

# Как я использовал PYTHON

Алексей Лавров

27 июня 2014 г.

## Содержание

<b>Введение</b> . . . . .	<b>2</b>
<b>1 С чего стоит начинать</b> . . . . .	<b>3</b>
<b>2 Что дальше</b> . . . . .	<b>3</b>
<b>3 Построение графиков</b> . . . . .	<b>3</b>
<b>4 Русские подписи для графиков</b> . . . . .	<b>4</b>
<b>5 Сохранение графиков</b> . . . . .	<b>4</b>
<b>6 Конкретный пример</b> . . . . .	<b>4</b>
<b>Установка и полезные ссылки</b> . . . . .	<b>5</b>

## Введение

Как это ни странно, но проучившись в техническом вузе шесть лет, я не повстречался с языком программирования PYTHON на общем курсе информатики. Были C, ASSEMBLER x86, немного C++, а PYTHON не было. Резонно будет заметить, что у нас существуют курсы по выбору, на которых можно выучить этот язык. Начиная ходить на *«Современную практику коллективной разработки программного обеспечения»*, который вел наш выпускник, но с началом практических работ просто закопался в синтаксисе. Пользу от умения программировать на тот момент я не понимал.

Однако интерес к PYTHON с тех пор у меня остался. И вот в один из дней мне попала ссылка на новую MOOC платформу – **edX**. Одним из двух курсов, который привлек мое внимание, оказался *Introduction to Computer Science and Programming Using Python*. Недолго думая, решил подписаться, рассчитывая одновременно убить двух зайцев: во-первых, начать изучать PYTHON с курса «для чайников», во-вторых, оценить уровень зарубежных лекций.

В этом небольшом документе я бы хотел описать, где впоследствии мне пригодился этот язык и как можно начать им пользоваться самому, не закапываясь глубоко в документацию. Сразу отмечу, что на текущий я пользовался PYTHON в основном для автоматизации обработки данных и построения графикой для научной работы.

## 1 С чего стоит начинать

В качестве первой книги советую прочитать [«Think Python»](#), Downey (2012), которая, кроме того, что содержит чуть более 200 страниц, еще и свободно распространяется. Из этой книги вы узнаете синтаксис, базовые классы данных и сможете поэкспериментировать с графикой и анимацией, используя написанный автором пакет *Swampy*.

Затем можно заглянуть в книгу, которая используется как основная на курсе edX, а именно [«Introduction to Computation and Programming Using Python»](#), Guttag (2013). Если вам интересны базовые возможности для визуализации данных (графики, диаграммы и пр.), то из нее можно взять только главы 12, 13, 15, 16 (при условии, что Downey (2012) вы прочитали до этого).

## 2 Что дальше

Наиболее полным руководством по PYTHON (и в то же наиболее затянутым) является книга [«Learning python»](#), Lutz (2013). Можно найти ответ практически на любой вопрос, касающийся языка.

PYTHON широко используется в научной среде. Если вы хотите узнать о возможностях языка для численных расчетов, то рекомендую ознакомиться с [«A primer on scientific programming with Python»](#), Langtangen (2009).

## 3 Построение графиков

Итак, допустим, вы прочитали Guttag (2013) не очень давно и помните содержание глав. Однако вам захотелось построить что-то отличное от примеров, использованных в книге. Тогда вам прямая дорога к [документации пакета matplotlib](#). Если хотите быстро запрогать и построить то, что у вас уже есть в мыслях, то написание кода по аналогии с [примерами на сайте](#) – ваш выход. Однако, если вы любите копаться в деталях и подкручивать различные параметры, осознавая фундаментальные вещи и чувствуя себя высокклассным прогером – беритесь за [«Matplotlib for Python developers»](#), Tosi (2009).

## 4 Русские подписи для графиков

Когда дойдет время до презентаций или публикаций графиков, вы столкнетесь с проблемой шрифтов. Примеры из вышеописанных книг с русскими буквами работать не будут. Вместо них будут просто квадраты (даже если используете *utf-8*). Решение проблемы хорошо описано [здесь](#).

## 5 Сохранение графиков

Для пользователей  $\text{\LaTeX}$  использование `PYTHON` будет вдвойне приятно, т.к. построенные графики можно автоматически сохранять в формате `.pdf` и `.pgf`. Последнее возможно, если у вас на машине стоит компилятор для  $\text{\LaTeX}$  с необходимыми пакетами. В документации пишут, например, об использовании `TEXLIVE`. Я же пользуюсь `MIKTEX` в силу исторических причин, что, однако, не делает меня менее счастливым. Все генерируется отлично.

Подробнее [написано в документации](#).

## 6 Конкретный пример

## Установка и полезные ссылки

Я пользуюсь версией интерпретатора PYTHON 2.7, которую можно найти [на официальном сайте проекта](#). После установки вы получите базовую функциональность, но все дополнительные пакеты придется ставить отдельно. Для удобства можно воспользоваться менеджером пакетов [pip](#), а можно ставить вручную, пользуясь [этим полезным архивом](#).

В качестве интерактивной оболочки я пользуюсь [IPYTHON](#). Она значительно удобнее IDLE, которая идет по умолчанию. Кратко и по делу об этой оболочке написано в книге [«Learning IPython for Interactive Computing and Data Visualization»](#), Rossant (2013).

## Список литературы

- Downey, Allen (2012). *Think Python*. "O'Reilly Media, Inc."
- Guttag, John V (2013). *Introduction to Computation and Programming Using Python*. MIT Press.
- Langtangen, Hans Petter (2009). *A primer on scientific programming with Python*. т. 6. Springer.
- Lutz, Mark (2013). *Learning python*. "O'Reilly Media, Inc."
- Rossant, Cyrille (2013). *Learning IPython for Interactive Computing and Data Visualization*. Packt Publishing Ltd.
- Tosi, Sandro (2009). *Matplotlib for Python developers: build remarkable publication quality plots the easy way*. Packt Publishing Ltd.