Время и даты в Python

Цели занятия

- Узнаем, какие типичные проблемы со временем ведут к ошибкам при обработке и интерпретации данных.
- Разберём на реальных примерах, как работать с ошибками в Python.
- Разберём, как работать с датами в Python.
- Познакомимся с форматом Unixtime.

Видео 1.

Типичные проблемы со временем в сфере IT

Существует ряд типичных проблем со временем, обусловленных:

- заблуждениями программистов относительно времени,
- техническими особенностями операционных систем,
- различными толкованиями и отсутствием единых стандартов,
- часовыми поясами.

Вывод

Проблемы со временем в сфере IT существуют с 70-х годов, когда вычислительная техника стала активно развиваться. Эти проблемы приводят к ошибкам при чтении и интерпретации данных.

Видео 2.

Обработка ошибок. Блоки try, except, traceback, finally

Как читать ошибки в коде Python

Шаг 1. Начать читать с последней строчки кода, где указывается вид и описание

ошибки.
Шаг 2. Определить место, где произошла ошибка: в какой строке, в какой функции.

Какие существуют типы ошибок в Python

- **ZeroDivisionError** — деление на ноль.
- **ImportError** — не удалось импортирование модуля или его атрибута (надо установить библиотеку).
- **IndexError** — индекс не входит в диапазон элементов.
- **KeyError** — несуществующий ключ (в словаре, множестве или другом объекте).
- **MemoryError** — недостаточно памяти.
- **SyntaxError** — синтаксическая ошибка (вы опечатались или не закрыли скобку).
- **TypeError** — операция применена к объекту несоответствующего типа.
- **ValueError** — функция получает аргумент правильного типа, но некорректного значения.
- **Warning** — предупреждение (текст на красном фоне в юпитере — это предупреждение, а не ошибка).

Что делать, если нужна помощь в чтении ошибки

1. Скопировать последнюю строчку, в которой указан тип и причина ошибки, и направить в техподдержку.
2. Отправить скриншот или текстом всё сообщение об ошибке, где будут указаны все пути, в которых эта ошибка воспроизводится.
3. Отправить весь файл с кодом либо код, который приводит к этой ошибке.

Как предотвращать ошибки. Блоки, чтобы цикл с расчётом не «падал»

Способ 1. Стандартный блок try, except.

Способ 2. Блок traceback.

Способ 3. Блок finally.

Вывод

Существуют 2 основных способа чтения ошибок в Python. Можно выделить некоторые типы ошибок в Python. Чтобы получить помощь при работе с ошибками, рекомендуется применять один из трёх сценариев. Важно не только обрабатывать возникшие ошибки, но и уметь предотвращать их с помощью блоков в Phyton.

Видео 3.

Datetime. Работа с датой и временем

Даты и время в Python

- Дата для Python — это набор символов, которые необходимо расшифровать.

- Для расшифровки даты необходимо импортировать библиотеку datetime.

Как импортировать библиотеку datetime для работы с датой и временем в Python

- 2 способа импорта datetime.

Как в Python форматировать строку в дату с помощью datetime.strptime

- В Python не применяется модуль автоматического распознавания дат в целях исключения ошибок со временем.
- Для расшифровки в Python пользователю необходимо самостоятельно задать формат даты с помощью datetime.strptime.

Что можно делать с датой и временем в Python:

- сравнивать,
- складывать/вычитать,
- прибавлять интервал,
- определять номер недели,
- проходить високосный год.

Вывод

С датами в Python можно совершать различные действия только после того, как строка стала датой. Чтобы превратить строку в дату нужно корректно импортировать библиотеку datetime в Python с помощью datetime.strptime и отформатировать строку.

Видео 4.

Timedelta: как прибавлять интервал к датам

Какие операции можно выполнять в классе timedelta

Для определения промежутка времени применяем **timedelta:**

- с помощью **timedelta** можно складывать и вычитать даты;
- **timedelta** перебирает время до микросекунд;
- **timedelta** предусматривает автоматический перебор дат и времени с учётом високосного года при сложении и вычитании.

Алгоритм работы с датами

1. Строки в Python вручную создаем в формате «год - месяц - день - час - минута - секунда - микросекунда».
2. Переводим строки в даты с помощью datetime.strptime**.**
3. Переходим в timedelta.
4. Прибавляем интервалы времени.
5. Возвращаем формат даты обратно в строку с помощью datetime.strftime.

Что такое метод while

Метод **while** позволяет упростить алгоритм перебора дат.

Вывод

Для определения промежутка времени применяем **timedelta.** При совершении операций с датами следует придерживаться определенного алгоритма. Для упрощения перебора дата можно применять метод **while.**

Видео 5.

Unixtime как альтернатива Datetime

Datetime или Unixtime?

- Что такое Unixtime
- В каких случаях и с какой целью при работе с датами оптимально применять Unixtime, а в каких Datetime.

Преимущества применения Unixtime

- Целое число воспринимается просто, не нужно думать о формате
- Формат занимает меньше места на диске и оперативной памяти, чем строки datetime.
- Unixtime измеряет количество секунд с 01.01.1970 г., тем самым делая их независимыми от часового пояса, тогда как datetime сохраняет дату и время без часового пояса.
- формат **Unixtim**е легко перевести в **datetime** с помощью **fromtimestamp**.

Вывод

Для оптимизации работы и повышения производительности труда Datetime в некоторых случаях можно заменять на формат Unixtime.

Общие итоги

1. Рассмотрели типичные проблемы со временем, ведущие к ошибкам при обработке и интерпретации данных.
2. Разобрали на реальных примерах, как работать с ошибками в Python: чтение ошибок, обработка информации, как запрашивать помощь при работе с ошибками.
3. Разобрали на практике, как работать с датами в Datetime: как из строк получать полноценный объект даты и времени, как импортировать библиотеку, как заполнять datetime, как сравнивать, вычитать и прибавлять даты.
4. Познакомились с форматом Unixtime как альтернативой Datetime.