

Введение в программирование на Python

Цель занятия

После освоения темы вы:

- узнаете принципы организации кода;
- установите интерпретатор Python на компьютер;
- установите среду разработки PyCharm;
- напишите простой код на Python.

План занятия

1. О языке Python.
2. Установка интерпретатора Python на Windows.
3. Установка интерпретатора Python на Linux.
4. Работа в IDE PyCharm. Первая программа.
5. Введение в Python.
6. Ввод и вывод данных.
7. Примеры простейших программ.

Используемые термины

Компьютерная программа (далее — программа) — совокупность команд на понятном компьютеру языке для выполнения некоторых функций.

Язык программирования — набор правил для написания программы.

Интерпретатор — программа, которая выполняет код, написанный на языке программирования, например, интерпретатор Python позволяет писать программу в интерактивном режиме.

Инструкция — команда, определяющая, какие операции программа выполнит над данными.

Переменная — именованная область оперативной памяти, в которой хранятся некоторые данные определённого типа.

Конспект занятия

1. О языке Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

Python является одним из первых языков, в которых упор сделан на его простоту: язык построен таким образом, чтобы программный код было удобно читать.

Для более подробного ознакомления с Python существуют два сайта:

- официальный сайт (<https://www.python.org/>);
- документация (<https://docs.python.org/3/>).

В рейтингах языков программирования Python занимает одно из первых мест (например, в [рейтинге TIOBE Programming Community](#)).

Python поддерживает все современные парадигмы программирования:

- структурное;
- объектно-ориентированное (ООП);
- функциональное;
- императивное;
- аспектно-ориентированное.

Основные архитектурные черты языка:

- динамическая сильная неявная типизация;
- автоматическое управление памятью;
- полная интроспекция;
- механизм обработки исключений;
- поддержка многопоточных вычислений;
- удобные высокоуровневые структуры данных.

Запуск программ происходит следующим образом (рисунок 1): код записывается в среде разработки, затем с помощью интерпретатора выполняется запуск программы на компьютере. Таким образом, интерпретатор представляет собой переводчик между языком программирования и компьютером. После обработки в интерпретаторе программа выполняется на компьютере.



Рисунок 1. Интерпретатор Python.

Особенности Python:

- Python — самый стильный язык программирования в мире;
- Python не допускает двоякого написания кода;
- в Python код можно написать только одним способом;
- в Python отсутствуют лишние конструкции;
- более читаемого кода нет ни в одном другом языке программирования.

Язык программирования Python достаточно молодой, на сентябрь 2022 года актуальной является версия 3.10. До этого существовали версии 1.X и 2.X, которые не поддерживаются. В Python не сформировалась обратная совместимость версий языка. То есть, если программа написана на языке версии 2, она может не запуститься на интерпретаторе версии 3. И наоборот.

Язык применяется во всей индустрии ИТ и во многих ИТ-гигантах: Яндекс, VK, Google и другие. В образовании Python получил наибольшее распространение в США, Python изучают в школах и университетах. Часто Python рекомендуют как первый язык программирования.

Плюсы языка Python:

- открытый код,
- простота синтаксиса,
- наличие мощных библиотек,
- многоплатформенность,
- бесплатность,
- проста в изучении,
- многочисленное сообщество поддержки.

Минусы языка Python:

- невысокая скорость выполнения программ,
- медленная эволюция языка.

Для написания программы удобно использовать интегрированную среду разработки – IDE (англ. Integrated Development Environment). Примеры IDE для написания кода на Python:

- [IDLE](#)
- [PyDev](#)
- [Aptana](#)
- [Wing](#)
- [PyCharm](#)
- [Eclipse](#)
- [Emacs](#)
- [Notepad++](#)

2. Установка интерпретатора Python на Windows

Для начала нужно скачать интерпретатор Python с [официального сайта](#). Скачивать интерпретатор с других ресурсов не рекомендуется, на них могут быть вирусы.

На сайте python.org выберите в меню *Download*→*Download for Windows*, далее нажмите на кнопку «Python 3.X.X».

По окончании загрузки установщика, запустите его двойным щелчком мыши, чтобы начать установку Python.

Обязательно установите флажок на «Add Python X.Y to PATH» в мастере настройке, в противном случае возможны проблемы с доступом инсталлятора Python к командной строке.

Нажмите «Установить» для начала установки. Установка должна пройти быстро, после чего Python будет готов к запуску на вашей системе. К следующему шагу нужно убедиться, что все было выполнено надлежащим образом.

По окончании установки, Python должен быть установлен на вашем компьютере. Чтобы убедиться в том, что все было выполнено правильно, можно протестировать, имеет ли Python доступ к командной строке (cmd) Windows.

Для этого откройте командную строку Windows, запустив `cmd.exe`. Введите команду `pip` и нажмите «Enter». Если всё было выполнено верно, вы увидите справочный текст менеджера пакетов Python под названием «pip». Если вы получите уведомление об ошибке, повторите этапы установки Python, и убедитесь в том, что вы обладаете рабочей версией Python.

Большая часть проблем, с которыми вы можете столкнуться, будут иметь то или иное отношение к неправильной настройке PATH. Переустановка и подтверждение того, что опция «Add Python to PATH» была активирована, должны исправить эту проблему.

3. Установка интерпретатора Python на Linux

Мы рассмотрим наиболее универсальный метод установки — установка через терминал операционной системы.

Для этого необходимо запустить сам терминал и в командной строке выполнить команду для обновления установщика пакетов — `apt update`. Будет выполнена установка некоторых пакетов. Если возникли предупреждения, проигнорируйте их, это достаточно стандартный вывод.

Далее для установки самого интерпретатора необходимо выполнить команду: `sudo apt install python3.8`. Последние цифры в команде соответствует номеру устанавливаемой версии интерпретатора Python.

Для того, чтобы убедиться в корректности установки, достаточно выполнить команду: `python3 -V`. В результате будет выведена версия установленного интерпретатора.

4. Работа в IDE PyCharm. Первая программа

PyCharm — среда, используемая для написания программ на Python. В IDE есть инструменты для анализа кода, графическая отладчик, встроенное модульное тестирование и поддержка веб-разработки.

Для установки PyCharm сначала нужно установить интерпретатор Python.

Чтобы установить IDE PyCharm перейдите на [сайт](#) среды разработки. Далее на главной странице по нажатию на кнопку «Download» перейдите на страницу загрузки PyCharm.

Дальше необходимо выбрать версию PyCharm:

- Professional — платная версия с полным набором функций. Она идеально подходит для профессиональной разработки.
- Community — бесплатная версия, ею можно пользоваться благодаря набору базовых возможностей. Для курса будет достаточно возможностей версии Community.

После нажатия на кнопку «Скачать» скачивание начнется автоматически.

Далее необходимо запустить установку, в окне установщика нажмите кнопку «Next».

При установке есть возможность выбрать папку для установки PyCharm, можно оставить папку установки по умолчанию.

В окне параметров установки поставьте галочки напротив пунктов:

- *PyCharm Community Edition*
- *Add "Open Folder as Project"*
- *Add "bin" folder to the PATH*
- *.py*

Далее запускаем установку, нажав кнопку «*Install*». В конце установки программа может попросить перезагрузить компьютер.

После запуска PyCharm, если вы только что установили IDE и запускаете ее в первый раз, может высветиться окно с предложением установки или импорта предыдущей конфигурации. Можно оставить настройки в появившемся окне по умолчанию.

При первом запуске PyCharm есть возможность открыть существующий проект или создать новый. Поскольку мы только начинаем изучение Python, выбираем создание нового проекта. Затем выбираем директорию для сохранения проекта и нажимаем кнопку «*Create*».

После выполнения указанных действий откроется окно для написания программы.

Далее можно настроить среду разработки, например, сменить тему. Для этого переходим в главное меню: *File*→*Settings*. В появившемся окне на вкладке *Appearance* изменим тему, например, для светлой темы можно выбрать *IntelliJ Light*.

При создании нового проекта в окне PyCharm создается файл `main.py`, в котором уже содержится некоторая информация: быстрые клавиши и простейшая программа.

Важно! В правом нижнем углу окна программы PyCharm должно появиться название интерпретатора с указанием его версии. Наличие указанной строки означает, что вы сделали все верно, среда разработки видит и может работать с интерпретатором Python. Если название интерпретатора отсутствует, необходимо нажать мышью на область, где должна быть записана строка с информацией о нем. Так вы перейдете в настройки. Далее в настройках укажите интерпретатор вручную. Также возможна полная переустановка интерпретатора и среды разработки.

Используйте этот код, чтобы проверить работу PyCharm:

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

Для запуска написанной программы в любом месте кликните правой кнопкой мыши и выберите «*Run*». Программа выполнится, а вывод будет показан на экране. Это значит, что PyCharm установлен и готов к работе.

5. Введение в Python

Программа (проект) на языке Python представляет собой несколько модулей — отдельных файлов, содержащих код на языке Python.

Сам **код** — набор инструкций, то есть последовательность определенных команд, которые выполняются интерпретатором. Самые простые программы содержат один модуль. Более сложные программы содержат несколько модулей. Модули могут быть связаны между собой, а могут выполнять не зависящие друг от друга функции.

То есть **модуль** — ряд связанных между собой операций.

Инструкция — указание компьютеру, определяющие, какие операции компьютер выполнит над данными.

Инструкции (или команды) могут быть простые и составные:

- простые состоят из одной строки кода,
- составные содержат вложенные инструкции.

Выражения в составе инструкций определяют, над какими именно данными будут выполнены действия, описанные в инструкции.

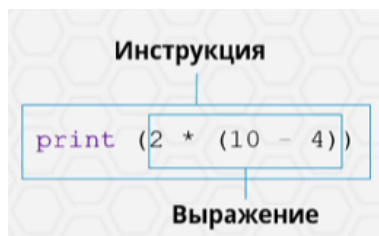


Рисунок 2. Пример команды, содержащей инструкцию.

Операции — любые действия над операндами.

Операнды — некоторые данные:

- литералы,
- выражения,
- переменные.



Рисунок 3. Пример литералов в выражении.

Важно! Приоритет выполнения операций соответствует принятому в математике.

Переменная — именованная область оперативной памяти, в которой хранятся некоторые данные определённого типа. Тип данных может быть: целые и вещественные числа, строки и другие.

При определении переменной резервируется ячейка оперативной памяти, и в нее записывается значение переменной. Пример присвоения значения переменной:

```
a = 25
```

Важно! При задании имени переменной можно использовать только ряд допустимых символов:

- заглавные и строчные буквы английского алфавита (A–Z, a–z),
- цифры (0–9),
- знак подчеркивания (_).

Не допускается имя переменной, начинающееся с цифры. Также имена переменных чувствительны к регистру: `name` и `Name` — это две разные переменные.

В Python есть «красивый» способ обмена переменных значениями, который отличает его от других языков программирования:

```
a, b = b, a
```

В результате такого кода переменные обменяются своими значениями.

Особенности синтаксиса языка Python:

- не содержит операторных скобок (`begin..end` в Pascal или `{..}` в C),
- блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием,
- однострочные комментарии начинаются со знака фунта `#`, многострочные — начинаются и заканчиваются тремя двойными кавычками `"""`,
- чтобы присвоить значение переменной используется знак `=`, а для сравнения — `==` (два знака равенства),
- для увеличения значения переменной, или добавления к строке используется оператор `+=` (знаки плюс и равно), а для уменьшения — `-=` (знаки минус и равно).

6. Ввод и вывод данных

Ввод и вывод данных рассмотрим на примере задачи о сложении двух чисел: пользователь вводит в консоль два числа, компьютер сам считает их сумму, результат также выводится в консоль.

Для ввода значения переменной с клавиатуры используется функция `input()`. То есть программа ждет, пока пользователь введет значение и нажмет *Enter*. После чего

введенное значение записывается в переменную, или как говорят, связывается с именем переменной.

Важно! Результат функции `input()` – строка символов вне зависимости от содержимого ввода. Чтобы преобразовать вводимые с клавиатуры символы в целое число, надо указать тип вводимой переменной: `a = int(input())`.

При использовании функции `input()` возможно использование подсказки:

```
a = int(input("Введите число: "))
```

Для вывода значения переменной используется функция `print()`:

```
print(a) #значение переменной  
print("Ответ:", a) #значение и текст
```

Возможно более сложное использование функции `print()` – одновременный вывод текста и нескольких переменных:

```
print("Ответ:", a+b)  
print(a, "+", b, "=", c)
```

Параметр `sep` в функции `print()` позволяет настроить работу разделителей. Например, чтобы «очистить» вывод от пробелов используйте команду:

```
print(a, "+", b, "=", c, sep="")
```

Пример программы сложения двух чисел.

```
a = int(input())  
b = int(input())  
c = a + b  
print(c)
```

Важно! Результат вычислений суммы записан в переменную `c`, то есть под результат выделена отдельная ячейка памяти. В дальнейшем результат сложения (переменная `c`) может использоваться в программе.

Пример программы сложения двух чисел (более удобный для пользователя вариант):

```
print("Введите два числа:")  
a = int(input())  
b = int(input())  
c = a + b  
print(a, "+", b, "=", c, sep="")
```

Команды для ввода и вывода данных:

- ввод данных (пользователь должен что-то напечатать на клавиатуре) — `input()`,
- вывод данных (программа должна что-то показать на экране) — `print()`.

Важно! Избегайте использования пробелов сразу перед открывающей скобкой, после которой начинается список аргументов функции.

- Правильно: `print("Привет!")`
- Неправильно: `print ("Привет!")`

В среде разработки PyCharm код программы записывается в верхнем окне, а консоль для ввода-вывода располагается в нижней части. Также в консоль выводится сообщения об ошибках.

7. Примеры простейших программ

Программа «повторения» сводится к повтору введенного пользователем текста.

Для того, чтобы программа считала строку с консоли используем команду `input()`. Запишем результат в переменную

```
t = input()
```

Так как в команде `input()` можно использовать подсказку, дополним ее

```
t = input("Введите текст")
```

Для вывода используем команду `print()`

```
print(t)
```

Мы можем изменить программу, сделав чтобы введенный текст повторялся дважды. В этом случае нужно еще раз повторить команду `print(t)`.

Текст программы «повторения» полностью:

```
t = input("Введите текст")
print(t)
print(t)
```

Программа «приветствие пользователя» запрашивает имя пользователя, а затем приветствует пользователя по введенному имени.

Будем использовать для вывода команду

```
print("Как вас зовут?")
```

После вывода запрашиваем имя и записываем его в переменную `Name`

```
Name = input()
```

Снова используем команду, на этот раз выведем имя

```
print("Рады Вас видеть,", Name, "!")
```

Текст программы «приветствие пользователя» полностью:

```
print("Как вас зовут?")
```

```
Name = input()
```

```
print("Рады Вас видеть,", Name, "!")
```

Дополнительные материалы для самостоятельного изучения

1. [Официальный сайт Python](#) (англ.)
2. [Среда разработки IDE PyCharm](#) (англ.)
3. [Интерпретатор, интерактивная оболочка](#) (англ.)