

Monster Project

Suggested Progression

1. When the page loads, the following displays:
 - a. Hero picture
 - b. Random monster picture
 - c. Hero health (starts at 100)
 - d. Monster health (starts at 100)
 - e. Number of monsters defeated (starts at 0)
 - f. "Fight" button appears
 - g. Selection box filled with your heroes name's
 - h. Appropriate header
2. Change hero button
 - a. Grab and store the value of the dropdown select box
 - b. Compare the selection to each of your hero's names
 - c. Change hero index appropriately
 - d. Change the src of the appropriately
3. setNewMonster()
 - a. Generate random number within the range of your monster array, assign it to your monster index
 - b. Change the src of tag appropriately
 - c. Update your monster's health appropriately
4. Generate a random monster when the page loads
 - a. When the <body> loads, call the appropriate function to display a new monster
 - b. Check by refreshing your webpage...does the monster change?
5. Adjusting health in play() function
 - a. When "Fight" button is clicked, both monster and hero health's are adjusted
 - i. Generate random numbers within the ranges
 - ii. Subtract from appropriate character's health
 - iii. Display new information in corresponding <div>

6. Create hero message and monster message
 - a. `monsterMessage()`
 - i. Concatenate Strings to create a sentence displaying the damage dealt by the monster
 - ii. Use the same damage value given in parameter!
 - iii. Return your String
 - b. `heroMessage()`
 - i. Concatenate Strings to create a sentence displaying the damage dealt by the hero
 - ii. Use the same damage value given in parameter!
 - iii. Return your String
 - c. In `play()`, after calculating the hero's and monster's attack, use what the above functions return to concatenate and display a String with the results.
7. When monsters reach 0 health in `play()` function
 - a. Check if the monster's health is at or below 0
 - i. Update monsters defeated
 - ii. If health is below 0, set it to 0 (you can't have negative health)
 - iii. Display appropriate message
 - iv. Change innerHTML of button container `<div>` to be a new button
 1. Should say "Next Monster"
 2. Should have `onClick` event handler to display a new monster
8. When hero reaches 0 health in `play()` function
 - a. Check if the hero's health is at or below 0
 - i. If health is below 0, set it to 0 (you can't have negative health)
 - ii. Display appropriate message
 - iii. Change innerHTML of button container `<div>` to be a new button
 1. Should say "Play again"
 2. Should have `onClick` event handler to reset the game
9. When you defeat 5 monsters
 - a. Check if you have defeated 5 monsters
 - b. Display appropriate message
 - c. Update and display monsters defeated
 - d. Change innerHTML of button container `<div>` to be a new button
 - i. Should say "Play again"
 - ii. Should have `onClick` event handler to reset the game
10. `resetEntireGame()`
 - a. Reset and display hero health
 - b. Reset and display `monstersDefeated`
 - c. Generate a random monster