

**Due** Feb 6 by 9:30am      **Points** 100      **Submitting** a website url

## Mod 1: The DOM Based Game



## Project Details

Mandatory To Pass:

MVP - Minimum Viable Product

Your dom project must meet these requirements:

1. Built with HTML, CSS and JavaScript (must be visual and use the DOM not the Console)
2. Hosted on Github pages
3. Commits to Github every day
4. A [README.md](#) file with explanations of the technologies used, the approach taken, a link to your live site, installation instructions, unsolved problems, etc. [Here is a great guide on how to write a readme](#) ↗

If you build a story:

1. **Must have two Classes and one instance of each class**
  - Example: make a Dog Class and create a Dog named Sam
  - Example: make a Person Class and create a Person named John
2. **Story must be controlled by buttons**
3. **Must use Flexbox**
4. **Must have a reset button**

If you build a you Game must have:

1. **Must be a two player game** (either against the computer or against another player)
  - Example: Blackjack: A player plays against the dealer. The dealer is the computer - Example: Connect Four: Two players pass the game between themselves to take turns
2. **A win state** - a way for the player to win the game
  - High score can be considered a win state
3. **A lose state** - a way for the player to lose the game
  - Example: Blackjack - a player must be able to lose all of their money with losing hands and cannot play if their bankroll is at 0
  - Example: Connect Four - the other player has won or there are no possible plays left
4. **A way to keep playing if the game is not over**
5. **Multiple rounds to play** - a round must begin, end, and there must be a way to check if the game should continue or the overall game is won or lost
  - Example: Blackjack: a player takes turns playing a hand versus a computer - the player's hand can either win, lose or tie the dealer. If the player has enough money in their bankroll they can keep playing. A player must be able to win several rounds and increase their bankroll - Example: Connect Four: two (non-computer) players take turns adding chips to the board. The game will check if a player won or if the board is full and there are no more plays possible. A player gets four chips in a row (vertically or horizontally)- one person wins, one loses, there are no further plays in this case

### Stretch Goals (Not Mandatory):

## Recommended Features

- A way to reset the board and play again
- CSS to give your game a personal and fun style
- Responsive mobile design
- Work with your instructor to determine additional stretch goals

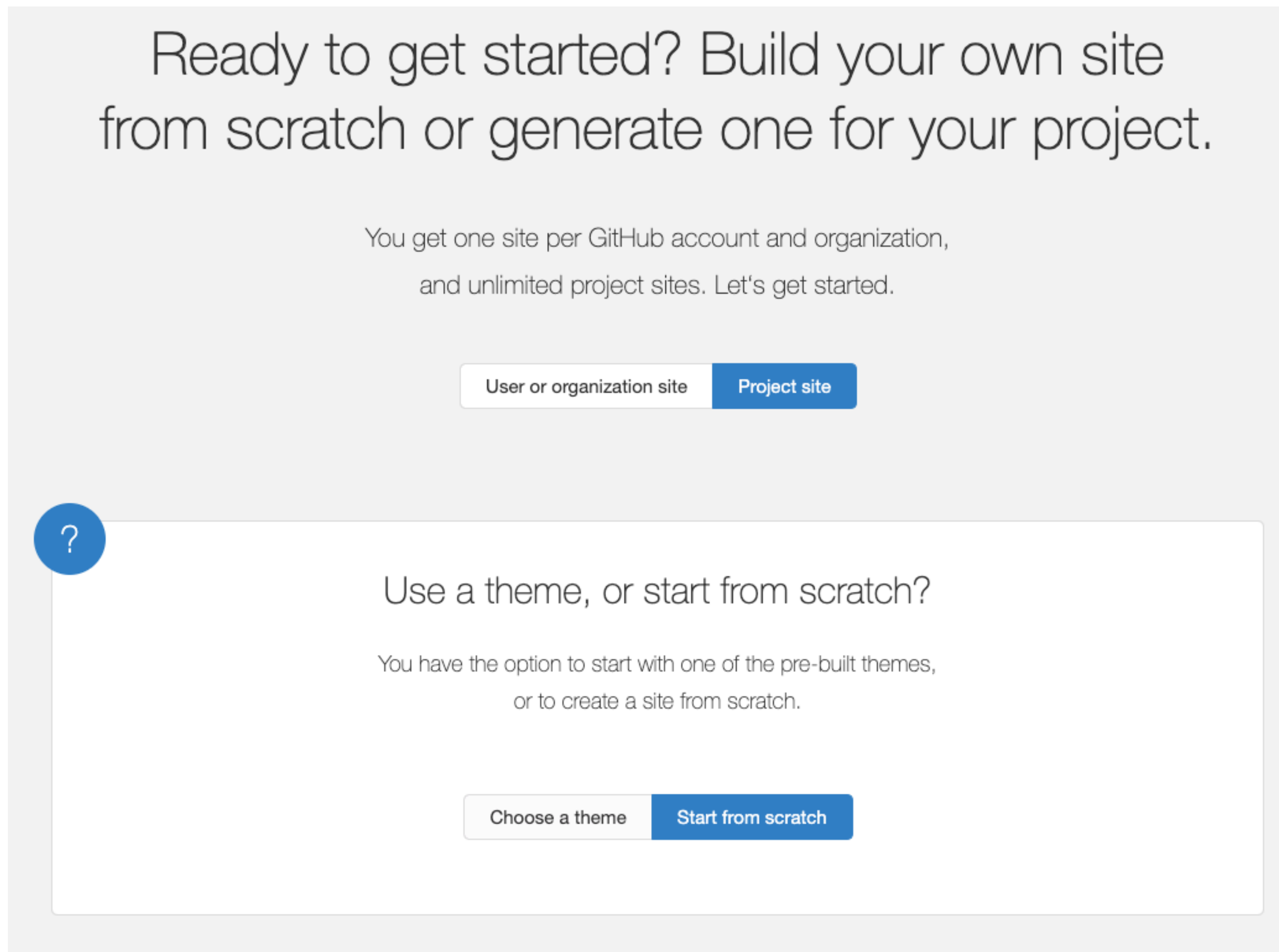
## Make A New Repo

You will be using GitHub, **not** GitHub Enterprise!

**Do not** begin your project within a class repo.

**Do not** clone your project into a class repo.

1. After you come up with your project, [make a new Github repo for your project](#). Remember to keep your repo set to public so you can deploy it.
2. From there, follow the instructions for a Project Site outlined by Github themselves on [github.io](#). To get to the correct instructions, select 'Project site' and then 'Start from scratch'



Note: You can create your `index.html` on Github or you can create it from the terminal (like we've done in class) and push it up, but you will need at least an `index.html` to deploy your site.

► Extra: want your own domain name?

## Technical Demonstration

All projects will be presented to the class. Your presentation should:

- Be approximately 5 minutes in length
- Show off all features of the app
- Explain the technical details
- Explain the technical challenges
- Explain which improvements you might make You will be sharing your game and your code. Be prepared to answer questions from the instructors and other students.

## Meetings with instructors

**Mandatory**

You will turn in a created REPO with a readme that explains what your project should do.

## How to Submit Your Project

You will present your project and show your code to classmates and instructors.

## Where to go for help during project week

1. Seek out help online
2. Seek out help with your classmates
3. Seek out help with our class TA