

Основы Python



Видео 1:

Знакомство с Python

Цели занятия:

- Познакомиться с базовыми понятиями языка Python, его преимуществами и проанализировать, насколько обоснована его критика.
- Разобрать принципы работы в Jupyter Notebook.
- Выяснить, почему обязательно нужно писать комментарии в коде и в чём особенности простых арифметических операций в Python 3.
- Познакомиться с переменными и узнать, как их называть.
- На практике увидеть применение переменных, условных конструкций и логических операторов, написание кода.

Преимущества Python перед другими языками программирования:

1. Простой синтаксис.
2. Большое количество библиотек на все возможные случаи.
3. Широкое применение и возможность работать с большими данными.
4. Подходит для веб-разработки.
5. Применяется в системном администрировании.

Основная критика Python:

1. Высокоуровневый язык, часть написана на языке C.
2. Не подходит для production-кодов, но позволяет быстро составить прототип сложной системы.
3. На Python тяжело делать высоконагруженные проекты.
4. Привлекать специалистов по Python может быть трудно, но оправдано в плане бюджетов и сроков.

Название языка Python происходит от названия комик-группы Monty Python.

Вывод:

У языка программирования Python есть ряд преимуществ перед другими языками. Он отлично подходит для задач обработки данных и позволяет с наименьшими усилиями в сжатые сроки написать рабочий код, который решает конкретную задачу.

Видео 2:

Jupyter Notebook

Основной инструмент для работы с Python — Jupyter Notebook.

Преимущества Jupyter Notebook:

1. Построчное выполнение кода.
2. Удобное отображение таблиц и промежуточных результатов.
3. Визуализация любой сложности.

Рекомендуется запускать Jupyter Notebook через дистрибутив Anaconda:

- Содержит большое количество библиотек для обработки кода.
- Содержит большое количество виджетов.
- При запуске Jupyter Notebook можно писать код в удобном веб-интерфейсе.

Markdown

Язык разметки, популярный в разных системах. Его используют, чтобы оформлять отчёты, создавать ссылки, вставлять таблицы, меню.

Основные команды:

- Shift + Enter — выполнить код в ячейке и перейти на следующую.
- Ctrl + Enter — выполнить и остаться.
- Табуляция — автозаполнение длинных названий.
- Esc + m или просто m — перевести ячейку из кода в комментарии.
- b — добавить новую ячейку.
- Ctrl + Shift + P (Command + Shift + P) — список горячих клавиш.

Важно: Jupyter понимает эти команды и в русской раскладке.

Все команды можно сделать через верхнюю панель под меню File, Edit.

Jupyter поддерживает:

- HTML-код. Его можно использовать для оформления отчётов.

- Разметку, позволяющую писать формулы: греческие символы, математические операции, матрицы и др.

Вывод:

Jupyter Notebook — основной инструмент для работы с Python. У него есть ряд преимуществ перед другими инструментами. Удобный в использовании, он содержит огромное количество библиотек, виджетов и поддерживает все необходимые функции для оформления кода.

Видео 3:

Комментарии в коде. Арифметические операции

Комментарии в коде

Комментарии — поясняющий текст, который присутствует в коде программы, но не выполняется интерпретатором.

Комментарии используют, чтобы добавить объяснение для определённого блока кода. Рекомендуется всегда писать комментарии при работе с кодом.

Важно:

Однострочный комментарий начинается с символа #

Многострочные комментарии пишут в тройных кавычках `''' '''` `""" """`

Арифметические операции

В Python используют следующие арифметические операции с числами:

- Сложение (+)
- Вычитание (-)
- Умножение (*)
- Деление (/)
- Целочисленное деление (//)
- Возведение в степень (**)
- Взятие остатка от деления (%)

Последовательность выполнения операций регулируют скобками: операции в скобках будут выполняться в первую очередь.

Несколько вычислений записывают через запятую.

Важно: в Python 2 деление по умолчанию происходит целочисленно.

Выведение результата вычисления на экран:

1. Через функцию `print`: каждое вычисление пишут в скобках, выводится результат **всех вычислений**.
2. Без функции `print`: выводится результата только **последнего вычисления**.

Вывод:

При работе с кодом рекомендуется всегда писать комментарии — пояснения определённого блока кода.

В Python используют стандартные арифметические операции, а последовательность из нескольких вычислений задают скобками. Существует два основных способа вывести их на экран, каждый из которых обладает своими особенностями.

Видео 4:

Переменные

Переменная — произвольный объект, которому дано имя. Переменной может быть число, строка и практически что угодно в Python.

Переменные необходимы для хранения данных и промежуточных результатов вычислений.

$a = 10 + 20$

$b = a \times 30$

$c = a / b$

Как называть переменные:

- Имя переменной может состоять только из цифр, букв и знаков подчёркивания.
- Имя переменной должно описывать её суть.
- Имя переменной не может начинаться с цифры.
- Длинные названия переменных:
 - snake_case — стандарт, используемый в Python;
 - CamelCase — больше распространён в других языках программирования.

Важно: snake и Snake — разные переменные.

Можно задавать переменные по-русски латиницей, однако считается, что лучше называть переменные английскими аналогами.

В Jupyter Notebook функция автодополнения через табуляцию автоматически предлагает названия переменных, в том числе длинных.

Объявление переменной

Объявить переменную — значит присвоить имени какой-то объект (целое число, строку).

Сначала переменную необходимо объявить, а затем производить действия: например, вывод на экран.

Строки

Строки — текст.

Как можно задавать строки:

- В кавычках (одинарных или двойных). Внутри одного типа кавычек важно использовать кавычки другого типа (' He said: "Hello!" ').
- Экранированием — использовать кавычки одного типа и слеш ("Hello \"World!"). Слеш показывает, что следующий символ является просто текстом.

Вывод:

Переменная — объект, которому дано имя. Переменной может быть число, строка и что угодно. Имена переменных должны быть интуитивно понятны и описывать суть переменной. Если название очень длинное, можно использовать функцию автодополнения через табуляцию. Перед тем, как совершить какие-то действия с переменной, её нужно объявить. При объявлении переменных в качестве строк важно помнить про последовательность кавычек.

Видео 5:

Условные конструкции

Операторы сравнения

Для сравнения чисел используют следующие операторы сравнения:

- >
- <
- == - используется для проверки равенства двух переменных;
 - = используется для присвоения значения переменной;
- >=
- <=
- != оператор неравенства.

В результате операций сравнения возвращается булево значение (True/False) — аналог ИСТИНА и ЛОЖЬ в Excel.

Можно делать сразу несколько сравнений. В этом случае сравнения могут быть записаны в цепочку.

Сравнение строк

По умолчанию строки сравниваются по алфавиту. Длинные строки тоже сортируются по алфавиту.

При сравнении дат используется алфавитный порядок дат в формате YYYY-MM-DD.

Функция len

Функция len считает количество элементов объекта.

В случае строк функция len считает количество символов в строке.

Условные конструкции

Условные конструкции используются для реализации сценариев при сравнении объектов.

Условные конструкции — это:

- способ управлять выполнением программы;
- способ запрограммировать принятие решений;
- логическое выражение, после которого пишутся команды, выполняющиеся, если условие истинно.

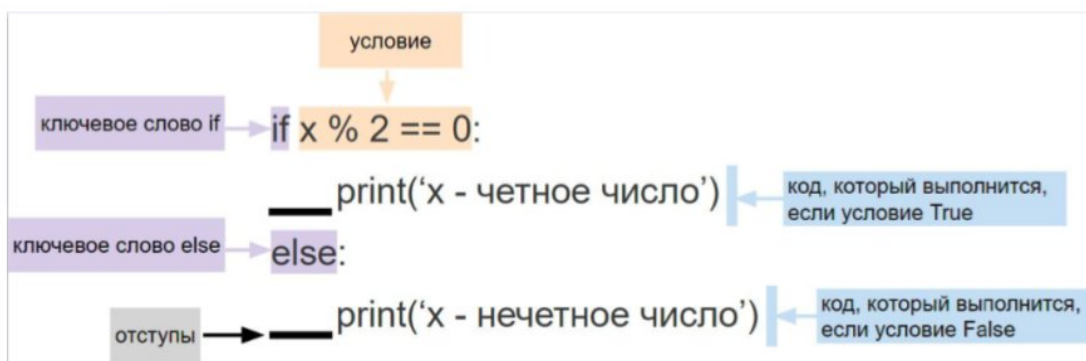
Чтобы проверить условные конструкции, используют следующие операторы:

1. if — проверяет на верность условие, которое написано после него.
2. elif — используется для дополнительной проверки выполнения условия.
3. else — выполняется, когда предыдущее условие не выполнено.

01 if

02 elif

03 else



Логические операторы

1. AND — логическое И, возвращает True, когда оба условия верны.
2. OR — логическое ИЛИ — True, когда хотя бы одно условие выполняется.
3. NOT — логическое НЕ — возвращает противоположный результат.

Логические проверки можно объединять в длинные цепочки, чтобы проверить сразу несколько условий. Каждую из проверок рекомендуется писать в отдельных скобках с соблюдением дополнительных отступов.

Можно реализовывать логические операторы AND и OR в рамках одной проверки.

Функция input в Jupyter Notebook позволяет задать меню для ввода определённого значения.

Вывод:

В Python используются различные операторы сравнения, и часто они могут быть записаны в одну цепочку. При сравнении строк по умолчанию используется сортировка по алфавиту, а при сравнении дат — по формату YYYY-MM-DD. Условные конструкции — это логические выражения, которые используются для реализации сценариев при сравнении объектов. Их задают с помощью операторов if, elif и else, а также логических операторов AND, OR, NOT. Такие логические проверки могут быть записаны в длинные цепочки, чтобы проверить несколько условий одновременно.

Общие итоги:

1. Познакомились с базовыми понятиями языка Python, его преимуществами и проанализировали, насколько обоснована его критика.
2. Разобрались, как работать в Jupyter Notebook.
3. Поняли, почему нужно обязательно писать комментарии в коде, и узнали особенности арифметических операций в Python 3.
4. Познакомились с переменными и узнали, как их называть.
5. На практике увидели применение переменных, условных конструкций и логических операторов, написание кода.