE (Python GUI) => File => New File**



Итак, давайте напишем первую программу, которая выведет сообщение «Hello, World!»

Для этого достаточно набрать следующий код:

```
print("Hello, World!")
```

print – функция (команда) вывода.

Запись в тетрадь:

Функция вывода:

```
print("текст")
```

Первая программа:

```
print("Hello, World!!")
```

Запись в тетрадь!

Функция вывода:

```
print("текст")
```

Запуск программы:

Клавиша F5

Или в меню:

Run => Run Module

Второе, что мы изучим – это переменную и оператор присваивания. (Пишем новую программу).

```
message = 'Hello, World!'  
print(message)
```

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы. В программе мы создали переменную с именем `message`, присвоили ей значение-строку `'Hello, World!'`, и, следовательно, эта переменная приняла строковый тип.

Знак `<=>` - это оператор присваивания.

### Переменная и оператор присваивания

```
message = 'Hello, World!'  
print(message)
```

`message` – переменная  
`=` – оператор присваивания

**Переменная** – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

Имена переменных могут состоять из:

- Латинские буквы (строчные и заглавные буквы различаются!)
- Русские буквы (не рекомендуется)
- Цифры (имя не может начинаться с цифры и состоять только из цифр)
- Знак подчеркивания `_`

Нельзя использовать в именах переменных:

- Пробелы
- Знаки `+, -, >, <, =, (), !` и др.
- Ключевые слова языка Python

## Имена переменных

Имена переменных могут состоять из:

- Латинские буквы (строчные и заглавные буквы различаются!)
- Русские буквы (не рекомендуется)
- Цифры (имя не может начинаться с цифры и состоять только из цифр)
- Знак подчеркивания `_`

Нельзя использовать в именах переменных:

- Пробелы
- Знаки `+, -, >, <, =, (), !` и др.
- Ключевые слова языка Python

Нельзя использовать как имена переменных ключевые слова языка Python.

Ключевые слова – это слова языка программирования, которые имеют специальное, раз и навсегда закрепленное за ними значение. К ним относятся имена функций, операторов и другое. Например, функция «print» - ключевое слово, которое нельзя использовать в качестве имени переменной. Позже мы изучим и другие функции.

Нельзя использовать как имена переменных  
ключевые слова языка Python:

False	<b>class</b>	<b>finally</b>	<b>is</b>	<b>return</b>
None	<b>continue</b>	<b>for</b>	<b>lambda</b>	<b>try</b>
True	<b>def</b>	<b>from</b>	<b>nonlocal</b>	<b>while</b>
<b>and</b>	<b>del</b>	<b>global</b>	<b>not</b>	<b>with</b>
<b>as</b>	<b>elif</b>	<b>if</b>	<b>or</b>	<b>yield</b>
<b>assert</b>	<b>else</b>	<b>import</b>	<b>pass</b>	<b>print</b>
<b>break</b>	<b>except</b>	<b>in</b>	<b>raise</b>	

Перейдём к знакомству с математическими операциями. (Создаём новый файл).

Создадим две целочисленные переменные и попросим компьютер их сложить.

```
a = 78001457
```

```
b = 2546880
```

```
c = a + b
```

```
print(c)
```

## Математические операции

```
a = 78001457
b = 2546880
c = a + b
print(c)
```

```
a = 78
b = 25
c = (a-b) * (a+b) / 27
print(c)
```

Переменной `c` можно присвоить целое математическое выражение:

```
c = (a-b) * (a+b) / 27
```

Другие математические операции:

<code>x + y</code>	Сложение
<code>x - y</code>	Вычитание
<code>x * y</code>	Умножение
<code>x / y</code>	Деление
<code>x // y</code>	Получение целой части от деления
<code>x % y</code>	Остаток от деления
<code>-x</code>	Смена знака числа
<code>abs(x)</code>	Модуль числа
<code>divmod(x, y)</code>	Пара ( <code>x // y</code> , <code>x % y</code> )
<code>x ** y</code>	Возведение в степень



### Другие математические операции:

$x + y$	Сложение
$x - y$	Вычитание
$x * y$	Умножение
$x / y$	Деление
$x // y$	Получение целой части от деления
$x \% y$	Остаток от деления
$-x$	Смена знака числа
$abs(x)$	Модуль числа
$divmod(x, y)$	Пара ( $x // y, x \% y$ )
$x ** y$	Возведение в степень

Функция ввода.

Для того чтобы присвоить переменной значение, введённое с клавиатуры, используется функция `input()`. Напишем и запустим следующую программу:

```
name = input("Введите своё имя: ")
print("Привет, ", name)
```

Измените программу так, чтобы она выводила в конце восклицательный знак.

Запись в тетрадь:

Ввод строки:

```
s = input("Введите строку: ")
```

"Введите строку: " – обращение к пользователю (не обязательно, но очень желательно)

### Функция ввода

```
name = input("Введите своё имя: ")
print("Привет, ", name)
```



Измените программу так, чтобы она выводила в конце восклицательный знак.

Запись в тетрадь!

Ввод строки:

```
s = input("Введите строку: ")
```

"Введите строку: " – обращение к пользователю (не обязательно, но очень желательно)

По умолчанию все введённые данные интерпретатор Питона понимает, как строки, поэтому, если мы хотим получить число, то строку придётся преобразовать в число.

Преобразование к целочисленному типу и ввод целого числа:

Запись в тетрадь:

Ввод целого числа:

```
n = int(input("Введите число: "))
```

То есть на функцию ввода мы навешиваем ещё одну функцию преобразования в целое число.

Запись в тетрадь:

Функция преобразования к целочисленному типу:

```
n = int(s)
```

Функция преобразования к строковому типу:

```
s = str(n)
```

По умолчанию все введённые данные интерпретатор Питона понимает, как строки. Поэтому, если мы хотим получить число, то строку придётся преобразовать в число.

Запись в тетрадь!

Ввод целого числа:

```
n = int(input("Введите число: "))
```

Функция преобразования к целочисленному типу:

```
n = int(s)
```

Функция преобразования к строковому типу:

```
s = str(n)
```

Задание. Напишите программу, которая получает на вход два числа и выводит их сумму.

```
a = input("Введите число a: ")
```

```
b = input("Введите число b: ")
```

```
sum = a+b
```

```
print("a+b= ", sum)
```

Почему программа работает не правильно? (Потому что все введённые данные компьютером понимаются как строки) Что исправить в программе, чтобы она работала правильно?

Правильный вариант:

```
a = int(input("Введите число a: "))
b = int(input("Введите число b: "))
sum = a+b
print("a+b= ", sum)
```



Задание. Напишите программу, которая получает на вход два числа и выводит их сумму:

```
a = input("Введите число a: ")
b = input("Введите число b: ")
sum = a+b
print("a+b=", sum)
```

Почему программа работает неправильно?  
Что исправить в программе, чтобы она работала правильно?

Задача. В каждой строке определить тип и значение переменной:

```
a = 5
n = input()          #пользователь вводит цифру 8
c = int(n)
d = a*c
d = d-a
s = "Рамамбахарумамбуру"
d = n+a
m = n+s
```

Запись в тетрадь:

```
# Комментарии к программе, компьютер их не читает
```

Задача. В каждой строке определить тип и значение переменной:

```
a = 5
n = input()    #человек вводит цифру 8
c = int(n)
d = a*c
d = d-a
s = "Рамамбахарумамбур"
d = n+a
m = n+s
```

Запись в тетрадь!

```
# Комментарии к программе, компьютер
# их не читает
```

Генератор случайных чисел

Запись в тетрадь:

Функция генерации случайного целого числа из отрезка [x,y]:

```
import random
a = random.randint(x,y)
```

### Генератор случайных чисел

Запись в тетрадь!

Функция генерации случайного целого числа из отрезка [x,y]:

```
import random
a = random.randint(x,y)
```

4. Самостоятельная работа  
на компьютерах (13 мин)

УУД:

- Личностные
- Регулятивные
- Познавательные
- Знаково-символические
- Коммуникативные

Учащиеся стараются самостоятельно решить задачи:

- 1) Вывести на экран три введенных с клавиатуры числа в порядке, обратном их вводу.
- 2) Ввести с клавиатуры два числа и вывести целую часть от деления первого на второе.
- 3) Ввести с клавиатуры основание и высоту треугольника и вывести площадь треугольника.
- 4) Ввести с клавиатуры два катета треугольника и вывести гипотенузу. (Квадратный корень – это возведение в степень  $(1/2)$  )
- 5) Сгенерировать случайное двузначное число, вывести на экран это число, а также сумму и произведение его цифр.

Для получения цифр используйте целочисленное деление на 10 и взятие остатка от деления на 10. Пример для числа 47:

$$47//10=4$$

$$47\%10=7$$



### Задания

- 1) Вывести на экран три введенных с клавиатуры числа в порядке, обратном их вводу.
- 2) Ввести с клавиатуры два числа и вывести целую часть от деления первого на второе.
- 3) Ввести с клавиатуры основание и высоту треугольника и вывести площадь треугольника.
- 4) Ввести с клавиатуры два катета и вывести гипотенузу. (Квадратный корень – это возведение в степень  $(1/2)$  )
- 5) Сгенерировать случайное двузначное число, вывести на экран это число, а также сумму и произведение его цифр.

Для получения цифр используйте целочисленное деление на 10 и взятие остатка от деления на 10.

Пример для числа 47:

$$47//10=4 \quad 47\%10=7$$

Выставление оценок за работу на уроке.

5. Домашнее задание (5 мин)	<u>УУД:</u> - Личностные - Коммуникативные
Домашнее задание: Написать программы: <ol style="list-style-type: none"><li>1) Ввести основания и высоту трапеции и вывести площадь трапеции.</li><li>2) Получить случайное трехзначное число, вывести это число и сумму его отдельных цифр.</li></ol>	

3) Программа, которая рассчитывает возраст человека в часах.