PYTHON329 HW №13

Metadata

type	home work
hw_theme	рефакторинг, функции, pyinstaller
hw_num	12 - 13
topic	рефакторинг "Города" на функции, создание модуля, упаковка в exe
st_group	python 329
date	2023-08-27
block	python
module	4
lesson_no	24 - 28

Что мы прошли? 🧐

Привет! 👋

В этом уроке рассматриваются следующие темы:

1. Создание и работа со словарями:
   - Определение словарей и их особенности.
   - Создание словарей с различными ключами и значениями.
   - Добавление элементов в словарь.
2. Методы словарей:
   - `.get()` для получения значения по ключу.
   - `.values()` для получения всех значений словаря.
   - `.keys()` для получения всех ключей словаря.
   - `.items()` для получения всех пар ключ-значение словаря.
   - `.pop()` для удаления элемента по ключу.
3. Применение словарей в реальных задачах:
   - Работа с фильмами и актерами.
   - Использование генераторов списков для поиска актеров в фильмах.
   - Использование циклов и условий для поиска и фильтрации данных.
4. Использование сторонних библиотек:
   - Импорт `pprint` для красивого вывода словарей.
5. Упаковка Python-программы:
   - Описание библиотеки `pyinstaller` для упаковки Python-программ в исполняемые файлы.
   - Установка и использование `pyinstaller` для создания исполняемого файла.

Этот код демонстрирует работу со словарями, методами словарей, поиском данных в словарях и использование сторонних библиотек для более читаемого вывода данных. Также он включает информацию о возможности упаковать Python-программу в исполняемый файл с помощью `pyinstaller`.

Данный код представляет собой часть скрипта для игры в города и включает в себя рефакторинг, использование модулей, работу с файлами и использование условия `if __name__ == '__main__':` для управления запуском скрипта.

В этом коде были использованы следующие технологии и концепции:

1. Модули и библиотеки:
   - `json` модуль для работы с JSON-файлами.
   - `pprint` из модуля для красивого вывода данных.
2. Чтение и обработка файлов:
   - Чтение данных из файла `cities_set.json` с помощью `json.load()`.
   - Формирование множества `cities_set` из прочитанных данных.
3. Структуры данных:
   - Использование множества `set` для хранения городов и букв.
4. Циклы и условия:
   - Основной игровой цикл с условием `while True`.
   - Проверки на последнюю букву, плохую букву и соответствие букв.
   - Проверка на наличие города в множестве.
5. Работа с функциями:
   - Намечены функции `read_json`, `get_bad_first_letters`, `check_is_bad_first_letter`, `check_is_letters_match`, но они пока не реализованы.
6. Условие `if __name__ == '__main__':`:
   - Вызов функции `main()` в блоке `if __name__ == '__main__':`.

Также в коде содержатся множество комментариев, которые описывают основные шаги и логику игры.

Домашнее задание 📄

В этот раз нас ждёт непростая, но очень интересная задача. Мы сделаем рефакторинг игры в "Города" на функции.

На текущий момент у всех вас игра в разной реализации. Кто-то сделал максимально сложные варианты, с логированием последних 5 результатов игры и предусмотрел, или попытался предусмотреть условие `возьми предпоследнюю букву, если на последнюю ничего нет`.

У кого-то простая, но полностью рабочая игра.

Сейчас стоит задача переписать игру на функции, вынести их в отдельный файл `utils.py` .Напишите все функции там. Создайте функцию `main()` которая будет запускать игру и будет содержать цикл, пользовательский ввод, вызовы функций, промежуточные переменные, и всё что не поместилось в функции. Т.е. весь код игры.

Рядом создайте `main.py` куда импортируйте всё из `utils.py` и сделайте

```
if __name__ == '__main__':
    main()
```

💪 Сложная версия

- Вынесите все настройки в третий файл `settings.py`
- Поместите туда константы с именами файлов, сделайте константы с сообщениями в консоль.
- Импортируйте все необходимое от туда в `utils.py`
- Добавьте функцию логирования. Она будет печатать результат матча из константы и в консоль, и так же заносить её в лог (в котором не более 5 результатов)
- Лог для простоты можно сделать txt файлом, считывая от туда все строки, и считая их количество 😊

Критерии проверки 👌

Обычная версия ✅

- Всё что можно перенесено на функции, кроме самых простых вещей
- Функции называются по правилам нейминга функций в Пайтон
- PEP-8 и чистый код
- Комментирование
- Отдельный файл для функций, отдельный для запуска
- **Это задание для 2х пар, оно будет одинаковым и для субботы, и для воскресенья, пожалуйста, загрузите ОДИНАКОВЫЙ код и туда, и туда.**

Сложная версия 🧠

- Все требования, которые есть и для обычной версии
- третий файл `settings.py` с константами
- Логирование

Связано

PYTHON 329. Журнал
Академия TOP
Python 329. Python. Lesson 28
Python 329. Python. Lesson 26