ECA
CHECK-IN

Welcome to Hack Club!

Workshop: Memory Game

11/8/2021

BENEFITS

Memory Game

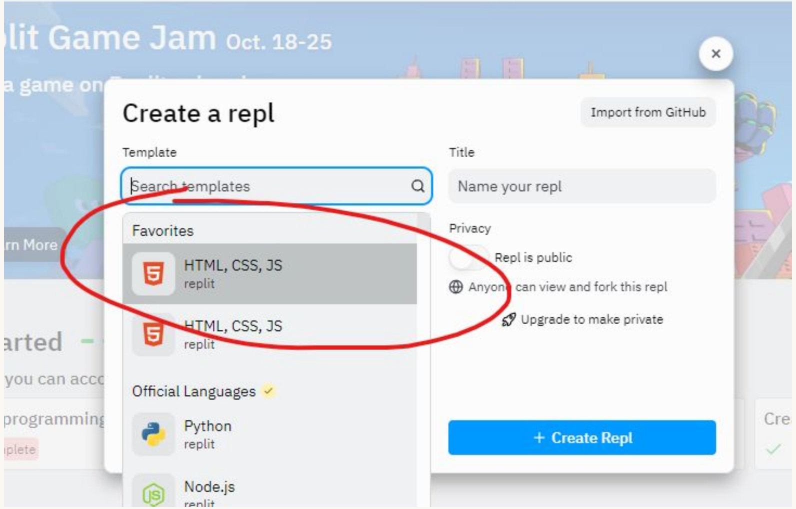
Objective

Memory Game rules

- You will start by flipping over one card
- If the next card you flip matches, both will be marked as “matched”
- If the next card you flip does not match, both will be flipped back
- Goal is to correctly match all cards

Setting Up

Create a new repl



Create a repl

Template

Search templates

Favorites

HTML, CSS, JS
replit

HTML, CSS, JS
replit

Official Languages ✓

Python
replit

Node.js
replit

Title

Name your repl

Privacy

Repl is public

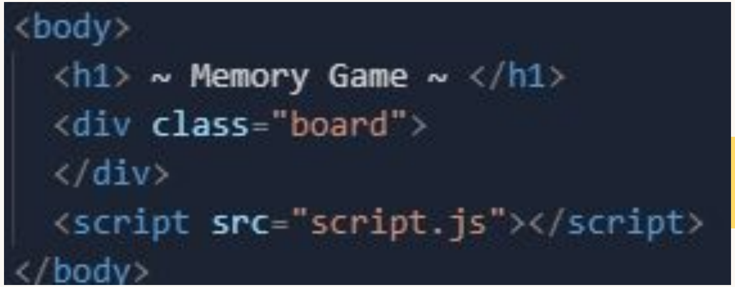
Anyone can view and fork this repl

Upgrade to make private

Setup Basic HTML

We will illustrate our charts on the <canvas> element.

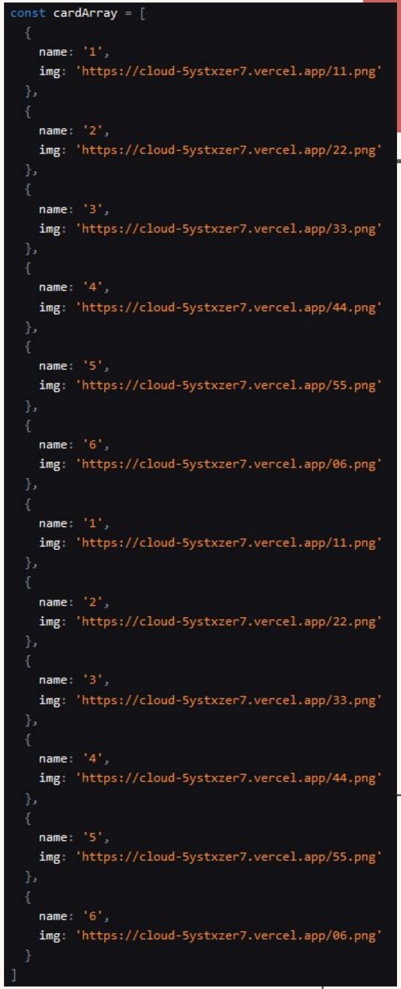
1. Create a heading using the <**h1**> tag
2. Then, create a <**div**> container and assign it the class “board”



```
<body>
  <h1> ~ Memory Game ~ </h1>
  <div class="board">
  </div>
  <script src="script.js"></script>
</body>
```

Setup JS

1. Set window.onload to a function to do stuff when the page loads
2. Create an array of cards (we're going to base the rest of our code off of this)



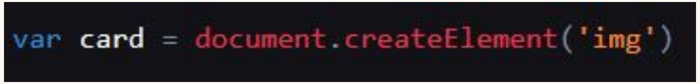
```
const cardArray = [
  {
    name: '1',
    img: 'https://cloud-5ystxzer7.vercel.app/11.png'
  },
  {
    name: '2',
    img: 'https://cloud-5ystxzer7.vercel.app/22.png'
  },
  {
    name: '3',
    img: 'https://cloud-5ystxzer7.vercel.app/33.png'
  },
  {
    name: '4',
    img: 'https://cloud-5ystxzer7.vercel.app/44.png'
  },
  {
    name: '5',
    img: 'https://cloud-5ystxzer7.vercel.app/55.png'
  },
  {
    name: '6',
    img: 'https://cloud-5ystxzer7.vercel.app/06.png'
  },
  {
    name: '1',
    img: 'https://cloud-5ystxzer7.vercel.app/11.png'
  },
  {
    name: '2',
    img: 'https://cloud-5ystxzer7.vercel.app/22.png'
  },
  {
    name: '3',
    img: 'https://cloud-5ystxzer7.vercel.app/33.png'
  },
  {
    name: '4',
    img: 'https://cloud-5ystxzer7.vercel.app/44.png'
  },
  {
    name: '5',
    img: 'https://cloud-5ystxzer7.vercel.app/55.png'
  },
  {
    name: '6',
    img: 'https://cloud-5ystxzer7.vercel.app/06.png'
  }
]
```

Create 3 constant variables

1. board -> document.querySelector(‘.board’)
2. placeholder -> ‘https://cloud-5ystxzer7.vercel.app/7placeholder.png’
3. blank -> ‘https://cloud-5ystxzer7.vercel.app/6blank.png’

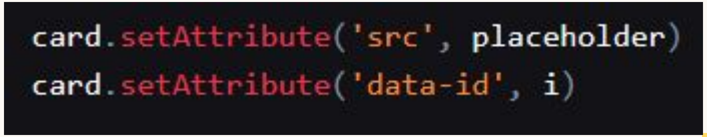
createBoard() function

1. Create a function with the name createBoard
2. Create a for loop the iterates over the array of cards
3. Create a img element called card using document.createElement(‘img’)

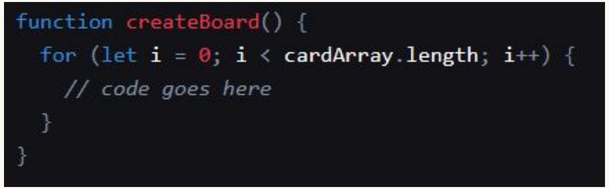


```
var card = document.createElement('img')
```

4. Set the following attributes of the card using card.setAttribute()
    a. ‘src’ -> placeholder
    b. ‘data-id’ -> placeholder



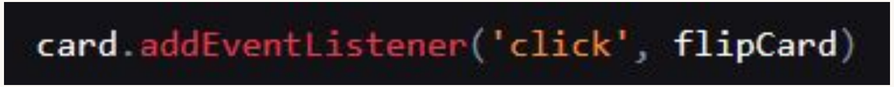
```
card.setAttribute('src', placeholder)
card.setAttribute('data-id', i)
```



```
function createBoard() {
  for (let i = 0; i < cardArray.length; i++) {
    // code goes here
  }
}
```

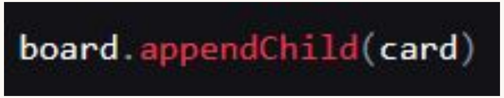
createBoard() continued

5. Add an event listener using card.addEventListener()
    a. ‘click’ -> flipCard (function we’re gonna implement later)



```
card.addEventListener('click', flipCard)
```

6. Append the card to the board using .appendChild()



```
board.appendChild(card)
```

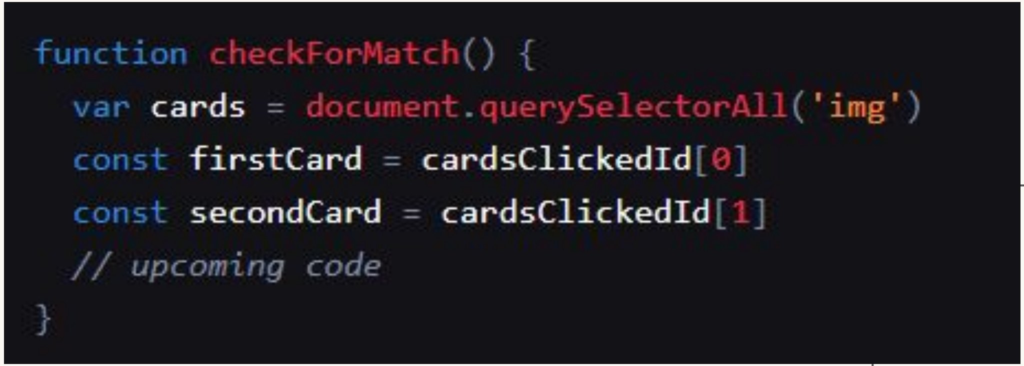
flipCard() function

1. Declare some global arrays
    1. cardsClicked
    2. cardsClickedId
    3. cardsMatched
2. Create the flipCard() function
3. Get the cardId by using this.getAttribute('data-id')
4. Get the name of the card using cardId and add to cardsClicked
5. Add the cardId to cardsClickedId

Match Card

When the second card is clicked, we need to check if it’s a match.

- Let’s create a **checkForMatch()** function

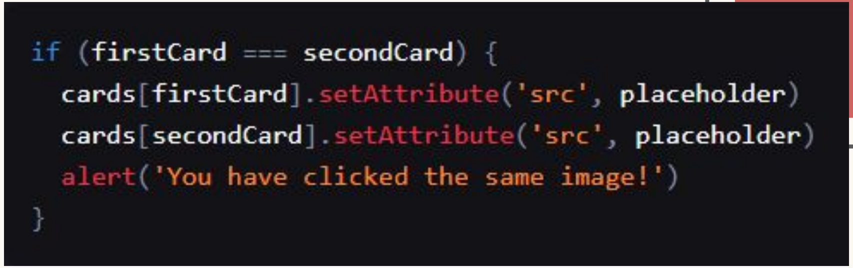


```
function checkForMatch() {
  var cards = document.querySelectorAll('img')
  const firstCard = cardsClickedId[0]
  const secondCard = cardsClickedId[1]
  // upcoming code
}
```

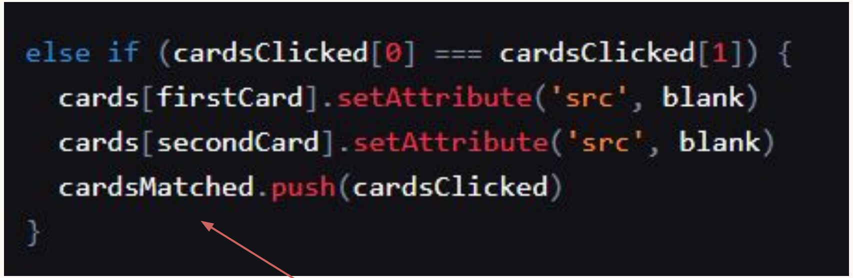
Match Card cont.

In the checkForMatch() function…

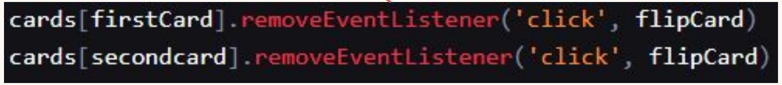
1. Check if the cards are the same (if the same card is clicked twice)
2. Else, check if the two cards match (different cards, same image)
   a. Remove click listener for matched pairs
3. Else (if cards don’t match) flip the cards back.



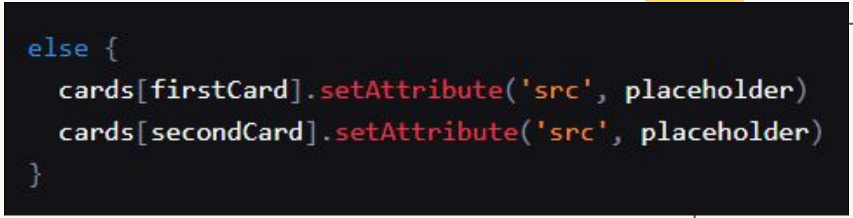
```
if (firstCard === secondCard) {
  cards[firstCard].setAttribute('src', placeholder)
  cards[secondCard].setAttribute('src', placeholder)
  alert('You have clicked the same image!')
}
```



```
else if (cardsClicked[0] === cardsClicked[1]) {
  cards[firstCard].setAttribute('src', blank)
  cards[secondCard].setAttribute('src', blank)
  cardsMatched.push(cardsClicked)
}
```



```
cards[firstCard].removeEventListener('click', flipCard)
cards[secondcard].removeEventListener('click', flipCard)
```



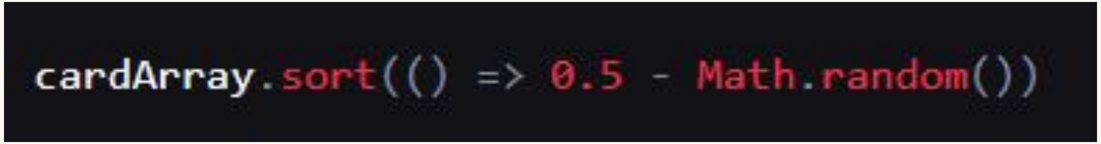
```
else {
  cards[firstCard].setAttribute('src', placeholder)
  cards[secondCard].setAttribute('src', placeholder)
}
```

Sorting the cards

You might notice that the cards are in the same order every time. We have to shuffle the **cardArray** every time before creating the board.

We can do this using the **sort()** method.

Add the following line after the **cardArray**, before the constants we created.



```
cardArray.sort(() => 0.5 - Math.random())
```

Hacking time

Possible Improvements

Add more levels (themes?)

Add more levels or a bigger grid! Maybe add different themes? Be creative!

Improve visuals

Use HTML and CSS to make the game look better - fonts? better ui? etc.

More features!

Add scoring! Timer! Special game modes? Make it your own.