JavaScript Rock Paper Scissors Instructions

Rock paper scissors is a classic two player game. Each player chooses either rock, paper, or scissors. The items are compared, and whichever player chooses the more powerful item wins.

The possible outcomes are:

- Rock destroys scissors.
- Scissors cut paper.
- Paper covers rock.
- If there's a tie, then the game ends in a draw.

Our code will break the game into four parts:

1. Get the user's choice.
2. Get the computer's choice.
3. Compare the two choices and determine a winner.
4. Start the program and display the results.

1	The user should be able to choose 'rock', 'paper', or 'scissors' when the game starts. Using `const` and arrow function syntax, create a function named `getUserChoice` that takes a single parameter `userInput`.
Hint	`const getUserChoice = userInput => {` `};`
2	Since a user can pass in a parameter, such as 'Rock' or 'rock' with different capitalisations, begin by utilizing JavaScript's `toLowerCase()` function to make the `userInput` all lowercase. You can use code like this: `userInput = userInput.toLowerCase();`
Hint	Insert the above code below the line that declares `userInput`. However, the code should still be inside the `getUserChoice()` function.
3	When getting the user's choice, you should also check to make sure that the user typed a valid choice: 'rock', 'paper', or 'scissors'. Inside `getUserChoice()`, write an `if/else` statement that makes sure the `userInput` is either `'rock'`, `'paper'`, or `'scissors'`. If it does, then `return` the `userInput`. If not, use `console.log` to print an error message to the console.

Hint	You can utilise your knowledge of the `||` logical operator to write this condition. Your code may look something like:
	
	`if (userInput === 'rock' || userInput === 'paper' || ... ) {`
	`  return userInput;`
	`} else {`
	`  console.log('Error!');`
	`}`

4	Test the function by calling it with valid and invalid input, and printing the results to the console. You can delete this when you know your function works.

Hint	`console.log(getUserChoice('Paper')); // should print 'paper'`
	`console.log(getUserChoice('fork')); // should print 'Error!' and `undefined``

5	Now we need to have the computer make a choice. Create a new function named `getComputerChoice` with no parameters. Inside its block, utilize `Math.random()` and `Math.floor()` to get a whole number between 0 and 2. Then, depending on the number, `return` either `'rock'`, `'paper'`, or `'scissors'`.

Hint	Since there are three choices, you can create a random number between 0 and 2 with code like this:
	
	`Math.floor(Math.random() * 3);`
	Then, you can utilize an `if/else` or a `switch` statement to `return` the computer's choice, like so:
	
	`switch (randomNumber) { case 0: return 'rock'; case 1: return 'paper'; ... }`

6	Test the function by calling it multiple times and printing the results to the console. You can delete this when you know your function works.
Hint	`console.log(getComputerChoice()); // should print 'rock', 'paper', or 'scissors'`
7	Now it's time to determine a winner. Create a function named `determineWinner` that takes two parameters named `userChoice` and `computerChoice`. This function will compare the two choices played and then `return` if the human player won, lost, or tied. Let's deal with the tie condition first. Within the `determineWinner()` function, write an `if` statement that checks if the `userChoice` parameter equals the `computerChoice` parameter. If so, `return` a string that the game was a tie.
Hint	You can see if two things equal each other with the `===` operator. `if (userChoice === computerChoice) { return 'The game is a tie!'; }`
8	If the game is not a tie, you'll need to determine a winner.

	Begin by writing an `if` statement that checks if the `userChoice` is `'rock'`. Inside the `if` statement's block, write another `if/else` statement. The inner `if/else` should check if the `computerChoice` is `'paper'`. If so, `return` a message that the computer won. If not, `return` a message that the user won.

Hint	If the game is not a tie, we know that when the `userChoice` is `'rock'`, `computerChoice` must be either `'paper'` or `'scissors'`. If the `computerChoice` is `'paper'`, then paper covers rock and the computer wins. If the `computerChoice` is not `'paper'`, it must be `'scissors'`, and then the user wins.
	
	`if (userChoice === 'rock') {`
	`  if (computerChoice === 'paper') {`
	`  return 'The computer won!';`
	`  } else {`
	`  return 'You won!';`
	`  }`
	`}`

9	Next, write another `if` statement for if the `userChoice` is `'paper'`. Inside this `if` statement, the `computerChoice` must be either `'scissors'` or `'rock'`. Write logic that will `return` a winner.

Hint	This code should be similar to the code from Step 6. It may look like this:
	
	`if (userChoice === 'paper') {`
	`  if (computerChoice === 'scissors') {`
	`  return 'The computer won!';`
	`  } else {`
	`  return 'You won!';`
	`  }`
	`}`

10	Next, write yet another `if` statement for if the `userChoice` is `'scissors'`. Inside of this `if` statement, the `computerChoice` must either be `'rock'` or `'paper'`. Write logic that will `return` a winner.

Hint	This code should be similar to the code from Steps 6 and 7. It may look like:
	
	`if (userChoice === 'scissors') {`
	`  if (computerChoice === 'rock') {`
	`  return 'The computer won!';`
	`  } else {`
	`  return 'You won!';`
	`  }`
	`}`

11	Don't forget to test your function!

Hint	The expected output depends on what you wrote in the `determineWinner()` function.
	
	`console.log(determineWinner('paper', 'scissors')); // prints something like 'The computer won!'`
	`console.log(determineWinner('paper', 'paper')); // prints something like 'The game is a tie!'`
	`console.log(determineWinner('paper', 'rock')); // prints something like 'The user won!'`

12	Everything is set up. Now you need to start the game and log the results. Create a function named `playGame`. Inside the `playGame()` function, create a variable named `userChoice` set equal to the result of calling `getUserChoice()`, passing in either `'rock'`, `'paper'`, or `'scissors'` as an argument. Create another variable named `computerChoice`, and set it equal to the result of calling `getComputerChoice()`. Under both of these variables, use `console.log` to print them to the console.

Hint	Since our functions `return` values, you can store their returned values in variables, which can be used elsewhere.
	
	`const playGame = () => {`
	`  const userChoice = getUserChoice('scissors');`
	`  const computerChoice = getComputerChoice();`
	`  console.log('You threw: ' + userChoice);`
	`  console.log('The computer threw:' + computerChoice);`
	`};`

13	Finally, let's determine who won. Inside the `playGame()` function, call the `determineWinner()` function. Pass in the `userChoice` and `computerChoice` variables as its parameters. Make sure to put this function call inside of a `console.log()` statement so you can see the result. Then, to start the game, call the `playGame()` function on the last line of your program.

Hint	`const playGame = () => {`
	`  const userChoice = getUserChoice('scissors');`
	`  const computerChoice = getComputerChoice();`
	`  console.log('You threw: ' + userChoice);`
	`  console.log('The computer threw:' + computerChoice);`
	`  console.log(determineWinner(userChoice, computerChoice));`
	`};`
	
	`playGame();`

14	**Stretch & Challenge** Make this game better by adding a secret cheat code. If a user types `'bomb'` as their choice, then make sure they win, no matter what.
Hint	In `getUserChoice()`, add a fourth condition that checks if the `userInput` is `'bomb'`. At the beginning of `determineWinner()`, add another `if` statement that makes the user win if the `userChoice` is `'bomb'`.