

Программирование на Python

Урок №1

План урока:

1. Краткий рассказ о том, что такое язык программирования и зачем для него требуется IDE
2. Запуск Python
3. Что такое “строка кода”
4. Команда вывода на экран
5. Первая программа “Hello world”
6. Что такое “переменная”
7. Программа “Поздороваться с пользователем по имени”

Результат:

Скриншот программы “Hello world”

```
print("Hello world")
```

Скриншот программы “Поздороваться с пользователем по имени”

```
name = input()  
print("Hello", name)
```

Приветствие:

Здравствуйте! Сегодня мы познакомимся с языком программирования Python. На сегодняшнем занятии мы научимся вводу информации и выводу ее на экран, а так же напишем свою первую программу.

Запуск Python:

Слова русского языка можно прочесть вне зависимости от того, какую тетрадку мы выбрали, так и программы на Python могут быть записаны в разных “тетрадках”, которые называются IDE. IDE расшифровывается как Integrated Development Environment, а переводится как Интегрированная Среда Разработки. Нашей “тетрадкой” на протяжении курса будет стандартная среда разработки от создателей Python. Давайте запустим ее. Сначала надо нажать на иконку IDE



IDLE
(Python 3.7
32-bit)

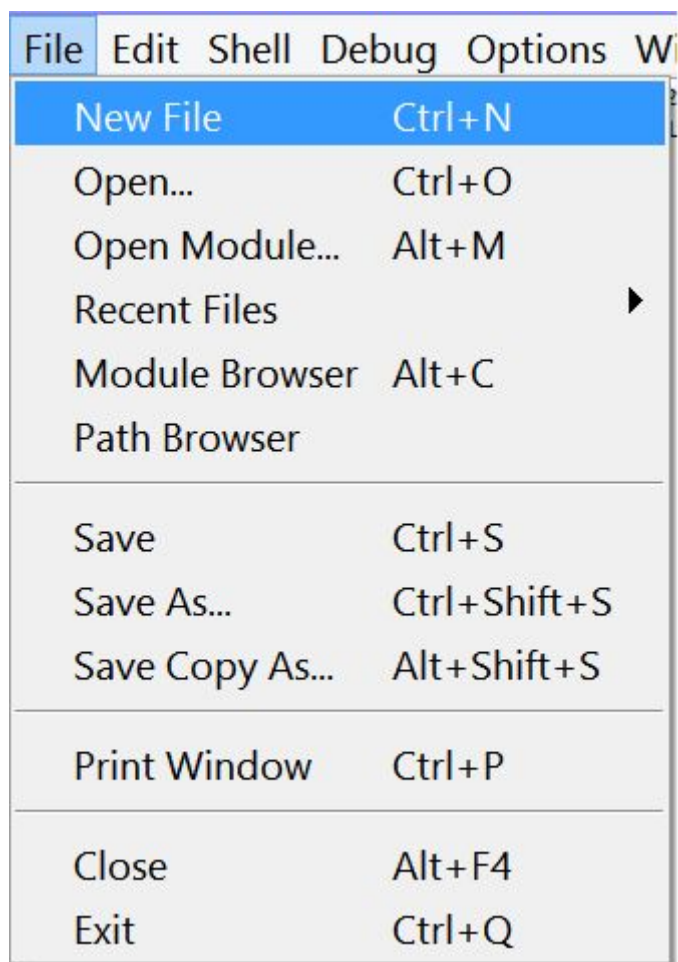
Кстати, как вы можете заметить, называется она IDLE, что созвучно с IDE, но расшифровывается как Integrated Development and Learning Environment и переводится как Интегрированная Среда Разработки и Обучения. Также, IDLE можно прочесть и как обычное слово Idle(читается как айдл), что переводится как “Ленивый”. Это небольшая шутка от разработчиков.

Теперь вы видите перед собой открытую IDE.

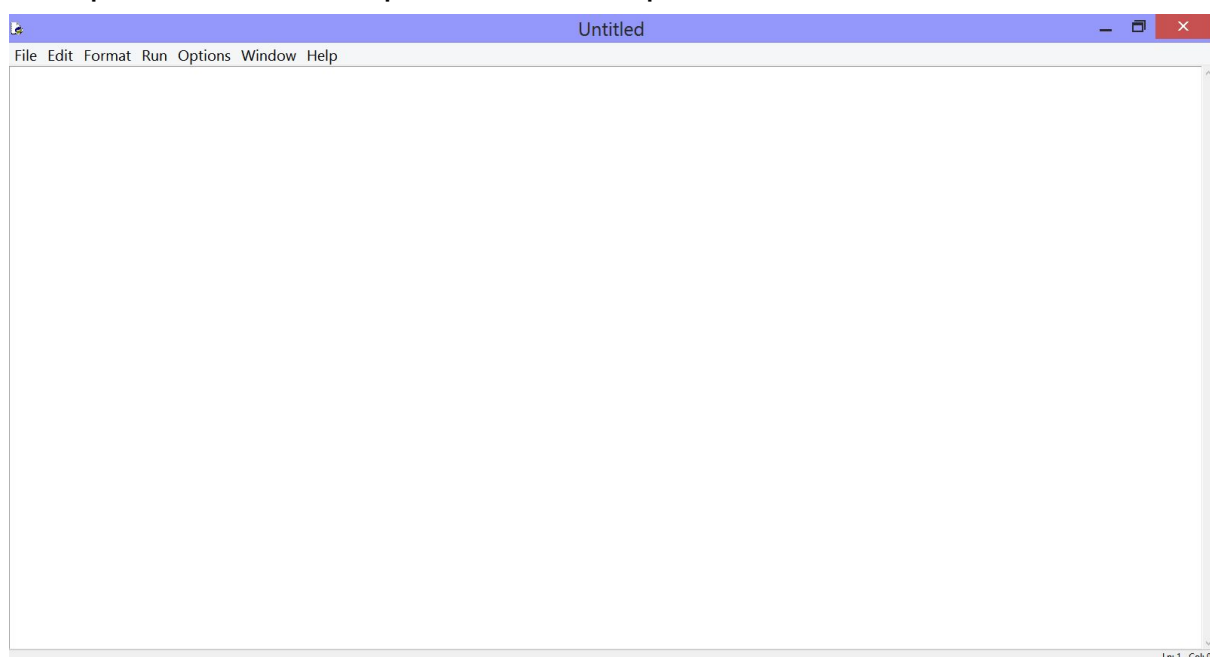


Версия Python, которую видно в заголовке, может быть другой, в зависимости от той, которую вы установили.

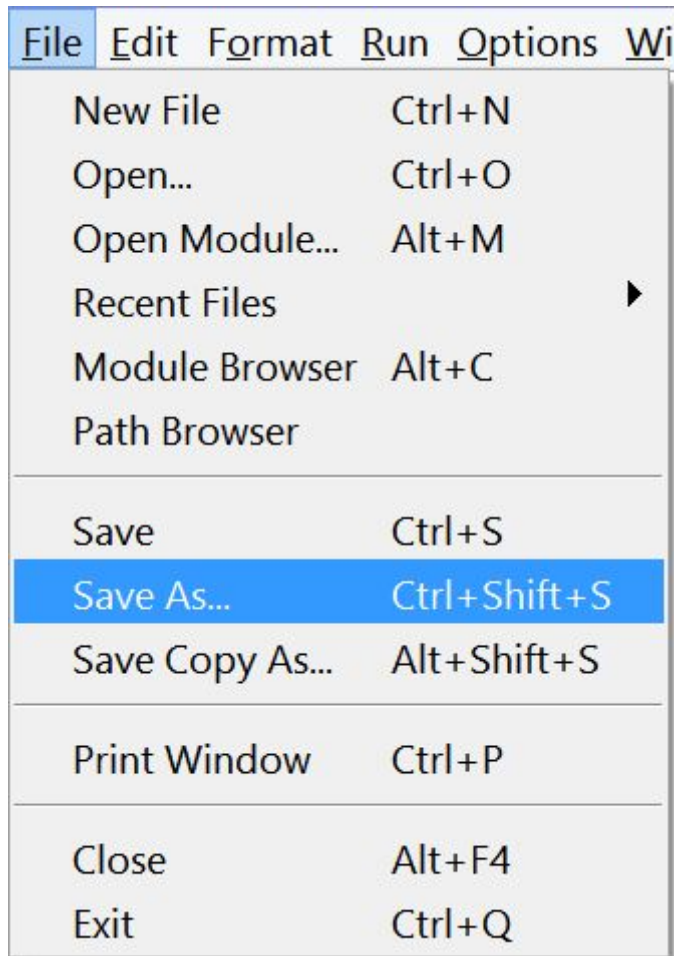
Давайте создадим свой первый проект. Для этого надо щелкнуть по вкладке **File** и выбрать **New file**:



Теперь вы видите открытый новый проект:



Теперь лучше всего будет сразу сохранить проект в нужную папку, нажав на **File** и потом на **Save as...**:



Давайте назовем наш проект “Hello world” и каждое отдельное задание будем делать в отдельном проекте, который будем сохранять, чтобы в конце мы смогли оглядеть проделанный путь.

Что такое “строка кода”:

Так же, как в русском языке мы говорим предложениями, в языках программирования мы разговариваем с компьютером строчками кода. Одна строчка — одна команда. И так же, как в русском языке, в Python слишком длинные “предложения” будут только запутывать.

Команда вывода на экран:

Давайте введем команду

```
print("Hello world")
```

и посмотрим что получилось.

A screenshot of a Python 3.7.1 Shell window. The title bar is blue and says "Python 3.7.1 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following: "Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", ">>>", "===== RESTART: C:\Users\Илья\Desktop\Hello world.py =====", "Hello world", and ">>>". The text "Hello world" is highlighted in blue. The status bar at the bottom right says "Ln: 6 Col: 4".

Как вы можете видеть, открылось новое окно и текст, подсвеченный синим цветом и есть то, что мы ввели внутри команды **print** и заключили в кавычки. Команда **print** позволяет выводить любой текст, который мы заключили в кавычки на экран.

Переменная:

Переменную можно представлять как ящик, на котором написано имя ящика, которое пишется для того, чтобы не спутать его с другими. Внутри ящика мы складываем информацию, которая нам будет нужна в дальнейшем. К примеру, мы захотим сделать ящик, в котором будем хранить свое имя для того, чтобы компьютер мог здороваться с нами по имени. Какое имя дать ящику? Ну, давайте дадим ящику “говорящее” имя, чтобы мы всегда могли понять что же лежит внутри него. Давайте дадим название на английском

языке, так как это общепринятый язык для записи программ. Назовем ящик “**name**”, что в переводе значит “имя”. После того, как мы дали имя ящику, его уже нельзя сменить. Давайте положим в наш ящик, к примеру, имя Илья. Мы напишем наше имя на листке и положим в ящик. В ящике в данный момент лежит имя Илья и любой может зайти и посмотреть это, а значит компьютер знает как с нами здороваться и будет здороваться с нами по имени Илья. Теперь мы захотим поменять имя внутри ящика. Как мы это сделаем? Вытащим листок, который там уже был, и положим новый. К примеру, с именем Настя. Теперь в ящике лежит имя Настя и компьютер будет здороваться с нами по имени Настя. Имя ящик не изменилось, а информация в нем может меняться постоянно. Давайте попробуем создать переменную **name** и внести в ее данные своё имя:

```
name = "Ilya"
```

Теперь давайте с помощью команды

```
name = "Ilya"  
print(name)
```

выведем данные, которые хранятся в переменной на экран. Как видите, переменная, в отличие от текста, пишется внутри **print** без кавычек. Почему это так, мы разберем на следующем занятии. Если мы запустим программу, то увидим, что в выводе на экран написано наше имя. Осталось сделать так, чтобы компьютер добавлял “**Hello**” перед ним и наша задача будет выполнена. Мы можем попробовать написать **print(“Hello”)** и **print(name)** на разных строках, но тогда компьютер будет выводить эти слова также на разных строках.

```
name = "Ilya"  
print("Hello")  
print(name)
```

```
Hello  
Ilya  
>>>
```

Так что нужен другой способ. Давайте напишем “**Hello**” и **name** внутри одного **print**, но через запятую.

```
name = "Ilya"
print("Hello", name)
```

И теперь компьютер здоровается с нами по имени!

```
Hello Ilya
>>>
```

Строки и переменные, разделенные запятыми внутри одного **print**, выводятся подряд через пробел.

Если мы захотим повторить опыт с ящиком и попробовать положить в переменную **name** сначала Илью, а потом Настю:

```
name = "Ilya"
name = "Nastya"
print("Hello", name)
```

то результатом будет приветствие компьютером Насти:

```
Hello Nastya
>>>
```

Если выводить на экран **name** до того, как мы изменим значение внутри нашего ящика-переменной, то компьютер использует то значение, которое имеет в данный момент:

```
name = "Ilya"
print("Hello", name)
name = "Nastya"
print("Hello", name)

Hello Ilya
Hello Nastya
>>>
```

Задания:

1. Написать программу, которая будет выводить "Hello Ilya. How are you?", где вместо Ilya будет стоять имя пользователя из переменной **name**.
2. Написать программу, которая будет не только здороваться по имени с пользователем, но и будет выводить его возраст на экран, где возраст будет лежать в переменной **age**.
3. Написать программу, которая будет выводить анкету пользователя, где все данные будут записаны в соответствующие переменные. Вся анкета должна быть написана в одной строчке.

Усложненные задания:

1. Написать программу, которая будет выводить анкету пользователя, аналогичной той, что была описана в задании 3, но каждой раздел данных должен идти с новой строки.
2. Написать программу, которая будет выводить сначала данные в анкете, а потом их пояснять.

Тайминг

Тема	Время с начала занятия, мин
Вводная часть	5
Краткий рассказ об интерфейсе программы	10
Что такое строка кода	20
Команда вывода на экран	35
Переменная	45
Сохранение проекта	50
Первая программа	60
Задания	70
Контрольные вопросы	80
Работа в тетради	90

Контрольные вопросы:

1. Что такое строка кода?
2. Что такое переменная?
3. С помощью какой команды мы можем вывести информация на экран?
4. Как с помощью команды вывода на экран выводить текст и переменную одновременно и в одну строчку?
5. Как выводить текст в несколько строк?

Ответы:

1. Это одно “предложение” языка программирования, оно записывается в одну строчку и следующие строчки уже пишутся с новой строки.
2. Это именной контейнер, в котором мы храним информацию. Информацию в нем можно свободно менять.
3. `print()`
4. Писать текст и переменную разделяя запятой внутри одного `print()`
5. Использовать несколько команд `print()`