PYTHON 329 HW №11

Metadata

type	home work
hw_theme	match - case
hw_num	11
topic	Изменяем проверки IF на match - case в игре Города
st_group	python 329
date	2023-08-22
block	python
module	3
lesson_no	22

Что мы прошли? 🧐

Привет! 👋
На прошедших парах (№21 и №22) мы узнали:

GTP:
😅 предоставленный код касается нескольких тем, связанных с оператором `match case` в Python, а также вводит концепцию функций. Вот разбивка основных тем и разделов кода, рассмотренных в предоставленном коде:

1. **Оператор `Match Case`:**
   Код демонстрирует использование оператора `match` в Python, который аналогичен конструкции `switch-case` в других языках программирования. В нем показано, как использовать сопоставления случаев для обработки различных сценариев на основе шаблонов.
2. **Обращение и Проверка на Палиндром:**
   Код демонстрирует, как перевернуть строку с помощью срезов, а затем проверяет, является ли данное слово палиндромом (читается одинаково слева направо и справа налево). Для реализации проверки на палиндром используется оператор `match`.
3. **Функции:**
   Код вводит концепцию функций в Python и объясняет их основный синтаксис и использование.
4. **Параметры Функций:**
   Код охватывает различные аспекты параметров функций, включая передачу нескольких параметров, использование именованных параметров и предоставление значений параметров по умолчанию.
5. **Функция Записи JSON:**
   Код содержит пример пользовательской функции с именем `write_json`, предназначенной для записи данных JSON в файл с настраиваемой кодировкой, обработкой ASCII и форматированием отступов.

В целом, код предоставляет примеры, чтобы проиллюстрировать, как использовать операторы `match case` и функции в Python. Похоже, что код также содержит объяснения и комментарии, направленные на помощь читателю в освоении представленных концепций.

Домашнее задание 📃

Замените все проверки, которые возможно заменить с `IF - ELSE` на `MATCH - CASE`

Сложная версия 💪

Можете начать рефачить код на функции!

> ⓘ **Рефа́кторинг (англ. refactoring), или перепроекти́рование кода, перераба́тка кода, равноси́льное преобразова́ние алгоритмов — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы. В основе рефакторинга лежит последовательность небольших эквивалентных (то есть сохраняющих поведение) преобразований. Поскольку каждое преобразование маленькое, программисту легче проследить за его правильностью, и в то же время вся последовательность может привести к существенной перестройке программы и улучшению её согласованности и чёткости.**

Критерии проверки 👌

- Чистый код
- Нейминг
- PEP-8
- Замена проверок на match-case

Связано

PYTHON 329. Журнал
Академия TOP
Python 329. Python. Lesson 22