



Урок 1.

Введение в язык программирования Python. Установка среды разработки

История языка Python	2
Python 2 и Python 3	3
Понятие интерпретатора	4
Понятие Jupyter Notebook	5
Понятие переменной	6
Команды print и input	7
Задание для закрепления	8
Комментарии	9
Синтаксис	10
Синтаксические ошибки	11
Задание для закрепления	12
Практическая работа	13
Полезные материалы (ко всему курсу)	14



История языка Python

Python - это язык программирования, созданный Гвидо ван Россумом в конце 1980-х годов.

Ван Россум хотел разработать язык, который был бы простым в использовании, читаемым и понятным для разработчиков. Он назвал его Python в честь своего любимого комедийного шоу "Монти Пайтон и священный Грааль".

С момента своего создания Python получил широкое распространение и стал одним из самых популярных языков программирования.



Подумайте, в чем преимущества Python перед другими языками и в каких сферах он используется?



Python 2 и Python 3

Python имеет две основные версии - Python 2.x и Python 3.x.

Изначально был выпущен Python 2, который использовался многими разработчиками на протяжении долгого времени.

Однако в 2008 году была выпущена Python 3, которая внесла некоторые существенные изменения в язык.

Одним из ключевых различий между Python 2 и Python 3 является синтаксис.

Python 3 внесло несколько улучшений и упрощений в синтаксис, делая его более современным и последовательным. Одним из примеров является использование скобок для функций `print()` в Python 3.

Еще одно важное отличие между Python 2 и Python 3 - это обработка строк и поддержка Unicode.

Python 3 по умолчанию работает с кодировкой Unicode, что упрощает работу с различными языками и символами.

Различия между Python 2 и Python 3 также включают изменения в области управления памятью, стандартной библиотеки, улучшения производительности и другие технические детали.

Важно отметить, что Python 2 и Python 3 несовместимы между собой, поэтому при переходе на Python 3 могут потребоваться некоторые изменения в существующем коде.

Сегодня рекомендуется использовать Python 3 для новых проектов, так как он продолжает развиваться и получает активную поддержку от сообщества разработчиков.

Однако, некоторые старые проекты все еще используют Python 2, и для поддержки таких проектов остается возможность использования Python 2.



Подумайте про то, зачем вообще обновляют версию языка, какие могут быть сложности в процессе этого?

Что такое обратная совместимость и к каким проблемам это может приводить?



Понятие интерпретатора



Интерпретатор - это программа, которая выполняет код Python.

Он считывает и интерпретирует инструкции построчно, преобразуя их в машинный код или байт-код, который может быть непосредственно выполнен компьютером.

В случае Python наиболее распространенным интерпретатором является CPython, который является референсной реализацией языка. CPython написан на языке C и обеспечивает высокую производительность и надежность.

Кроме CPython, существуют и другие интерпретаторы Python:

- Jython (реализация на Java)
- IronPython (реализация на платформе .NET)
- PyPy (Just-in-Time компилятор).

Они обеспечивают дополнительные возможности и интеграцию с соответствующими платформами.

Использование интерпретатора Python позволяет разработчикам писать и тестировать код без необходимости компиляции. Он также обладает мощным интерактивным режимом, который позволяет выполнять код построчно и получать немедленные результаты.



Подумайте, как объяснить на пальцах как работает компиляция, в чем плюсы и минусы подхода с компиляцией и с интерпретацией?



Понятие Jupyter Notebook

Jupyter Notebook - это интерактивная среда разработки и обмена документами, которая позволяет создавать и выполнять код Python в виде ячеек. Он предоставляет удобное окружение для разработки, тестирования и демонстрации кода Python.

Jupyter Notebook состоит из веб-интерфейса, который позволяет создавать и редактировать ноутбуки, а также встроенного интерпретатора Python, который выполняет код в каждой ячейке ноутбука. Каждая ячейка может содержать код, текстовые описания, формулы, изображения, графики и другие элементы.

Одним из ключевых преимуществ Jupyter Notebook является его интерактивность. Разработчик может выполнять код в ячейках по одной или несколько сразу, что позволяет получать немедленные результаты и видеть вывод прямо в ноутбуке. Это удобно для тестирования и экспериментирования с кодом.

Jupyter Notebook также поддерживает множество расширений и плагинов, что позволяет использовать различные библиотеки, инструменты и визуализации для более продвинутой разработки и анализа данных. Например, можно использовать библиотеки для визуализации данных, машинного обучения или создания интерактивных графиков.



Понятие переменной



Переменная в программировании представляет собой символическое имя, которое используется для хранения и обращения к данным в памяти компьютера.

Переменные позволяют нам сохранять значения и использовать их в различных частях программы.

В Python для создания переменной не требуется объявление типа. Достаточно указать имя переменной и присвоить ей значение.

Например, `x = 5` создаст переменную `x` и присвоит ей значение 5.



Команды print и input



Команда `print` используется для вывода информации на экран.

Она позволяет нам отображать значения переменных, текстовые сообщения и результаты вычислений.

Например, `print("Hello, world!")` выведет на экран сообщение "Hello, world!".



Команда `input` используется для ввода данных пользователем.

Она позволяет нам запрашивать данные у пользователя и сохранять их в переменных.

Например, `name = input("Введите ваше имя: ")` запросит у пользователя ввод имени и сохранит его в переменной `name`.



Задание для закрепления

1. Наберите в редакторе код из задания символ в символ и нажмите «Выполнить».

Python

```
print('Hello, World!')
```

Внимание: если вы напишете heLLo, woRld! вместо Hello, World!, то это будет считаться другим текстом, потому что заглавные и строчные буквы — это разные символы, отличающиеся регистром. В программировании регистр практически всегда имеет значение, поэтому привыкайте всегда обращать на него внимание!

2. Сделать вывод своего имени.

Python

```
name = input("Введите ваше имя: ")
```




Комментарии



Комментарий — текст в коде программы, который не влияет на функциональность и добавляется программистами для себя и своих коллег.

Практически все языки программирования позволяют оставлять в коде комментарии. Они никак не используются интерпретатором. Они нужны исключительно для людей, чтобы программист оставлял пометки для себя и для других программистов.

С их помощью добавляют пояснения, как работает код, какие ошибки нужно исправить или не забыть что-то добавить позже:

Python

```
# Удалить строку ниже после реализации задачи по регистрации
print(10)
```

Комментарии в Python начинаются со знака # и могут появляться в любом месте программы. Они могут занимать всю строку. Если одной строки мало, то создается несколько комментариев:

Python

```
# For Winterfell!
# For Lanisters!
```

Комментарий может находиться на строке после кода:

Python

```
print('I am the King') # For Lannisters!
```



Синтаксис



Синтаксис - это набор правил, описывающий комбинации символов алфавита, считающиеся правильно структурированной программой (документом) или её фрагментом.

Конец строки является концом инструкции (точка с запятой не требуется).

Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков. И про читаемость кода не забывайте. Отступ в 1 пробел, к примеру, не лучшее решение. Используйте 4 пробела (или знак табуляции, на худой конец).

Вложенные инструкции в Python записываются в соответствии с одним и тем же шаблоном, когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

Python

Основная инструкция:

 Вложенный блок инструкций

Иногда возможно записать несколько инструкций в одной строке, разделяя их точкой с запятой:

Python

```
a = 1; b = 2; print(a, b)
```



Синтаксические ошибки



Синтаксическая ошибка — нарушение грамматических правил языка программирования.



SyntaxError (ошибка парсинга) — тип ошибок в Python, возникающих при наличии синтаксических ошибок в коде.

Если программа на Python написана синтаксически некорректно, то интерпретатор выводит на экран соответствующее сообщение. Также он указывает на файл и строку, где произошла ошибка.

Синтаксическая ошибка возникает в том случае, когда код записали с нарушением грамматических правил. В естественных языках грамматика важна, но текст с ошибками обычно можно понять и прочитать. В программировании все строго. Мельчайшее нарушение — и программа даже не запустится. Примером может быть забытая ;, неправильно расставленные скобки и другие детали.

Вот пример кода с синтаксической ошибкой:

```
Python
print('Hodor')
```

Если запустить код выше, то мы увидим следующее сообщение:

```
Python
$ python index.py
File "index.py", line 1
    print('Hodor)
          ^
SyntaxError: EOL while scanning string literal
```

С одной стороны, ошибки синтаксиса — самые простые, потому что они связаны с грамматическими правилами написания кода, а не со смыслом кода. Их легко исправить: нужно лишь найти нарушение в записи. С другой стороны, интерпретатор не всегда может



четко указать на это нарушение. Поэтому бывает, что забытую скобку нужно поставить не туда, куда указывает сообщение об ошибке.



Задание для закрепления

Выведите на экран:

Python

Python **is** very interesting to learn



Практическая работа

Выведите на экран друг за другом три имени: Robert, Stannis, Renly.

Для каждого имени используйте свой собственный вызов `print()`.

В результате на экране должно отобразиться:

```
Python
```

```
Robert
```

```
Stannis
```

```
Renly
```



Полезные материалы (ко всему курсу)

Смотрим в этих материалах темы, относящиеся к материалам урока:

1. [Официальная документация Python](#)
2. [Учебник "Learn Python the Hard Way"](#)
3. [Видеоуроки "Sentdex"](#)
4. [Видеоуроки "Corey Schafer"](#)

Убедитесь, что при изучении Python вы выбираете актуальные материалы, которые соответствуют последней версии языка (Python 3).