Textual RPG (Role-Playing Game)

Array methods

Required methods for the following exercises: `map`, `filter`, `reduce`, `spread`.

1. **Retrieve Player Names**
   - **Input:** An array of player objects.
   - **Task:** Return a new array that contains only the names of each player.
   - **Example:**
     - Input: `[{ name: "Hero" }, { name: "Warrior" }, { name: "Mage" }]`
     - Expected Output: `["Hero", "Warrior", "Mage"]`
2. **Filter Players by Level**
   - **Input:** An array of player objects, a minimum level value.
   - **Task:** Return a new array containing only players with a level equal to or greater than the provided value.
   - **Example:**
     - Input: `[{ level: 1 }, { level: 2 }, { level: 3 }], minLevel = 2`
     - Expected Output: `[{ level: 2 }, { level: 3 }]`
3. **Calculate Total Health**
   - **Input:** An array of player objects.
   - **Task:** Calculate and return the total health of all players.
   - **Example:**
     - Input: `[{ health: 100 }, { health: 120 }, { health: 80 }]`
     - Expected Output: `300`
4. **Combine Player Names and Levels**
   - **Input:** An array of player objects.
   - **Task:** Return a new array with strings combining the player name and their level. Each string should be formatted as "{name} - Level {level}".
   - **Example:**
     - Input: `[{ name: "Hero", level: 1 }, { name: "Warrior", level: 2 }]`

        - Expected Output: `["Hero - Level 1", "Warrior - Level 2"]`

5. **Update Player Health**
    - **Input:** An array of player objects, a new health value, and a player name.
    - **Task:** Update the health of the specified player in the array and return the updated array.
    - **Example:**
        - Input: `[{ name: "Hero", health: 100 }, { name: "Warrior", health: 120 }], newHealth = 150, playerName = "Hero"`
        - Expected Output: `[{ name: "Hero", health: 150 }, { name: "Warrior", health: 120 }]`

6. **Filter Players by Location**
    - **Input:** An array of player objects, a location string.
    - **Task:** Return a new array containing only players who are in the specified location.
    - **Example:**
        - Input: `[{ location: "forest" }, { location: "mountain" }, { location: "castle" }], location = "forest"`
        - Expected Output: `[{ location: "forest" }]`

7. **Get Average Player Level**
    - **Input:** An array of player objects.
    - **Task:** Calculate and return the average level of all players.
    - **Example:**
        - Input: `[{ level: 1 }, { level: 2 }, { level: 3 }]`
        - Expected Output: `2`

8. **Sort Players by Health**
    - **Input:** An array of player objects.
    - **Task:** Return a new array sorted by player health in descending order.
    - **Example:**
        - Input: `[{ health: 100 }, { health: 120 }, { health: 80 }]`
        - Expected Output: `[{ health: 120 }, { health: 100 }, { health: 80 }]`

9. **Retrieve Player Inventories**

- **Input:** An array of player objects.
- **Task:** Return a new array that contains only the inventories of each player.
- **Example:**
  - Input: `[{ inventory: ["sword", "health potion"] }, { inventory: ["axe", "shield"] }]`
  - Expected Output: `[["sword", "health potion"], ["axe", "shield"]]`