



---

# EXERCISE 5

---

WEB PROGRAMMING



Name : Aman Kumar Singh

Reg. No.: 23BPS1011

FEBRUARY 23, 2025

## Q1. Traffic Light

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Traffic Light</title>
    <style>
        body{
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
            background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(9,9,121,1) 0%, rgba(0,212,255,1) 100%);
        }
        .main{
            padding: 25px;
            height: 375px;
            width: 150px;
            background-color: black;
            border-radius: 50px;
        }
        .light{
            margin: 25px;
            height: 100px;
            width: 100px;
            background-color: gray ;
            border-radius: 50px;
        }
        .poll{
            margin-left: 95px;
            height: 300px;
            width: 10px;
            border: 2px solid black;
            background-color: black;
        }
    </style>
</head>
<body>
    <div>
        <div class="main">
            <div class="light" id="red"></div>
            <div class="light" id="yellow"></div>
            <div class="light" id="green"></div>
        </div>
        <div class="poll"></div>
    </div>
    <script>
        let colors = ["red", "yellow", "green"];
        let index = 0;
```

```
function changeLight() {
    document.getElementById(colors[(index + 2) % 3]).style.background = "gray";
    document.getElementById(colors[index]).style.background = colors[index];
    index = (index + 1) % 3;
}
setInterval(changeLight, 1000);
</script>
</body>
</html>
```





## Q2. Flames

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flames Game</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f8ff;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }

        .container {
            text-align: center;
            border: 2px solid #333;
            padding: 20px;
            border-radius: 10px;
            background-color: white;
            box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);
        }

        h1 {
            color: #333;
            font-size: 2.5em;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Flames Game</h1>
    </div>
</body>
</html>
```

```

        }

    .input-container {
        margin-bottom: 20px;
    }

    input {
        padding: 10px;
        margin: 5px;
        width: 200px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    button {
        padding: 10px 20px;
        background-color: #4CAF50;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 1em;
    }

    button:hover {
        background-color: #45a049;
    }

    .result {
        margin-top: 20px;
        font-size: 1.5em;
        font-weight: bold;
    }

```

</style>

</head>

<body>

```

<div class="container">
    <h1>FLAMES Game</h1>
    <div class="input-container">
        <input type="text" id="name1" placeholder="Enter First Name" />
        <input type="text" id="name2" placeholder="Enter Second Name" />
        <button onclick="playGame()">Play Game</button>
    </div>
    <div id="result" class="result"></div>
</div>

<script>
    function playGame() {
        const name1 = document.getElementById("name1").value.toLowerCase().replace(/\s+/g,
        '');
        const name2 = document.getElementById("name2").value.toLowerCase().replace(/\s+/g,
        '');

```

if (name1 === "" || name2 === "") {

```
        alert("Please enter both names!");
        return;
    }

    let name1Array = name1.split(' ');
    let name2Array = name2.split(' ');

    name1Array.forEach((letter, index) => {
        if (name2Array.includes(letter)) {
            name2Array.splice(name2Array.indexOf(letter), 1);
            name1Array.splice(index, 1);
        }
    });

    const totalRemainingLetters = name1Array.length + name2Array.length;
    let flames = ['F', 'L', 'C', 'B', 'E', 'S'];

    let index = 0;
    while (flames.length > 1) {
        index = (index + totalRemainingLetters - 1) % flames.length;
        flames.splice(index, 1);
    }

    const result = flames[0];
    let resultText = '';

    switch (result) {
        case 'F':
            resultText = 'Friendship';
            break;
        case 'L':
            resultText = 'Love';
            break;
        case 'C':
            resultText = 'Close Friends';
            break;
        case 'B':
            resultText = 'Best Friends';
            break;
        case 'E':
            resultText = 'Enemies';
            break;
        case 'S':
            resultText = 'Sister / Brother';
            break;
    }

    document.getElementById("result").innerText = `Result: ${resultText}`;
}
</script>
</body>
</html>
```

### FLAMES Game

### FLAMES Game

**Result: Best Friends**

#### Q3. Guess Game

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Guess Game</title>
    <style>
        body {
```

```
background-color: #f0f4f8;
font-family: 'Arial', sans-serif;
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
margin: 0;
color: #333;
}

.container {
    background-color: #fff;
    padding: 30px 50px;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
    text-align: center;
    width: 300px;
}

h1 {
    color: #4CAF50;
    font-size: 24px;
    margin-bottom: 20px;
}

input {
    padding: 10px;
    font-size: 18px;
    width: 80%;
    border: 2px solid #ddd;
    border-radius: 5px;
    margin-bottom: 20px;
    transition: border-color 0.3s ease;
}

input:focus {
    border-color: #4CAF50;
    outline: none;
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    padding: 10px 20px;
    font-size: 18px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease, transform 0.2s ease;
}

button:hover {
    background-color: #45a049;
    transform: translateY(-2px);
```

```
        }

    button:active {
        transform: translateY(2px);
    }

.message {
    margin-top: 20px;
    font-size: 18px;
    font-weight: bold;
}

.correct {
    color: #4CAF50;
}

.incorrect {
    color: #f44336;
}

```

</style>

```
</head>
<body>
    <div class="container">
        <h1>Guess the number (1 to 10)</h1>
        <input type="number" id="guess" placeholder="Enter your guess" min="1" max="10">
        <button onclick="checkGuess()">Submit</button>
        <p class="message" id="feedback"></p>
    </div>

<script>
    let randomNumber = Math.floor(Math.random() * 10) + 1;
    let attempts = 0;
    let guess = document.getElementById('guess');
    let feedback = document.getElementById('feedback');

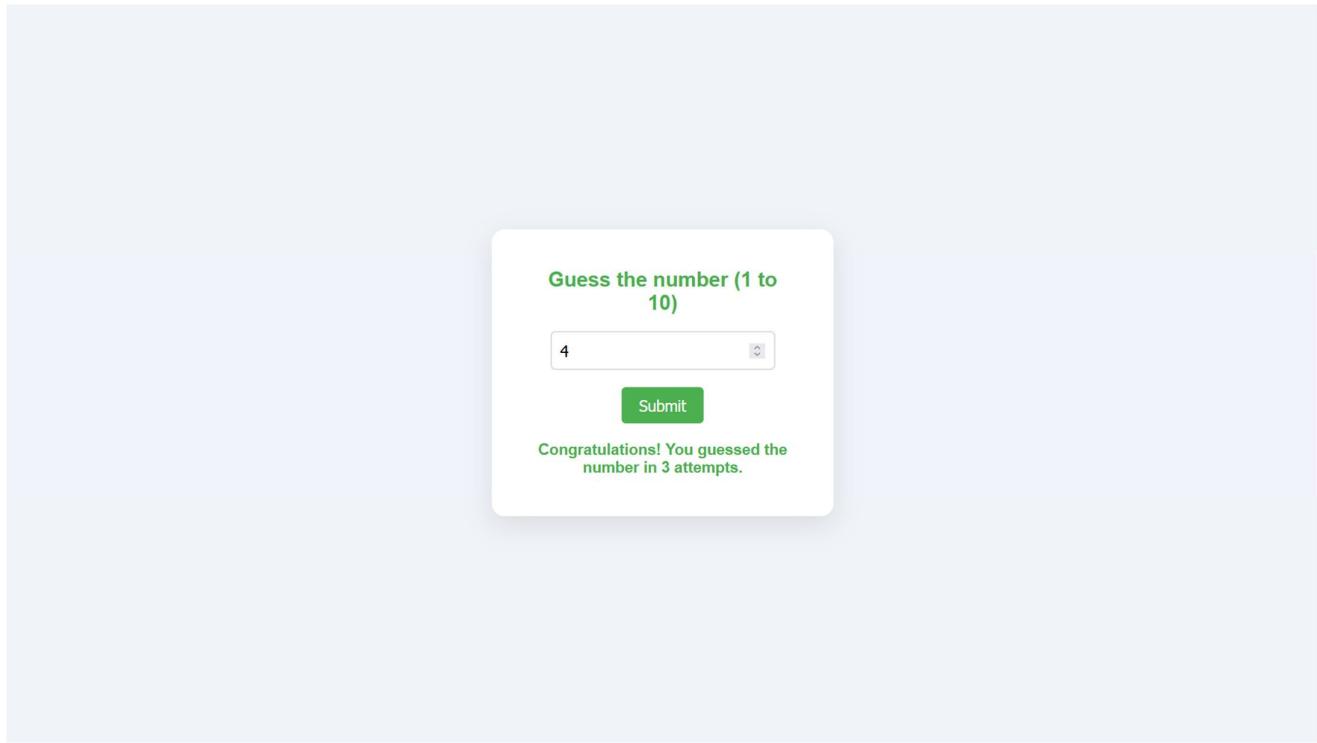
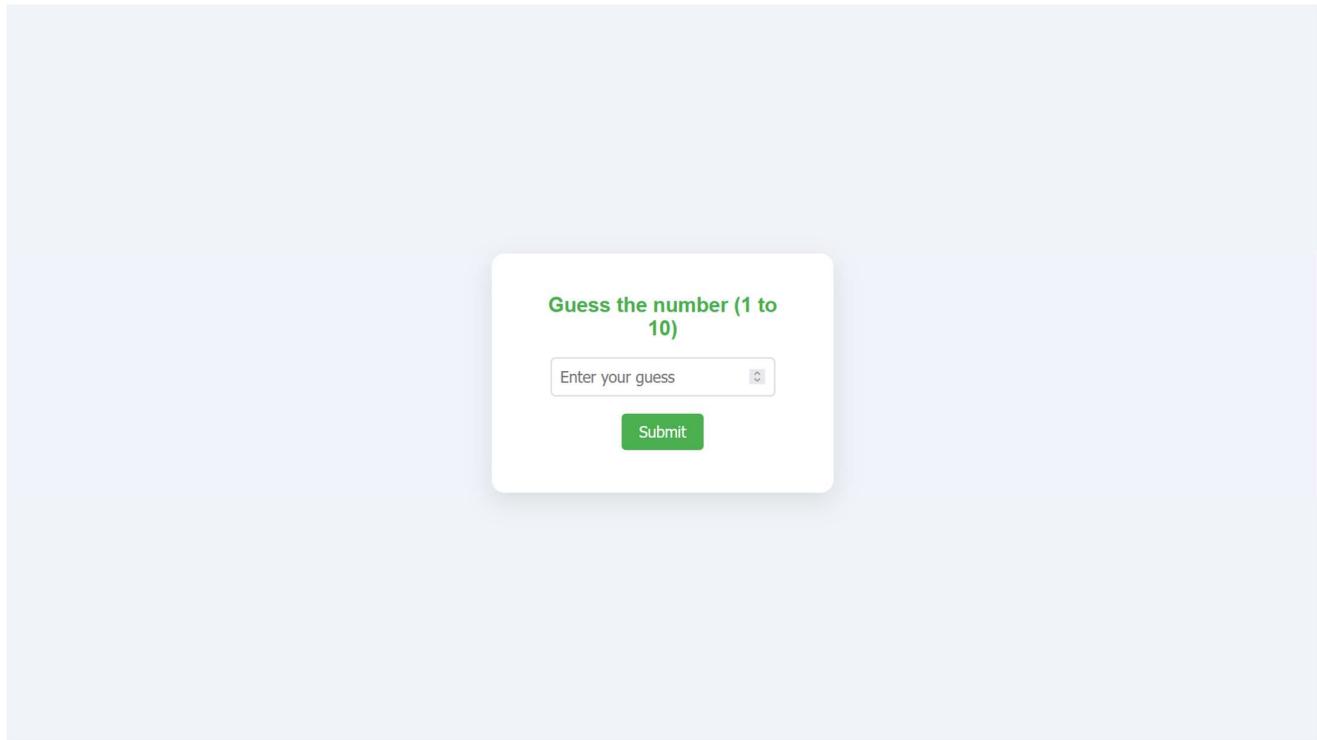
    function checkGuess() {
        attempts++;
        let userGuess = parseInt(guess.value);

        if (isNaN(userGuess) || userGuess < 1 || userGuess > 10) {
            feedback.textContent = 'Please enter a valid number between 1 and 10.';
            feedback.className = 'message incorrect';
            return;
        }

        if (userGuess === randomNumber) {
            feedback.textContent = `Congratulations! You guessed the number in ${attempts} attempts.`;
            feedback.className = 'message correct';
        } else if (userGuess > randomNumber) {
            feedback.textContent = 'Too high! Try again.';
            feedback.className = 'message incorrect';
        } else {
            feedback.textContent = 'Too low! Try again.';
        }
    }
</script>

```

```
        feedback.className = 'message incorrect';
    }
}
</script>
</body>
</html>
```



#### Q4. TO-DO App

```
<!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>To Do App</title>
    <style>
        h1 {
            text-align: center;
            color: #333;
        }
        input {
            width: 100%;
            padding: 10px;
            margin: 10px 0;
            font-size: 16px;
        }
        button {
            padding: 10px;
            font-size: 16px;
            background-color: #333;
            color: #fff;
            border: none;
            cursor: pointer;
        }
        ul {
            list-style-type: none;
            padding: 0;
        }
        li {
            padding: 10px;
            margin: 10px 0;
            background-color: #f9f9f9;
        }
        .delete {
            background-color: #f00;
            color: #fff;
            border: none;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <h1>Todo App</h1>
    <input placeholder="Enter your task" />
    <button>Add Task</button>
    <ul>
        <li>Eat <button class="delete">Delete</button> </li>
        <li>Sleep <button class="delete">Delete</button> </li>
    </ul>
    <script>
```

```

let btn = document.querySelector("button");
let ul = document.querySelector("ul");
let inp = document.querySelector("input");

btn.addEventListener("click", function() {
    let item = document.createElement("li");
    item.innerText = inp.value;

    let delBtn = document.createElement("button");
    delBtn.innerText = "Delete";
    delBtn.classList.add("delete");

    item.appendChild(delBtn);
    ul.appendChild(item);
    console.log(` ${inp.value} added as task`);
    inp.value="";
});

ul.addEventListener("click", function(event) {
    if(event.target.nodeName == "BUTTON") {
        let listItem = event.target.parentElement;
        listItem.remove();
        console.log(` deleted ` );
    }
});
</script>
</body>
</html>

```

## Todo App

Add Task

Eat
Delete

Sleep
Delete

## Todo App

Add TaskDelete

---

EatDelete

---

SleepDelete

---

Web Programming Exercise 5Delete

---

DBMS DA 1Delete

### Q5. Draw Shapes

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Draw Shapes</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f9f9f9;
            color: #333;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
        }

        h1 {
            color: #4CAF50;
            font-size: 36px;
            margin-bottom: 30px;
            text-align: center;
        }

        .shape-container {
            display: flex;
            justify-content: center;
        }
    </style>

```

```
        align-items: center;
        gap: 15px;
        margin-bottom: 20px;
    }

    .shape {
        background-color: #4CAF50;
        color: white;
        padding: 10px;
        font-size: 18px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        transition: background-color 0.3s ease, transform 0.2s ease;
    }

    .shape:hover {
        background-color: #45a049;
        transform: translateY(-3px);
    }

    .shape:active {
        transform: translateY(3px);
    }

    div {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 400px;
        width: 400px;
        border: 2px solid #ddd;
        background-color: #fff;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        border-radius: 10px;
        margin-bottom: 20px;
    }

    svg {
        width: 100%;
        height: 100%;
        border-radius: 10px;
    }

    body {
        background-color: #f0f4f8;
    }

</style>
</head>
<body>
    <h1>Draw Shapes</h1>
    <div>
        <svg id="svgCanvas"></svg>

```

```
</div>

<div class="shape-container">
    <button onclick="drawCircle()" class="shape">Circle</button>
    <button onclick="drawRectangle()" class="shape">Rectangle</button>
    <button onclick="drawLine()" class="shape">Line</button>
    <button onclick="drawTriangle()" class="shape">Triangle</button>
</div>

<script>
    let svg = document.querySelector('#svgCanvas');

    function drawCircle() {
        svg.innerHTML = '<circle cx="200" cy="160" r="80" stroke="black" stroke-width="3" fill="red" />';
    }

    function drawRectangle() {
        svg.innerHTML = '<rect x="100" y="100" width="200" height="100" stroke="black" stroke-width="3" fill="blue" />';
    }

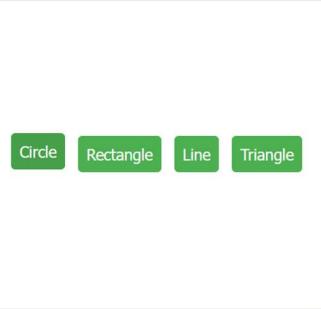
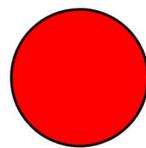
    function drawLine() {
        svg.innerHTML = '<line x1="100" y1="100" x2="300" y2="200" style="stroke:black;stroke-width:2" />';
    }

    function drawTriangle() {
        svg.innerHTML = '<polygon points="200,50 100,250 300,250" style="fill:lime;stroke:purple;stroke-width:1" />';
    }
</script>
</body>
</html>
```

## Draw Shapes



## Draw Shapes



## Draw Shapes



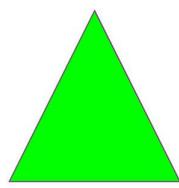
Circle Rectangle Line Triangle

## Draw Shapes



Circle Rectangle Line Triangle

## Draw Shapes



Circle   Rectangle   Line   Triangle

### Q6. Emoji Generator

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Random Emoji Generator</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background: linear-gradient(135deg, #f06, #4a90e2);
            color: white;
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
            height: 100vh;
            margin: 0;
            text-align: center;
        }
        h1 {
            font-size: 3em;
            margin-bottom: 20px;
            text-transform: uppercase;
            letter-spacing: 2px;
            font-weight: bold;
        }
        button {
```

```
background-color: #ff69b4;
color: white;
border: none;
padding: 15px 30px;
font-size: 1.2em;
cursor: pointer;
border-radius: 8px;
transition: all 0.3s ease;
box-shadow: 0 8px 15px rgba(0, 0, 0, 0.3);
}

button:hover {
    background-color: #ff1493;
    transform: scale(1.1);
    box-shadow: 0 12px 18px rgba(0, 0, 0, 0.3);
}

div {
    font-size: 150px;
    margin-top: 30px;
    transition: all 0.5s ease;
}

div.emoji {
    animation: bounce 1s ease-in-out infinite;
}

@keyframes bounce {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-20px); }
}
</style>
</head>
<body>
<h1>Generate a Random Emoji</h1>
<button onclick="generateEmoji()">Generate</button>
<div id="print" class="emoji"></div>

<script>
    function generateEmoji(){
        let arr = ["❤️", "👉", "😊", "👤", "👉", "👉", "👈", "🎂", "ଓ", "ଓ",
        "ଓ", "ଓ", "ଓ", "ଓ", "ଓ"];
        let randomIndex = Math.floor(Math.random() * arr.length);
        document.getElementById("print").innerHTML = arr[randomIndex];
    }
</script>
</body>
</html>
```

## GENERATE A RANDOM EMOJI

Generate

## GENERATE A RANDOM EMOJI

Generate





#### Q7. Moving Eyes

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Moving Eyes</title>
    <style>
        body {
            background-color: #FCE300;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .eye-container {
            display: flex;
            gap: 30px;
        }
        .eye {
            width: 100px;
            height: 100px;
            background-color: white;
            border-radius: 50%;
            position: relative;
            display: flex;
            justify-content: center;
            align-items: center;
            border: 10px solid #444;
        }
        .eye::before {
            content: '';
            position: absolute;
            top: -10px;
            left: -10px;
            width: 0;
            height: 0;
            border-left: 10px solid transparent;
            border-right: 10px solid transparent;
            border-top: 20px solid white;
        }
        .eye::after {
            content: '';
            position: absolute;
            top: -10px;
            left: -10px;
            width: 0;
            height: 0;
            border-left: 10px solid transparent;
            border-right: 10px solid transparent;
            border-top: 20px solid #444;
        }
    </style>

```

```
        }
    .pupil {
        width: 40px;
        height: 40px;
        background-color: brown;
        border-radius: 50%;
        position: relative;
        display: flex;
        justify-content: center;
        align-items: center;
    }
    .inner-pupil {
        width: 15px;
        height: 15px;
        background-color: black;
        border-radius: 50%;
        position: absolute;
    }
</style>
</head>
<body>
    <div class="eye-container">
        <div class="eye">
            <div class="pupil">
                <div class="inner-pupil"></div>
            </div>
        </div>
        <div class="eye">
            <div class="pupil">
                <div class="inner-pupil"></div>
            </div>
        </div>
    </div>
</script>
const eyes = document.querySelectorAll(".eye");
const pupils = document.querySelectorAll(".pupil");

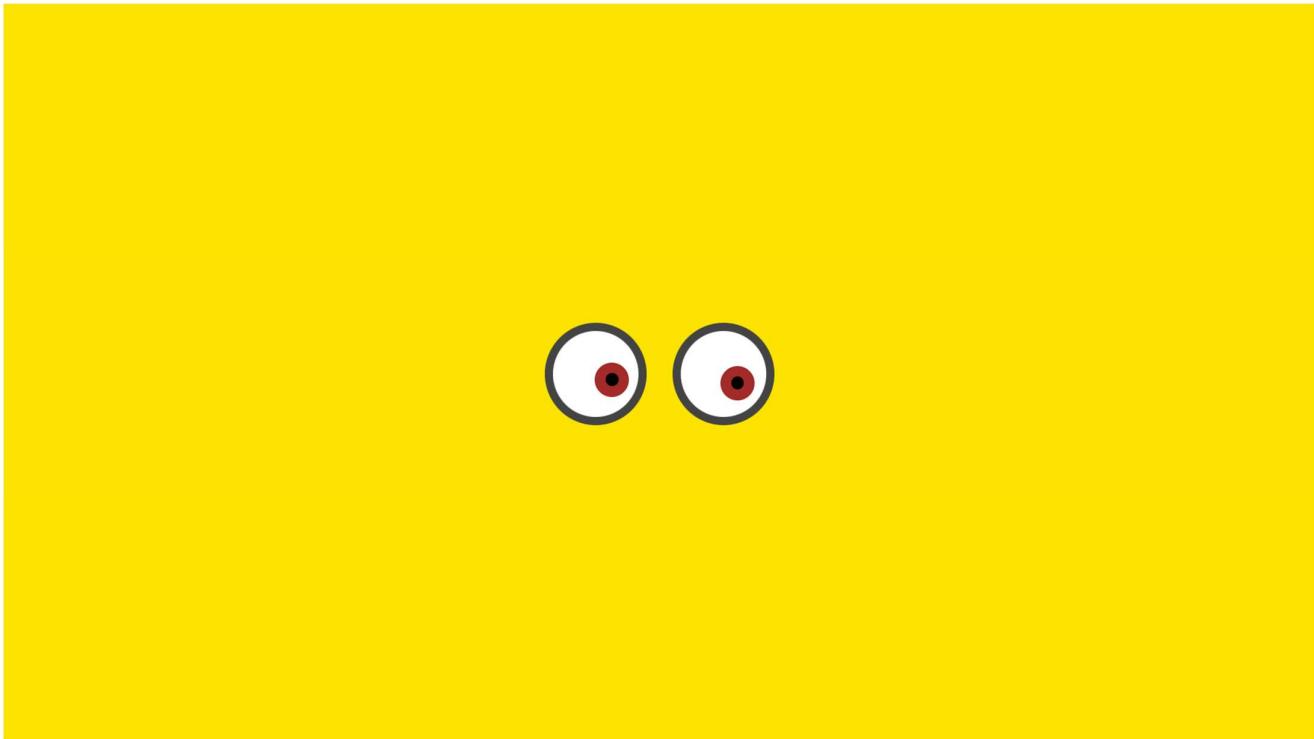
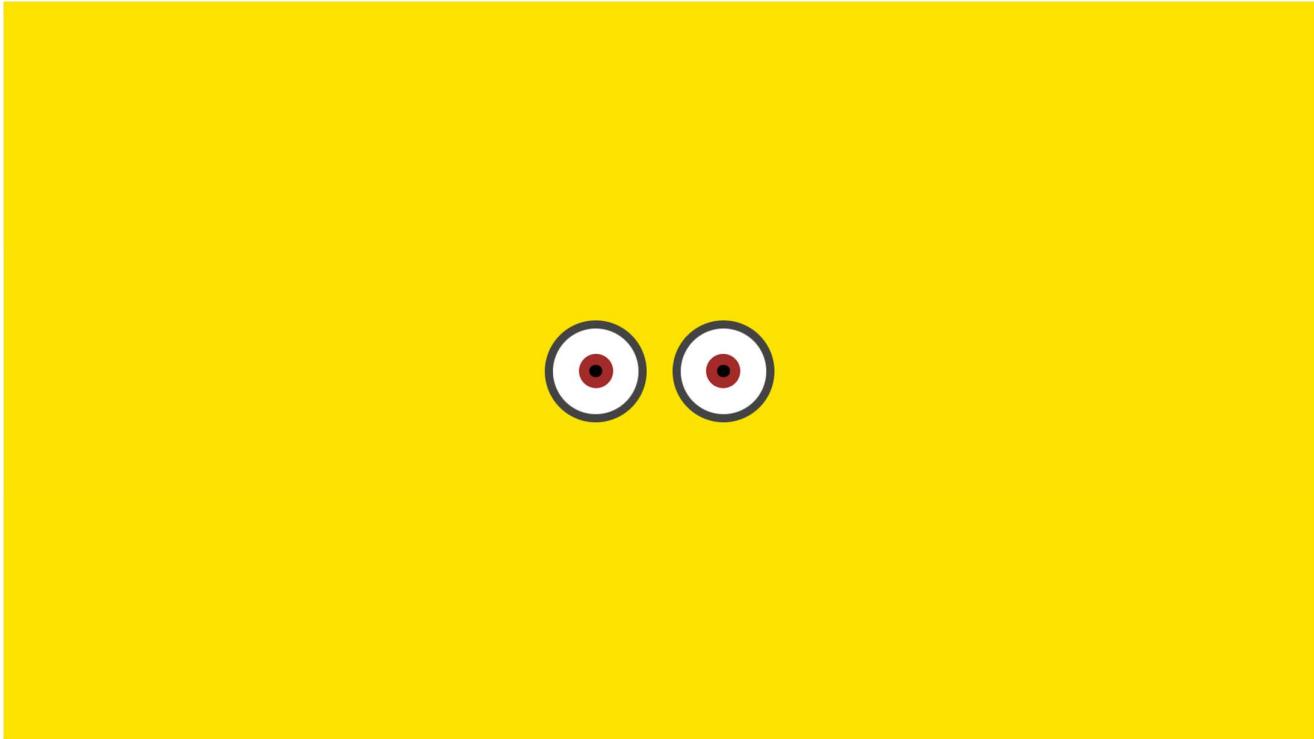
document.addEventListener("mousemove", (event) => {
    const { clientX: mouseX, clientY: mouseY } = event;

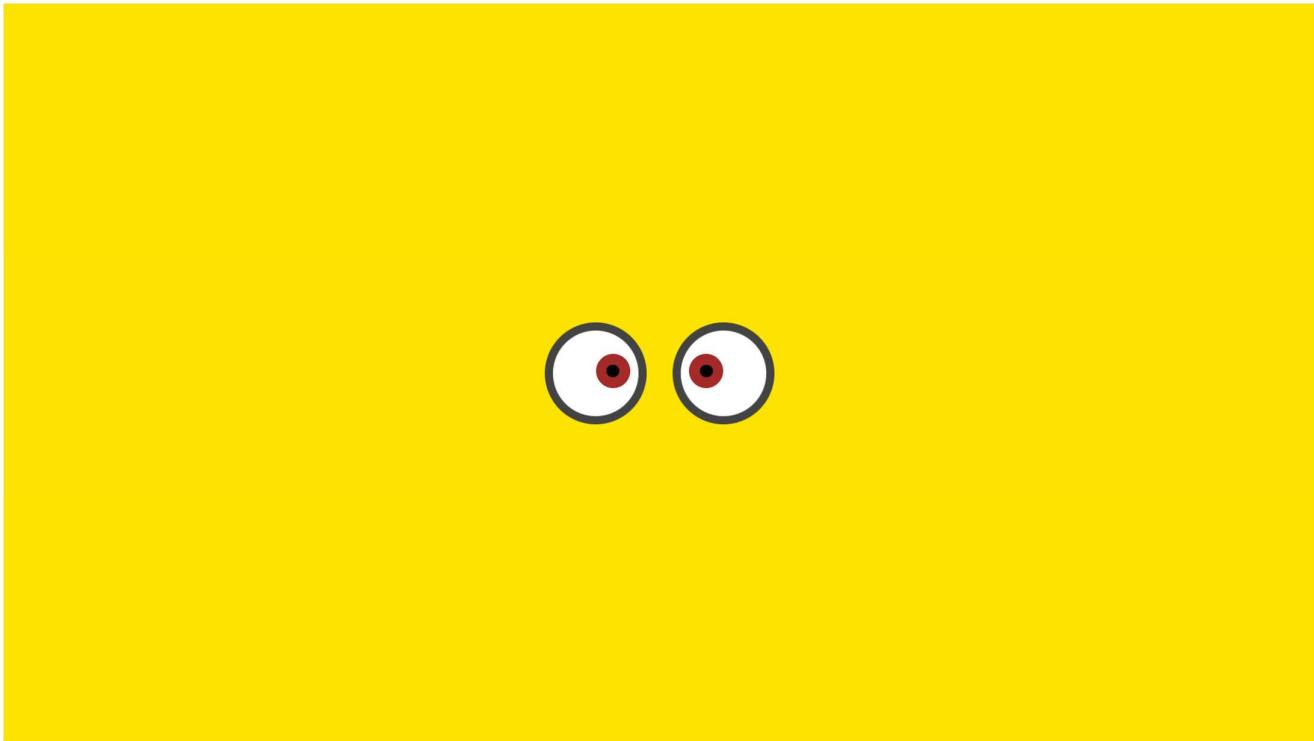
    eyes.forEach((eye, index) => {
        const rect = eye.getBoundingClientRect();
        const eyeCenterX = rect.left + rect.width / 2;
        const eyeCenterY = rect.top + rect.height / 2;

        const deltaX = mouseX - eyeCenterX;
        const deltaY = mouseY - eyeCenterY;
        const angle = Math.atan2(deltaY, deltaX);

        const maxMove = 20;
        const pupilX = Math.cos(angle) * maxMove;
        const pupilY = Math.sin(angle) * maxMove;
    });
})
```

```
        pupils[index].style.transform = `translate(${pupilX}px, ${pupilY}px)`;  
    });  
});  
</script>  
</body>  
</html>
```





#### Q8. Spotlight Effect

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mouse Spotlight Effect</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body, html {
      height: 100%;
      width: 100%;
      overflow: hidden;
      background: black;
    }

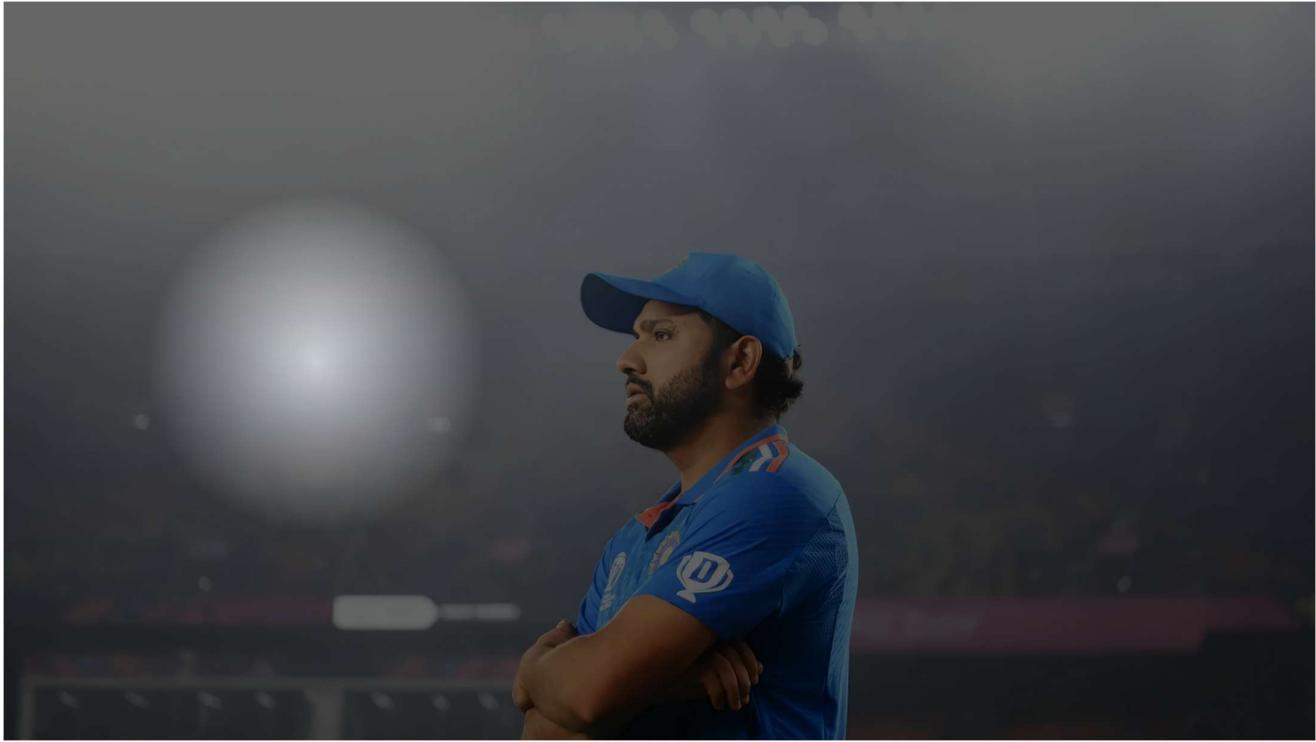
    .container {
      position: relative;
      height: 100vh;
      width: 100%;
      background-image: url(https://wallpapercave.com/wp/wp13235278.jpg);
      background-size: cover;
      background-position: center;
    }
  </style>
</head>
<body>
  <div class="container"></div>
</body>
</html>
```

```
.spotlight {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: radial-gradient(circle 250px at var(--x, 50%) var(--y, 50%), rgba(255, 255, 255, 0.3) 0%, rgba(0, 0, 0, 0.6) 80%);
  pointer-events: none;
  transition: 0.05s ease-out;
}
</style>
</head>
<body>
  <div class="container">
    <div class="spotlight"></div>
  </div>

  <script>
    const spotlight = document.querySelector('.spotlight');

    document.addEventListener('mousemove', (e) => {
      requestAnimationFrame(() => {
        spotlight.style.setProperty('--x', `${e.clientX}px`);
        spotlight.style.setProperty('--y', `${e.clientY}px`);
      });
    });
  </script>
</body>
</html>
```





#### Q9. Image Slider

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Slider</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        .slider-container {
            position: relative;
            max-width: 100%;
            margin: auto;
            overflow: hidden;
            height: 100vh;
        }

        .slider {
            display: flex;
            transition: transform 0.5s ease-in-out;
            height: 100%;
        }

        .slide {
            min-width: 100%;
            height: 100%;
            object-fit: contain;
        }
    </style>

```

```
        }

    button {
        position: absolute;
        top: 50%;
        transform: translateY(-50%);
        background-color: rgba(0, 0, 0, 0.5);
        color: white;
        border: none;
        padding: 10px;
        cursor: pointer;
        font-size: 24px;
        z-index: 10;
    }

    .prev {
        left: 10px;
    }

    .next {
        right: 10px;
    }

```

</style>

```
</head>
<body>

<div class="slider-container">
    <div class="slider" id="slider">
        
        
        
        
        
        
        
        
        
        
    </div>
    <button class="prev" id="prev">&#10094;</button>
    <button class="next" id="next">&#10095;</button>
</div>

<script>
```

```
let currentIndex = 0;
const slides = document.querySelectorAll('.slide');
const totalSlides = slides.length;

const prevButton = document.getElementById('prev');
const nextButton = document.getElementById('next');

function showSlide(index) {
    const slider = document.getElementById('slider');
    if (index >= totalSlides) {
        currentIndex = 0;
    } else if (index < 0) {
        currentIndex = totalSlides - 1;
    } else {
        currentIndex = index;
    }
    slider.style.transform = `translateX(${-currentIndex * 100}%)`;
}

prevButton.addEventListener('click', () => {
    showSlide(currentIndex - 1);
});

nextButton.addEventListener('click', () => {
    showSlide(currentIndex + 1);
});

</script>

</body>
</html>
```





&lt;

&gt;