

YEAH Hours: Assign7

Match the Flag! 🌐🌐🌐

CS106AX Autumn 2025 🍁

Slides based on materials by Ryan Eberhardt, CS106AX 2019

Welcome to Week 8/9 of CS106AX!



Autumn



Week 7



Winter

Excited to be back in JavaScript? My favorite language 💖

Lecture Recap!

Web Programming, Interactors



~10 minutes

Recap and examples on the classroom whiteboard; feel free to ask any questions!

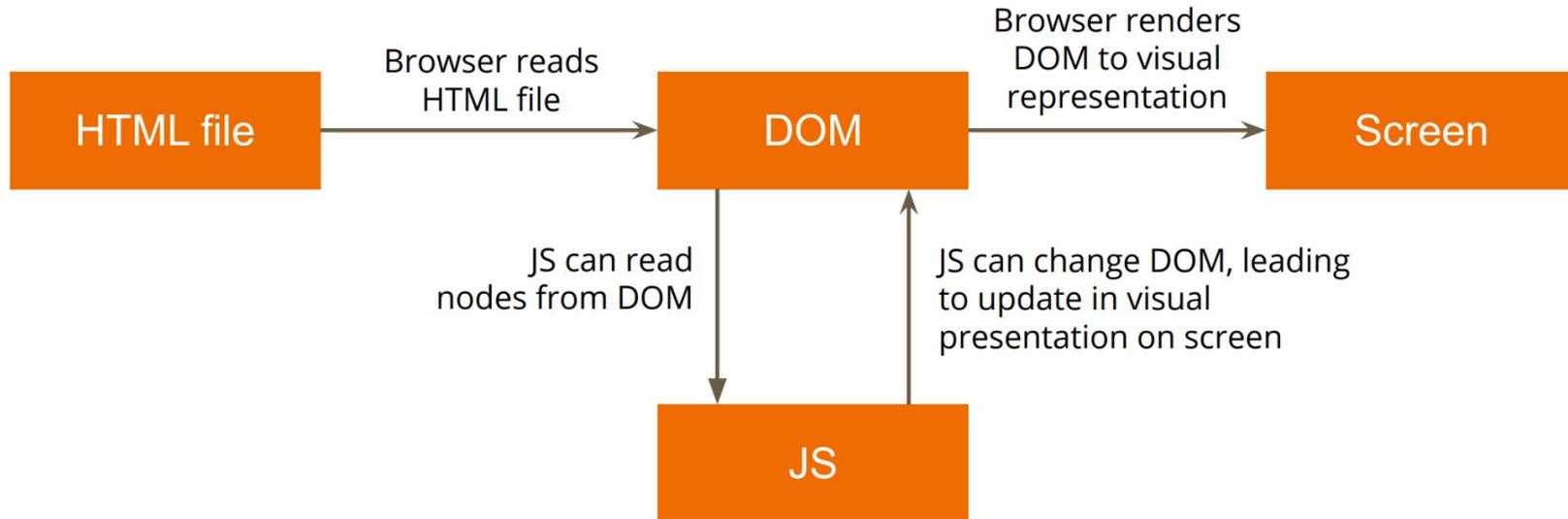


Jerry Picture of the Week 🧡



Big Picture: DOM Tree Interaction

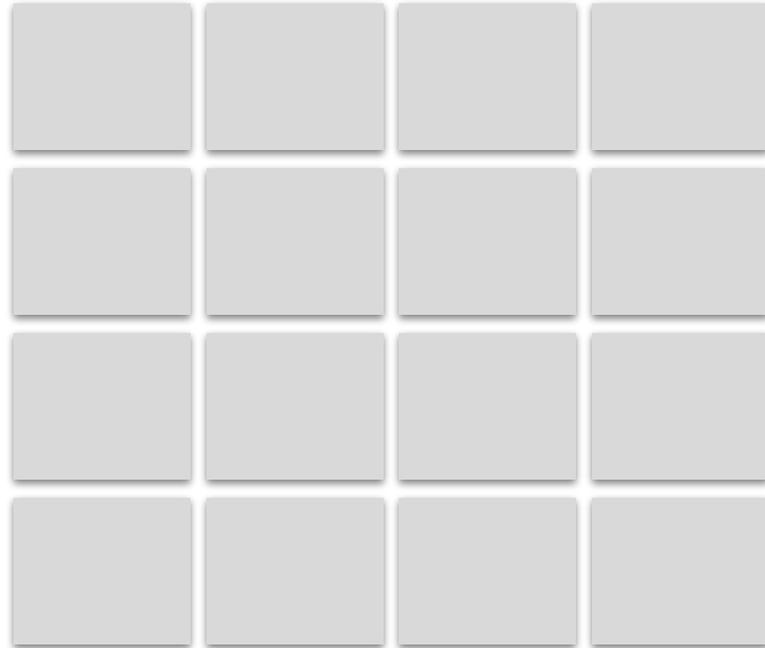
From materials by Ryan Eberhardt





Match the Flag Memory Game

A memory game where a player is presented with a **4 x 4** grid of covered up tiles, under which are one of 8 different flags – they come in pairs.





Match the Flag Memory Game

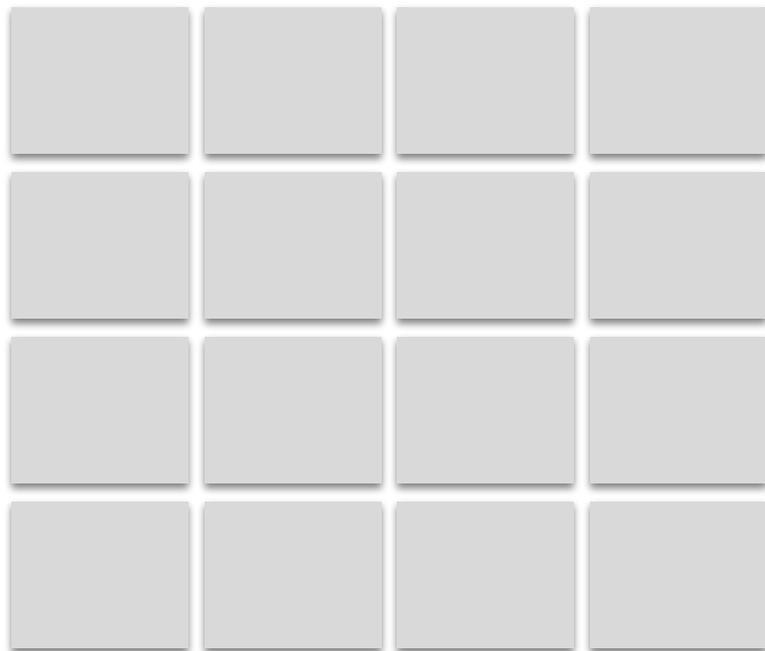
Underneath the tiles, **8 country flags** that comes in pairs! When starting, the user doesn't know the flags under each tile: they must figure it out.





Match the Flag Memory Game

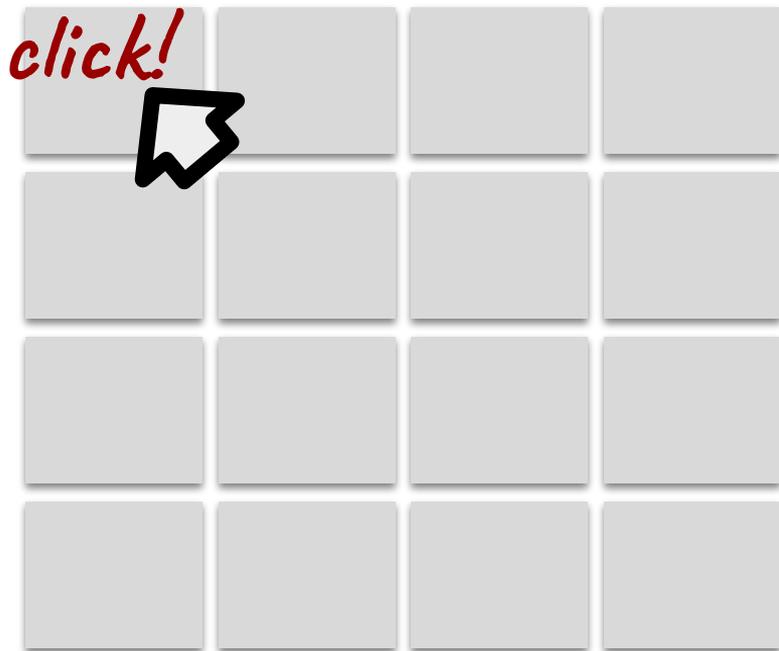
Each turn, the player clicks on two tiles; the program sees if they match.





Match the Flag Memory Game

Each turn, the player clicks on two tiles; the program sees if they match.

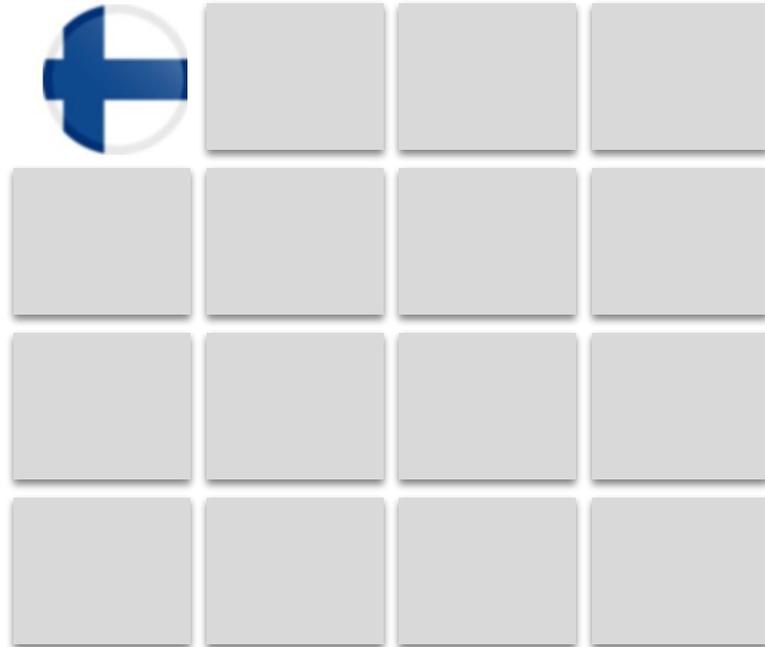




Match the Flag Memory Game

Each turn, the player clicks on two tiles; the program sees if they match.

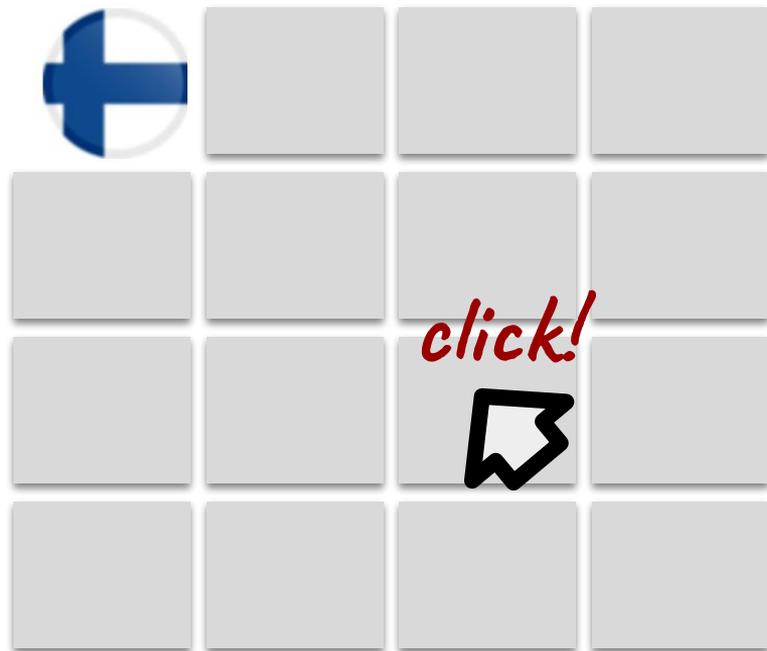
Reveal!





Match the Flag Memory Game

Each turn, the player clicks on two tiles; the program sees if they match.

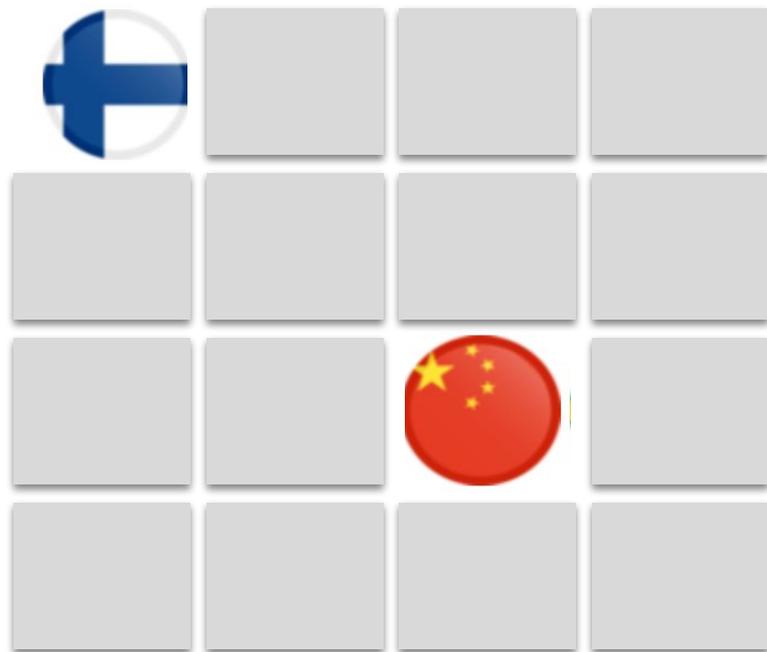


*Click on
another one*



Match the Flag Memory Game

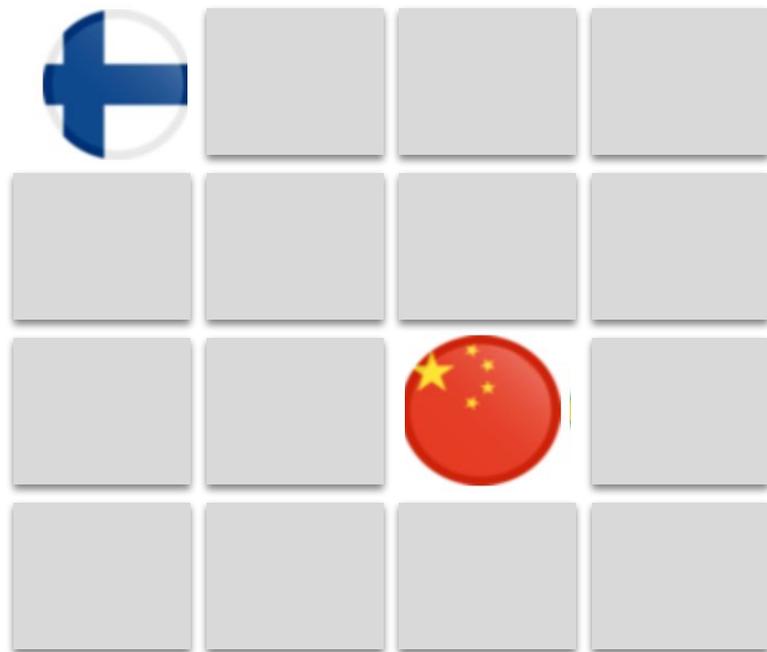
Each turn, the player clicks on two tiles; the program sees if they match.



Reveal!

Match the Flag Memory Game

✗ NO MATCH: Tiles are not the same, so we don't remove them from the grid. We flip back / re-conceal the two tiles, letting the user try again.





Match the Flag Memory Game

✗ NO MATCH: Tiles are not the same, so we don't remove them from the grid. We flip back / re-conceal the two tiles, letting the user try again.



Flip back!



Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath.





Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath.



reveal!



Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath.



*Click on
another one*



Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath.



Reveal!



Match the Flag Memory Game

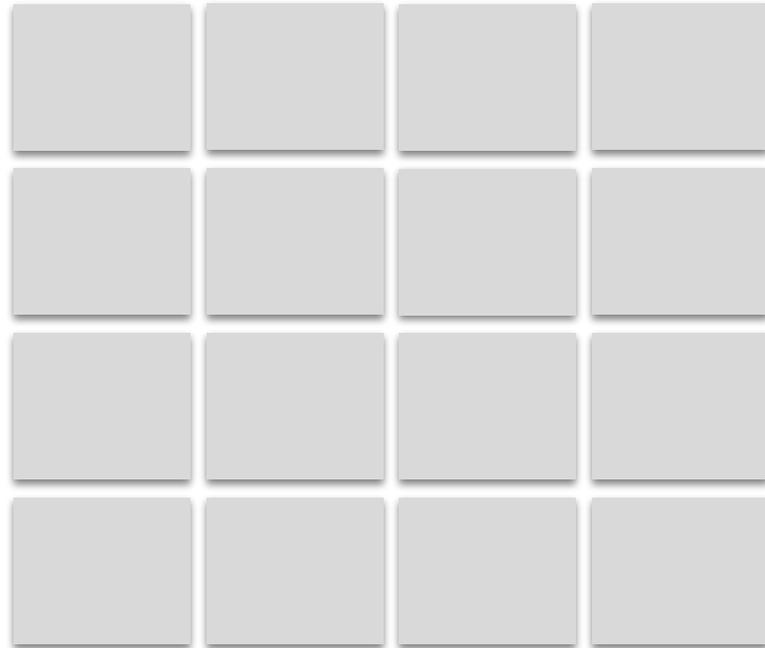
✗ NO MATCH: Tiles are not the same, so we flip them back / re-conceal the tiles, and try again. However, that flag of Finland looks familiar ... 🤔





Match the Flag Memory Game

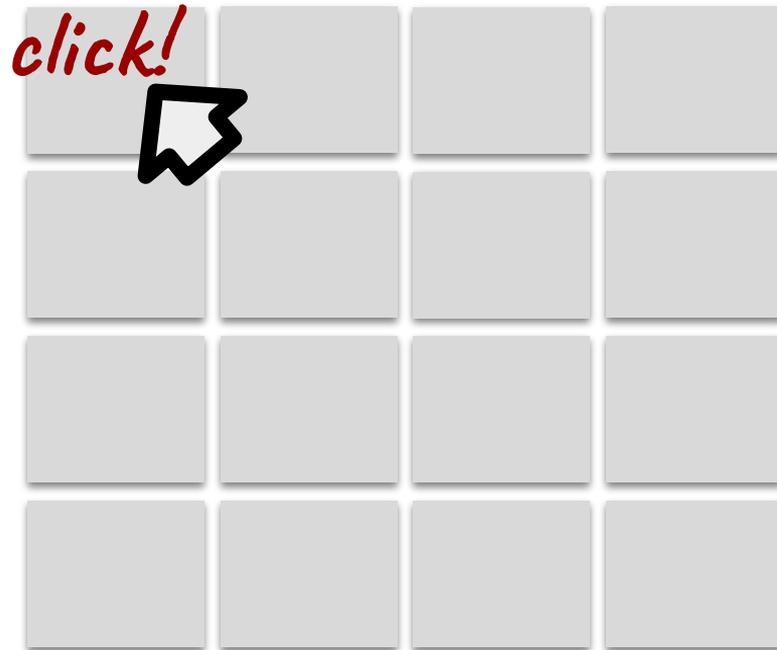
✗ NO MATCH: Tiles are not the same, so we flip them back / re-conceal the tiles, and try again. However, that flag of Finland looks familiar ... 🤔





Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath. *And I seem to remember a flag of Finland also being in the top left corner.*

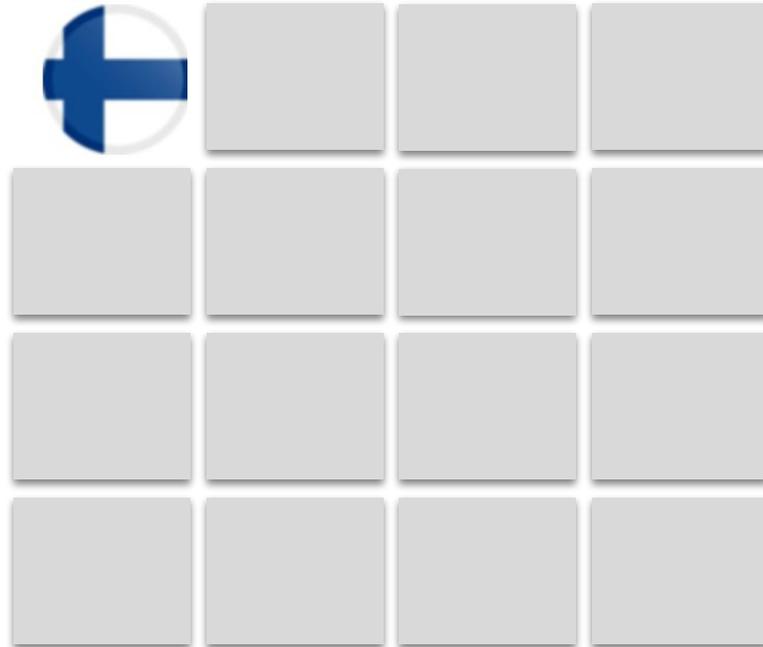




Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath. *And I seem to remember a flag of Finland also being in the top left corner.*

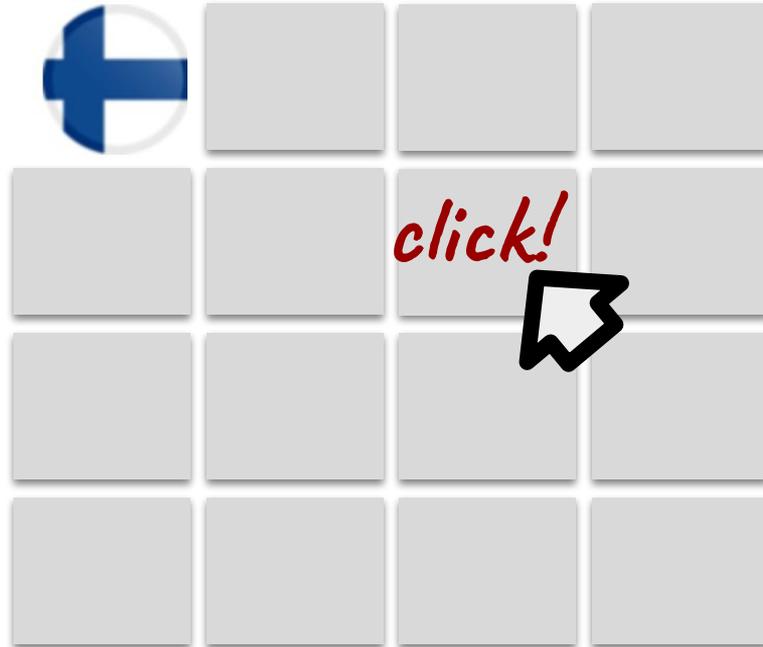
Reveal!





Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath.

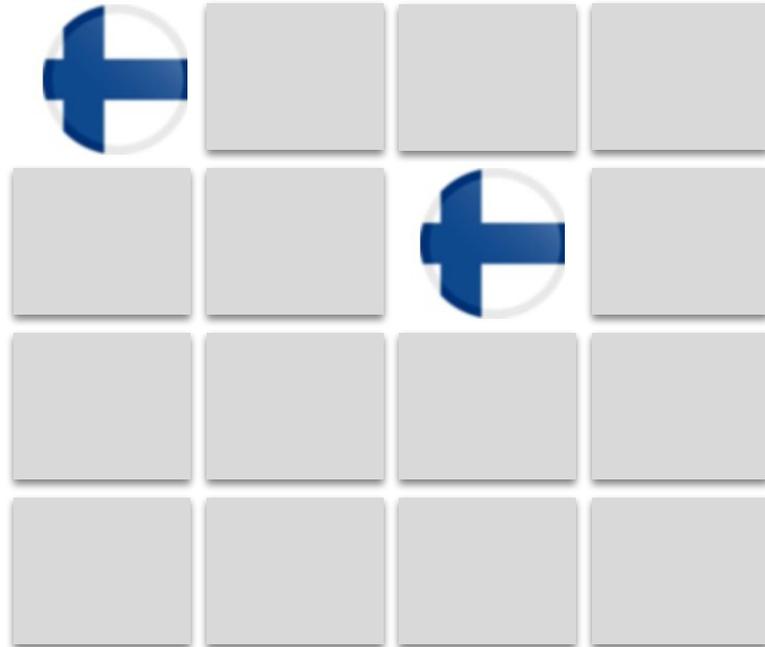


*Click on
another one*



Match the Flag Memory Game

Try Again: Click on two tiles to see which flags are underneath.



Reveal!

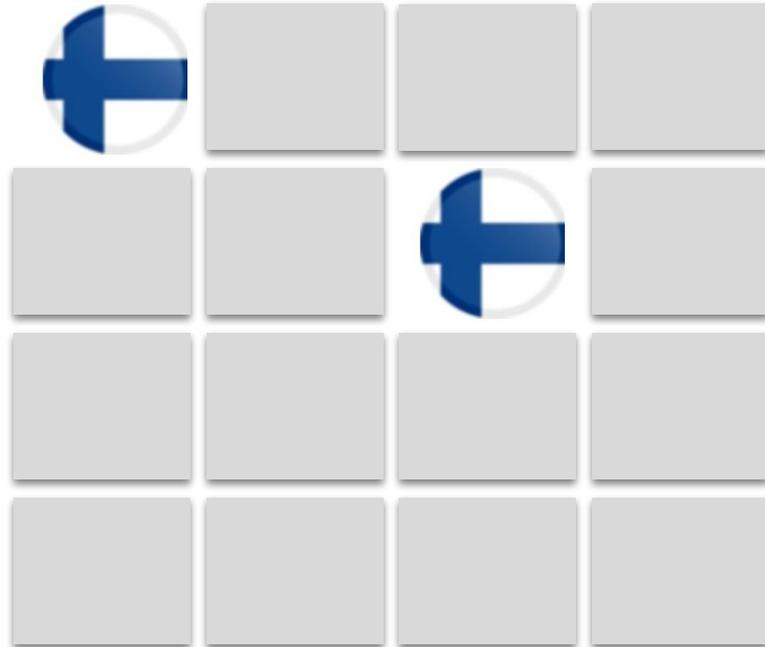
*And they're the
same flag 🎉*



Match the Flag Memory Game



MATCH: Revel in the victory, then clear the two flags from the screen.

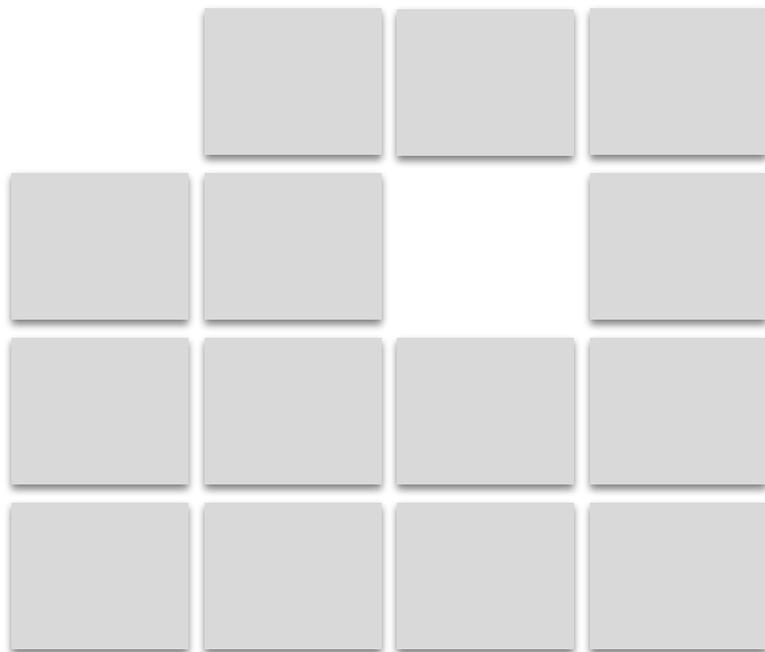




Match the Flag Memory Game



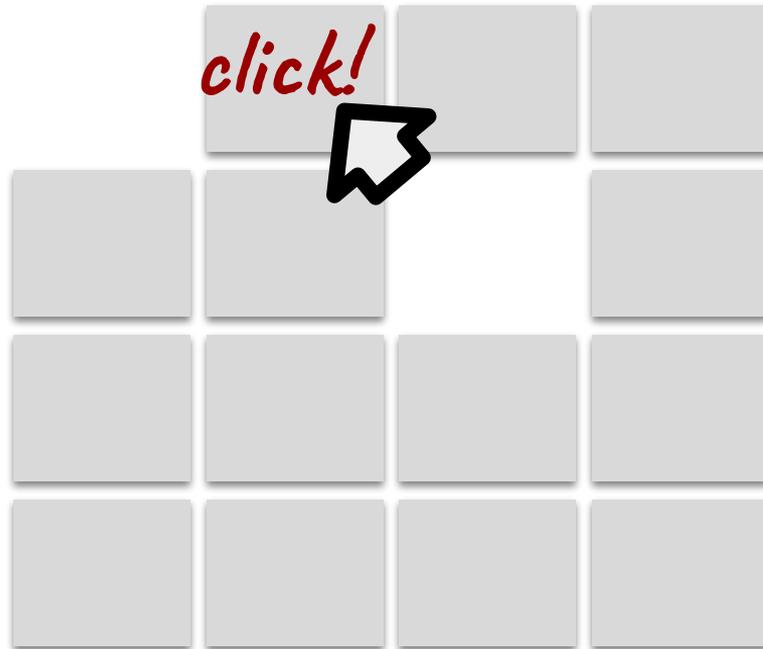
MATCH: Revel in the victory, then clear the two flags from the screen.





Match the Flag Memory Game

 **Goal:** Now, keep repeating this – clicking a pair of flags and going for a match – until all flags have been cleared / all pairs have been discovered.

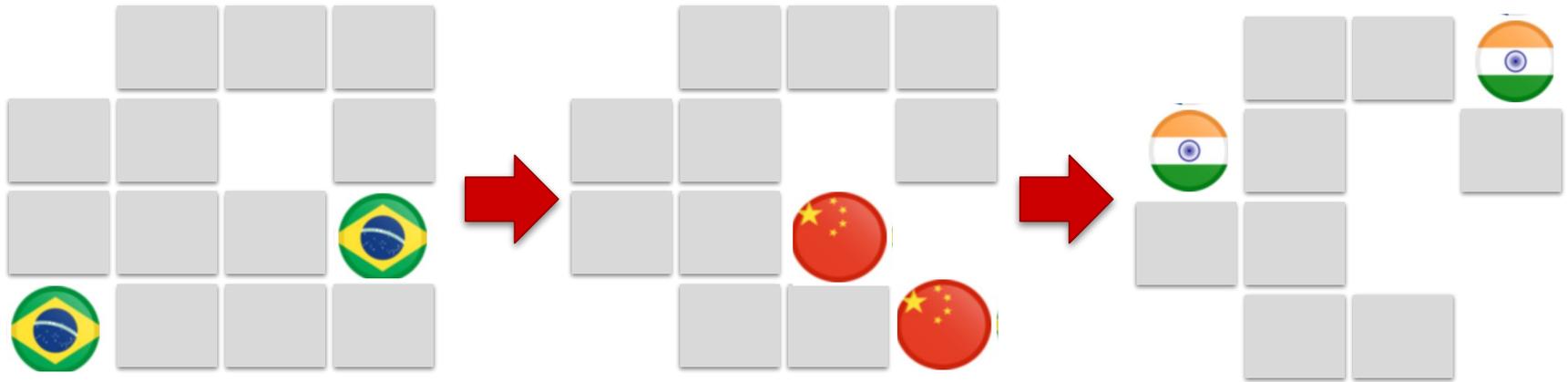




Match the Flag Memory Game



Goal: Now, keep repeating this – clicking a pair of flags and going for a match – until all flags have been cleared / all pairs have been discovered.



Leaving out many mismatches along the way 🙄



Match the Flag Implementation Milestones

Milestone 1	Create, Randomize List of Images 
Milestone 2	Add Images to HTML Board 
Milestone 3	Conceal Images on Board At First 
Milestone 4	Attach Event-Listeners to Images 
Milestone 5	Implement Flag-Matching Gameplay 

Milestone 1: Create, Randomize List of Images



We create an array of 16 image filename strings, 2 per country, and randomize the ordering with `shuffle()`

We can find a `COUNTRIES` array constant in the starter code, along with an `images` directory storing images of each country flag.

```
const COUNTRIES = [  
  "Belize", "Brazil", "China", "Colombia",  
  "Egypt", "Finland", "Greece", "India"  
];
```



```
["images/China.png", "images/Brazil.png", ...]
```

Milestone 2: Add Images to HTML Board

For each filename in the Milestone 1 array, create an `img` node and add it to the `board` div declared in the HTML file.

Use `document.createElement` to create the node. Set the image using `element.setAttribute("src", imageName)`

```
match-the-flags.html — ~/Desktop/assign-7-solution
match-the-flags.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Match The Flag</title>
6     <link rel="stylesheet" type="text/css" href="match-the-flags.css"/>
7     <script type="text/javascript" src="match-the-flags-constants.js"></script>
8     <script type="text/javascript" src="match-the-flags.js"></script>
9   </head>
10  <body>
11    <div id="board"></div>
12  </body>
13 </html>
14
~/Desktop/assign-7-solution/match-the-flags.html 1:1  LF UTF-8 HTML GitHub Git (0)
```

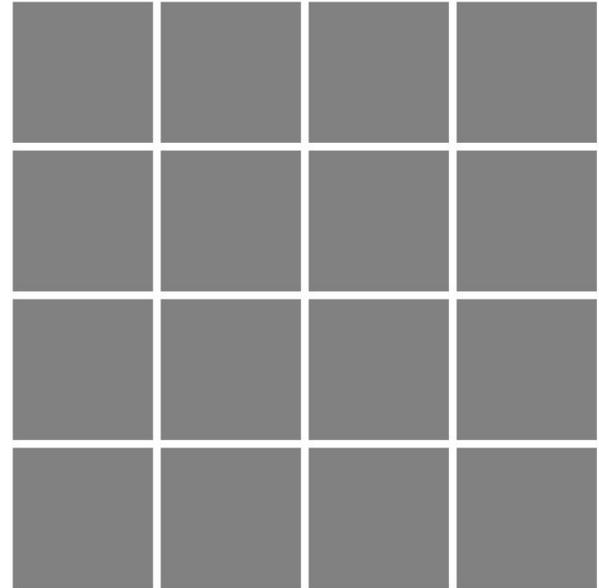


Milestone 3: Conceal Images on Board

We want to cover each tile by showing the cover image (`images/cover.png`) instead of the country flag image.

Thus, we have to replace the `src` attribute with this new cover image for each node.

However, we still want to keep track of which country flag is underneath the tile / should be revealed when we click on it!



✓ Advice: We recommend adding a custom attribute `data-country-image` to each node, storing the original filename of the country flag image.

Milestone 3: Conceal Images on Board

✓ You can confirm this is done correctly by looking for the `data-country-image` attributes in the Chrome inspector, or via `console.log()` to manually print out the image nodes / attributes



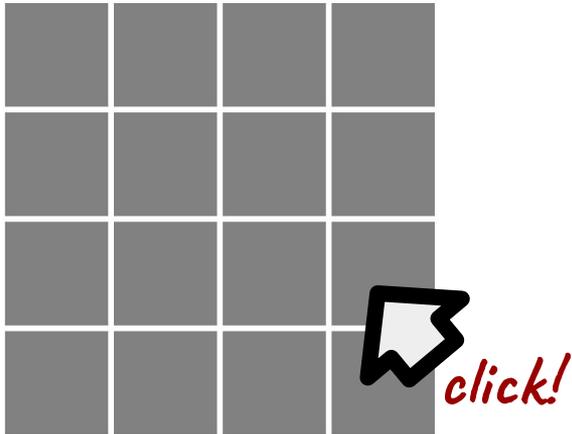
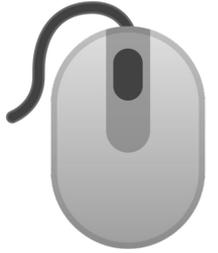
```
Inspector Console Debugger Network Style Editor Performance
Search HTML
<!DOCTYPE html>
<html> event scroll
  <head> </head>
  <body class="vsc-initialized">
    <div id="board">
       event
       event
    </div>
  </body>
</html>
html > body.vsc-initialized > div#board
```

Milestone 4: Attach Event-Listeners to Images



Add a click-handling function to each `img` node that toggles between the gray cover image and the country flag.

Don't create any new nodes! You only have to modify the existing `img` node that was clicked on, and modify its `src` attribute.

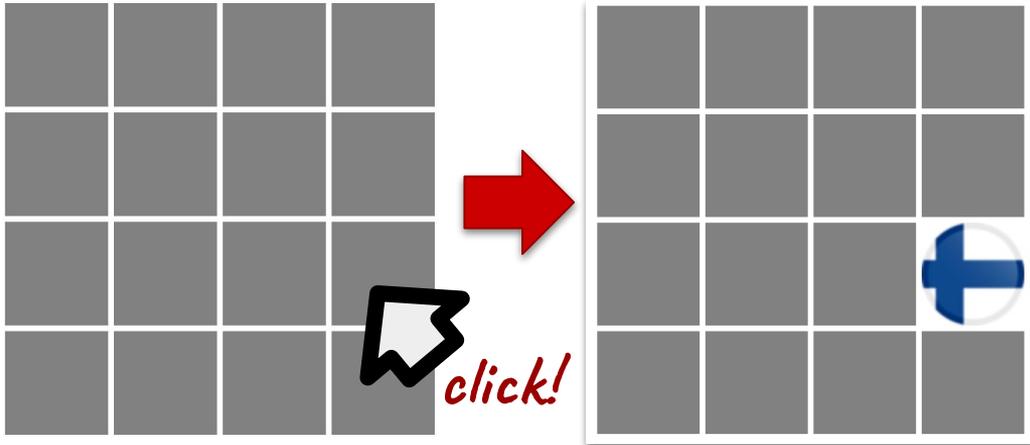
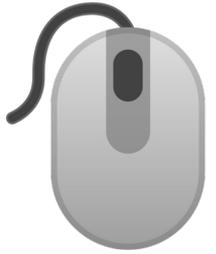


Milestone 4: Attach Event-Listeners to Images



Add a click-handling function to each `img` node that toggles between the gray cover image and the country flag.

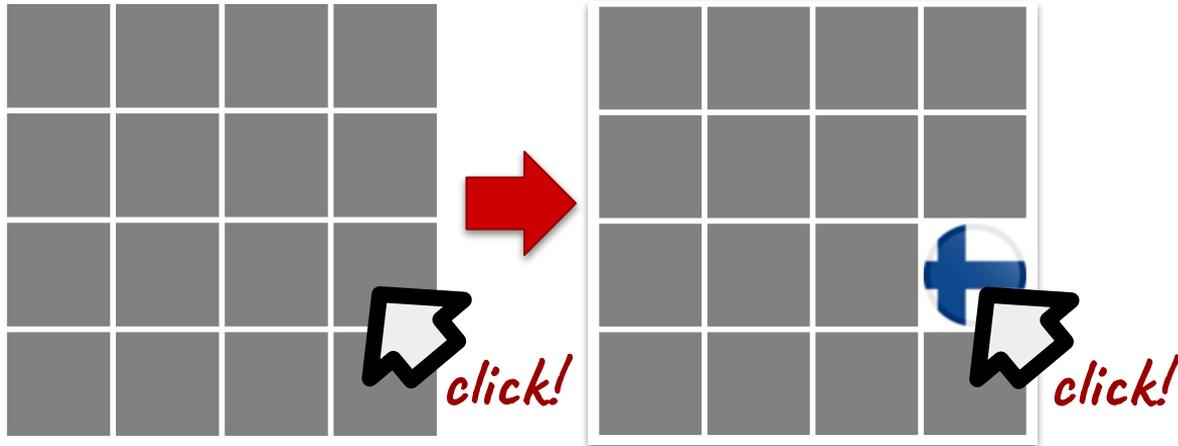
Don't create any new nodes! You only have to modify the existing `img` node that was clicked on, and modify its `src` attribute.



Milestone 4: Attach Event-Listeners to Images

Add a click-handling function to each `img` node that toggles between the gray cover image and the country flag.

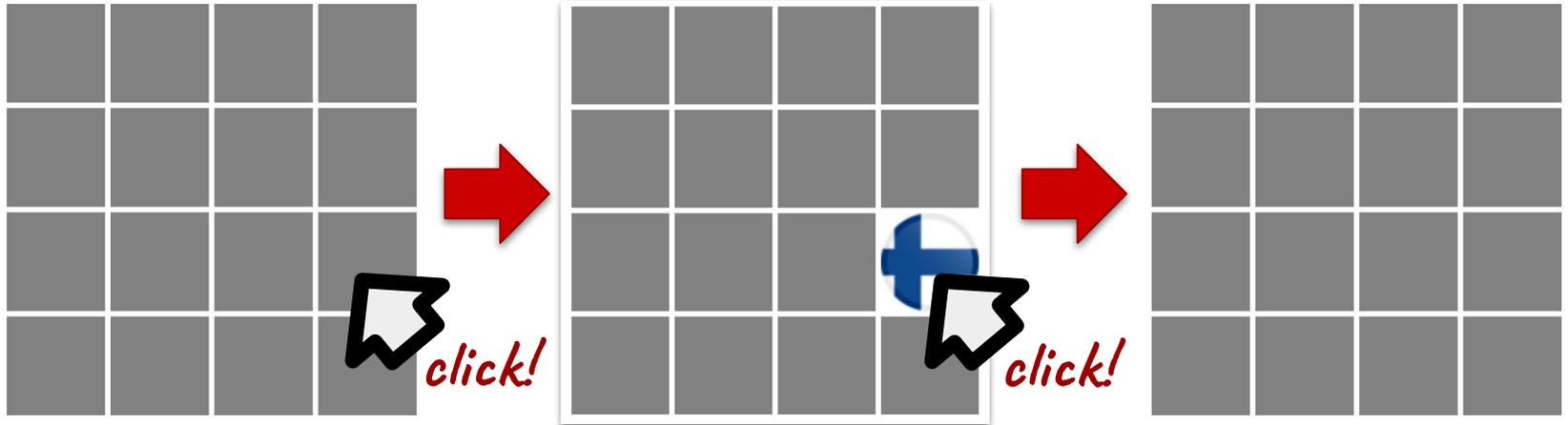
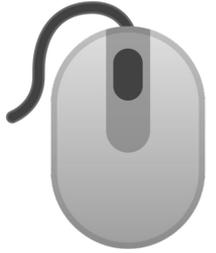
Don't create any new nodes! You only have to modify the existing `img` node that was clicked on, and modify its `src` attribute.



Milestone 4: Attach Event-Listeners to Images

Add a click-handling function to each `img` node that toggles between the gray cover image and the country flag.

Don't create any new nodes! You only have to modify the existing `img` node that was clicked on, and modify its `src` attribute.

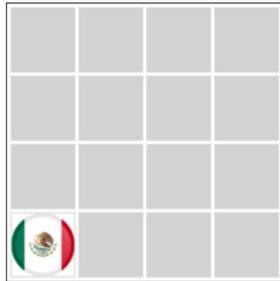


Milestone 5: Implement Flag-Matching Gameplay 🎮

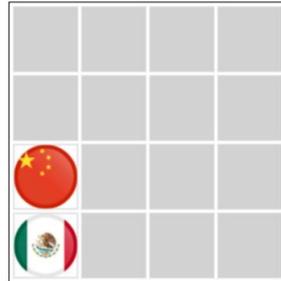
Lastly, we implement the actual gameplay of **Match the Flag!** So that the program keep tracks of revealed tiles and whether they match.



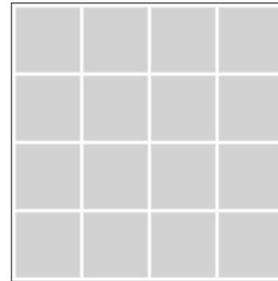
When a 2nd tile is clicked, we should check whether it's the same flag as the 1st.



after first click



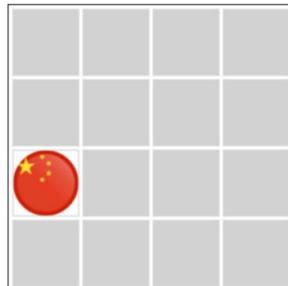
after second click



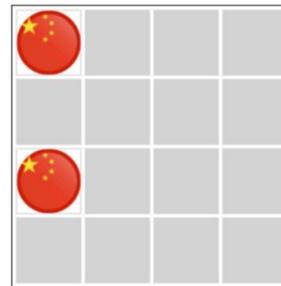
after one second delay

✗ If the flags are different:

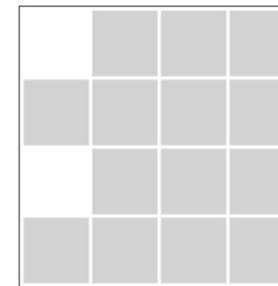
Set a 1-second timeout that covers the tiles back up; during this timeout, clicks do nothing



after first click



after second click



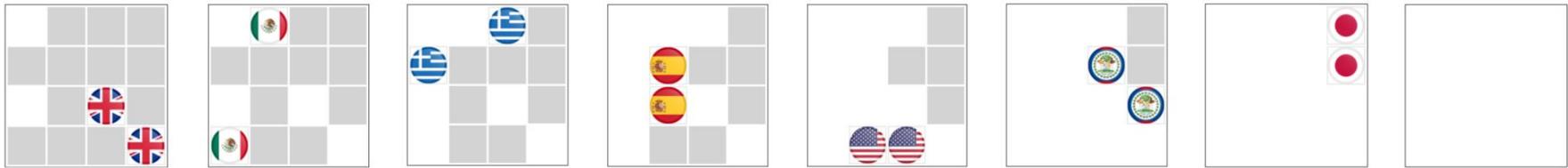
after one second delay

✓ If the flags are the same:

Set a 1-second timeout that changes both tiles to a transparent **MATCHED_IMAGE**.

Milestone 5: Implement Flag-Matching Gameplay 🎮

Eventually, after enough game turns ... all tiles will become the transparent `MATCHED_IMAGE`, upon which clicks do nothing – the tile's already been matched.



🎃 This is the trickiest milestone! **A few edge cases to consider:**

- ❑ If you click a tile, then the same tile again for the 2nd click, do nothing (i.e. don't cover tile back, and certainly don't treat the 2nd click as a correct "match")
- ❑ If you click during the 1-second timeout (while 2 country flags are being shown), don't do anything. There should never be > 2 country flags shown at once.
- ❑ Clicking on a matched / transparent tile should do nothing.

Interactive examples on the classroom board!

Mouse Event-Listeners, Timers,
Working on the Flag Game's DOM Tree



Keep up the amazing work! :)

Please feel free to reach out if you have any questions, e.g., in the office hours following the overview portion of the YEAH session.

