

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ ИЗУЧЕНИЯ ЯЗЫКА PYTHON В СРЕДНЕЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ШКОЛЕ

Modern tendencies of studying Python in middle school

Шабалин Юрий Константинович, техник Управления информационных технологий МПГУ.

 yuk.shabalin@mpgu.edu

Статья рассматривает язык Python и причины, по которым он набирает популярность среди преподавателей информатики, ищущих современные средства обучения программированию в средней школе. Показаны преимущества языка Python как переходного звена от начального к продвинутому уровню программирования. Обсужден выбор Python для средней школы, рассмотрены преимущества и недостатки языка.

The article focuses on the Python language and the reasons why it is gaining popularity among Informatics teachers, searching for modern means of teaching programming in middle school. Python is introduced as a transition stage from primary to advanced level programming. Article explains why to choose Python, reviewed advantages and disadvantages of the language.

Ключевые слова: **программирование, обучение, Python, средняя школа.**

Keywords: **programming, learning, Python, middle school.**

Развитие информационного общества обуславливает ситуацию, когда программирование становится необходимым инструментом деятельности очень широкого круга работников, не только сферы информационных технологий. Умение программировать на начальном уровне обеспечивается курсом информатики в основной школе: учащиеся осваивают минимальный уровень владения выбранным языком программирования и могут применить знания для решения тех задач, в которых они смогут выделить алгоритмические структуры.

Поскольку изучение информатики в основной школе является начальной ступенью в подготовке специалистов практически любой современной отрасли, то, от качества полученных на этом этапе знаний, умений и навыков зависит очень многое. Одним из критериев качества можно считать актуальность получаемых знаний, их соответствие текущим и будущим запросам общества, работодателей, поэтому одной из первоочередных задач при обучении программированию становится проблема выбора подходящего инструмента.

Сегодня в качестве инструмента обучения программированию, на наш взгляд, наиболее актуально использовать язык Python. Он сочетает в себе простоту изучения и низкий порог вхождения как для новичков, так и для уже имеющих опыт в программировании учеников, а также многогранность и мощность традиционно используемых профессиональными программистами языков Java и семейства C. При всей своей популярности, Pascal и Basic сложно назвать современными, они хорошо подходят

для знакомства с программированием и оттачиванию навыков, но давно потеряли актуальность и были вытеснены продуктами, предлагающих более широкий набор возможностей при значительно более простой реализации кода [5]. Резкий переход от Pascal к Java или Ruby не эффективен: эти языки ориентированы на опытных пользователей, способных самостоятельно отслеживать свои ошибки при написании кода. Промежуточное положение занимает Python: лёгкий переход с начальных языков программирования при изучении Python помогает восполнить пробелы при изучении этих языков, а также подготавливает ученика к более нишевым языкам, если он проявит интерес в их изучении.

Python устанавливается на компьютеры под управлением любой из популярных ОС, а в дистрибутивах Linux он поставляется по умолчанию [2]. Строгие требования к структуризации кода воспитывают культуру программирования, а широкое использование в научной и web-среде делают полученные учащимся знания не только современными, но и применимыми для решения актуальных задач [1].

Язык Python не является слишком простым или слишком сложным: ученик может самостоятельно отлаживать код. Обязательным является условие понимания задачи и некоторые познания в английском языке: компиляторы Python далеко не такие «строгие», как Pascal, и будут исполнять код, только если его невозможно обработать на начальном уровне (опечатки) или на уровне согласования данных. При этом многие служебные слова и встроенные функции имеют краткие названия или аббревиатуры английских слов.

Всё больше преподавателей выбирают Python для изучения в средней школе [6]. Помимо названных преимуществ у языка есть еще одно: итоговая аттестация школьников и олимпиады по программированию всероссийского уровня часто включают задания, требующие от ученика знаний и умений применять Python.

Когда время на решение задачи ограничено, а сама задача является вызовом для программиста, он должен тратить как можно меньше времени на написание кода. В этом Python незаменим, показывая свою скорость и простоту с первых уроков.

Рассмотрим классическое «Привет, мир!» на этом языке:

```
print ('Hello, world!')
```

Заметим, что представленная строчка является готовой программой. Она не содержит переменных, обозначений фрагментов кода или типизации данных. Команда определяет результат. Но в этом заключается и нежелательность использования Python в качестве первого языка программирования: зачастую задачу не решить, если не понимать всех процессов, скрывающихся за простой, на первый взгляд, командой.

Основная идея Python – удобный и легко воспринимаемый код [7]. Исходный код программы на Python можно читать, как обычный английский текст. Но для этого код необходимо правильно оформить. В Python

структуризация кода не только важна для удобства, она необходима для правильной работы программы. Например, для $i = 0$: [2]

<pre>while i<10: i = i + 1 print (i)</pre>	<pre>while i<10: i = i + 1 print (i)</pre>
---	---

Сравним два фрагмента. Первый выведет на экран число 10, второй же напишет числа от 1 до 10. Описание такого фрагмента в другом языке программирования потребовало бы дополнительных символов или служебных слов, `Begin end, { }`, но в Python эти слова и символы заменяют отступы. Для того, чтобы включить код в область действия цикла, нужно сделать отступ длиной в 4 пробела. Это укажет компилятору, что выделенный отступом код является частью какой-либо структуры. Такой способ оформления кода является одной из причин популярности языка, а у уставших от нагромождения символов учеников обычно вызывает восторг.

Изучение Python целесообразно начинать, обзорно повторив изученное, например, на Pascal: структуризация кода, переменные, ввод и вывод данных, типы данных, операторы, циклы, функции. Постепенно углубляясь в особенности языка, можно перейти к любой теме, от графиков до создания программ с графическим интерфейсом, при этом экономится время учителя и учеников [2].

Универсальность языка позволяет демонстрировать на его примере как объектно-ориентированное, так и структурное, функциональное, модульное и другие парадигмы программирования. Ученик увидит Python не как еще одно окно, которое из текста делает программу, а как инструмент, позволяющий реализовать любую идею и решить любую задачу, которую можно решить с помощью программирования. Изучая Python, учащийся узнаёт типы данных и виды циклов, локальные и глобальные переменные, списки, кортежи, словари, функции работы со строками, использование библиотек, математические функции языка, атрибуты объектов, модульность программ и многое другое [4].

Перечислим некоторые недостатки Python, которые хоть и не создают помех при изучении, но заметно снижают темп введения языка учебный процесс средней школы:

- Несовместимость программ, написанных на разных версиях языка: Python 2.7 обладает широчайшей библиотекой подключаемых модулей и уже написанных программ, которые сложно или невозможно перенести на актуальные версии языка.
- Требования к «граммотному» коду: написанный с грамматической или логической ошибкой код всё равно может быть выполнен, если не нарушает механизмов работы компилятора и транслятора. В сложном проекте «отлавливание» и исправление подобных ошибок может

занять много времени, если программист изначально не уделяет проверке кода внимания.

- Расход памяти и быстродействие сравнительно с С и Java невысоки, даже несмотря на обширное использование Python в сетевых проектах. Интерпретируемость языка отрицательно сказывается на требуемой для работы программ памяти. Например, создание исполняемых (executable) файлов из кода Python включает в себя интеграцию части компилятора в конечный запускаемый файл, что увеличивает размер последнего. Из-за этого не так много крупных программных решений, написанных целиком на python. Зачастую этот язык используют для написания фрагментов кода. С другой стороны, это облегчает портирование приложений между ОС windows и Linux.
- Учащихся можно «избаловать» простым языком Python, что приведет к низкой мотивации изучения других языков [3].

Несмотря на указанные недостатки, Python остаётся наиболее удобным языком для перехода с начального уровня программирования на более продвинутый. Программисты, владеющие Python, востребованы и потребность в них ближайшие годы будет только расти. Учителю необходимо показать этот язык как надёжный и многогранный инструмент, но не сделать его универсальным, единственным достойным изучения.



БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бухарова Г.А., Комельских П. С. «Важность и необходимость внедрения языка программирования python в процесс обучения студентов» <http://elar.rsvpu.ru/handle/123456789/5>.
2. Кузнецов А.В., Зыбцева К.А. «Python как базовый язык для обучения программированию» <http://elibrary.ru/item.asp?Id=23244708>.
3. Поляков К.Ю. «Язык Python Глазами учителя», «Информатика» № 9, Изд. Дом 1 сентября, 2014 г.
4. Супрунов С. «Python – в школу!» <http://samag.ru/archive/article/1644>.
5. Федорова Н.Е. «Структура, содержание и методические подходы к преподаванию языка программирования python в школе» <http://elibrary.ru/item.asp?Id=23020834>.
6. Школы, использующие Python <https://wiki.python.org/moin/schoolsusingpython>.
7. «Dive into Python» <http://www.diveintopython.net/>.