

RPG Battle Log Parser Challenge

Challenge Preamble

This challenge is designed to help you practice basic Python concepts such as variables, loops, conditional statements, and file handling. You will develop a program that simulates a player's health system based on reading battle events from a text file.

Your task is to parse a battle log text file line by line, interpret each event, and update the player's health points (HP) accordingly. The player starts with a maximum HP of 250, which cannot be exceeded, and it will decrease whenever the player takes damage from monsters.

Part 1 – Battle Log Parsing: Theory

In this program, the player's HP changes based on reading and interpreting lines of text from a battle log file (for example, `battle_log.txt`). Each line represents a specific event such as taking damage, hitting a monster, or drinking a potion.

- Key Concepts:
 - Read file lines using.
 - Interpret keywords like 'took', 'hit', and 'takes'.
 - HP starts at 250 and cannot exceed that value.

Try to think about the challenge and design your own version before reading the step-by-step tutorial below.

Part 2 – Battle Log Parsing: Code Walk-through

Function: `main()`

This function manages the entire program: reading the battle log, interpreting events, updating HP, and printing results.

1. Step-by-step explanation:

1. Open `battle_log.txt`.

The python function `open()` is used to do that, it can be paired with a “`with ... as`” clause to automatically close the file as the scope finishes.

2. Read the lines.

There are several ways to do that, one of those is to use the `readlines()` method of the `io` object returned by `open()` to transfer all lines into a list of lines, which can be used to iterate.

3. Initialize `hp = 250`.

4. Loop through data and split each line into words.

Again, there are multiple way to do that, index-based, enumerate-based but the easiest to remember for basic usage is the object iterative way where an element of the list is given with the loop function. The basic syntax for this type of loop is

```
for element in list:
```

5. Use `if/elif` to decide what happens:

1. 'Player took x dmg' → reduce the hp variable by x.
2. 'Player takes a potion...' → increase the hp by the value, capping at 250.
3. 'Player hit the' → no action to do

Here there can be several way of testing when to select a specific case. When we analyze the given input file, we notice that the 2nd word in each line is what we want to check. Useful methods to use would be `split()` method from a str object. This one divide one string (a line from the text file) into a list of words based of standard dividers (usually a space).

This can also be done more elegantly using a `match ... case`, a new feature that was added recently in python

6. Print the final HP after all lines.
7. The answer should be **238 hp**.

Part 3 – Bonus Challenge : Listing All Enemies

Now that your program can correctly calculate the player's final health, let's extend it with a small upgrade — **list every enemy the player fought during the adventure!**

Each battle line in the log contains the name of the monster.
For example:

```
Final Health after all battles: 238
```

```
Enemies encountered: ['Kragobloth', 'Slazfang', 'Drimaw',  
'Vornlurker']
```