**MEMORY GAME**

**Made with- HTML,CSS,JavaScript**

**Topics covered in this game:-**

1. **Push()**
2. **querySelector()**
3. **setAttribute()**
4. **getAttribute()**
5. **createElement()**
6. **appendChild()**
7. **Math.random()**
8. **Sort()**
9. **For Loops**

**This game is developed by the basic concepts of HTML,CSS, and JavaScript.**

1. **Push()-** The **push() method** adds one or more elements to the end of an array and returns the new length of the array.
2. **querySelector()-** The **querySelector**() is a method of the Element interface. The **querySelector**() allows you to find the first element that matches one or more CSS selectors. You can call the **querySelector**() method on the document or any HTML element. ... If no element matches the CSS selectors, the **querySelector**() returns null .
3. **setAttribute()-** The **setAttribute**() method is used to set or add an attribute to a particular element and provides a value to it. If the attribute already exists, it only set or changes the value of the attribute. ... The attribute name automatically converts into lowercase when we use it on an HTML element.
4. **getAttribute()- getAttribute**() method is used to get the value of the attribute of the element. By specifying the name of the attribute, it can get the value of that element. To get the values from non-standard attributes we can **use getAttribute**() method. Syntax: It returns the string containing the attribute's value.
5. **createElement()-**The **createElement**() method creates an Element Node with the specified name. After the element is created, use the element. appendChild() or element. insertBefore() method to insert it to the document.
6. **appendChlid()-** The **appendChild**() method allows you to add a node to the end of the list of child nodes of a specified parent node. In this method, the childNode is the node to **append** to the given parent node. The **appendChild**() returns the appended child.
7. **Math.random()-**The **Math**. **random**() function returns a floating-point, pseudo-**random** number in the range 0 to less than 1 (inclusive of 0, but not 1) with approximately uniform distribution over that range — which you can then scale to your desired range.\
8. **Sort()-**The **sort()** method allows you to **sort** elements of an array in place. Besides returning the **sorted** array, the **sort()** method changes the positions of the elements in the original array. By default, the **sort()** method **sorts** the array elements in ascending order with the smallest value first and largest value last.
9. **For Loops-**The for...of statement creates a **loop** iterating over iterable objects, including: built-in String , Array , array-like objects (e.g., arguments or NodeList ), TypedArray , Map , Set , and user-defined iterables.

**HTML Code used-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Memory Game</title>

    <link rel="stylesheet" href="style.css"></link>

    <script src="script.js" charset="utf-8"></script>

</head>

<body>

     <h3>Score:<span id="result"></span></h3>

     <div class="grid"></div>

</body>

</html>

**CSS Code used-**

.grid{

    display: flex;

    flex-wrap: wrap;

    height: 300px;

    width: 400px;

}

**JavaScript Code used-**

document.addEventListener('DOMContentLoaded', () => {

  //card options

  const cardArray = [

    {

      name: 'fries',

      img: 'images/fries.png'

    },

    {

      name: 'cheeseburger',

      img: 'images/cheeseburger.png'

    },

    {

      name: 'ice-cream',

      img: 'images/ice-cream.png'

    },

    {

      name: 'pizza',

      img: 'images/pizza.png'

    },

    {

      name: 'milkshake',

      img: 'images/milkshake.png'

    },

    {

      name: 'hotdog',

      img: 'images/hotdog.png'

    },

    {

      name: 'fries',

      img: 'images/fries.png'

    },

    {

      name: 'cheeseburger',

      img: 'images/cheeseburger.png'

    },

    {

      name: 'ice-cream',

      img: 'images/ice-cream.png'

    },

    {

      name: 'pizza',

      img: 'images/pizza.png'

    },

    {

      name: 'milkshake',

      img: 'images/milkshake.png'

    },

    {

      name: 'hotdog',

      img: 'images/hotdog.png'

    }

  ]

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

  const grid = document.querySelector('.grid')

  const resultDisplay = document.querySelector('#result')

  let cardsChosen = []

  let cardsChosenId = []

  let cardsWon = []

  //create your board

  function createBoard() {

    for (let i = 0; i < cardArray.length; i++) {

      const card = document.createElement('img')

      card.setAttribute('src', 'images/blank.png')

      card.setAttribute('data-id', i)

      card.addEventListener('click', flipCard)

      grid.appendChild(card)

    }

  }

  //check for matches

  function checkForMatch() {

    const cards = document.querySelectorAll('img')

    const optionOneId = cardsChosenId[0]

    const optionTwoId = cardsChosenId[1]

    if(optionOneId == optionTwoId) {

      cards[optionOneId].setAttribute('src', 'images/blank.png')

      cards[optionTwoId].setAttribute('src', 'images/blank.png')

      alert('You have clicked the same image!')

    }

    else if (cardsChosen[0] === cardsChosen[1]) {

      alert('You found a match')

      cards[optionOneId].setAttribute('src', 'images/white.png')

      cards[optionTwoId].setAttribute('src', 'images/white.png')

      cards[optionOneId].removeEventListener('click', flipCard)

      cards[optionTwoId].removeEventListener('click', flipCard)

      cardsWon.push(cardsChosen)

    } else {

      cards[optionOneId].setAttribute('src', 'images/blank.png')

      cards[optionTwoId].setAttribute('src', 'images/blank.png')

      alert('HARD LUCK')

    }

    cardsChosen = []

    cardsChosenId = []

    resultDisplay.textContent = cardsWon.length

    if  (cardsWon.length === cardArray.length/2) {

      resultDisplay.textContent = '!!!!!BINGO!!!!!'

    }

  }

  //flip your card

  function flipCard() {

    let cardId = this.getAttribute('data-id')

    cardsChosen.push(cardArray[cardId].name)

    cardsChosenId.push(cardId)

    this.setAttribute('src', cardArray[cardId].img)

    if (cardsChosen.length ===2) {

      setTimeout(checkForMatch, 500)

    }

  }

  createBoard()

})

**By- Saumya Swain**