

REVIEW DOM MANIPULATION BY BUILDING A ROCK, PAPER, SCISSORS GAME

Introduction:

In the previous projects you learned how to work with basic DOM manipulation. Now it is time to review what you have learned by building a Rock, Paper, Scissors game.

In this mini project, you will review conditionals, functions, `getElementById`, and more. This project will give you an opportunity to solve small problems and get a better understanding of the basics.

Step 1:

The first step is to build out the function that will generate a random choice for the computer.

The `getRandomComputerResult` function will be used to get the computer's choice. Inside that function, you should see an `options` array with "Rock", "Paper", and "Scissors".

Your task is to complete the `getRandomComputerResult` function so that it returns a random option from the `options` array.

Tips

- You can use `Math.random()` and `Math.floor()` to help you get a random whole number. This will represent the index number for the `options` array.
- You can use the random index to access the option from the `options` array.

Step 2:

In the game, there will be multiple rounds. The first to reach three points wins the game.

In this step, you will focus on determining if the player has won the round.

Complete the `hasPlayerWonTheRound` function. This function has two parameters: `player` and `computer`. The function should return `true` if the player has won the round, and `false` if the player has lost or tied the round.

Here are the criteria for the player to win a round:

- If the player chooses `"Rock"` and the computer chooses `"Scissors"`
- If the player chooses `"Scissors"` and the computer chooses `"Paper"`
- If the player chooses `"Paper"` and the computer chooses `"Rock"`

A few function calls have been provided for you to test your function.

Step 3:

Now it is time to get the results of the round. Complete the `getRoundResults` function.

If the player wins the round, update the `playerScore` by 1 and return the message `"Player wins! [player's choice] beats [computer's choice]"`.

If the computer and player choose the same option, return the message `"It's a tie! Both chose [player's choice]"`.

If the computer wins the round, update the `computerScore` by 1 and return the message `"Computer wins! [computer's choice] beats [player's choice]"`.

`[computer's choice]` should be replaced with `computerResult` while `[player's choice]` should be replaced with the `userOption`.

Tips

- Remember you can use the `hasPlayerWonTheRound` function to check if the player wins the round.
- You can use template literals or regular string concatenation to build the message.

Step 4:

Now it is time to update the scores and the round results message.

Complete the `showResults` function. The `playerScoreSpanElement` and `computerScoreSpanElement` should be updated to show the updated scores of the player and computer.

The `roundResultsMsg` should also be updated with the result of the round.

Tips

- Remember that you learned how to work with the `innerText` property to update the text content of an element.
- You can use the `getRoundResults` function to get the result of the round.

Step 5:

If you try to play the game, you will see that you can play for an infinite amount of rounds. But the rules state that the first one to three points wins.

Inside your `showResults` function, you will need to check if the player or computer has reached three points. If either has reached three points, you should display a message indicating the winner.

For example, if the player has won the game, then the `winnerMsgElement` should be updated to `"Player has won the game!"`. If the computer has won the game, then the `winnerMsgElement` should be updated to `"Computer has won the game!"`.

If there is a winner, you will want to show the `resetGameBtn` button and hide the `optionsContainer` so the player can play again.

Tips

Use the `style.display` property on an element, with the value `"block"` or `"none"`, to show or hide the element.

Step 6:

If the player or computer has won the game, there should be an option to reset the game and play again.

Complete the `resetGame` function that accomplishes the following:

- Resets the player and computer scores to `0`.
- Updates the `playerScoreSpanElement` and `computerScoreSpanElement` to display the new scores.
- Hides the `resetGameBtn` button.
- Shows the `optionsContainer` so the player can play again.
- Clears the content for the `winnerMsgElement` and `roundResultsMsg` elements.

Tips:

- You can use the `innerText` property to update the content of an element. To clear the content of an element, you can set the `innerText` to an empty string.

Once you apply those changes, you will have completed the Rock, Paper, Scissors game!