

Азбука халтурщика-ARMатурщика

разработка встраиваемых систем
основы бытовой автоматики,
систем управления и сбора данных

© ruOpenWrt

© HackSpace «Чебураторный завод»

© Консорциум хоббитов России

22 ноября 2014 г.

Оглавление

Введение	8
I Основы электроники	9
1 Линейные схемы на пассивных элементах, основы электротехники	11
2 Симуляция и расчет схем в ngSPICE	12
3 KiCAD	13
3.1 Отрисовка схем в KiCAD	13
3.2 Библиотеки элементов	13
3.3 Передача схемы в ngSPICE	13
4 Простейшие полупроводниковые элементы	14
4.1 Оптоэлектроника	14

4.2	Схемы на биполярных транзисорах	14
4.3	Схемы на на полевых транзисорах	14
5	Операционные усилители	15
6	Источники питания	16
6.1	Батарейное питание	16
6.2	Линейные стабилизаторы	16
6.3	Импульсные преобразователи на ШИМ-контроллерах	16
6.4	Цепи защиты и гашения кондуктивных помех	16
7	Цифровая электроника	17
8	Компьютерные интерфейсы	18
8.1	Поколение 90х: COM, LPT, ISA	19
8.1.1	Резервный программатор AVR “пять проводков”	19
8.2	Сеть CAN	19
8.3	Интерфейсные модули USB	19
8.3.1	Универсальный высокоскоростной конвертер FTDI FT2232H	19
8.3.2	JTAG-адаптер	19
8.3.3	Отладочный модуль CAN	19
8.4	Интерфейсные модули Ethernet	19
9	ПЛИС	20
10	Датчики	21

11 Электропривод и исполнительные устройства	22
II Основы конструирования РЭС	23
12 Пакеты моделирования на основе OpenFOAM	24
13 Обеспечение теплового режима	25
14 Электромагнитная совместимость	26
14.1 Кондуктивные помехи	26
14.2 Компонентные модели и оптимизация кабельной сети	26
III Технология РЭС	27
15 Трассировка плат и подготовка производства в KiCAD	28
15.1 Технология ЛУТ (Лазерный УТюг)	28
15.2 Технология фоторезиста	28
15.3 Формат Gerber и подготовка промышленного производства	28
16 FreeCAD	29
16.1 Чертеж	30
16.2 Эскиз	30
16.3 Деталь	30
16.4 Сборка	30

16.5 Автогенерация конструкторской документации	30
16.6 Скрипты и пользовательские расширения	30
17 Эксплуатация станочного оборудования	31
18 Основы ЧПУ и цифрового производства	32
18.1 САМ-пакеты для FreeCAD	32
IV Основы теории систем автоматического управления	33
19 Математический аппарат	34
19.1 Передаточная функция	34
19.2 Устойчивость САУ	34
19.3 Сети Петри	34
19.4 Автоматы Маркова	34
20 Релейное управление	35
21 Пропорциональные САУ	36
22 ПИДn-регуляторы	37

V	Разработка ПО для встраиваемых систем	38
23	Вспомогательные скрипты на языке Python	39
23.1	Установка под Windows	41
23.2	Запуск	42
23.3	Дополнительные материалы	47
24	Make: управление сборкой проектов	48
25	VCS: системы контроля версий	49
25.1	CVS	49
25.2	Subversion	49
25.3	Git	49
25.3.1	GitHub	49
26	Основы Си и C₊	50
26.0.2	Установка MinGW (win32)	50
26.1	Особенности C ₊ в embedded	50
27	LLVM и разработка собственных компиляторов	51
27.1	Лексический и синтаксический анализ	51
27.2	Применение flex/bison для разбора текстовых форматов данных	51
27.3	Компилятор Паскаля	51
28	Сборка кросс-компилятора GNU toolchain	52

VI	Микроконтроллеры Cortex-Mx	53
VII	Периферия	54
VIII	Встраиваемый emLinux	55
29	cross	56
30	BuildRoot	57
31	Особенности OpenWrt	58
32	Библиотека SDL	59
32.1	Реализация microGUI	59
33	Приложения для X Window	60
34	Программирование сетевых приложений	61
35	Сборка кросс-компилятора GNU мальтийским крестом	62

IX IDE ECLIPSE 63

X Подготовка публикаций в \LaTeX 64

35.1 Установка MikTeX (win32)	68
35.2 Структура документа	68
35.2.1 Заголовочный файл или блок	68
35.2.2 Стили документа	68
35.2.3 Пакеты	68
35.2.4 Автор и название	68
35.2.5 Верстка титульных страниц	68
35.2.6 Оглавление	68
35.3 Верстка слайдов	68
35.4 Список литературы и цитирование	68
35.5 Команды секционирования: часть, глава, раздел,.. . . .	68
35.6 Таблицы	68
35.7 Формулы	68
35.8 Перекрестные ссылки и гипессылки	68
35.9 Листинги скриптов и текстовых данных	68
35.10 Подготовка иллюстраций	68
35.10.1 Графики GNUPLOT	68
35.10.2 Схемы и графы в GraphViz	68

Литература 69

Введение

Первоначально этот материал задумывался как комплект документации к платам BlackSwift и VoCore, но постепенно превратился в толстенный учебник для студентов ВУЗов и научных работников по специализациям, связанным с применением цифровой электроники и компьютерной техники.

Большой упор был сделан на использование открытого некоммерческого программного обеспечения, с целью удешевления учебного процесса, уменьшения себестоимости ваших проектов¹, и стимулирования вашего участия в развитии этих программных пакетов.

Лицензия на эту книгу пока не выбрана, так что она пока просто пишется в духе OpenSource: любой может использовать ее часть, изменять или дополнять, до тех пор, пока не накладываются какие-либо административные, финансовые или юридические ограничения на распространение и развитие оригинальной версии или ее открытых форков.

Приглашаем всех желающих участвовать в развитии этого учебного пособия на форум [ruOpenWrt](#), нам нужна обратная связь по качеству материала, результаты тестирования на вас или ваших студентах, дополнения и замечания.

¹ вряд ли ли у вас окажется лишняя пачка килобаксов на покупку пары коммерческих САПР, по крайней мере пока ваш стартап не взлетит в Top\$100K

Часть I

Основы электроники

Здесь идет список ссылок на онлайн лекции в edX, Coursera, и т.п.

Глава 1

Линейные схемы на пассивных элементах, основы электротехники

Глава 2

Симуляция и расчет схем в ngSPICE

Глава 3

KiCAD

3.1 Отрисовка схем в KiCAD

3.2 Библиотеки элементов

3.3 Передача схемы в ngSPICE

Глава 4

Простейшие полупроводниковые элементы

4.1 Оптоэлектроника

4.2 Схемы на биполярных транзисорах

4.3 Схемы на на полевых транзисорах

Глава 5

Операционные усилители

Глава 6

Источники питания

6.1 Батарейное питание

6.2 Линейные стабилизаторы

6.3 Импульсные преобразователи на ШИМ-контроллерах

6.4 Цепи защиты и гашения кондуктивных помех

Глава 7

Цифровая электроника

Глава 8

Компьютерные интерфейсы

8.1 Поколение 90х: COM, LPT, ISA

8.1.1 Резервный программатор AVR “пять проводков”

8.2 Сеть CAN

8.3 Интерфейсные модули USB

8.3.1 Универсальный высокоскоростной конвертер FTDI FT2232H

8.3.2 JTAG-адаптер

8.3.3 Отладочный модуль CAN

Глава 9

ПЛИС

Глава 10

Датчики

Глава 11

Электропривод и исполнительные устройства

Часть II

Основы конструирования РЭС

Глава 12

Пакеты моделирования на основе OpenFOAM

Глава 13

Обеспечение теплового режима

Глава 14

Электромагнитная совместимость

14.1 Кондуктивные помехи

14.2 Компонентные модели и оптимизация кабельной сети

Часть III

Технология РЭС

Глава 15

Трассировка плат и подготовка производства в KiCAD

15.1 Технология ЛУТ (Лазерный УТюг)

15.2 Технология фоторезиста

15.3 Формат Gerber и подготовка промышленного производства

Глава 16

FreeCAD

16.1 Чертеж

16.2 Эскиз

16.3 Деталь

16.4 Сборка

16.5 Автогенерация конструкторской документации

16.6 Скрипты и пользовательские расширения

Глава 17

Эксплуатация станочного оборудования

Глава 18

Основы ЧПУ и цифрового производства

18.1 САМ-пакеты для FreeCAD

Часть IV

Основы теории систем автоматического управления

Глава 19

Математический аппарат

19.1 Передаточная функция

19.2 Устойчивость САУ

19.3 Сети Петри

19.4 Автоматы Маркова

Глава 20

Релейное управление

Глава 21

Пропорциональные САУ

Глава 22

ПИДn-регуляторы

Часть V

Разработка ПО для встраиваемых систем

Глава 23

Вспомогательные скрипты на языке Python



Название языка произошло вовсе не от вида пресмыкающихся. Автор назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона». Впрочем, всё равно название языка чаще ассоциируют именно со змеёй, нежели с передачей — пиктограммы файлов в KDE или в Microsoft Windows и даже эмблема на сайте <http://www.python.org> (до выхода версии 2.5) изображают змеиные головы.

Python¹ — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Python удобно применять для написания различных вспомогательных скриптов. Часто его используют при разработке сложных программных систем для написания первых версий. В процессе работы над большими программами часто перерабатываются большие объемы кода, поэтому для ускорения разработки требуется максимально высокоуровневый язык. После того как архитектура программы стабилизируется, узким местом становится производительность, и программу переписывают на более низкоуровневом компилируемом языке, чаще всего C_+^+ .

Написание программ упрощают:

- **объектно-ориентированное программирование** облегчает разработку программ, позволяет переопределить стандартные операторы для пользовательских типов данных, упрощая синтаксис
- **динамическая типизация** не требуется заранее упределять переменные, они создаются простым присваиванием
- **обработка исключений** для секции кода можно определить обработчик ошибок
- **высокоуровневые структуры данных** — списки, словари (набор элементов ключ:значение), очереди
- богатая стандартная библиотека и множество дополнительных библиотек на все случаи

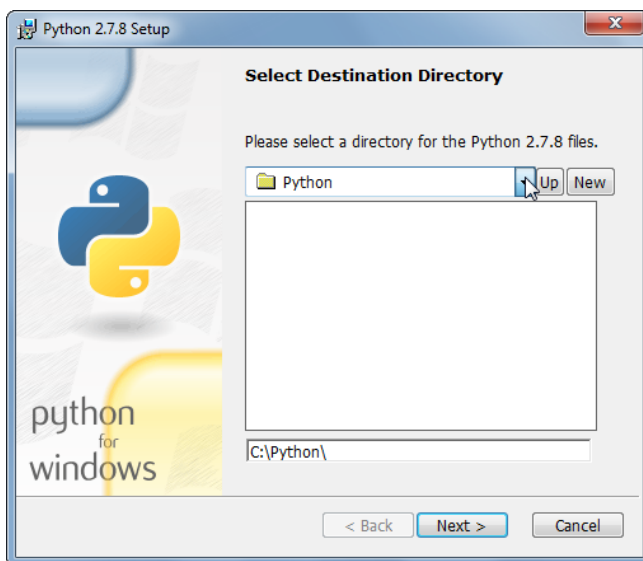
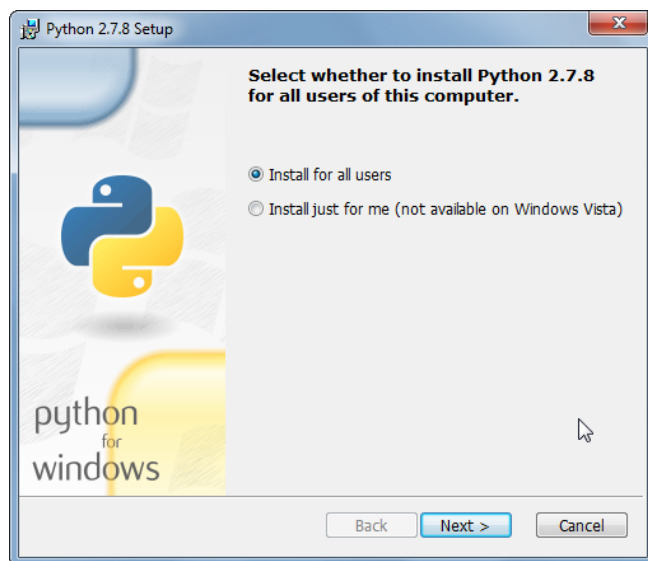
¹ в оригинале читается **пáйтон**, но давно русифицировался как **питóн**

23.1 Установка под Windows

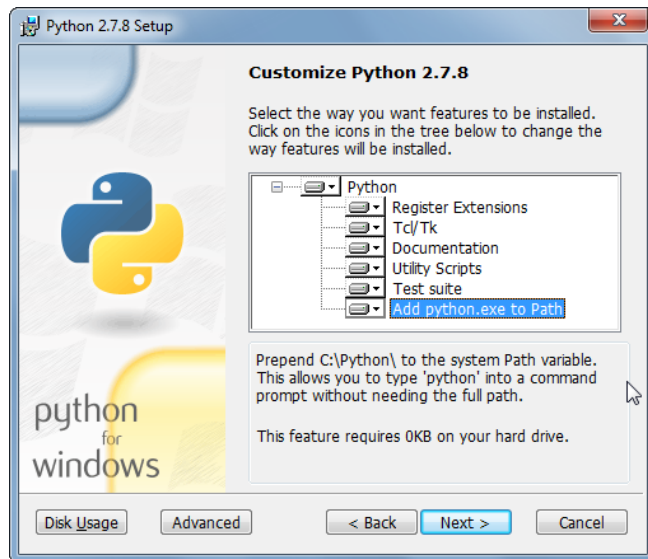
⊞ + R >> <http://www.python.org> >> Downloads >> Python 2.7.8

python-2.7.8.msi >> Setup >> for all users/for me


Destination Directory >> C:/Python/ >> Next

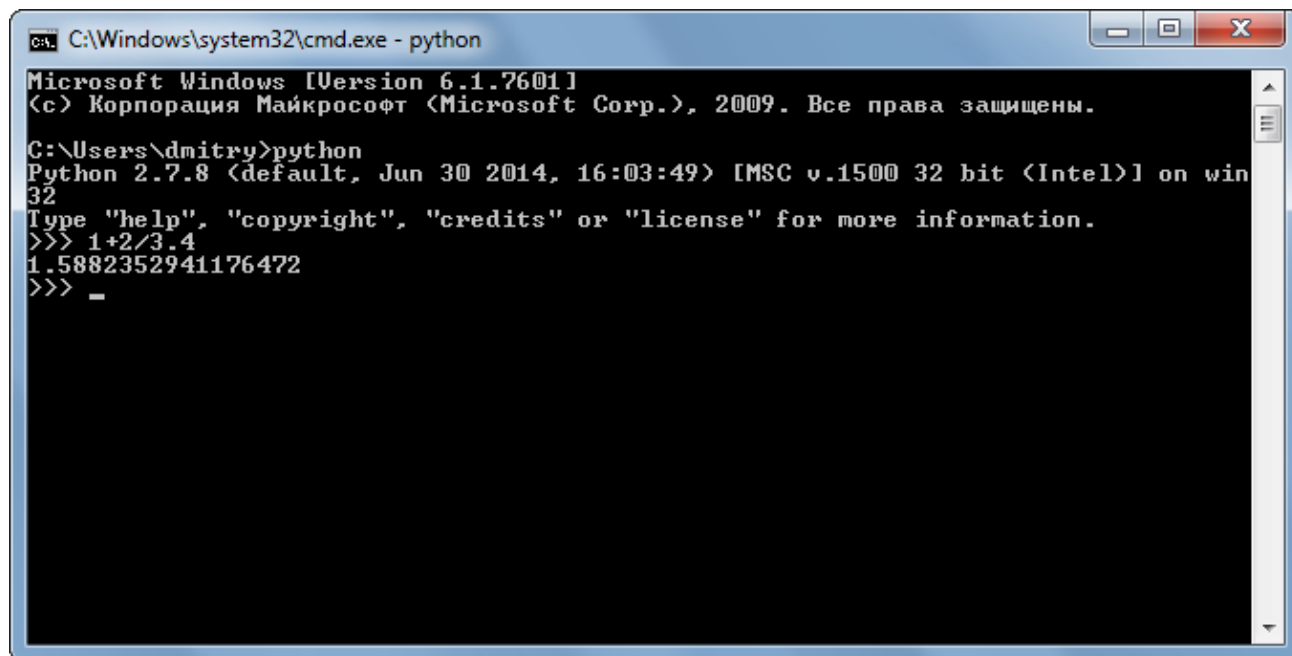


Customize » Python » Add python.exe to PATH » Next » Finish



23.2 Запуск

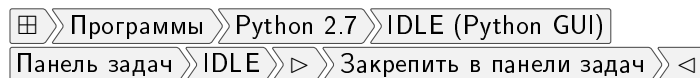
Из командной строки:  + R » cmd » python



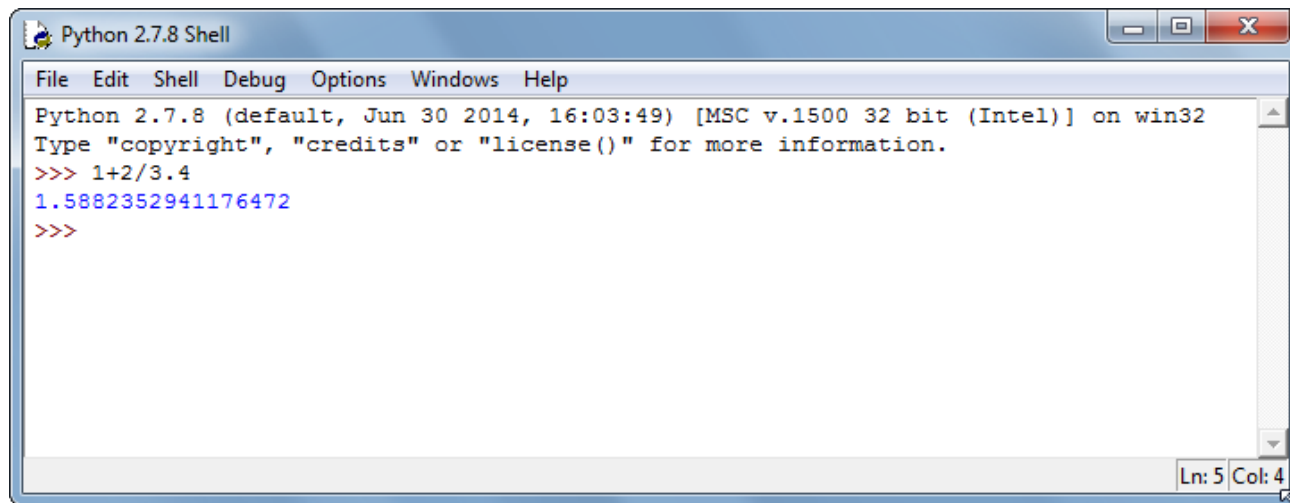
```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.

C:\Users\dmitry>python
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> 1+2/3.4
1.5882352941176472
>>> _
```

Простейшая среда IDLE²:



² на GUI-библиотеке Tkinter, идущей в комплекте



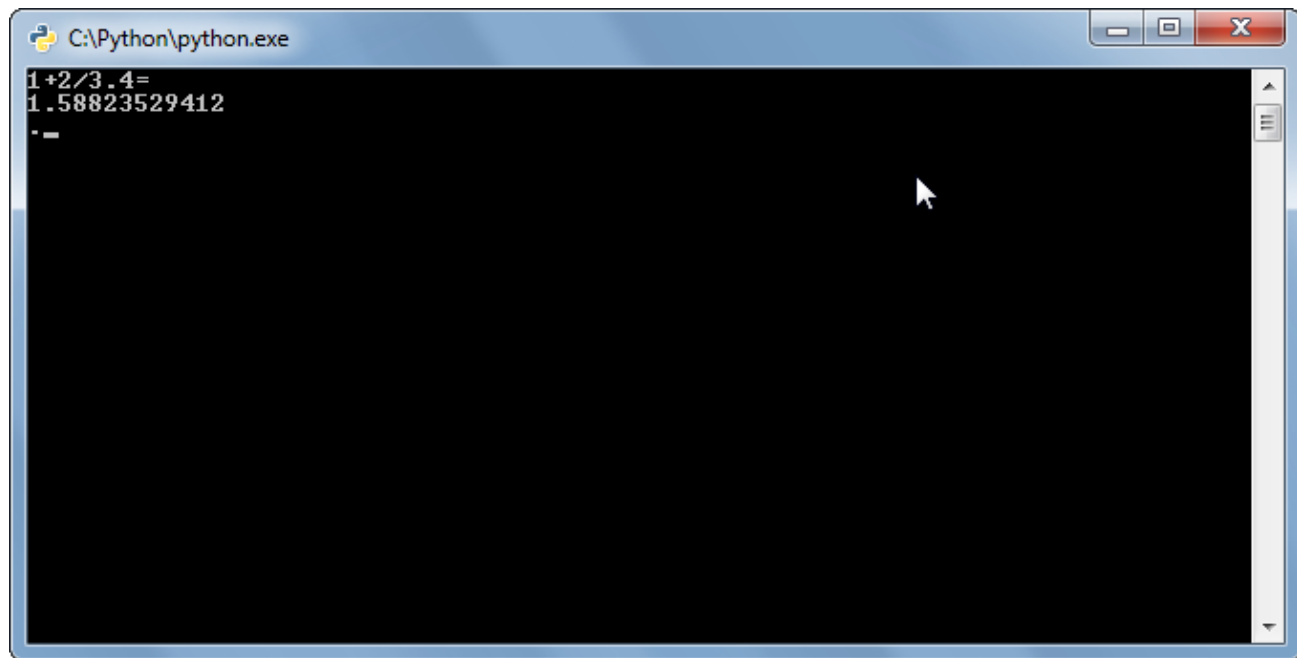
◁◁ по файлу скрипта:

 +  notepad /tmp/py.py

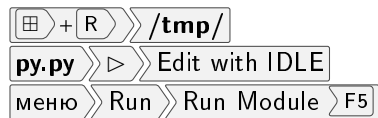
/tmp/py.py

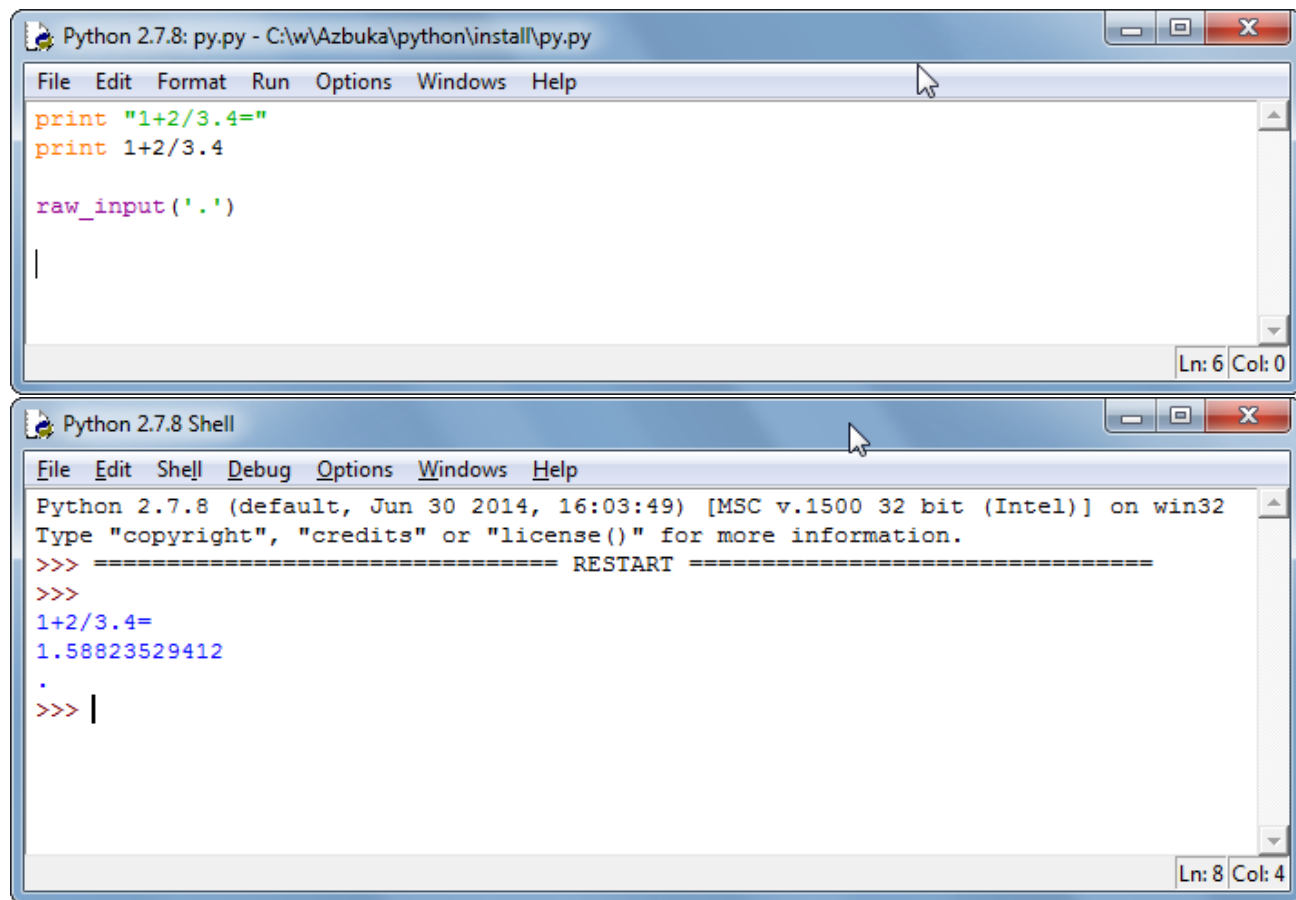
```
1 print "1+2/3.4="
2 print 1+2/3.4
3
4 raw_input( ' . ' )
```

 +  /tmp/py.py



Открытием файла скрипта в IDLE:





23.3 Дополнительные материалы

- [1] Г. Россум, Ф.Л.Дж. Дрейк, Д.С. Откидач, [Язык программирования Python](#)
- [2] Аллен Дауни [Думать на языке Python: Думать как компьютерный специалист](#)

Глава 24

Make: управление сборкой проектов

Глава 25

VCS: системы контроля версий

25.1 CVS

25.2 Subversion

25.3 Git

25.3.1 GitHub

Глава 26

ОСНОВЫ СИ И C_+^+

26.0.2 Установка MinGW (win32)

26.1 Особенности C_+^+ в embedded

Глава 27

LLVM и разработка собственных компиляторов

27.1 Лексический и синтаксический анализ

27.2 Применение flex/bison для разбора текстовых форматов данных

27.3 Компилятор Паскаля

Глава 28

Сборка кросс-компилятора GNU toolchain

Часть VI

Микроконтроллеры Cortex-Mx

Часть VII

Периферия

Часть VIII

Встраиваемый emLinux

Глава 29

cross

Глава 30

BuildRoot

Глава 31

Особенности OpenWrt

Глава 32

Библиотека SDL

32.1 Реализация microGUI

Глава 33

Приложения для X Window

Глава 34

Программирование сетевых приложений

Глава 35

Сборка кросс-компилятора GNU мальтийским крестом

Часть IX

IDE  ECLIPSE

Часть X

Подготовка публикаций в L^AT_EX

LaTeX (по-русски произносится **латéx**) — наиболее популярный набор макрорасширений (или макропакет) системы компьютерной вёрстки T_EX, который облегчает набор сложных документов. В типографском наборе форматируется как L^AT_EX.

Главная идея L^AT_EX состоит в том, что авторы должны думать о содержании, о том, что они пишут, не беспокоясь о конечном визуальном облике (печатный вариант, текст на экране монитора или что-то другое). Готовя свой документ, автор указывает логическую структуру текста (разбивая его на главы, разделы, таблицы, изображения), а L^AT_EX решает вопросы его отображения. Так содержание отделяется от оформления. Оформление при этом или определяется заранее (стандартное), или разрабатывается для конкретного документа.

В практическом смысле использование L^AT_EX позволяет (в порядке уменьшения важности):

- с помощью макросов и T_EX-программирования реализовывать любые стили и самую сложную верстку, существует множество готовых пакетов для верстки графических химических формул, разнообразных схем, транскрипционных знаков, внезапно электронных схем, цветных листингов и т.п.
- автоматизировать работу с документами: пересобирать выходные файлы через [Make](#), генерировать части документов с помощью своих скриптов²
- получить выходной документ в .pdf .html .txt .PostScript .djvu ... с кликабельными ссылками, анимированными, а иногда и интерактивными элементами
- не использовать файлы документов в закрытом формате

¹ копияста <https://ru.wikipedia.org/wiki/LaTeX>

² отчеты, стандартные формы, результаты работы любых программ

- легко держать набор файлов в [VCS](#)
- не покупать текстовый процессор

Особенно важен пункт про сложную верстку: она всегда нужна в крупных технических публикациях, особенно в учебной литературе, или отчетных работах. Вам обязательно понадобится вставлять графики экспериментальных данных, тематически специфичные схемы, листинги, выходные данные работы ваших программ и т.п.

Традиционно \LaTeX любим математиками, и всеми кто готовит публикации с большим количеством формул и перекрестных ссылок: после небольшого обучения формулы вводятся с листа со скоростью набора текста, особенно если ваш редактор умеет автодополнение, и никакой мышью возни.

Естественно всякие чисто автоматические вещи типа автонумерации ссылок и формул, сборки оглавлений и индексов, цветовая подсветка синтаксиса в листингах программ, размещение плавающих иллюстраций и т.п. выполняются автоматически \TeX -процессором в пакетном режиме, и на выходе получается красивый печатный или электронный (.pdf) документ.

Единственная область, не удобная в \LaTeX -верстке — создание сложных таблиц. Для этого были созданы визуальные редакторы, позволяющие отрисовать структуру таблицы мышью, а затем заполнить готовый шаблон данными.

35.1 Установка MikTeX (win32)

35.2 Структура документа

35.2.1 Заголовочный файл или блок

35.2.2 Стили документа

35.2.3 Пакеты

35.2.4 Автор и название

35.2.5 Верстка титульных страниц

35.2.6 Оглавление

35.3 Верстка слайдов

35.4 Список литературы и цитирование

35.5 Команды секционирования: часть, глава, раздел,...

35.6 Таблицы

35.7 Формулы

Литература

- [1] Г. Россум, Ф.Л.Дж. Дрейк, Д.С. Откидач, М. Задка, М. Левис, С. Монтаро, Э.С. Реймонд, А.М. Кучлинг, М.-А. Лембург, К.-П. Йи, Д. Ксиллаг, Х.Г. Петрилли, Б.А. Варсав, Дж.К. Ахлстром, Дж. Роскинд, Н. Шеменор, С. Мулендер.

Язык программирования Python. / 2001 — 454 с.

Python является простым и, в то же время, мощным интерпретируемым объектно-ориентированным языком программирования. Он предоставляет структуры данных высокого уровня, имеет изящный синтаксис и использует динамический контроль типов, что делает его идеальным языком для быстрого написания различных приложений, работающих на большинстве распространенных платформ. Книга содержит вводное руководство, которое может служить учебником для начинающих, и справочный материал с подробным описанием грамматики языка, встроенных возможностей и возможностей, предоставляемых модулями стандартной библиотеки. Описание охватывает наиболее распространенные версии Python: от 1.5.2 до 2.0.

© Stichting Mathematisch Centrum, 1990–1995

© Corporation for National Research Initiatives, 1995–2000

© ВеOpen.com, 2000

© Д.С. Откидач, 2001

[2] Аллен Дауни

Думать на языке Python: Думать как компьютерный специалист

версия 1.1.24+Kart (Python 3.2), перевод версия 1.06