UNIT 03
LESSON 03.09

Guess Mystery Number

Random Numbers

if-else if-else logic

show-hide buttons

buttons calling functions

Open 03.09-Number-Guessing-Game.html

In this lesson we will write a number guessing program:

- user clicks PLAY button
- button click calls function playGame
- function generates a random integer from 1-100
- input box and GUESS button replace PLAY button
- player enters number from 1-100
- player clicks GUESS button
- button calls function evalGuess
- function gets value from input box and converts it to a real number
- compares user guess number to the random number
- h2 tag of id "feedback" displays all messages
- when user gets number right, a PLAY AGAIN button appears.

First we need a few global variables. They need to be global, because they must be available to more than one function:

Declare randNum to hold the random number; playerGuess to store the player's number; and guessesLeft to count down from 7.

```
let randNum = 0;
let playerGuess = 0;
let guessesLeft = 7;
```

Next, get the DOM elements used for the game. There are two buttons, but the GUESS button is hidden on page load, so all we see is the PLAY button.

The input box for entering the guess number is also hidden on page load. Both hidden elements will appear when the user clicks PLAY button:

```
    let playBtn = document.getElementById('play-btn');

    // hide on page load, show on PLAY click
    let guessBtn = document.getElementById('guess-btn');
```

Have each button listen for its respective function to call when clicked.

```
    playBtn.addEventListener('click', playGame);

    guessBtn.addEventListener('click', evalGuess);
```

Also get the "guess box" (number input box), which is also hidden on page load. The hidden states are handled in the CSS, with the display:none property.

```
    // hide on page load, show on PLAY click
    let guessBox = document.querySelector('input');
```

Get the h2 that displays the feedback:

```
    let feedback = document.getElementById('feedback');
```

Now for the functions. Declare the playGame function, which runs when the player clicks the PLAY button. The function does the following:

- generates the random number from 1-100

- hides "this", which is itself, the PLAY button

- shows (display = "block") the guess box (number input field) and the GUESS button.

- sets the input box to an initial value of 0, so the user knows to click there and enter a number.

- gives the user some feedback: 'Guess the number!'

```
    function playGame() {
        randNum = Math.ceil(Math.random() * 100);
        this.style.display = 'none';
        guessBox.style.display = 'block';
        guessBtn.style.display = 'block';
        guessBox.value = 0;
        feedback.textContent = 'Guess the mystery number from 1-100!';
    }
```

The evalGuess function runs when the user clicks the GUESS button. This function gets the value from the input box and converts it to a real number using the `Number()` method.

```
function evalGuess() {
    playerGuess = Number(guessBox.value);
}
```

Now for the big if-else if-else block:

- if the guess is too low, user gets feedback (too LOW) and guesses again

- else if the guess is too high, user gets feedback (too HIGH) and guesses again

- else guess matches mystery number, so user is congratulated

- GUESS button is replaced by PLAY AGAIN button

- user gets 7 guesses, which count down and are displayed with each guess

- when the user gets it right, the resetGame() function is called

- there is no enforcement of guess limit. Game just keeps going after 7 guesses.

```
function evalGuess() {
    playerGuess = Number(guessBox.value);

    if (playerGuess < randNum) {
        guessesLeft--;
        feedback.innerHTML = '<p>Your guess is too low! <br>You have '
+ guessesLeft + ' guesses left!</p>';
    } else if (playerGuess > randNum) {
        guessesLeft--;
        feedback.innerHTML = '<p>Your guess is too high! <br>You have
' + guessesLeft + ' guesses left!</p>';
    } else {
        guessesLeft--;
        feedback.innerHTML = '<p>Congrats!<br>The mystery number is
indeed ' + randNum + '! <br>You got it in ' + (7 - guessesLeft) + '
guesses!</p>';
        resetGame();
    }
}
```

The resetGame function is called when the user guesses correctly. The function restores all the settings as they are on page load. The only difference is that the PLAY button now says PLAY AGAIN (it is the same button, with different text).

```
function resetGame() {
    guessesLeft = 7;
    guessBtn.style.display = 'none';
    guessBox.style.display = 'none';
    playBtn.style.display = 'block';
    playBtn.textContent = 'PLAY AGAIN';
    guessBox.value = 0;
}
```

END: Lesson 03.09

NEXT: Lesson 04.01