

# Лекция 1

Язык программирования Python введение.

Хайрулин Сергей Сергеевич

email: [s.khairulin@g.nsu.ru](mailto:s.khairulin@g.nsu.ru), [s.khayrulin@gmail.com](mailto: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

# 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 3.8.5 -> в предрелиз 3.9, в разработке 3.10
  - Python 3 не гарантирует совместимости кода с Python 2

# Установка

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Python Source

Python 3.8.5

Not the OS you are looking for? Python can be used on many operating systems and environments. [View the full list of downloads.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

**Get Started**  
Whether you're new to programming or an experienced developer, it's easy to learn and use Python.  
[Start with our Beginner's Guide](#)

**Download**  
Python source code and installers are available for download for all versions!  
Latest: Python 3.8.5

**Docs**  
Documentation for Python's standard library, along with tutorials and guides, are available online.  
[docs.python.org](#)

**Jobs**  
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.  
[jobs.python.org](#)

[Latest News](#) [Upcoming Events](#)

# 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++...)

# Программирование/Языки программирования

Процесс разработки программ, для вычислительных устройств с помощью языков программирования.

Язык программирования - 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

## Интерпретируемые

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

## Компилируемые

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

# 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

# Почему Python?

- Постоянно растущая популярность языка дает импульс 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

# 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.  
>>> 
```

## Использование интерактивной строки 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

```
import dis  
  
dis.dis('my_function(my_variable, 2)')  
0 LOAD_NAME          0 (my_function)  
2 LOAD_NAME          1 (my_variable)  
4 LOAD_CONST         0 (2)  
6 CALL_FUNCTION      2  
8 RETURN_VALUE
```

# Практическая Часть