React Native Devc

Week 1 - Homework - Number Guessing Game in HTML

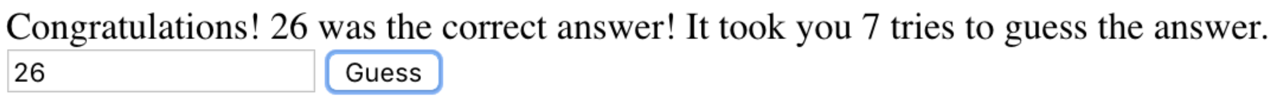
Due Date: 23:59 on Sunday, July 21st, 2019.

Overview: Build a Number Guessing Game in HTML

Project Brief

This assignment is about building a simple `Number Guessing Game in HTML`.

Milestone 1: Setting up your UI elements.



Congratulations! 26 was the correct answer! It took you 7 tries to guess the answer.

There are 3 required UI Elements for this Assigment:

- Message Box: This container will display the notification of the game.

```
<div id="message"></div>
```

- Number Input Box: This box allow you to type in your guessed number.

```
<input type="number" id="number">
```

- Guessing Button: This button will activate your guessing function.

```
<button id="guessButton"></button>
```

Milestone 2: Bind the event to elements.

You can bind the guessing event to the button by setting an onclick attribute for the button. For example,

```
let button = document.getElementById('guessButton');
button.addEventListener('click', function() {
   // What happens when you click the button?
   // Hint: Perhaps what happens next in Milestone 4.
});
```

Milestone 3: Get Value From User Input

Now after you click the guess button, you have to get the number that user has typed in order to compare the `user input value` and `random value`. You can get the value from the `<input /> element` by using `let userGuess = document.getElementById('yourInputId').value`. But you will need to convert the input value into `Integer` because the value type is `String`. Reason we are doing parsing is the machine can't compare `Integer` and `String` because they didn't have the same data type.

Milestone 4: Show the correct message

Last time, you `alert`ed or possibly `console.log`ed when the user guessed. Now, you'll be writing to your message div.

By now, you should know how to get to the message element: use `document.getElementById`. Now, you can set the `innerHTML` to the appropriate message, which can be:

- Sorry your guess is too low
- Sorry your guess is too high
- Congratulations!! You guessed correctly.

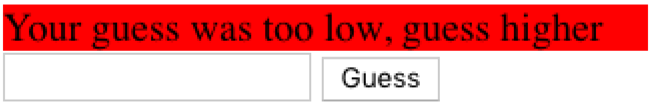
Milestone 5 (Optional): Show user's past guesses

Great!, now our game is working fine. Let just step up the game a little bit, why don't we show the `past guesses` of user like in the last assignment.

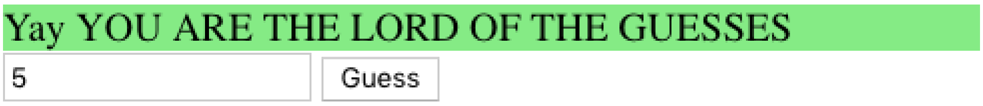
Milestone 6 (Optional): Change message container color

Whenever the user guess wrong, we should color the `message` to `red`, and if they are right, color them `green`.

If they are wrong

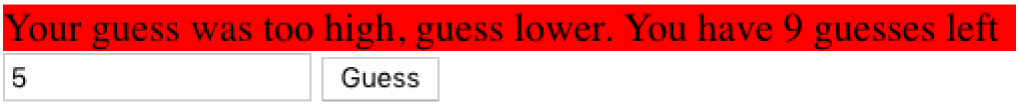

Your guess was too low, guess higher

If they are right

> 
> Yay YOU ARE THE LORD OF THE GUESSES
>
> 5
>
>

Milestone 8 (Optional): Limit person to 10 guesses, show many guesses remaining.

Let's limit the user guesses down to 10. And let show the user how many guesses they have left in `message container`.


Your guess was too high, guess lower. You have 9 guesses left

5