

Лабораторная работа 1 - Установка и настройка среды. Основы синтаксиса

Где применяется Python?

На Питоне можно делать много чего: приложения под ПК, нейросети, веб сайты и даже игры.

Если говорить про наиболее частое использование языка, то это такие сферы как: веб разработка, работа с данными, машинное обучение и нейронные сети, разработка ботов и написание скриптов под разные системы.

Несколько основных функциональных достоинств Python:

- Умеет работать с расширениями xml/html;
- Поддерживает управление http-запросами;
- Обладает графическим интерфейсом;
- Помогает создавать веб-скрипты;
- Может работать с FTP;
- Способен оперировать картинками, видео и звуковыми файлами;
- Используется в разработке робототехники;
- Отвечает за создание научных, вычислительных площадок и ещё много всего другого.

Можно сказать, что Python отлично подходит для выполнения большей части ежедневных задач программистов. Участвует как в создании обычных бекапов или чтении писем с почты, так и в разработке крупных игр. Питон практически не имеет ограничений по сферам использования, что позволяет использовать его в проектах колоссальных масштабов.

Python часто применяется китами мира IT – Google и Яндекс. Если добавить к этому простоту создания программ, несложно понять, почему Питон заслуживает место в списке лучших языков.

Перед изучением языка Python сперва необходимо выполнить установку среды разработки и установку самого Питона на компьютер. В ходе видео мы установим Питон и программу PyCharm для написания кода в будущих уроках.

- Установка Python и IDE (Integrated Development Environment)
<https://www.python.org>

Язык Python будет готов к использованию сразу же после установки. Вам останется лишь протестировать его работоспособность.

Во вкладке «Пуск» должна появиться новая папка, в которой можно запустить IDLE. Если такой папки нет, то попробуйте найти программу вручную.

IDLE - это встроенная среда разработки, в которой можно работать с языком Питон. Также вы можете проверить работоспособность в консоле (терминале). Для этого

Введение в программирование Python

откройте консоль и введите «python3». Если ничего не произошло, тогда попробуйте ввести просто слово «python» и нажать Enter.

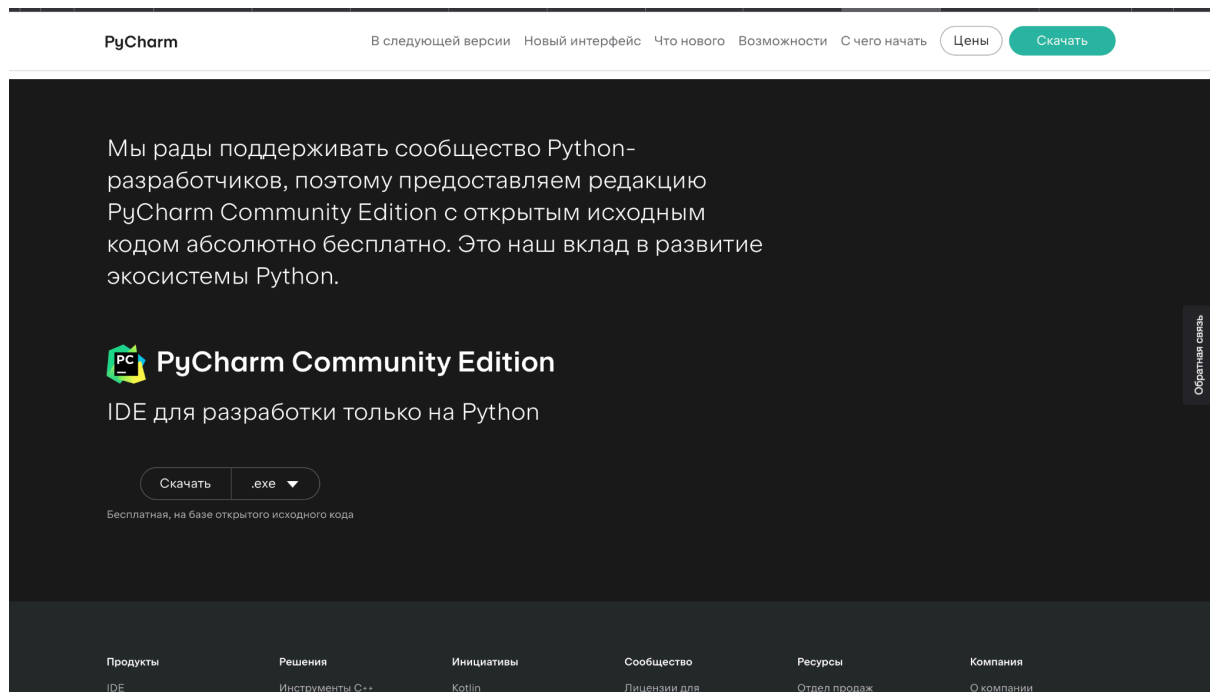
Среда разработки

В качестве среды разработки вы можете использовать любую подходящую программу. В ходе курса мы будем использовать одну из наиболее надежных программ – PyCharm. Если вам больше по душе другая программа, то можете использовать её. Главное необходимо убедиться что ваша программа имеет поддержку языка Питон.

Указание пути к файлу

Чтобы указать путь к файлу на Windows необходимо использовать пример следующей конструкции: «C:\Users\USER\Downloads\ФАЙЛ.py».

- Установка PyCharm Community Edition
- <https://www.jetbrains.com/ru-ru/pycharm/download/?section=windows>



Переменные и типы данных

Переменные невероятно важны, так как позволяют хранить информацию и использовать её в дальнейшем. Вначале может быть не совсем понятно зачем вообще что-то записывать в переменную, если можно просто оперировать значениями без них. Понимание переменных придет немного позже, когда мы начнем создавать более сложные программы и нам потребуется хранить информацию в каком-либо месте.

Типы переменных в языке Python не объявляются очевидно, тем не менее они присутствуют. Интерпретатор понимает что записывается в переменную и на основании этого добавляет тип к этой переменной.

Во время выполнения программы есть возможность перезаписывать переменные, а также менять их тип. Если вначале переменная была с типом float, то потом её можно преобразовать в другой тип, к примеру, в string.

Введение в программирование Python

```
first_num = 23.2 # Тип данных float
```

```
first_num = "1" # Тип данных string
```

При объединение нескольких переменных с разными типами данных программа спровоцирует ошибку.

Пример:

```
first_num = "IloveYou"
```

```
second_num = 13
```

```
res = first_num + second_num # Скрипт выдаст ошибку
```

Всего в Python есть 4 базовых типа переменных:

- some = 1 Integer - целые числа;
- some = 1.12 Float - числа с плавающей точкой;
- some = "Привет" String - строки;
- some = True Boolean - тип данных принимающий либо False, либо True.

Есть и другие типы, но мы будем их разбирать в последующих уроках.

В одной строке можно создать сразу несколько переменных:

```
first = sec = third = 1 # Всем трём переменным будет присвоено значение 1
```

```
first, sec, third = "Hi", 75, 23.1 # Поочередное присвоение значений
```

Переменные и работа с ними, пример:

```
number = 5 # int
```

```
digit = -4.54356876 # float
```

```
word = "Результат:" # string
```

```
boolean = True # bool
```

```
str_num = '5' # string
```

```
# print(word + str(digit))
```

```
print(word + str(number + int(str_num)))
```

```
del number
```

```
number = 7
```

```
print("Результат:", number)
```

Получение данных, пример:

```
num1 = int(input("Введите первое число: "))
```

```
num2 = int(input("Введите второе число: "))
```

```
num1 *= 5
```

```
print("Result:", num1 + num2)
```

```
print("Result:", num1 - num2)
```

```
print("Result:", num1 / num2)
```

```
print("Result:", num1 * num2)
```

```
print("Result:", num1 ** num2)
```

```
print("Result:", num1 // num2)
```

Введение в программирование Python

```
word = "Hi"  
print(word * 2)  
word = True
```

Вывод информации, пример:

Для вывода информации используется функция `print`. Она принимает строку в качестве параметра и выводит её в терминале (консоли). Пример реализации функции:

```
print("Hello World")
```

#Пример

```
print("Результат:", 5 // 3)  
print("Результат:", max(5, 6, 3, 0, 2, -3, 5))  
print("Результат:", round(5 / 3))  
print("Результат: ", 7, 15, sep="", end="\n")  
print('Second \' \'t \' \'ni\'nn\'ne')  
input("Введите свой возраст: ")
```

Правильный запуск, пример:

- Создайте файл и запустите его через терминал. При запуске файла в терминале должно быть выведено сообщение: «Python is working».
- Создайте пустой Python файл «file.py». Чтобы запустить файл пропишите команду «python file.py» или же «python3 file.py».
- В файле пропишите код:
- `print("Python is working")`

Текстовые надписи, пример:

- Отобразите на экран следующую фразу: «Специальные символы (\, ", &, \n)».
- Важно чтобы все символы отобразились при запуске проекта.
- Решение задания:
- `print("Специальные символы (\\, \", &, \\n)")`

Самостоятельная работа 1

★ **Подготовьте отчет с скриншотами вашего исходного кода и с описанием алгоритма решения задачи в WORD!!!**

1. Нахождение числа

Создайте программу, которая будет принимать число (n), введенное пользователем, и выдавать результат в виде $(n + n * 2)$.

Пример:

Вводим 23

Получаем: 2346

2. Работа с переменными

Создайте переменную со значением 46 и переменную со значением "string".

Последнюю переменную умножьте на 5.

Выведите на экран обе переменные.

3. Простые переменные

Создайте переменные со значениями: 5, Е, Привет, 90.2.

Создайте переменную, которую нельзя будет изменить и установите ей значение 67.

Выведите переменную со значением "Привет" на экран.

4. Разделение числа на символы

Напишите программу, которая будет получать от пользователя число с 4 числами.

Реализуйте разделение этого числа на отдельные цифры.

Пример:

Число 5934

Результат 5, 9, 3, 4

Важно: число вводит пользователь с клавиатуры.

5. Получение данных от пользователя

Создайте программу, что будет запрашивать данные пользователя: имя, фамилию, возраст и выводить их на экран.

6. Получение данных

Выведите надпись: «Как зовут вашего друга?».

Получить значение от пользователя и выведите его в консоль.

7. Простая математика

Выясните сколько семерок в числе 136.

8. Математические операции

Создайте программу, которая будет запрашивать три переменные у пользователя и после их получения выводить на экран математически операции над ними:

- сумма
- вычитание
- деление
- умножение
- остаток при делении

9. Типы переменных

Создайте три переменных со следующими типами данных:

- число
- число с точкой
- число в формате строки

Выполнить перемножение между данными переменными

Важно: вам потребуется приводить типы данных к одному общему, дабы выполнить это задание.