

Кузнецов Александр Евгеньевич  
г. Барнаул  
[kuznetcov\\_ae@uni-altai.ru](mailto:kuznetcov_ae@uni-altai.ru)  
Зыбцева Ксения Александровна  
г. Барнаул  
[susazybka@mail.ru](mailto:susazybka@mail.ru)

## PYTHON КАК БАЗОВЫЙ ЯЗЫК ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

На сегодняшний день очевидным является факт необходимости обучения школьников и студентов программированию. Однако до сих пор нет единого мнения о том, какому языку программирования обучать, какой язык лучше подходит в качестве первого – обычно предлагаются Pascal, C/C++ или Basic. Каждый из них имеет свои преимущества и недостатки с точки зрения методики обучения, тем не менее, в части С экзамена ЕГЭ по информатике школьникам предлагают использовать именно эти языки. С некоторых пор этот список пополнился языком Python.

Python (по-русски произносится «Пайтон» или «Питон») – это высокоуровневый язык программирования общего назначения, поддерживающий большое количество парадигм программирования [2]. По данным одного из самых известных рейтингов ТЮВЕ, Python с 2008 года прочно удерживается в восьмерке наиболее популярных языков программирования [5]. В 2014 году Python стал самым популярным языком, используемым для обучения основам программирования в университетах США [1]. Его используют такие известные компании, как Google, Яндекс, Европейская организация по ядерным исследованиям (CERN), Национальное управление по воздухоплаванию и исследованию космического пространства США (NASA) и др.

Язык является интерпретируемым, свободно распространяемые реализации интерпретатора есть для большинства современных платформ (Windows, Linux, Mac OS X, FreeBSD, Android, iOS и др.), тем самым снимаются вопросы, связанные с лицензированием.

Рассмотрим некоторые особенности языка Python.

### **Отступы – часть синтаксиса**

В отличие от большинства других языков программирования, отступы в языке Python используются для обозначения блочной структуры. Это позволяет писать более компактный и грамотный код, кроме того, дисциплинирует программистов – правильно работающий код практически невозможно написать «нечитабельным», программист просто вынужден использовать отступы в своей программе, что делает её более понятной.

Данная особенность позволяет также избежать ряда ошибок, характерных для других языков программирования. В частности, на Паскале можно написать следующий код:

```
i:=0;
while i < 10 do
    writeln(i);
    i = i + 1;
```

В данном случае произойдет заикливание, так как последний оператор не входит в тело цикла. Для Python такая ситуация практически не возможна – аналогичный код будет работать нормально, если операторы, составляющие тело цикла, записаны с равными отступами:

```
i = 0
while i < 10:
    print(i)
    i = i + 1
```

Использование отступов делает ненужными операторные скобки, ограничивающие блок (фигурные скобки в С-подобных языках, пара begin-end в Паскале).

### **Динамическая типизация**

Python использует динамическую типизацию переменных. С одной стороны это означает, что переменные не нужно объявлять – тип переменной определяется автоматически, когда ей присваивается значение. С другой стороны – каждая переменная может даже в одной программе указывать на значения разных типов.

Это связано с тем, что в языке Python нет переменных как таковых – вместо них используются ссылки на объекты [7]. В отличие от некоторых других языков программирования, оператор = не является оператором присваивания значения переменной, он связывает ссылку на объект с объектом, находящимся в памяти. Если ссылка на объект уже существует, ее легко можно связать с другим объектом, указав этот объект справа от оператора =. Если ссылка на объект еще не существует, она будет создана оператором =.

### **Типы данных**

В Python имеются встроенные типы: булевый, строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. В языке нет некоторых, характерных для других языков, типов: символьного типа, массивов, нет разделения целых чисел на int и long – размер целого числа ограничен только имеющейся памятью компьютера, а не фиксированным числом байтов. Однако есть типы, позволяющие создавать коллекции данных: список, кортеж (неизменяемый список), словарь, множество и другие.

### **Библиотеки**

Одно из главных достоинств Python – наличие богатой встроенной библиотеки модулей и обширного множества сторонних библиотек. Здесь имеются средства для работы со многими сетевыми протоколами и форматами интернета, например, модули для написания HTTP-серверов и клиентов, для

разбора и создания почтовых сообщений, для работы с XML и т. п. Набор модулей для работы с операционной системой позволяет писать кросс-платформенные приложения. Существуют модули для работы с регулярными выражениями, текстовыми кодировками, мультимедийными форматами, криптографическими протоколами, архивами, сериализации данных, поддержка юнит-тестирования и др. Имеются модули для создания приложений с графическим интерфейсом. Всё это доступно «из коробки» – сразу после установки интерпретатора.

Большое количество сторонних библиотек позволяет расширить возможности языка: существуют библиотеки для научных и символьных вычислений, создания графических интерфейсов, работы с базами данных, программирования игр и др.

Для написания программ на Python имеется достаточно большое количество средств разработки. Одно из самых простых поставляется в комплекте с интерпретатором – это написанный на самом Python редактор кода с подсветкой синтаксиса – IDLE. Также есть более функциональные редакторы, например Spyder (рис. 1) [4], Pyzo [3].

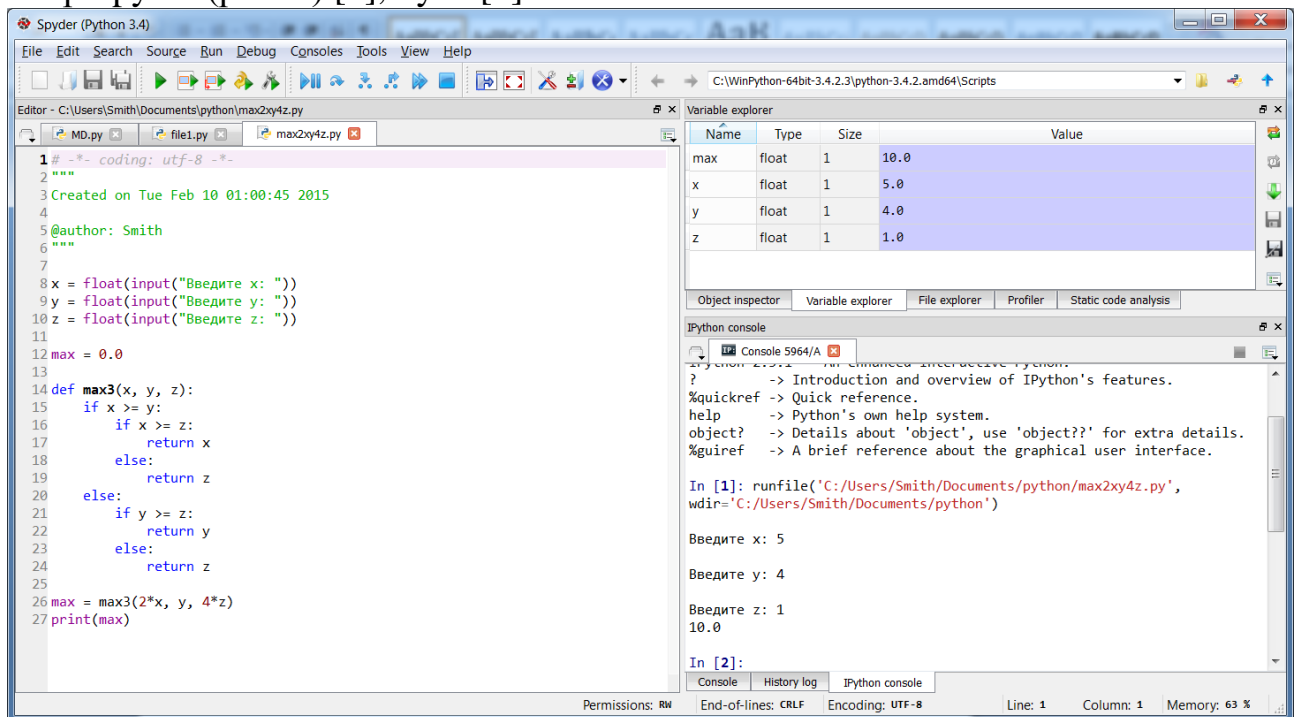


Рис. 1. Рабочее окно Spyder

После обсуждения некоторых особенностей языка перейдем к главной мысли настоящей статьи.

На наш взгляд, Python лучше, чем какой либо другой язык подходит для обучения программированию. На это есть ряд причин [6]:

- понятный синтаксис, отступы как часть синтаксиса позволяют быстро освоиться с основами языка;
- наличие развитой структуры данных: списки, кортежи, словари, множества и др.
- поддержка нескольких парадигм программирования;

- Python широко применяется в профессиональных разработках, то есть не является чисто учебным языком без перспектив применения в реальной жизни;
- богатая библиотека встроенных и сторонних модулей;
- возможность разработки приложений с графическим интерфейсом;
- наличие нескольких свободно распространяемых интерпретаторов для всех популярных операционных систем.

Помимо обучения основам программирования, на базе Python можно построить изучение многих других курсов.

Так, с использованием этого языка можно изучать объектно-ориентированное программирование (дисциплина «ООП») и функциональное программирование.

В рамках дисциплины «Веб-программирование» Python может полноценно заменить изучение языка PHP, так как предоставляет богатые возможности в этой сфере.

С использованием свободной библиотеки NumPy и подобных можно проводить различные научные вычисления, что может помочь при изучении дисциплин «Компьютерное моделирование», «Численные методы» и др.

Во всех этих случаях нет необходимости изучать новые языки (например, Matlab/Scilab для компьютерного моделирования или PHP для веб-программирования), учащиеся уже владеют необходимым инструментом. Это позволяет сразу погрузиться в изучение особенностей предмета и сэкономить время на изучении языков и программного обеспечения.

Таким образом, именно на базе Python можно построить комплекс дисциплин, который с одной стороны позволит снизить нагрузку на обучающихся (нет необходимости изучать много языков), а с другой – позволит изучать широкий спектр технологий, используя знакомый инструмент.

### **Библиографический список используемой литературы:**

1. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities [Электронный ресурс] / Режим доступа: <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext> (Дата обращения 13.04.2015);
2. Python, Wikipedia [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/Python> (Дата обращения 13.04.2015)
3. Pyzo, официальный сайт [Электронный ресурс] / Режим доступа: <http://www.pyzo.org/>
4. Spyder, Wikipedia [Электронный ресурс] / Режим доступа: [https://ru.wikipedia.org/wiki/Spyder\\_\(IDE\)](https://ru.wikipedia.org/wiki/Spyder_(IDE)) (Дата обращения: 13.04.2015)
5. TIOBE Index [Электронный ресурс] / Режим доступа: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (Дата обращения 13.04.2015);

6. Поляков К.Ю. Язык Python глазами учителя [Электронный ресурс] / Режим доступа: <http://kpolyakov.spb.ru/download/inf-2014-09a.pdf> (дата обращения 13.04.2015)
7. Программирование на Python 3. Подробное руководство. – Пер. с англ. – СПб.: Символ-Плюс, 2009. – 608 с., ил.