

Лекция 1

Язык программирования Python введение.

Хайрулин Сергей Сергеевич

email: s.khairulin@g.nsu.ru, s.khayrulin@gmail.com

Ссылка на [материалы](#)

План

- Лекции/практические занятия
 - Тест
- Дифференцированный зачет в конце семестра
 - Защита задания

Литература

Начальный уровень

- Mark Pilgrim. Dive into Python - <http://www.diveintopython.net/>
- Марк Лутц. Изучаем Python, 4-е издание // Символ-Плюс 2011.
- ...

Стандарт/Документация

- PEP-8 - <https://www.python.org/dev/peps/pep-0008/>
- <https://www.python.org/>
- <https://github.com/python/cpython>

Экспертный уровень

- **Лучано Рамальо: Python. К вершинам мастерства**
- Mitchell L. Model. Bioinformatics Programming Using Python // O'Reilly 2010.

Программирование/Языки программирования

Вопросы

- Что это такое программирование?
- Что такое язык программирования?
- Для чего вообще вот это все?
- Ваши вопросы...?

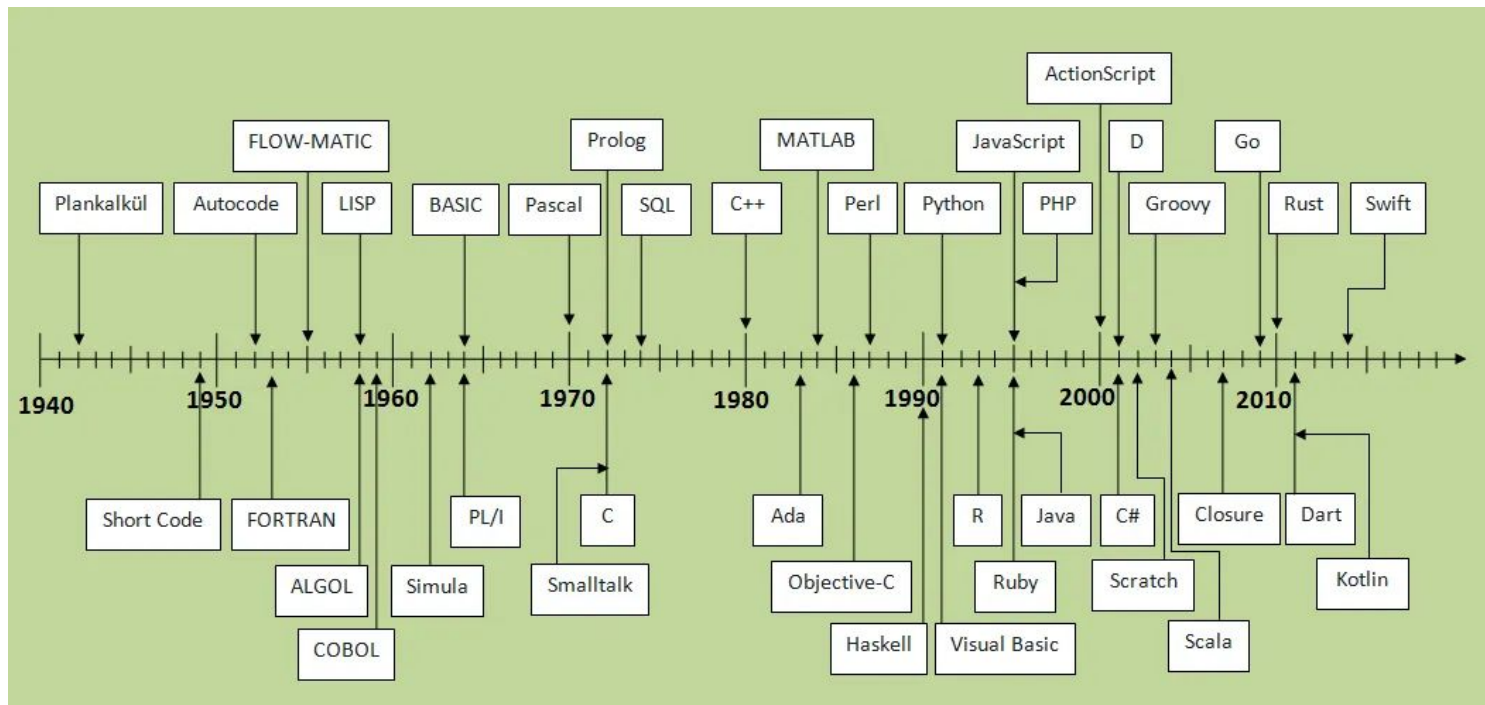
Программирование/Языки программирования

Формально

Программирование - процесс разработки программ, для вычислительных устройств с помощью языков программирования.

Язык программирования (ЯП) - формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под ее управлением (wiki).

Языки программирования(ЯП)



И это далеко не все...

яп

Зачем так много?

ЯП

С увеличением вовлеченности ЭВМ в промышленность экономику, появляется необходимость в специалистах и инструментах, с помощью которых можно **массово** создавать прикладные программы. Появляется **очевидная** необходимость в разработке языков программирования.

Все равно не понятно почему так много?

ЯП

```
extern printf ; Request symbol "printf".
global main   ; Declare symbol "main".

section .data
    str: DB "Hello World!", 0x0A, 0x00

section .text
main:
    PUSH str      ; Push string pointer onto stack.
    CALL printf   ; Call printf.
    POP eax       ; Remove value from stack.
    MOV eax,0x0   ; \_Return value 0.
    RET           ; /
```

ASM

```
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

Java

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

Python

Классификация ЯП

Эволюционно языки можно подразделять на два типа

- Языки низкого уровня -> низкоуровневое взаимодействие с конкретной платформой и устройствами ЭВМ - к таким языкам можно отнести машинный код, assembler

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 14 00 00 00 00 00 00 00 00 00 00 00 00 .....H.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....@.....@.....
00 00 00 18 00 00 00 00 00 00 00 00 00 00 00 00 .....FE FF FF 6F .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....@.....0.....0....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 t@.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....P.....6.@.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F@....V@.....GCC:
2D 31 75 62 75 6E 74 75 35 29 20 34 2E 36 2E (Ubuntu/Linaro 4.6.3-1ubuntu5) 4.6.
68 73 74 72 74 61 62 00 2E 69 6E 74 65 72 70 3...syntab..strtab..shstrtab..interp
67 6E 75 2E 62 75 69 6C 64 2D 69 64 00 2E 67 ..note.ABI-tag..note.gnu.build-id..g
74 72 00 2E 67 6E 75 2E 76 65 72 73 69 6F 6E nu.hash..dynsym..dynstr..gnu.version
2E 64 79 6E 00 2E 72 65 6C 61 2E 70 6C 74 00 ..gnu.version_r..rela.dyn..rela.plt.
64 61 74 61 00 2E 65 68 5F 66 72 61 6D 65 5F .init..text..fini..rodata..eh_frame_
2E 64 74 6F 72 73 00 2E 6A 63 72 00 2E 64 79 hdr..eh_frame..ctors..dtors..jcr..dy
64 61 74 61 00 2E 62 73 73 00 2E 63 6F 6D 6D namic..got..got.plt..data..bss..comm
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ent.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

Классификация ЯП

С увеличением сложности программируемых систем появляется необходимость в разработки более комплексных и гибких инструментов для разработки прикладного ПО.

- Языки высокого уровня

Классификация ЯП(высокоуровневые ЯП)

К таким языкам можно отнести

- C++
- Java
- C#
- Python
- Go
- Rust
- JS
- ...

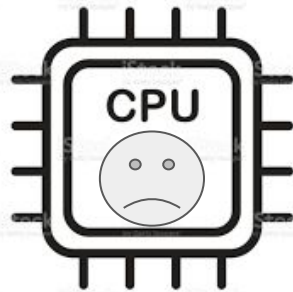
Еще классификации

Вообще говоря, дальнейшая классификация идет относительно ЯП высокого уровня

- Строго типизированный/динамическая типизация
- Интерпретируемый/Компилируемый
- Относительно парадигм программирования
 - поддерживает ООП?
 - императивный
 - функциональный
 - процедурный
 - ...
- ...

Компиляция/трансляция

```
def hello_world():  
    print("Hello World!")  
  
if __name__ == '__main__':  
    hello_world()
```



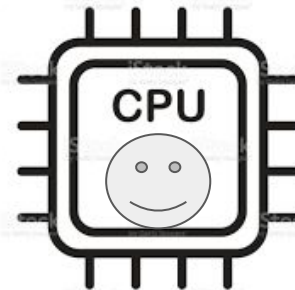
Центральный процессор (CPU) не понял, что от него хотят, ведь он говорит, только на ограниченном языке машинных инструкций, и понимает лишь слова содержащие 0 и 1.

Компиляция/трансляция

```
def hello_world():  
    print("Hello World!")  
  
if __name__ == '__main__':  
    hello_world()
```

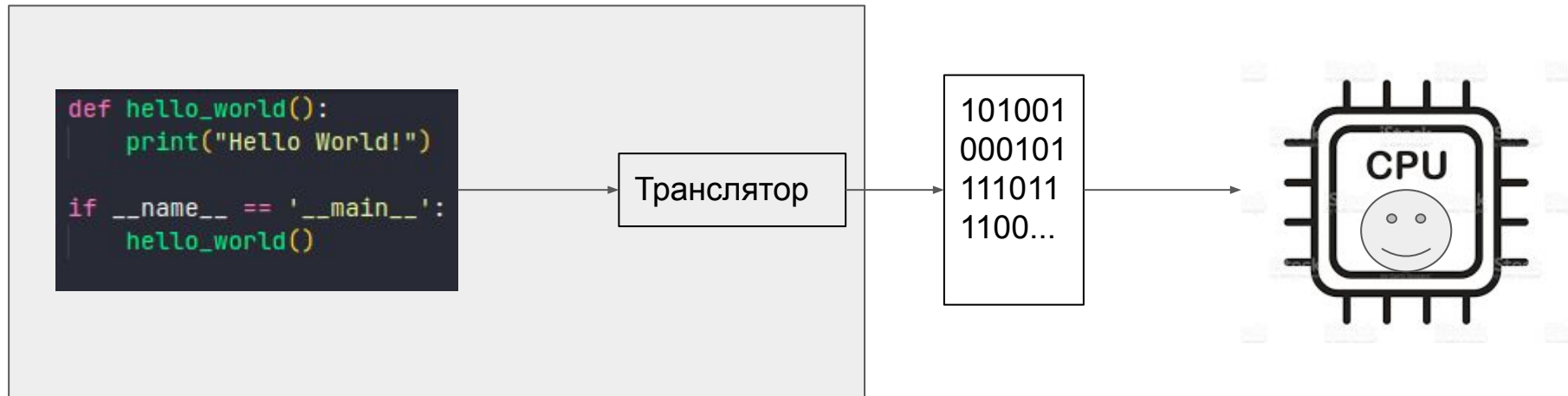
Компилятор

101001
000101
111011
1100...



В дело вмешался компилятор, который смог перевести инструкции ЯП в понятный для (CPU) язык машинных инструкций

Компиляция/трансляция



Теперь вместо компилятора мы поместим специализированную программу транслятор, который также способен переводить код в машинные инструкции. Разница в том, что компилятор выдает готовый к исполнению код, в то время как транслятор, выполняет код программы строчка за строчкой, то есть транслятор переводит инструкции в момент выполнения программы

Интерпретируемые ЯП

- + Кроссплатформенность, пошаговое отслеживание выполнения программы, модификация программы во время исполнения, меньшие затраты времени на разработку и отладку, простой способ создания переносимых программ, не требует затрат на компиляцию небольших программ ...
- Основным недостатком является более медленное выполнение программ. Необходимость везде и всегда таскать с собой рантайм(среду выполнения кода)*.

* Последнее замечание конечно нивелируется тем, что Python весьма популярен и стандартная библиотека как и сам интерпретатор поставляется обычно со всеми более или менее популярными дистрибутивами ОС “из под капота”.

Интерпретируемые ЯП

Интерпретируемые

- Python
- Java
- Bash
- Ruby
- Perl
- C#
- ...

Компилируемые

- C
- C++
- Assembler
- Go
- Pascal
- Rust
- Swift
- ...

Почему Python?

- Постоянно растущая популярность языка дает импульс для развития и совершенствования
- Большое комьюнити
- Огромное многообразие библиотек для различных нужд от программирования микроконтроллеров, до машинного обучения
- Машинное обучение
- Низкий порог вхождения
 - прост в освоение
 - можно быстро начать писать код для продакшена
- Универсальность

Рейтинги

Aug 2021	Aug 2020	Change	Programming Language	Ratings	Change
1	1		 C	12.57%	-4.41%
2	3	▲	 Python	11.86%	+2.17%
3	2	▼	 Java	10.43%	-4.00%
4	4		 C++	7.36%	+0.52%

<https://www.tiobe.com/tiobe-index/>

PULL REQUEST

Year: 2021 Quarter: 2

# Ranking	Programming Language	Percentage (YoY Change)	YoY Trend
1	JavaScript	19.014% (+0.225%)	
2	Python	16.351% (+0.243%)	
3	Java	12.817% (+2.086%)	
4	Go	7.574% (-1.348%)	

https://madnight.github.io/github/#/pull_requests/2021/2

Community

Repositories	2M+
Code	132M
Commits	28M+
Issues	6M
Discussions	16K
Packages	5K
Marketplace	172
Topics	4K
Wikis	280K
Users	129K



Python

Python is a dynamically typed programming language.

[See topic](#)

☆ Star

Showing 2,170,439 available repository results ?

Sort: Best match ▼



TheAlgorithms/Python



Sponsor

All Algorithms implemented in **Python**

[python](#)

[hacktoberfest](#)

[education](#)

[algorithm](#)

[practice](#)

[interview](#)

[sorting-algorithms](#)

[learn](#)

[algos](#)

[algorithm-competitions](#)

[sorts](#)

[algorithms-implemented](#)

[community-driven](#)

[searches](#)



118k



Python

MIT license

Updated 11 hours ago

1 issue needs help

МНОГО библиотек/инструментов

[Help](#)[Sponsors](#)[Log in](#)[Register](#)

Find, install and publish Python packages
with the Python Package Index



Or [browse projects](#)

326,533 projects

2,864,328 releases

4,843,946 files

535,633 users

Машинное обучение

Обилие специализированных библиотек для работы с ML/AI/DS - делает python очень привлекательным \Rightarrow выбору Python в качестве основного инструмента.



Порог вхождения

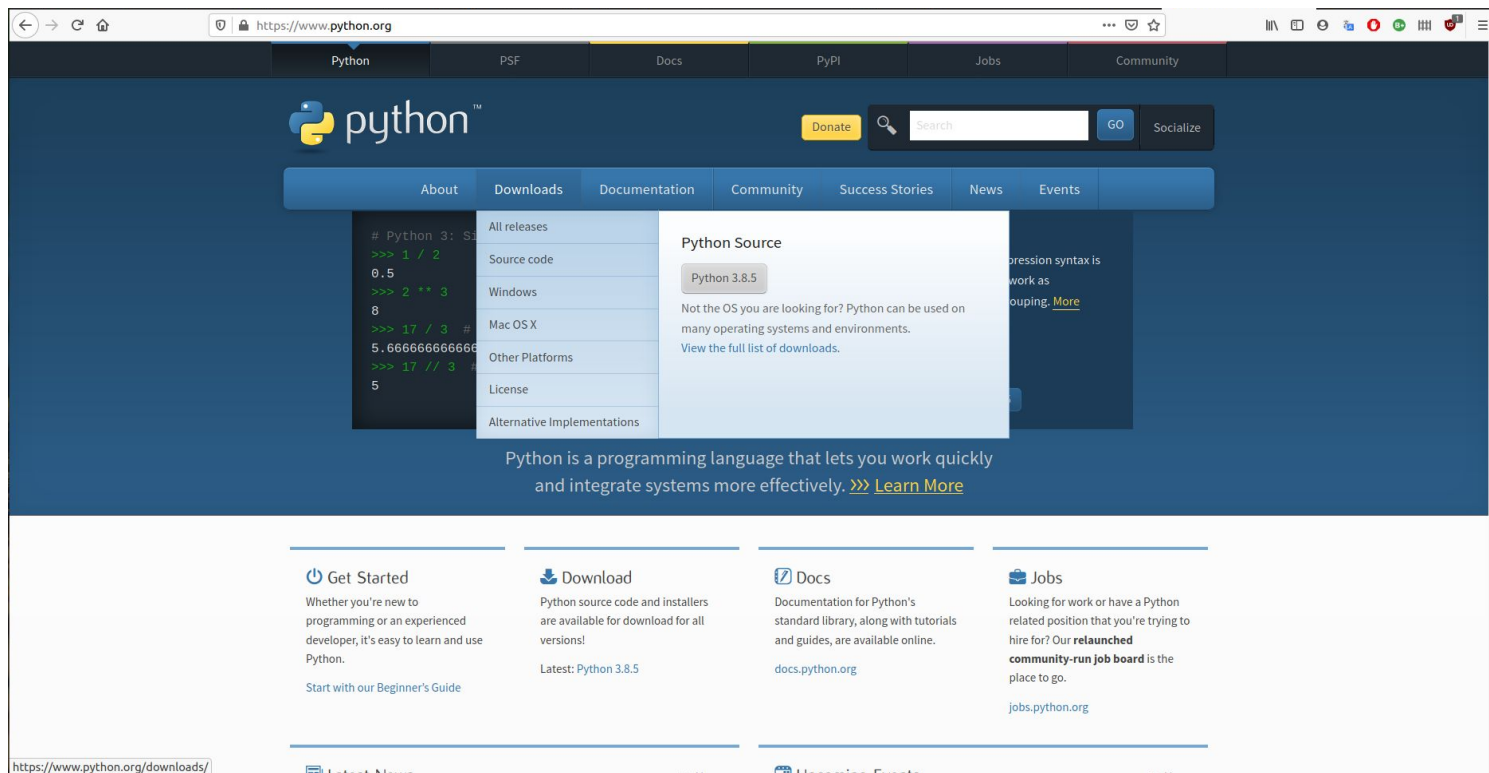
Начать писать на Python при понимании основных конструкций очень просто
⇒ Можно быстро и, следовательно, дешево, реализовать программное решение

Универсальность

Python - подходит для широкого круга задач, как задачи, связанные с машинным обучением, так и написание высоконагруженных сервисов.

- Instagram
- Youtube
- Google
- Facebook
-

Установка



<https://python.org>

Версии Python

- Python 2 вышел 2010 году последняя версия 2.7.16 - исправлялись только баги с января 2020 года поддержка прекращена.
- Python 3 появился в 2008, является актуальной версией языка. Текущая стабильная версия 3.9 в разработке 3.10
 - Python 3 не гарантирует совместимости кода с Python 2

НАСТОЯТЕЛЬНО РЕКОМЕНДУЮ РАБОТАТЬ С PYTHON >= 3.x

Что входит в стандартную библиотеку

Стандартная библиотека предоставляет большой список функций и типов данных, которыми можно оперировать не устанавливая при этом сторонние библиотеки. Полный список можно найти перейдя по ссылке <https://docs.python.org/3/library/>.

Введение

- запуск интерактивной строки интерпретатора
 - для Linux|MacOS в командной строке запуск `python` | `python3`
 - Windows обычно после установки `python` добавляет в системную переменную `PATH`

```
[22:36:03] serg :: serg-pc → ~»  
python3  
Python 3.8.2 (default, Jul 16 2020, 14:00:26)  
[GCC 9.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 
```

Интерактивная строка интерпретатора

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> import
```

Использование интерактивной строки интерпретатора

```
python
Python 3.8.10 (default, Jun 2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>> 1 + 1
2
>>> 10 + 1 - (7 - 2) * 3
-4
>>> a = 1
>>> b = 2
>>> c = a + b
>>> c
3
```

Используй

Среды разработки (IDE)

- Visual Studio Code <https://code.visualstudio.com/> - бесплатная IDE, мультязычная, много плагинов все бесплатные
- PyCharm - <https://www.jetbrains.com/pycharm/> - условно бесплатная для студентов доступна enterprise версия
- Spyder - <https://www.spyder-ide.org/> - бесплатная IDE
- Любой текстовый редактор (vim, nano, atom, notepad++...)

Практическая Часть

1. Проверить наличие python в вашей операционной системе
2. Если python установлен, выяснить какую версию интерпретатора вы используете, не стесняйтесь использовать internet
 - a. Если версия python < 3 перейти к шагу 3
 - b. Если python отсутствует перейти к шагу 3
3. Если python не установлен, установите наиболее свежую версию
4. Запустить интерактивную строку интерпретатора
5. Решить несколько простых задач
 - a. Вычислить значение выражения $1 + 2 \cdot 3$
 - b. Вычислить значение выражения $(1 + 2) \cdot 3$
 - c. Вычислить значение выражения $1 + 2/3$
 - d. Вычислить значение выражения $1 + 2/3 + 2$
 - e. Вычислить значение выражения $1 + 2/(3 + 2)$
 - f. Попробуйте разделить число на ноль, что выведется на экран