Санкт-Петербургский государственный инженерно-экономический университет
Кафедра вычислительных систем и программирования

Введение в язык программирования Python

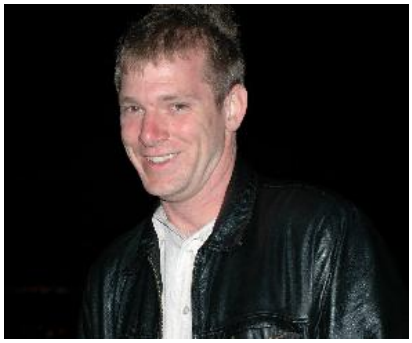
Новая версия: http://pycode.ru/edu/why-python/

Джейми Завински, лисп-хакер, один из первых разработчиков Netscape, http://www.jwz.org

Проектирование – это непрерывный процесс, никогда не знаешь, насколько проект хорош, пока программа не будет готова. Поэтому я предпочитаю как можно скорее снять пробу, получить что-нибудь на экране и посмотреть на это со всех сторон.

Думаю, одна из самых важных вещей, по крайней мере для меня, когда создаешь что-то с нуля, заключается как раз в том, чтобы как можно скорее довести программу до состояния, когда ты, программист, сможешь ее использовать. Когда что-то уже есть на экране и есть кнопка, связанная с некоторой функцией, появляется ощущение, какая кнопка будет следующей.

Брэд Фицпатрик, программист, создатель сервиса блогов LiveJournal, с августа 2007 года работает в Google,
http://www.bradfitz.com
http://brad.livejournal.com



Я беру исходный код Android просто так, без видимой причины. То же самое с Chrome: когда его код стал открытым, я сделал зеркало репозитория и стал изучать код. И тоже самое сделал с Firefox и с Open Office. Пользуешься какой-то программой, а потом получаешь доступ к ее исходному коду и можешь на него взглянуть.

Моя кривая обучения выглядит так: я быстро изучаю что-нибудь, пока не смогу работать с этим достаточно быстро и качественно. Обычно это где-то 80-90%, тогда я работаю весьма продуктивно, мне не приходится постоянно что-то искать, вот и отлично. После этого я продвигаюсь уже медленнее. И только если мне что-то очень уж приглянулось, говорю себе: «Пора поглубже изучить документацию этого языка – мануалы – и обшарить все потайные углы и щели».

Самый важный навык программиста – мыслить как ученый; за один раз изменять только что-то одно. Терпение и стремление понять коренную сущность вещей. Научитесь разрабатывать программу шаг за шагом, так чтобы была возможность проверять ее на каждом шаге.

Чтобы писать код, лицензия не нужна. Я не сторонник кучи предписаний, но не хотел бы, чтобы некоторые из PHP-программистов, допускающие атаки через XSS, занимались системами управления полетами.

Джо Армстронг, один из основателей языка программирования Erlang, http://www.sics.se/~joe/

Мне нравится разбираться в том, как все работает. Хороший способ проверки – реализовывать все самому. Для меня программирование - это не ввод кода в машину, это процесс понимания. Программирование есть понимание. Я люблю понимать, как все устроено. Зачем я создаю интерфейс для X Windows? Чтобы разобраться, как работает X-протокол.

Реализация чего-нибудь – сильный мотивирующий фактор. Рекомендую всем. Хотите понять Си – напишите для него компилятор. Хотите понять Лисп – напишите для него компилятор или интерпретатор. Некоторые говорят: «Компилятор – это же так трудно». Совсем нет. Это легко. Есть масса мелочей, которые не трудны и которые нужно освоить. Надо разбираться в структурах данных. Надо разбираться в хеш-таблицах и в парсинге. В генерировании кода. В техниках интерпретации. Каждый из этих предметов не особенно сложен. Начинающие думают, что это все большие и сложные темы, и поэтому к ним не подступают. Все, что вы не делаете, трудно, все, что вы уже сделали, легко [1].

Язык программирования Python

На сегодняшний день разработчики поддерживают параллельно две версии языка программирования Python: Python 2.X и Python 3.X. Хотя Python 3 рассматривается как будущее языка программирования, тем не менее, Python 2 по-прежнему широко используется. Версия 3 – это в значительной степени тот же язык программирования, но она практически несовместима с программным кодом, написанным для прежних версий интерпретатора [2].

Начинающие программисты могут сосредоточиться на версии Python 3.

Почему программисты используют Python?

- *Качество программного обеспечения*: удобочитаемость (вследствие единообразия оформления – отступы в Python являются частью синтаксиса, в отличие, например, от языка C++), ясность. Код легко читается, а значит, легче многократно его использовать. Поддержка объектно-ориентированного программирования (ООП).

- *Высокая скорость разработки*: меньше объем кода по сравнению с C, C++, Java, программы запускаются сразу, минуя этап компиляции.

- *Переносимость программ*: большая часть программ на Python выполняется без изменений на всех основных платформах (переносимые графические интерфейсы и т.д.).

- *Библиотеки поддержки*: поставляется стандартная библиотека с массой дополнительных возможностей, допускается расширение за счет своих библиотек либо сторонних (инструменты создания веб-сайтов[1], программирование математических вычислений - NumPy[2], разработка игровых программ[3] и т.д.). [4]

- *Интеграция компонентов*: программный код на Python имеет возможность вызывать функции из библиотек на языке C/C++, интегрироваться с программными компонентами на языке Java, взаимодействовать с COM и .NET.

Python позиционируется как объектно-ориентированный язык сценариев.

- *Недостатки*: скорость выполнения программ по сравнению с компилируемыми языками (C или C++). Python транслирует инструкции

[1] Проект Django: https://www.djangoproject.com/
[2] Сайт проекта NumPy: http://numpy.scipy.org/
[3] Возможность создавать 3D-модели: http://www.vpython.org/
[4] Более подробная информация о расширениях: http://pypi.python.org/pypi

исходного программного кода в *байт-код*, а затем интерпретирует этот байт-код. Байт-код обеспечивает переносимость программ, поскольку это платформонезависимый формат. Гораздо важнее, что преимущество в скорости разработки порой важнее потери скорости выполнения (с учетом быстродействия современных компьютеров).

Кто в наше время использует Python?

- Компания Google оплачивает труд создателя Python и использует язык в поисковых системах
- Сервис YouTube
- Компания Yandex в ряде сервисов[5]
- Программа BitTorrent написана на Python
- Веб-фреймворк App Engine от Google использует Python в качестве прикладного языка программирования
- Intel, Cisco, HP, IBM используют Python для тестирования аппаратного обеспечения
- NASA использует язык в научных вычислениях

Что можно делать с помощью Python?

Выделим наиболее типичные области применения языка Python.

- *Системное программирование:* инструмент для создания переносимых программ и утилит системного администрирования (инструментов командной строки), поддерживает все типичные инструменты операционных систем.

- *Графический интерфейс:* в состав Python входит стандартный ООП интерфейс к Tk GUI API – tkinter. При выборе библиотеки можно использовать другие инструменты – Qt (PyQt), GTK (PyGTK), MFC (PyWin32), .NET (IronPython), Swing (Jython – реализация языка Python на Java).

- *Веб-сценарии:* система HTMLGen позволяет создавать HTML-страницы на основе классов Python. Пакет mod_python предназначен для запуска сценариев на Python под управлением веб-сервера Apache. Полноценные пакеты веб-разработки – Django, TurboGears, web2py, Pylons, Zope.

- *Интеграция компонентов:* система Cython позволяет смешивать программный код на Python и C.

[5] Как Python стал делать погоду в Яндексе — Руслан Гроховецкий, PDF
Python в инфраструктуре поиска — Леонид Васильев, PDF

- *Приложения баз данных:* в языке имеются интерфейсы доступа (и переносимый прикладной программный интерфейс) ко всем основным реляционным базам данных – Sybase, Oracle, Informix, ODBC, MySQL и т.д.

- *Быстрое создание прототипов:* можно создавать прототипы на Python, а затем переносить выбранные компоненты на компилирующие языки (C и C++), отдельные компоненты можно оставить на Python, что упростит сопровождение и использование.

- *Программирование математических и научных вычислений:* расширение NumPy для математических вычислений.

- *Игры, искусственный интеллект, XML и многое другое:* создание игровых программ с помощью системы pygame, управлять роботом с помощью PyRo, анализ фраз на естественном языке с помощью NLTK[6].

Сильные стороны Python

- *Изначально ООП язык*, но ООП не является обязательным в Python. Как и C++, Python поддерживает оба стиля программирования – процедурный и объектно-ориентированный.

- Может использоваться и распространяться *совершенно бесплатно*, можно получить полные исходные тексты реализации Python.

- Стандартная реализация Python написана на *переносимом* ANSI C, благодаря чему он компилируется и работает практически на всех основных платформах (Linux, UNIX, MS Windows, Mac OS, BeOS, iPod, Symbian и т.д.)

Создатель Python Гвидо ван Россум, математик по образованию, назвал свое детище в честь комедийного сериала «Летающий цирк Монти Пайтона», который транслировался по каналу BBC. Это неизбежно добавляет юмора в примеры кода на языке Python. Например, слово «spam» взято из известной пародии Мони Пайтона, где герои сериала пытаются заказать блюдо в кафетерии, а их заглушает хор викингов, поющих о консервах фирмы Spam.

Хороший программист знает, что свой программный код он пишет для другого человека, который будет вынужден читать его в ходе сопровождения и использования.

Список используемой литературы

1. Кодеры за работой. П.Сейбел.
2. Изучаем Python. Лутц.

[6] http://nltk.org