

Textual RPG (Role-Playing Game)

Arrow Functions

1. Simple Arrow Function Conversion

- **Objective:** Convert a traditional function into an arrow function.
- **Exercise:** Given a function `getPlayerName` that returns the name of a player from your RPG game, convert it into an arrow function.
- **Initial Code:**

```
function getPlayerName(player) {  
  return player.name;  
}
```

2. Control Flow in Arrow Functions

- **Objective:** Understand how to use control flow statements (like if-else) in arrow functions.
- **Exercise:** Given a function `getPlayerHealth` that returns a string stating whether the player's health is "High", "Medium", or "Low" based on a specific health score. Convert this function into an arrow function.
- **Initial Code:**

```
function getPlayerHealth(health) {  
  if (health > 70) {  
    return "High";  
  } else if (health > 40) {  
    return "Medium";  
  } else {  
    return "Low";  
  }  
}
```

3. Arrow Functions with Multiple Operations

- **Objective:** Learn how to perform multiple operations in arrow functions before returning a value.
- **Exercise:** Given a function `getTotalInventoryValue` that takes an array of items (each item has a `value` property), calculates the total value of all items, and returns the result. Convert this function into an arrow function.
- **Initial Code:**

```
function getTotalInventoryValue(items) {  
  let totalValue = 0;  
  for (let i = 0; i < items.length; i++) {  
    totalValue += items[i].value;  
  }  
  return 'Total inventory value is ' + totalValue;  
}
```

4. Arrow Functions as Callbacks

- **Objective:** Understand how arrow functions can be used as callbacks.
- **Exercise:** Your RPG game sorts items in the inventory by their value before displaying. Currently, it uses a traditional function for sorting. Convert this sorting function into an arrow function.
- **Initial Code:**

```
let sortedItems = items.sort(function(a, b) {  
  return b.value - a.value;  
});
```

5. Implicit Return with Arrow Functions

- **Objective:** Understand the concept of implicit return in arrow functions.
- **Exercise:** There's a function `getItemNames` which returns an array of all item names. Convert this function into a single-line arrow function using the implicit return feature.
- **Initial Code:**

```
function getItemNames(items) {  
  return items.map(function(item) {  
    return item.name;  
  });  
}
```

