

Randomize the order of an array in JavaScript

To randomize the order of an array in JavaScript, you can use the following code:

```
function shuffleArray(array) { for (let i = array.length - 1; i > 0; i--) { const j = Math.floor(Math.random() * (i + 1)); [array[i], array[j]] = [array[j], array[i]]; } return array; }
```

This code defines a function called `shuffleArray` that takes an array as an argument. The function then loops over the array from the last index to the first, and for each index, it generates a random index using `Math.floor(Math.random() * (i + 1))`. It then swaps the values at the current index and the random index using destructuring assignment. Finally, it returns the shuffled array.

You can call the `shuffleArray` function by passing in an array like this:

```
const myArray = [1, 2, 3, 4, 5]; const shuffledArray = shuffleArray(myArray); console.log(shuffledArray); // [3, 2, 5, 1, 4] (example output)
```

This will randomly shuffle the order of the elements in the array and return the shuffled array.

This code is a simple game that generates a random word from a predefined array of words and jumbles the letters of the word.

The user's task is to guess the original word from the jumbled letters.

The HTML code consists of an input field of type "text" and a button with the label "Start". The input field has a CSS class of "hidden", which makes it invisible on the screen.

The JavaScript code selects the HTML elements using their CSS classes and sets up event listeners to respond to user actions. When the "Start" button is clicked, it triggers an event listener function that sets a Boolean variable `inplay` to true and changes the button's label to "Guess". It also calls a function `createWord()` that generates a random word from the `myArray` array, jumbles its letters using the `randomArray()` function, and logs the jumbled word to the console.

The `randomArray()` function takes an array as input and shuffles its elements randomly using the Fisher-Yates algorithm. The function uses a loop to iterate over the array from the end to the beginning, and for each element, it generates a random index between 0 and the current index, swaps the element at the current index with the element at the random index, and returns the shuffled array.

The code doesn't contain any logic for checking the user's input or verifying whether it matches the original word, but it provides a basic structure for building a word-guessing game.

```
<input type="text" class="hidden">
  <button type="button">Start</button>
  <script>
    const message =
document.querySelector(".message");
    const guess = document.querySelector("input");
```

```

    const button =
document.querySelector("button");
    let inplay = false;
    const myArray =
["javascript","website","html","document","course"];
    button.addEventListener("click", function () {
        console.log("clicked");
        if(!inplay){
            inplay = true;
            button.innerHTML = "Guess";
            guess.classList.toggle("hidden");
            console.log(createWord());
        }
    })

```

```

function createWord(){
    let randomIndex = Math.floor(Math.random()
* myArray.length);
    let tempWord = myArray[randomIndex];
    let rand = randomArray(tempWord.split(""))
;

    console.log(rand.join(""));
    return tempWord;
}

```

```

function randomArray(arr){
    for(let i = arr.length-1;i>0;i--){
        let temp = arr[i];
        let j = Math.floor(Math.random() *
(i+1));

        console.log(temp);
    }
}

```

```

        console.log(i);
        console.log(j);
        arr[i] = arr[j];
        arr[j] = temp;
    }
    return arr;
}

```

1. The HTML code defines an input field of type "text" with a CSS class of "hidden" and a button with the label "Start".
2. The JavaScript code starts by using the `document.querySelector()` method to select the HTML elements with the CSS classes of "message", "input", and "button" and assigning them to variables named `message`, `guess`, and `button`, respectively.
3. The code initializes a Boolean variable `inplay` to `false`.
4. An array named `myArray` is defined with five string elements: "javascript", "website", "html", "document", and "course".
5. An event listener is added to the "click" event of the "button" element using the `button.addEventListener()` method. The event listener is an anonymous function that logs the message "clicked" to the console and checks whether the Boolean variable `inplay` is `false`.
6. If `inplay` is `false`, the event listener sets `inplay` to `true`, changes the button's label to "Guess" using the `button.innerHTML` property, and removes the "hidden" CSS class from the input field using the `guess.classList.toggle()` method. The function `createWord()` is then called and its return value is logged to the console.
7. The `createWord()` function generates a random index between 0 and the length of `myArray` using the `Math.random()` and `Math.floor()` methods, and assigns it to a

variable named `randomIndex`. It then selects the element at that index in `myArray` and assigns it to a variable named `tempWord`.

8. The `split()` method is used to convert `tempWord` into an array of its individual characters, and the resulting array is passed as an argument to the `randomArray()` function.
9. The `randomArray()` function takes the input array and uses a loop to iterate over its elements from the end to the beginning. For each element, it generates a random index between 0 and the current index, assigns it to a variable named `j`, and swaps the element at the current index with the element at index `j`. The shuffled array is then returned.
10. The shuffled array is logged to the console using the `join()` method to convert it back into a string, and `tempWord` is returned from `createWord()`.
11. The code does not provide any further logic for the game, such as checking the user's input or displaying the results to the user. It simply generates a random word from `myArray`, shuffles its letters, and logs the shuffled word to the console.