

School:		
Date:	Teacher's name:	
Grade:	Number present:	absent:
Topic of the lesson: Introduction to the Python programming language.		
Learning objective(s) that this lesson is contributing to	know and use the components of the Python integrated development environment.	
Lesson objectives	create a project, save it, start compilation find and fix errors. solve more complex problems	
Assessment Criteria	1. Students in the picture name the interface elements. 2. create and save the project, start compilation, start the created 3. Determine why errors and correct	
Value links	Education of an emotionally positive orientation to practical activities, interest in computer science, personal responsibility for the results of their work	
Previous learning	Students have already developed projects in the IDE and in this lesson will repeat its capabilities. The activation of existing knowledge is carried out through group work	
Cross curricular links	with robotics	
Time	Planned activities	Resources
Beginning 5 min	<p>Today we will start a big new topic. Python programming. Answer the question, what is programming?</p> <p>Programming is the creation of computer programs. All programs: games, antiviruses, text editors on the computer were written by programmers. Of course, we will not be able to create such a large program as an antivirus or Microsoft Office Word editor, but we will try to make small games.</p> <p>Computer programs are written in special programming languages. A programming language is a language that a computer understands. Currently, there are a lot of programming languages. Who can name any programming languages?</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center; color: blue;">Программирование – это создание компьютерных программ.</p> <p style="text-align: center; color: blue;">Язык программирования – это язык, понятный компьютеру.</p> </div> <p style="text-align: center;">The most popular languages are now Java, JavaScript, C#, C, C++, Python, PHP, SQL, Ruby.</p>	<p>Presentation, tutorial</p> <p>Slide 1-3</p> <p>Slide 4</p>

Самые популярные языки программирования:

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

We will learn Python programming. This is a modern language, it is constantly evolving, being finalized. This language is used in projects such as Google, YouTube, Instagram, Yandex, Facebook and others. It is lightweight and easy to use.

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:



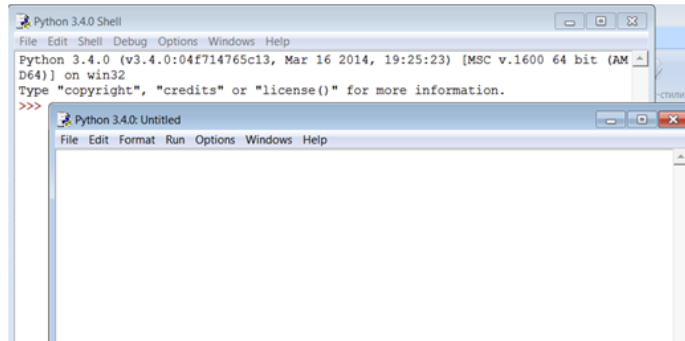
**Middle
30 Min**

Programs are written in special programming environments. Let's open the Python programming environment:

Start -> Python 3.4 -> IDLE (Python GUI) -> File -> New File

Открыть среду программирования Python:

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



So, let's write the first program that displays the message “Hello, World!”

To do this, just type the following code:

print(“Hello, World!”)

print – функция (команда) вывода.

Запись в тетрадь:

Функция вывода:

print(“text”)

Первая программа:

```
print (“Hello, World!!!”)
```

Запись в тетрадь!

Функция вывода:

```
print (“текст”)
```

Запуск программы:

Клавиша F5

Или в меню:

Run => Run Module

The second thing we will learn is the variable and the assignment operator. (We are writing a new program).

message = ‘Hello, World!’

print(message)

A variable is a quantity that has a name, type, and value. The value of the variable can be changed while the program is running. In the program, we

created a variable called message, assigned a string value to it **'Hello, World!'**, , and therefore this variable has adopted a string type.

The “=” sign is an assignment operator.

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.

Variable names may consist of:

- Latin letters (lowercase and uppercase letters are different!)
- Russian letters (not recommended)
- Numbers (a name cannot begin with a number and consist only of numbers)
- The underscore _

Cannot be used in variable names:

- Gaps
- Signs +, -, >, <, =, (), ! and etc.
- Python keywords

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**

You cannot use Python language keywords as variable names.

Keywords are words in a programming language that have a special meaning once and for all assigned to them. These include the names of functions, operators, and more. For example, the print function is a keyword that cannot be used as a variable name. Later we will explore other functions.

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	

Let's move on to acquaintance with mathematical operations. (Create a new file).

Create two integer variables and ask the computer to add them.

```
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)
```

Variable c can be assigned an integer mathematical expression:

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

Other mathematical operations:

$x + y$	Addition
$x - y$	Subtraction
$x * y$	Multiplication
x / y	Division
$x // y$	Getting the integer part from division
$x \% y$	Remainder of the division
$-x$	Change the sign of a number
$abs(x)$	The absolute value of a number
$divmod(x, y)$	Couple ($x // y, x \% y$)
$x ** y$	Exponentiation

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

Input function.

In order to assign a variable a value entered from the keyboard, the **input()** function is used. We will write and run the following program:

```
name = input("Enter your name: ")  
print("Hello, ", name)
```

Change the program so that it displays an exclamation point at the end.

Record in a notebook:

Line input:

```
s = input("Enter the string: ")
```

“Enter the string: “ - appeal to the user (not necessary, but very desirable)

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)

By default, the Python interpreter understands all the entered data as strings, so if we want to get a number, then the string will have to be converted to a number.

Converting to an integer type and entering an integer:

Record in a notebook:

Entering an integer:

```
n = int(input("Insert the number: "))
```

That is, we attach another function of conversion to an integer on the input function.

Record in a notebook:

Conversion function to integer type:

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

Convert to string type function:

```
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)
```

The task. Write a program that receives two numbers as an input and displays their sum.

```
a = input("Insert the number a: ")
```

```
b = input("Insert the number b: ")
```

```
sum = a+b
```

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

Why is the program not working correctly? (Because all entered data by a computer is understood as strings) What can I fix in a program so that it works correctly?

The correct option:

```
a = int(input("Insert the number a: "))
```

```
b = int(input("Insert the number 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?

Task. In each line, determine the type and value of the variable:

a = 5

n = input() # user enters a number 8

c = int(n)

d = a*c

d = d-a

s = "Рамамбахарумамбуру"

d = n+a

m = n+s

Record in a notebook:

Comments on the program, the computer does not read them

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
```

Random number generator

Record in a notebook:

The function of generating a random integer from the segment [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)
```

Students try to solve problems on their own:

1) Display the three numbers entered from the keyboard in the reverse order of their entry.

2) Enter two numbers from the keyboard and display the integer part from dividing the first by the second.

3) Enter the base and height of the triangle from the keyboard and display the area of the triangle.

4) Enter the two legs of the triangle from the keyboard and bring hypotenuse. (The square root is exponentiation (1/2))

5) Generate a random two-digit number, display this number, as well as the sum and product of its digits.

To get the numbers, use an integer division by 10 and take the remainder of the division by 10. Example for the number 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:

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

Grading

Balance wheel

**End
5 min**

Homework:

Install IDLE programming environment on the computer. (Downloading from the Internet according to the instructions or saving the installation file to removable media).

Write a program:

- 1) Enter the base and height of the trapezoid and display the area of the trapezoid.
- 2) Get a random three-digit number, print this number and the sum of its individual digits.
- 3) A program that calculates the age of a person in hours.



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.

**Differentiation – how do you plan to give more support?
How do you plan to challenge the more able learners?**

Assessment – how are you planning to check learners' learning?

Health and Safety

At the stage of consolidation, less capable students are offered various stages.

Formative assessment of previously acquired knowledge.

Compliance with safety precautions when moving students

	At the stage of practical work, students are evaluated according to the criteria	around the class during group work "Carousel", TB when working at the computer.
--	--	---