

## Конспект

### Введение в программирование на языке Python

#### Введение

#### Что такое программирование?

Программирование это, по сути, написание инструкций для компьютера, итоговая совокупность которых представляет собой программу.

Компьютерные программы контролируют все: от работы смартфонов до полета космических ракет.

#### Знание нужного языка

Чтобы написать компьютерную программу, необходимо все понятные человеку действия разложить на простые и ясные для компьютера инструкции, выразив их на таком языке, который он понимает

#### Внимательно!

Компьютер подчиняется инструкциям слепо и буквально. он не может сам мыслить, так что все действия, выраженные не однозначно, не четко, остаются без исполнения.

**Python** — это язык программирования, который широко используется в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML).

Разработчики используют Python, потому что он эффективен, прост в изучении и работает на разных платформах.

Вас приятно удивит то, как легко можно сосредоточиться на решении поставленной задачи, а не на синтаксисе и структуре языка, на котором вы программируете.

### **Официально Python представляют так:**

Python – это простой в освоении и мощный язык программирования. Он предоставляет эффективные высокоуровневые структуры данных, а также простой, но эффективный подход объектно-ориентированному программированию. Его элегантный синтаксис и динамическая типизация наряду с тем, что он является интерпретируемым, делают его идеальным языком для написания сценариев и быстрой разработки приложений в различных областях и на большинстве платформ.

Далее мы рассмотрим эти особенности более детально.

#### **История названия**

Гвидо ван Россум, создатель языка Python, назвал его так в честь телешоу на BBC под названием «Летающий цирк Монти Пайтона»<sup>1</sup>, а вовсе не потому, что он любит змей.

### **Особенности python.**

#### **Простой:**

Python – простой и минималистичный язык. Чтение хорошей программы на Python очень напоминает чтение английского текста, хотя и достаточно строгого! Такая псевдо-кодовая природа Python является одной из его самых сильных сторон. Она позволяет вам сосредоточиться на решении задачи, а не на самом языке.

#### **Лёгкий в освоении:**

Как вы увидите, на Python чрезвычайно легко начать программировать. Python обладает исключительно простым синтаксисом, как уже отмечалось выше.

#### **Свободный и открытый:**

Python – это пример свободного и открытого программного обеспечения – FLOSS (Free/Libre and Open Source Software). Проще говоря, вы имеете право свободно распространять копии этого

программного обеспечения, читать его исходные тексты, вносить изменения, а также использовать его части в своих программах. В основе свободного ПО лежит идея сообщества, которое делится своими знаниями. Это одна из причин, по которым Python так хорош: он был создан и постоянно улучшается сообществом, которое просто хочет сделать его лучше.

### **Язык высокого уровня:**

При написании программы на Python вам никогда не придётся отвлекаться на такие низкоуровневые детали, как управление памятью, используемой вашей программой, и т.п.

### **Портируемый:**

Благодаря своей открытой природе, Python был портирован на много платформ (т.е. изменён таким образом, чтобы работать на них). Все ваши программы смогут запускаться на любой из этих платформ без каких-либо изменений, если только вы избегали использования системно-зависимых функций. Python можно использовать в GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE и даже на PocketPC! Вы можете даже использовать такую платформу, как Kivu для создания игр для iOS (iPhone, iPad) и Android. 4.1.6

### **Интерпретируемый:**

Это требует некоторого пояснения. Программа, написанная на компилируемом языке программирования, как например, C или C++, преобразуется из исходного языка (т.е. C или C++) в язык, понятный компьютеру (бинарный код, т.е. нули и единицы) при помощи компилятора с применением разнообразных флагов и параметров. Когда вы запускаете такую программу, компоновщик/загрузчик копирует программу с диска в оперативную память и запускает её. Python же, напротив, не требует компиляции в бинарный код. Программа просто выполняется из исходного текста. Python сам преобразует этот исходный текст в некоторую промежуточную форму, называемую байткодом, а затем переводит его на машинный язык и запускает. Всё это заметно облегчает использование Python,

поскольку нет необходимости заботиться о компиляции программы, подключении и загрузке нужных библиотек и т.д. Вместе с тем, это делает программы на Python намного более переносимыми, так как достаточно их просто скопировать на другой компьютер, и они работают!

### **Объектно-ориентированный:**

Python поддерживает как процедурно-ориентированное, так и объектно ориентированное программирование. В процедурно-ориентированных языках программы строятся на основе процедур или функций, которые представляют собой просто-напросто многократно используемые фрагменты программы. В объектно-ориентированных языках программирования программы строятся на основе объектов, объединяющих в себе данные и функционал. Python предоставляет простые, но мощные средства для ООП, особенно в сравнении с такими большими языками программирования, как C++ или Java. 4.1.8

### **Расширяемый**

Если вам нужно, чтобы некоторая критическая часть программы работала очень быстро или вы вынуждены скрыть часть алгоритма, вы можете написать эту часть программы на C или C++, а затем вызывать её из программы на Python.

### **Встраиваемый:**

Python можно встраивать в программы на C/C++, чтобы предоставлять возможности написания сценариев их пользователям.

### **Обширные библиотеки:**

Стандартная библиотека Python просто огромна. Она может помочь в решении самых разнообразных задач, связанных с использованием регулярных выражений, генерированием документации, проверкой блоков кода, распараллеливанием процессов, базами данных, веб-браузерами, CGI, FTP, электронной почтой, XML, XML-RPC, HTML, WAV файлами, криптографией, GUI (графическим интерфейсом пользователя) и другими системно зависимыми вещами. Помните, что всё это доступно абсолютно везде, где

установлен Python. В этом заключается философия Python «Всё включено». Кроме стандартной библиотеки, существует множество других высококачественных библиотек, которые можно найти в Каталоге пакетов Python

## **Резюме**

Python – очень увлекательный и мощный язык. Он имеет хорошее соотношение производительности и возможностей, что делает написание программ на нём одновременно интересным и лёгким.

## **3. Установка VSC и Python**

Для установки Python в Visual Studio Code (VSC) выполните следующие шаги:

### **Шаг 1: Установите Python на ваш компьютер:**

Перейдите на официальный веб-сайт Python (<https://www.python.org/>) и загрузите последнюю версию Python для вашей операционной системы.

Запустите загруженный установщик Python и следуйте инструкциям по установке. Убедитесь, что установщик Python добавляет Python в переменную среды PATH, чтобы вы могли запускать Python из командной строки.

### **Шаг 2: Установите Visual Studio Code:**

Перейдите на официальный веб-сайт Visual Studio Code (<https://code.visualstudio.com/>) и загрузите установщик для вашей операционной системы.

Запустите загруженный установщик и следуйте инструкциям по установке, чтобы установить Visual Studio Code на ваш компьютер.

### **Шаг 3: Установите расширение Python для Visual Studio Code:**

Запустите Visual Studio Code.

Нажмите на иконку в левой боковой панели, чтобы открыть панель расширений (или используйте комбинацию клавиш Ctrl+Shift+X). В поисковой строке введите "Python". Найдите расширение "Python" от Microsoft и нажмите на кнопку "Установить".

#### **Шаг 4: Настройте путь к интерпретатору Python:**

Откройте Visual Studio Code.

Нажмите на кнопку с виньеткой в левом нижнем углу окна, чтобы открыть командную палитру (или используйте комбинацию клавиш Ctrl+Shift+P).

Введите "Python: Select Interpreter" и выберите эту опцию.

Если Python был успешно установлен и добавлен в переменную среды PATH, вы должны увидеть список доступных интерпретаторов Python. Выберите нужную версию Python.

Теперь вы должны быть готовы к работе с Python в Visual Studio Code.

#### **4. Вывести первый "привет, мир" показать как сохранять и создавать файл.**

По традиции, сложившейся в мире программистов, самой первой программой должна быть «Hello world!» (Привет, мир!). Именно эту строчку должен вывести на монитор компьютер, что в языке Python проще простого.

#### **Теперь создадим новый файл, пропишем код и выведем сообщение.**

Создадим папку на рабочем столе. Внутри папки нажмем правой кнопкой и открыть с помощью Code.

#### **Функции:**

В языке Python команда print(), или печать, это функция, а по сути она является миниатюрной, заранее написанной программой. Чтобы ее запустить, вам достаточно вписать в кавычки слово. Функция

`print()` делает так, что компьютер показывает на мониторе все, что вы печатаете в кавычках

пишем `print("Hello, world")` и запускаем через терминал и просто кнопкой.

### **Переменные:**

Чтобы создать переменную достаточно написать что-либо, допустим это будет `a`.

Переменная хранит в себе значения которые используются в коде в дальнейшем.

Чтобы поместить в нее значения достаточно просто поставить равно, добавить скобки и кавычки.

### **Ввод данных:**

Функция `input()`. За ввод в программу данных с клавиатуры в Python отвечает функция `input`. Когда вызывается эта функция, программа останавливает свое выполнение и ждет, когда пользователь введет текст.

Показать ввод и вывод простейших переменных.

## **5. Рассказать про вывод при помощи f строк (для ДЗ).**

### **ПОГРУЖЕНИЕ В F-СТРОКИ**

f-строки делают очень простую вещь — они берут значения переменных, которые есть в текущей области видимости, и подставляют их в строку. В самой строке вам лишь нужно указать имя этой переменной в фигурных скобках.

```
>>> name = "Дмитрий"
>>> age = 25
>>> print(f"Меня зовут {name} Мне {age} лет.")
>>> Меня зовут Дмитрий. Мне 25 лет.
```

## **6. Показать и рассказать про комментарии в коде.**

### **Комментирование Python кода**

Программирование отражает ваш образ мышления, чтобы описать отдельные шаги, которые вы предприняли для решения проблемы с помощью компьютера. Комментирование вашего кода помогает объяснить ваш мыслительный процесс, а также помогает вам и другим людям понять смысл вашего кода. Это позволяет вам легче находить ошибки, исправлять их, впоследствии улучшать код, а также повторно использовать его и в других приложениях.

Комментирование важно для всех видов проектов, независимо от того, маленькие они, средние или довольно большие. Это важная часть вашего рабочего процесса, и считается хорошей практикой для разработчиков. Без комментариев все может запутаться, очень быстро. В этой статье мы расскажем о различных методах комментирования, поддерживаемых Python, и о том, как его можно использовать для автоматического создания документации для вашего кода с использованием так называемых строк документации уровня модуля.

#### **Хорошие против плохих комментариев**

Как бы ни были важны комментарии, все еще можно писать плохие комментарии. Они всегда должны быть короткими, прямолинейными и добавлять информативную ценность.

Хорошее практическое правило - добавлять комментарии к любой строке кода (например, к списку) или к фрагменту кода, цель которого не очевидна. Это очень субъективно, и на самом деле это навык, который необходимо изучить.

#### **Однострочные комментарии**

Такой комментарий начинается с хеш-символа ( #) и сопровождается текстом, который содержит дополнительные пояснения.



```
# defining the post code  
postCode = 75000
```

Вы также можете написать комментарий рядом с оператором кода. Следующий пример показывает, это:

```
# define the general structure of the product with default values  
product = {  
    "productId": 0,      # product reference id, default: 0  
    "description": "",   # item description, default: empty  
    "categoryId": 0,     # item category, default: 0  
    "price": 0.00        # price, default: 0.00  
}  
  
# -----  
# demonstrates how to write ms excel files using python-openpyxl  
#  
# (C) 2015 Frank Hofmann, Berlin, Germany  
# Released under GNU Public License (GPL)  
# email frank.hofmann@efho.de  
# -----
```

Написание правильных комментариев в вашем коде Python не так сложно, и вам просто нужна сила выносливости. Это помогает всем, кто пытается понять ваш код, включая вас самих, когда вы в следующий раз вернетесь к нему. Я надеюсь, что совет, который дали вам здесь, облегчит вам создание более качественных комментариев и документации в вашем коде.

## **7. Числа.**

Числа в Python бывают трёх типов: целые, с плавающей точкой и комплексные.

- Примером целого числа может служить 2.
- Примерами чисел с плавающей точкой (или «плавающих» для краткости) могут быть 3.23 и 52.3E-4.

Обозначение E показывает степени числа 10. В данном случае 52.3E-4 означает  $52.3 \cdot 10^{-4}$ .

- Примеры комплексных чисел:  $(-5+4j)$  и  $(2.3 - 4.6j)$

## 8. Строки.

Строка – это последовательность символов. Чаще всего строки – это просто некоторые наборы слов. Слова могут быть как на английском языке, так и на любом другом, поддерживаемом стандартом Unicode, что означает почти на любом языке мира

Я могу с уверенностью сказать, что вы будете использовать строки почти в каждой вашей программе на Python. Поэтому уделите внимание тому, как работать со строками в Python.

## 9. Кавычки.

### 7.4.1 Одинарные кавычки

Строку можно указать, используя одинарные кавычки, как например, 'фраза в кавычках'. Все пробелы и знаки табуляции сохраняются, как есть.

### 7.4.2 Двойные кавычки

Строки в двойных кавычках работают точно так же, как и в одинарных. Например, "what's your name?".

### 7.4.3 Тройные кавычки

Можно указывать «многострочные» строки с использованием тройных кавычек (""" или '''). В пределах тройных кавычек можно свободно использовать одинарные и двойные кавычки. Например:

```
'''Это многострочная строка. Это её первая строка.
Это её вторая строка.
"what's your name?", - спросил я.
Он ответил: "Bond, James Bond."
'''
```

## Доп материалы.

Скачивание Python:

<https://www.python.org/downloads/>

Скачивание Visual Studio Code:

<https://code.visualstudio.com/Download>

Документация по установке:

<https://learn.microsoft.com/ru-ru/training/modules/python-install-vscode/>