JavaScript – Number Guessing Game
Form, Functions, innerHTML, Delay

This project using HTML, JavaScript, & Cascading Style Sheets combines several previous lessons with new concepts using Form features, JavaScript functions, Document Object Model features, and code to create a delay. Some additional features of the Math library are also used.

The basic construction is that of a simple game where a user tries to guess a number, 1 to 5, that is randomly selected by the computer.

The basic HTML Page is setup as usual with a separate CSS to define style. The Javascript code will be moved to a separate file, resulting in a rather abbreviated HTML file.

The CSS file is incorporated with the `<link href="numbersGame.css" rel="stylesheet">` tag in the header section.

The javacript file, and all its' functions are made available to the HTML page by referencing them in the `<script src ="numbersJScript.js"></script>` tag.

Form elements, such as buttons, are incorporated by using the features of the HTML <form> code. In this program, a button is created with the text "Play Game" put on the top and when it is clicked, it will execute the "play()" function. This function resides in the separate javascript file.

```
<form>
    <input type="button" value="Play Game" onclick="play()">
    <br><br>
</form>
```

Named locations are created on the webpage using tag <div id= > which allows content to be placed in a specific location on a web page.

```
<div id="userName"></div><br>
<div id="userMessage"></div><br>
<h3><div id="winnerMessage"></div></h3><br>
```

Although this page performs some complex actions, the HTML code itself is pretty minimal and doesn't give a good indication of what it actually does.

HTML Code

Code for the webpage is as shown here.

```
<!DOCTYPE html>
<html>
<!-- Numbers Guessing Game Using External File - T.Riherd 11/13/21 -->

<head>
    <meta charset="UTF-8">
    <title>Tom's Game</title>
    <link href="numbersGame.css" rel="stylesheet">
</head>

<body>
    <h1>Welcome to the Number Guessing Game!</h1>
    <p>How good are you at guessing numbers?</p>

<form>
    <input type="button" value="Play Game" onclick="play()">
    <br><br>
</form>

<!-- Create separate divisions in the HTML page -->
<div id="userName"></div><br>
<div id="userMessage"></div><br>
<h3><div id="winnerMessage"></div></h3><br>

<script src ="numbersJScript.js"></script>

<!-- * * * *    S T O P     H E R E   * * * * -->
</body>
</html>
```

Cascading Style Sheet

This will be the first use of a background image to be displayed on the web page. The file, `background_numbers.jpg`, is a basic graphic with a black background so that the webpage can also be set to have a black background. If needed, the image can be repeated to cover the background, which is what the "background-repeat" setting does.

```
1   /* CSS Style Sheet for Number Guessing Game */
2   body{
3       background-color: black;
4       background-image: url("background_numbers.jpg");
5       background-repeat: repeat;
6       text-align: center;
7       color:white;
8       }
```

JavaScript Code

The program is basically carried by the JavaScipt code file which is referenced in the HTML page. The game is divided into two sections.

Action starts on the webpage when the "Start Game" button is clicked. This will engage the javascript function play() which contains all the code for:

- Initializing the variables.
- Obtaining player name.
- Greeting the player.
- Instructing player to select a number.

In most browsers, the JavaScript will execute so quickly, that the player will not have a chance to view the greeting or "Select a number from 1 to 5." before the dialog box pops up for player input. JavaScript does not have a "delay" or "sleep" function to give the screen refresh process time to show these messages. This requires the function to be divided so that this text can be displayed before the next command are executed.

In order to place content on the webpage without rearranging the remainder of the page, the sections are referenced using the instruction:

```
document.getElementById("userMessage").innerHTML = myMessage;
```

This will place the value of "myMessage" on the webpage in the section designated as "userMessage".

Console log is used simply to observe what the program is doing. It also is convenient to know what number the computer has selected in order to test both branches of the program for identifying if the player guessed right or wrong.

A feature of this code that is not found in many web pages is the command:

```
setTimeout(play2, 1500)
```

This is the last instruction in the initial play() function and it is used to create a slight delay of 1.5 seconds (1500 milliseconds) before the program continues on to the 2nd part of the program.

Unlike most programming languages, JavaScript does not have the equivalent of a "sleep" or "delay" instruction. This is a work-around for creating a delay where our code needed it to allow players to see text on the screen before it is covered up by a dialog box.

```
gameNumber = Math.floor(Math.random() * 5) + 1;
```

The above instruction simply generates a random number between 0 and 4, truncates the decimal portion, and adds 1 to it in order to end up with numbers from 1 to 5.

JavaScript Code Part 1

Code for the webpage is as shown here.

```
1   // JavaScript File to play Number Guessing Game  11/13/2021
2   // NOW IN JAVASCRIPT MODE
3   var myMessage = "Hi Console!";
4   var getName = "name";
5   var getNumber = 0;
6   var gameNumber = 6;
7   var resultMsg = "Waiting to play";
8
9   console.log(myMessage);
10  myMessage = "Guessing numbers between 1 and 5.";
11  document.getElementById("userMessage").innerHTML = myMessage;
12
13  // Function Part 1 - Get Name & Computer Number
14  function play() {
15      myMessage = "User clicked button to play game.";
16      document.getElementById("userMessage").innerHTML = myMessage;
17      getName = prompt("Please enter your name.", "name");
18      document.getElementById("userName").innerHTML = "Hello, " + getName + "!";
19
20      gameNumber = Math.floor(Math.random() * 5) + 1;
21      console.log("Computer has chosen number: ", gameNumber);
22      document.getElementById("userMessage").innerHTML = "Think of a number from 1 to 5.";
23
24      // Create delay in order to show greeting and pick a number message.
25      //     (This required splitting the gameplay function into Parts 1 & 2)
26      setTimeout(play2, 1500)
27  }
28
```

JavaScript Code Part 2

The `setTimeout(play2, 1500)` command delays execution of this function until a designated number of microseconds have passed (1000 microseconds = 1 second). So the remainder of the game is put inside the `play2()` function. Since there are no more commands after this function, the page appears to "wait" for the desired amount of time.

```
29  // Function Part 2 - Get User Guess & Show Results
30  function play2() {
31      getNumber = prompt("Enter a number between 1 and 5.");
32      console.log(getName, " picked number: ", getNumber);
33      document.getElementById("userMessage").innerHTML = "You picked number : " + String(getNumber) +
34          " and Computer picked number : " + String(gameNumber);
35
36      if(getNumber != gameNumber) {
37          resultMsg = "You guessed wrong!"
38      } else {
39          resultMsg = "WOW! You guessed the number!";
40      }
41      document.getElementById("winnerMessage").innerHTML = resultMsg;
42  }
43
44  // END OF JAVASCRIPT
```

Enhancements to the Web Page

To make the webpage more user friendly, consider modifications that could be made to:

Not ask for player's name if they want to play the game more than once.
Clear the "winner" message when the user chooses to play the game again.
Track the number of times the user wins the game when played multiple times.