Setup

Step 1: create folder + files

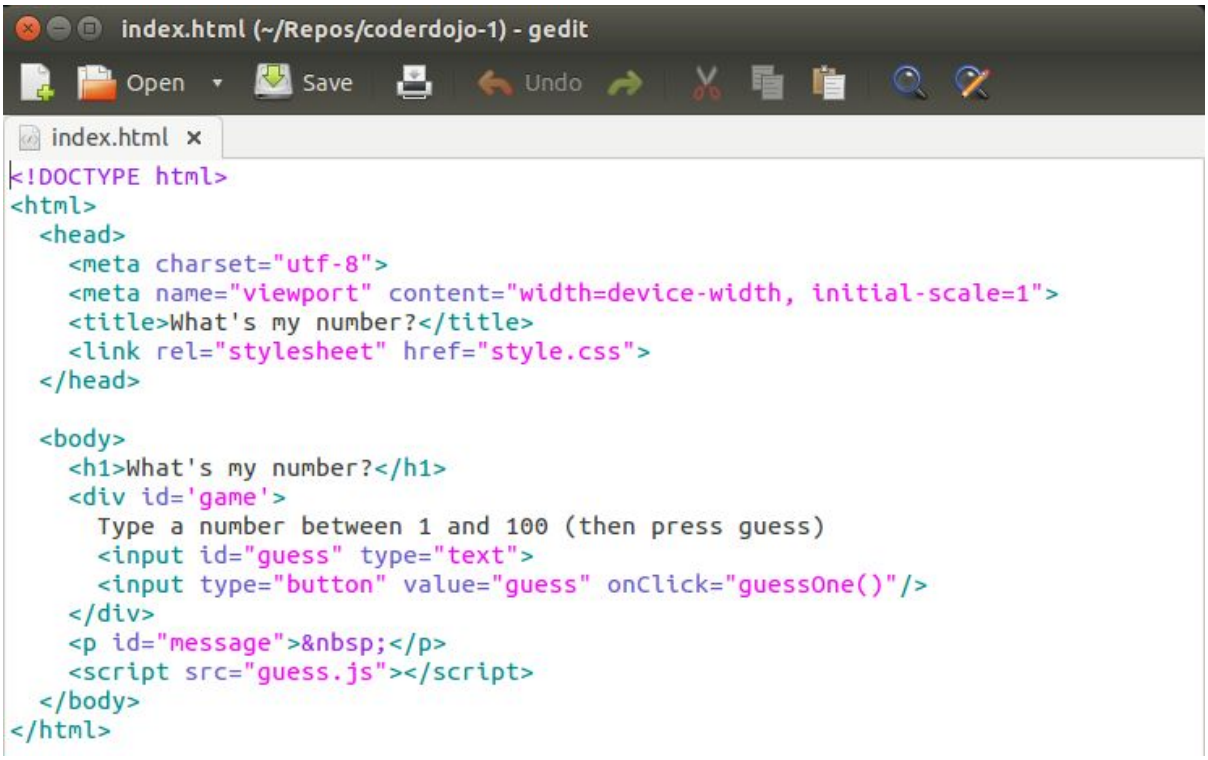
Create a folder and give it a name, f.e. 'number-guessing-game'
In the folder, create two files:
index.html
guess.js

Open the files with a text editor (Text Edit on Mac, Notepad on Windows, or Sublime/VIM if they are installed)

Step 2: Add code to index.html

Paste this code in index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>What's my number?</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <h1>What's my number?</h1>
    <div id='game'>
      Type a number between 1 and 100 (then press guess)
      <input id="guess" type="text">
      <input type="button" value="guess" onClick="guessOne()"/>
    </div>
    <p id="message"> </p>
    <script src="guess.js"></script>
  </body>
</html>
```

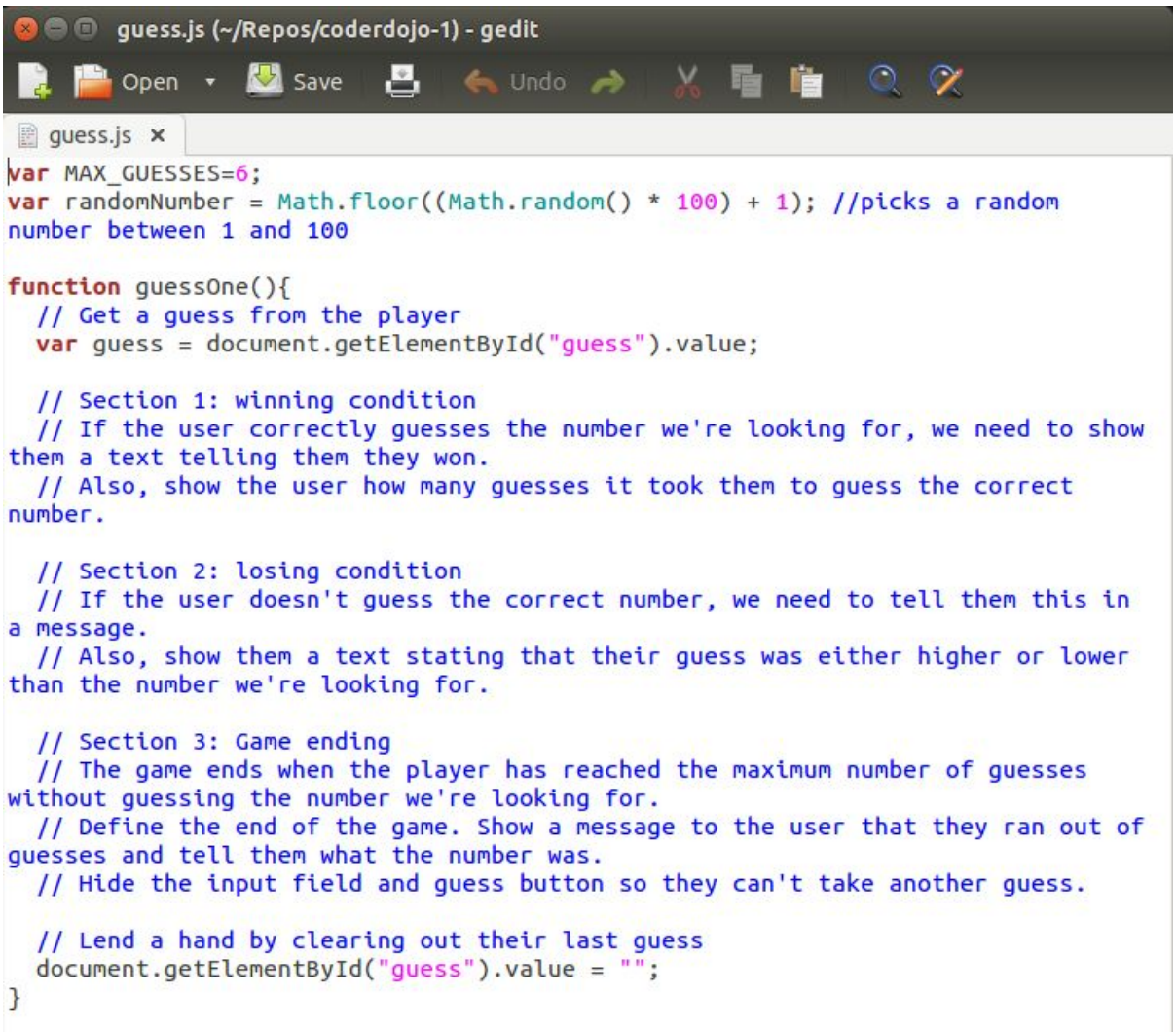


Step 3: Add code to guess.js

Paste this code in guess.js:

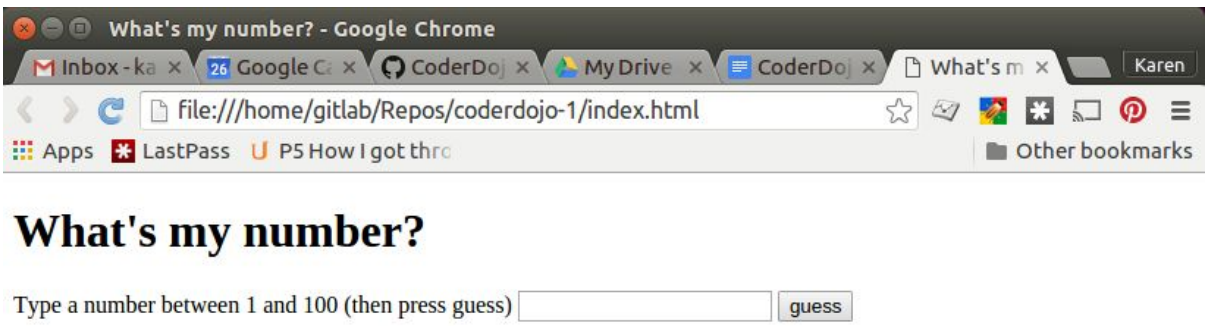
```
var MAX_GUESSES=6;
var randomNumber = Math.floor((Math.random() * 100) + 1); //picks a random number between 1 and 100

function guessOne(){
  // Get a guess from the player
  var guess = document.getElementById("guess").value;

  // Section 1: winning condition
  // If the user correctly guesses the number we're looking for, we need to show them a text telling them they won.
  // Also, show the user how many guesses it took them to guess the correct number.

  // Section 2: losing condition
  // If the user doesn't guess the correct number, we need to tell them this in a message.
  // Also, show them a text stating that their guess was either higher or lower than the number we're looking for.

  // Section 3: Game ending
  // The game ends when the player has reached the maximum number of guesses without guessing the number we're looking for.
  // Define the end of the game. Show a message to the user that they ran out of guesses and tell them what the number was.
  // Hide the input field and guess button so they can't take another guess.
```

```
  // Lend a hand by clearing out their last guess
  document.getElementById("guess").value = "";
}
```



Step 4: Check out the game in a browser

Open index.html in a browser to see what your game looks like.



Section 1: Winning Condition

Useful methods

Equality operators:

- < smaller than
- == equals
- > greater than

Function syntax:

```
function_name(variables) {
        do some operation with the variables;
};
```

If-statement syntax:

```
if (condition) {
        do something;
};
```

Get element on the page and put text in it:

```
document.getElementById("id").innerHTML = 'some text';
```

Answer

```
if (guess == randomNumber) {
        var result = "It took you " + guessCount + " guesses.";
        document.getElementById("message").innerHTML = result;
        return; // prevents saying 'ran out' if guessed in last round
}
```

Section 2: Losing condition

Useful methods

else-if syntax:

```
if (condition) {
        do x;
} else if (another condition) {
        do y;
} else {
        do z;
}
```

Answer

```
if (guess == randomNumber) {
        var result = "It took you " + guessCount + " guesses.";
        document.getElementById("message").innerHTML = result;
        return; // prevents saying 'ran out' if guessed in last round
} else if (guess < randomNumber) {
```

```
        var result = "Guess again. The number is higher than " + guess;
        document.getElementById("message").innerHTML = result;
} else {
        var result = "Guess again. The number is lower than " + guess;
        document.getElementById("message").innerHTML = result;
}
guessCount += 1;
```

Section 3: Game end

Useful methods

Hiding an element:

```
        document.getElementById("id").style.visibility = "hidden";
```

Answer

```
if (guessCount >= MAX_GUESSES) {
        var result = "Sorry, you ran out of guesses. The number was " + randomNumber;
        document.getElementById("message").innerHTML = result;
        document.getElementById("game").style.visibility = "hidden";
};
```

Section 4: Expand/Customize

1. Change the range of guesses to 1-1000. On the webpage, tell the player what the range is
2. Use constants to set the range and to give better messages to the player (f.e. "Your guess is lower than the correct number, but is in range.")
3. Change the name of the variable 'guess' to 'playerGuess'
4. Currently, the player can type in anything, also text (f.e. "zombies"). Use Javascript's parseInt function to convert any guess into a number
5. Change the styling of the webpage.

Let another player play your game and fix any bugs they find.