# challenge du week-end

## title: Number Guesser

## Overview

- Build a game where the user guesses a number between 1 and 100.

- Phase 4 is optional. If you execute phase 4, create a separate branch in Git for that work for evaluation (using Git is an extra challenge -- investigate how to initialize a Git repo and make a branch! We'll go over how to do this in class in a few days, so this is a great time to investigate on your own).

## Phase One: Basic Requirements

The application should have the following user interface components:

### Zero State:

- An input field for guessing the number
- A button for submitting a guess
- A button for clearing the input field
- A button that resets the game

### User's Guess State:

- Display the user's most recent guess
- Display results and feedback:
    - If their guess is too high, it should display: "That is too high"
    - If their guess is too low, it should display: "That is too low"
    - If the guess is correct, it should display: "BOOM!"

### Some things to consider:

- Input fields,regardless of their "type" attribute, store their values in strings. You'll need to

use `parseInt()` to turn it back into a number.

# Phase Two: More Better

- The input field should only accept numerical entries, within the defined min and max range
- The application should display an error message if the guess is not a number (e.g. `parseInt()` returns `NaN`).
- The application should display an error if the guess is outside of the range of possible answers.
- The clear button should be disabled if there is nothing to clear.
- The reset button should be disabled if there is nothing to reset.

# Phase Three: Level Up

- Add additional inputs that allow the user to specify the minimum/maximum range.
- Upon successful win, user's range is updated:
    - Every time the user wins a round increase the maximum number by 10.
    - Every time the user wins a round decrease the minimum number by 10.
    - Appropriate UI is incorporated such that user understands what is happening.

(**Pro-tip**: You'll need to adjust the input fields to accept the new minimum and maximum numbers.)

# Phase Four: Choose Your Own Adventure

You have liberty with the UI.

- Make it two player.
- Incorporate a scoring system based on how quickly user guesses correct number.
- Turn it into a game to teach basic arithmetic to kids.

**Layout comps**