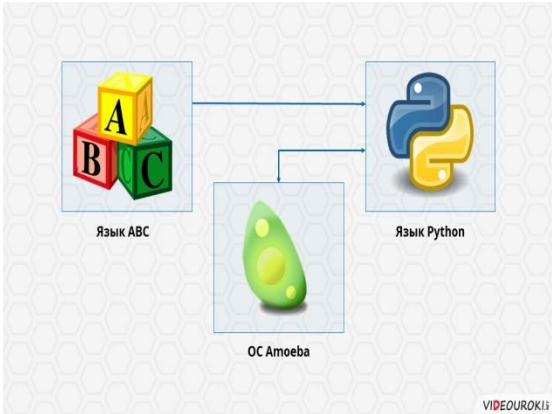


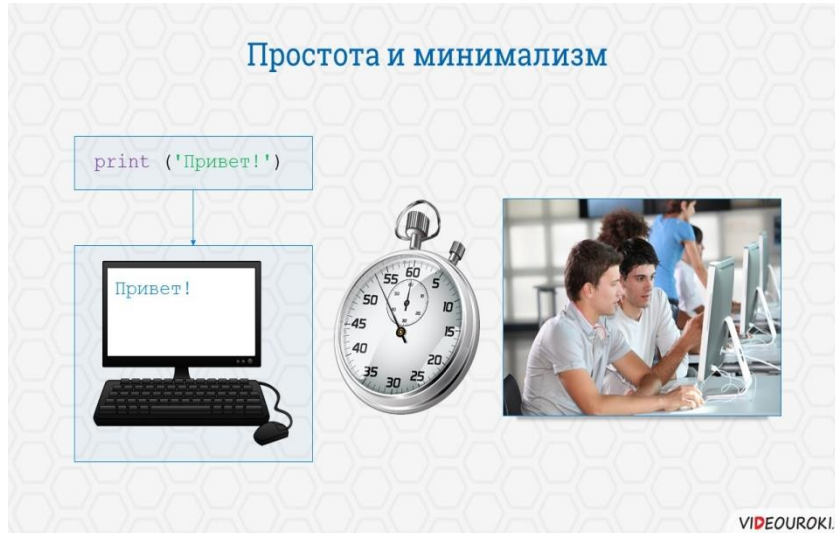
Школа:		
Дата:	ФИО учителя:	
Класс:	Участвовали:	Не участвовали:
Тема урока: Инструктаж по ТБ. История языков программирования. Компиляция и интерпретация		
Цели обучения, которые достигаются на данном уроке	знать и использовать компоненты интегрированной среды разработки программ Python	
Цели урока	Все учащиеся смогут: создавать проект, сохранять его, запускать компиляцию <ul style="list-style-type: none"> • Большинство учащихся смогут: находить и исправлять ошибки. <ul style="list-style-type: none"> • Некоторые учащиеся смогут: решать более сложные задачи <ul style="list-style-type: none"> • 	
Критерии оценивания	1. Учащиеся по рисунку называют элементы интерфейса. 2. создают и сохраняют проект, запускают компиляцию, запускают созданный 3. Определяют в чем ошибки и исправляют	
Воспитание ценностей	Воспитание эмоционально-положительной направленности на практическую деятельность, интереса к информатике, личной ответственности за результаты своей работы	
Предварительные знания	Учащиеся уже разрабатывали проекты в IDE и на этом уроке повторяют ее возможности. Активизация уже имеющихся знаний осуществляется через групповую работу	
Межпредметные связи	с робототехникой	
Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы
Начало урока <u>5</u> мин	<p>В современном мире престижно быть частью IT-индустрии. Всё меньше школьников хотят быть пожарными, летчиками и космонавтами — большинство мечтает стать вторым Цукербергом, Дуровым или Маском, развивать виртуальную реальность, создавать невероятные цифровые образы и продукты.</p> <p>Знание Python, одного из популярнейших языков программирования, открывает путь в ведущие IT-компании мира: Google, Яндекс, Mail.Ru, Youtube, Instagram.</p>  <p>Итак, язык программирования Python был задуман нидерландским программистом Гвидо ван Россумом в 1980-х годах. Ван Россум приступил к его созданию в декабре 1989 года в Центре математики и информатики в Нидерландах. Язык Python был задуман как потомок языка программирования ABC. В отличие от своего предка, Python должен был получить возможность обрабатывать исключения и взаимодействовать с открытой операционной системой Amoeba, разработанной в 1983 году в свободном университете Амстердама. В 1991 году ван Россум опубликовал код своего языка программирования. Он начал набирать популярность в интернете. В 1994 году, благодаря росту числа пользователей, была создана группа comp.lang.python – основной форум языка Python.</p> <p>Гвидо ван Россум</p> <p>Язык Python был назван так не в честь змеи. На момент создания языка, Гвидо ван Россум был увлечён просмотром комедийного</p>	Презентация, учебник Слайд 1-3 Слайд 4

сериала «Воздушный цирк Монти Пайтона». В честь главного героя этого сериала, Монти Пайтона, язык и получил своё название.

Сейчас Python – это активно развивающийся высокоуровневый многоцелевой язык программирования. Он поддерживает несколько, наиболее популярных сейчас, парадигм программирования, таких как структурное, объектно-ориентированное, функциональное программирование и другие... Популярности языка способствует то, что он соответствует стандартам Американского национального института стандартов и Международной организации по стандартизации. Язык Python входит в пятерку наиболее популярных языков программирования по итогам 2016 года. Эталонная реализация Pythona – это интерпретатор CPython, он совместим с большинством активно используемых на сегодняшний день платформ. Этот интерпретатор распространяется по свободной лицензии Python Software Foundation License.

Гвидо ван Россум и по сей день является активным участником разработки языка Python. Среди разработчиков он известен как «Великодушный пожизненный диктатор проекта». Это означает, что он продолжает наблюдать за разработкой языка и принимает окончательные решения по его развитию, когда это необходимо.

Почему же язык Python получил такую популярность? Начнём с того, что Python – это многоцелевой язык. Его можно одинаково хорошо использовать для разработки любых программ и их тестирования. Так, например, компания Google широко использует язык Python для своей поисковой системы. Большая часть популярного видеохостинга YouTube была написана на языке Python. Также язык Python применяется в анимационной графике, научных вычислениях и тестировании аппаратного обеспечения.



Python создавался так, чтобы программы на нём легко читались и также легко разрабатывались. Для того, чтобы написать программу на языке Python, нужно не так много кода. Например, следующая программа, состоящая из единственной строки, выводит на экран компьютера текстовое сообщение «Привет!».

```
print ('Привет!')
```

Так как для написания программ нужно меньше кода – они разрабатываются быстрее и достаточно легко читаются. Язык Python получился настолько простым в использовании, что его начали использовать для обучения студентов программированию, хотя изначально такой цели не ставилось.

Большая часть программ на языке Python выполняется без изменений, независимо от того, на какой платформе они используются. То есть, в большинстве случаев программа, написанная на языке Python, будет одинаково правильно работать на различных операционных системах, например, в Windows, Linux и Mac OS. Программы на языке Python легко интегрируются с компонентами, написанными на других языках программирования, таких как C, C++ и Java.

В стандартных библиотеках языка Python изначально реализовано множество возможностей, которые могут быть полезны при написании самых разных программ. Кроме расширений, язык Python также поддерживает библиотеки, написанные другими разработчиками, что ещё больше увеличивает доступную функциональность.

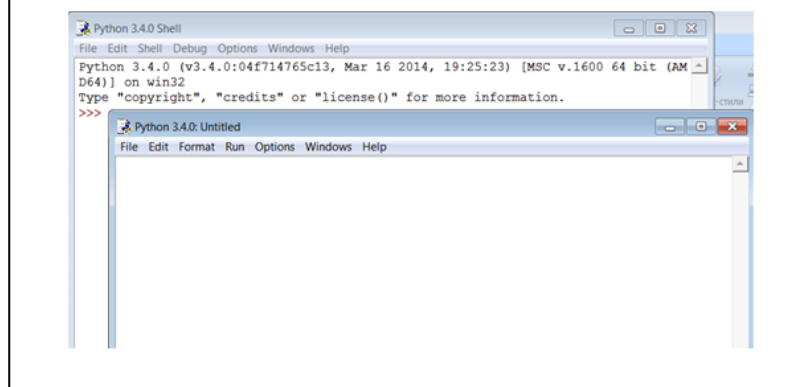
Середина урока
30 мин

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

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:

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

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 = "Рамамбахарумамбуру"
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)
```

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:

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

Оценивание Колесо баланса

Конец урока
—5_ мин

Домашнее задание:

Установить на компьютер среду программирования IDLE. (Скачивание из интернета по инструкции или сохранение установочного файла на съёмный носитель).

Написать программы:

- 1) Ввести основания и высоту трапеции и вывести площадь трапеции.
- 2) Получить случайное трехзначное число, вывести это число и сумму его отдельных цифр.
- 3) Программа, которая рассчитывает возраст человека в часах.



Домашнее задание:

Установить на компьютер среду программирования IDLE Python.

Написать программы:

- 1) Ввести основания и высоту трапеции и вывести площадь трапеции.
- 2) Получить случайное трехзначное число, вывести это число и сумму его отдельных цифр.
- 3) Программа, которая рассчитывает возраст человека в часах.

	<p>Установить на компьютер среду программирования IDLE Python.</p> <p>Написать программы:</p> <ol style="list-style-type: none">1) Ввести основания и высоту трапеции и вывести площадь трапеции.2) Получить случайное трехзначное число, вывести это число и сумму его отдельных цифр.3) Программа, которая рассчитывает возраст человека в часах.		
<p>Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?</p>	<p>Оценивание – как Вы планируете проверить уровень усвоения материала учащимися?</p>	<p>Охрана здоровья и соблюдение техники безопасности</p>	
<p>На этапе закрепления менее способным учащимся предлагаются различные подмости.</p>	<p>Формативное оценивание ранее полученных знаний. На этапе выполнения практической работы учащиеся оцениваются по критериям.</p>	<p>Соблюдение техники безопасности при передвижении учащихся по классу во время групповой работы «Карусель», ТБ при работе за компьютером.</p>	