

# Занятие 1. Установка языка Python и среды разработки

## Аннотация

Первый урок посвящен введению в язык Python и знакомству с теми техническими средствами, которые понадобятся нам для обучения.

## 1. Введение

**Язык программирования** — набор определенных правил, по которым компьютер может понимать команды (инструкции) и выполнять их. Текст программы на любом языке программирования, называется **программным кодом**

Языки программирования бывают **компилируемые** и **интерпретируемые**. Если программа написана на компилируемом языке (**C**, **C++**, **Pascal**), то перед выполнением её нужно полностью проверить на наличие синтаксических ошибок и уже после этого перевести в понятную для компьютера форму — машинный код. Это делает специальная программа, которая называется компилятором.

Если программа написана на интерпретируемом языке (**Python**, **PHP**, **Ruby**), она не переводится в машинный код целиком. Вместо этого специальная программа, которая называется интерпретатором, идет по коду, анализирует его и выполняет каждую отдельную команду.

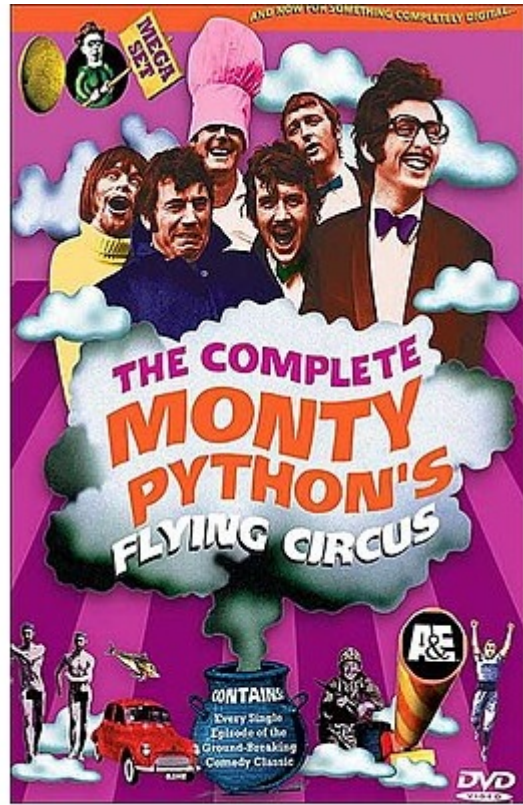
Существуют языки программирования, которые совмещают оба подхода (**C#**, **Java**). В таких языках код исходной программы сначала компилируется в промежуточный код (байт-код), а уже потом, во время выполнения, переводится в машинный код.

## Язык Python

Язык Python разработал голландский программист **Гвидо Ван Россум** (Guido van Rossum) в 1991 году. Гвидо был фанатом британского комедийного сериала «Monty Python's Flying Circus», откуда и пришло название языка.



Гвидо Ван Россум



Monty Python's Flying Circus

В настоящее время в русском языке для обозначения используют два варианта — «Питон» и «Пайтон».

Python относится к интерпретируемым языкам программирования: чтобы запустить написанную на Python программу, нужен интерпретатор Python (его можно скачать с сайта [python.org](https://www.python.org)). Подробнее об установке языка Python смотрите в видеоинструкции к уроку.

## 2. Установка и настройка

В этой части урока мы поможем вам с установкой Python 3.

### Windows

Перейдите по ссылке <https://www.python.org/downloads/> (<https://www.python.org/downloads/>) и скачайте установщик последней доступной версии Python3 для Windows.

Запустите установщик. На первом экране обязательно отметьте галочкой опцию **Add Python 3.9 to PATH** – это сделает Python 3 доступным в командной строке. Далее следуйте инструкциям, в процессе установки не снимайте галочки у предлагаемых для установки компонентов.

После установки Python 3 и программа IDLE будут доступны в меню Start (Пуск).

### MacOS

Мы опишем 2 возможных способа установить Python 3 на MacOS.

#### Способ 1:

Перейдите по ссылке <http://python.org/download/> (<http://python.org/download/>) и скачайте установщик последней доступной версии Python 3 для MacOS.

Запустите установщик, запустите его. Запустите Python.mpkg в открывшемся окне.

Во время установки вам нужно будет ввести административный пароль.

После установки в папке /Applications появится IDLE.

### **Способ 2:**

С помощью утилиты brew: [https://brew.sh/index\\_ru.html](https://brew.sh/index_ru.html) ([https://brew.sh/index\\_ru.html](https://brew.sh/index_ru.html))

Вам нужно установить brew, а затем набрать в терминале:

In [ ]:

```
brew install python3
```

После установки попробуйте запустить приложение Терминал (установлено по умолчанию) – и наберите python3 – должен запуститься интерактивный интерпретатор. Если команда python3 не запустила интерпретатор – попробуйте просто python.

## **Linux**

В различных дистрибутивах Linux способы установки Python 3 могут отличаться, вы можете собрать Python 3 из исходного кода. Но, например, на Ubuntu 16.10 установить Python 3 очень просто:

In [ ]:

```
$ sudo apt-get update  
$ sudo apt-get install python3.6
```

Если вы желаете собрать Python 3 самостоятельно, то вот пример сборки для Debian:

Из-под пользователя root:

In [ ]:

```
make install
```

Для других дистрибутивов собрать можно аналогично.

## **3. Выбор среды разработки (IDE)**

Писать код можно в любом текстовом редакторе (например, Блокнот), однако специализированные редакторы и полноценные среды разработки (IDE) добавляют процессу программирования массу удобств – таких как подсветка синтаксиса, автодополнение, интроспекция кода (возможность по клику перейти к месту объявления используемого класса или функции) и многие другие.

Если вы не уверены, в чем вам программировать – попробуйте PyCharm Community Edition. Это полноценная IDE. Чтобы установить PyCharm Community Edition, перейдите по [ссылке](https://www.jetbrains.com/ru-ru/pycharm/download/#section=windows) (<https://www.jetbrains.com/ru-ru/pycharm/download/#section=windows>) – вам будет предложено скачать на выбор Professional или Community версию среды разработки PyCharm. Professional версия – платная,

для полноценной работы с Python будет достаточно бесплатной Community версии. Скачайте установщик PyCharm Community Edition для вашей операционной системы. После этого запустите установщик и следуйте инструкциям.

Также можете посмотреть в сторону Visual Studio Code – для разработки на Python вам дополнительно потребуется установить плагин "Python".

Если вы программист с опытом, то возможно вам будет удобно писать код на Python в редакторах Vim или Emacs.

Среди других вариантов можно отметить редактор Atom, а также любимый многими за простоту и расширяемость Sublime Text.

Как вы можете видеть – редакторов и IDE много, выбирайте по вкусу. В видео подробно будет описан процесс создания нового Python проекта в PyCharm Community Edition – если не знаете какой редактор выбрать, советуем остановиться именно на нем.