```
'use strict';
///////// Lecture 73: Implementing the Game Logic
// Create a secret Number that the number is between 1 and 20.
let secretNum = Math.trunc(Math.random() * 20) + 1; // global
let score = 20; // global variable
let guessed = false; // global
document.querySelector('.number').textContent = '?';

// Upon click event of the 'check' button:
document.querySelector('.check').addEventListener('click', function () {
  const msg = document.querySelector('.message');
  // Print the value of the guessing to the screen and convert it to a number
  if (score > 0 && !guessed) {
    const guess =
      Number(document.querySelector('.guess').value);
    // Check if there is any guess
    if (!guess) msg.textContent = `No number!`;
    else if (guess > secretNum) outputMsg('Too High!');
    // guess is too high
    else if (guess < secretNum) outputMsg('Too Low!');
    else CorrectAnswer(); //guess is correct.
  } else msg.textContent = `You are out of tries!`;
});

///////// Lecture 74: Manipulating CSS Styles
// This function takes in no input, it returns void,
// but it changes the background colour to green when a user gets the right answer.
// Note that whenever you manipulate styles (CSS), you always need to specify a string. You cannot just write a number (like 30).
// If the actual attribute is 2 words, then write the attribute in camel case notation.

// 75 Coding Challenge #1: Implementing the 'Again' functionality
document.querySelector('.again').addEventListener('click', function () {
```

```
    if (guessed) {
      // Since player has guessed the number, we should see if his current score is a high score.
      saveHighScore();
      // Player has guessed the number, and is going to try again.
      document.querySelector('body').style.backgroundColor = `#222`;
      document.querySelector(`.number`).style.width = `15rem`;
      document.querySelector(`.number`).textContent = `?`;
      guessed = false;
    }
    score = 20;
    secretNum = Math.trunc(Math.random() * 20) + 1;
    document.querySelector('.score').textContent = score;
  });

  // Some utility function
  function CorrectAnswer() {
    document.querySelector('body').style.backgroundColor = `#60b347`;
    document.querySelector('.number').style.width = `30rem`;
    document.querySelector('.message').textContent = `Correct Number!`;
    document.querySelector('.number').textContent = secretNum;
    guessed = true;
  }
  /////////////// Lecture 77: Implementing High Scores
  function saveHighScore() {
    if (guessed) {
      const highscore = document.querySelector('.highscore').textContent;
      if (!highscore || highscore < score)
        document.querySelector('.highscore').textContent = score;
      // Set score as highscore
    }
  }
```

```
63  ////////////// Lecture 78: Refactoring / Restructuring
    •   Code
64  function outputMsg(str) {
65    document.querySelector('.message').textContent = str;
66    document.querySelector('.score').textContent = --
    •     score; //
67  }
68
```