

Introdução às tecnologias Web - ITW

Pratica 6 – Programação web utilizando Javascript

Exercício 1

Revisões e testes gerais

1. Construa uma página web , utilizando os exemplos apresentados no **slides 35 a 37** da aula teórica.
Para voltar a executar o código JS basta atualizar a página do navegador.
2. Coloque o código / script num ficheiro separado (de nome `myScript.js`, por exemplo) e verifique que o navegador utiliza o ficheiro e executa o código pretendido.
3. Experimente substituir a instrução `console.log` pelas instruções `document.write` e `alert`, para testar formas alternativas de apresentação de informação.



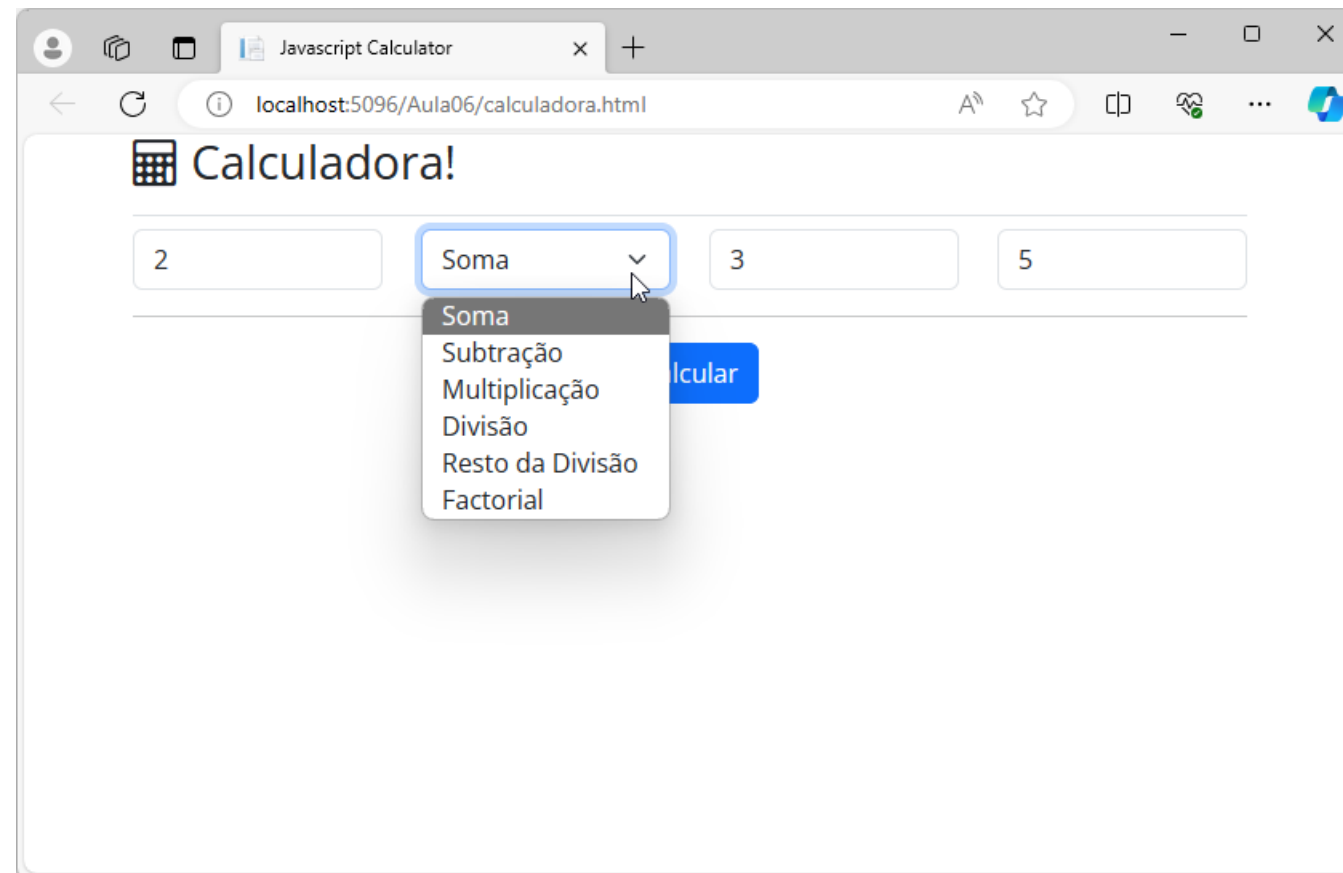
Exercício 2

Calculadora – revisão das aulas teóricas

1. Complete o exercício iniciado no **slide 55** da aula teórica implementando a função `calcula()` de forma a aplicar uma operação diferente, de acordo com o valor da variável `op`.
2. Adicione suporte para mais operações, tais como a multiplicação, divisão, resto da divisão inteira (%), factorial (!), ... tendo os devidos cuidados nos valores utilizados nas variáveis, como por exemplo divisão por zero ou factorial de números negativos.

Exercício 2

Calculadora – resultado esperado



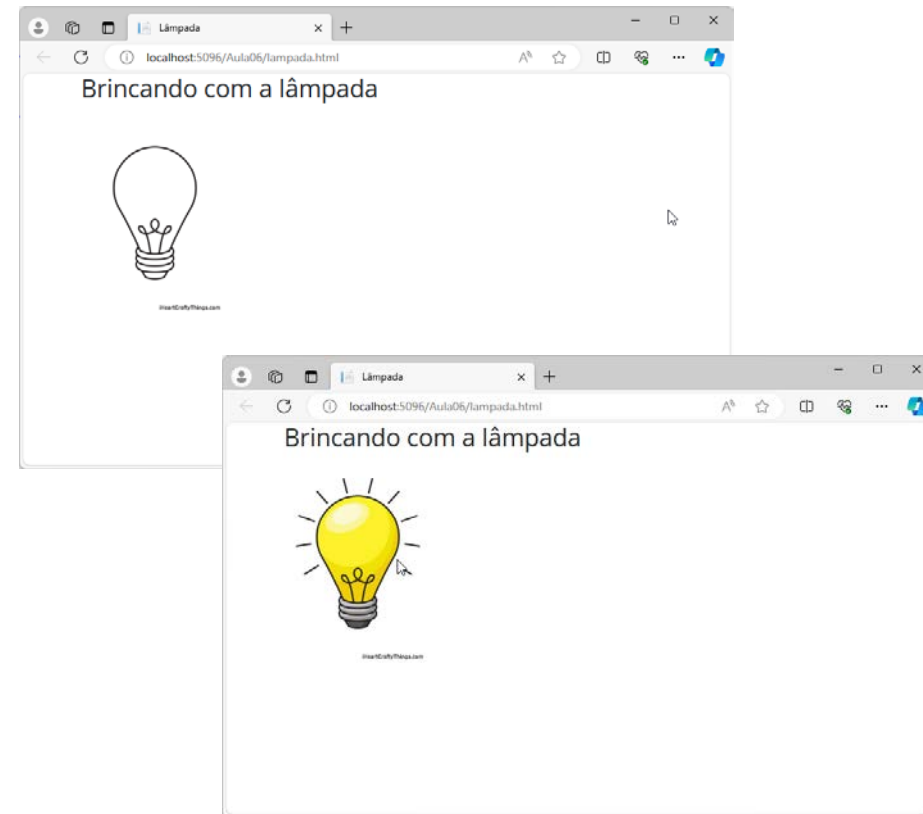
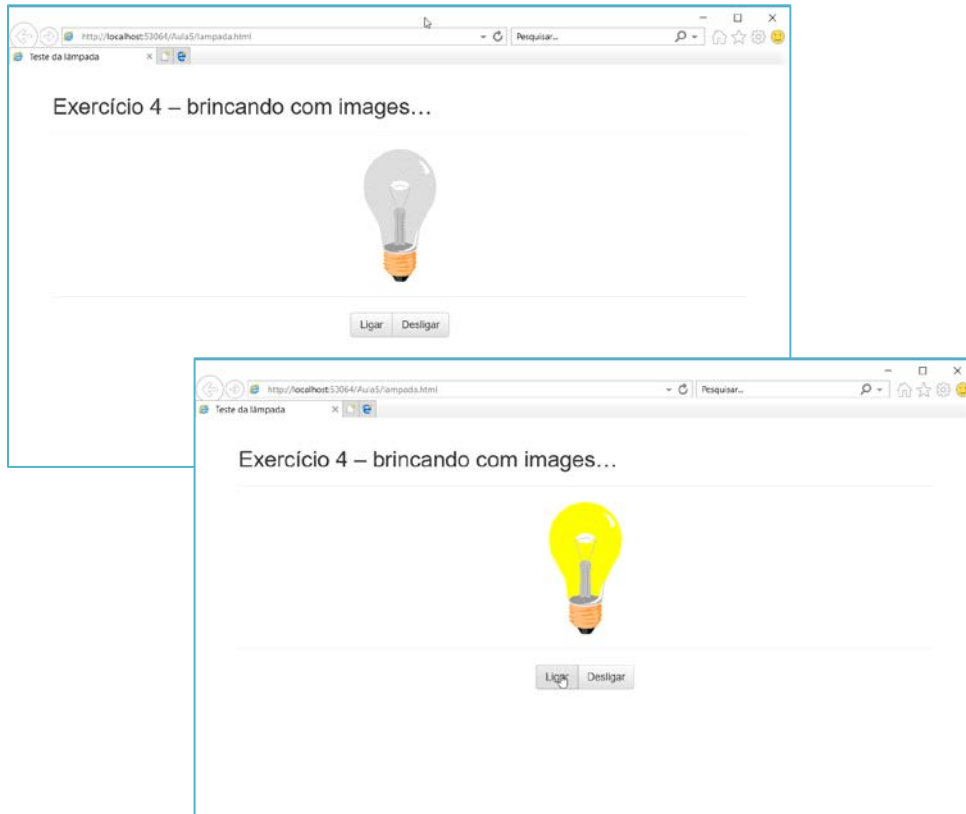
Exercício 3

Brincando com imagens...

<https://iheartcraftythings.com/wp-content/uploads/2021/04/Light-Bulb-DRAWING-%E2%80%93-STEP-10.jpg>

<https://iheartcraftythings.com/wp-content/uploads/2021/04/Light-Bulb-DRAWING-%E2%80%93-STEP-8.jpg>

Acenda e apague a luz ativando o interruptor certo... ou passando por cima da lâmpada



Exercício 4

Implementação de um “jogo”

Implemente um jogo com quatro retângulos/círculos de cores diferentes (vermelho, azul, amarelo e verde).

Os retângulos são baralhados sempre que o botão “baralhar” é ativado.

Sempre que o rato passa por cima de um dos retângulos, deverá ser dada informação sobre o seu nome e a sua posição.

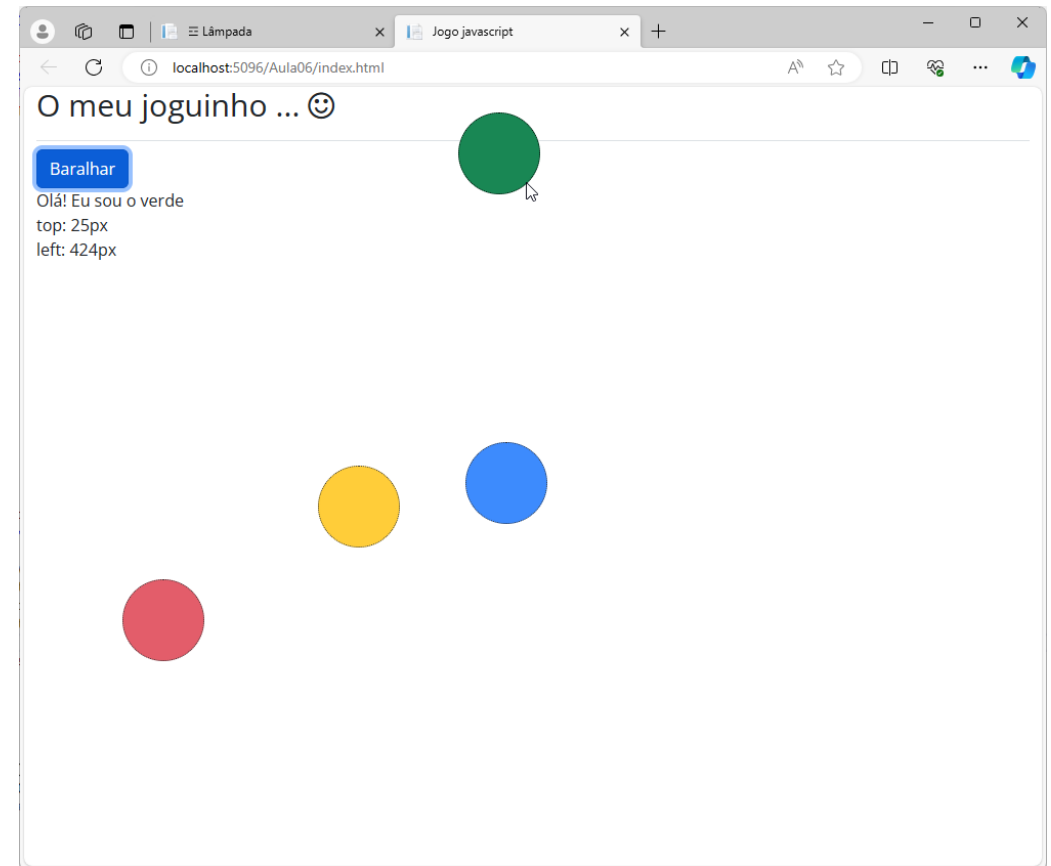
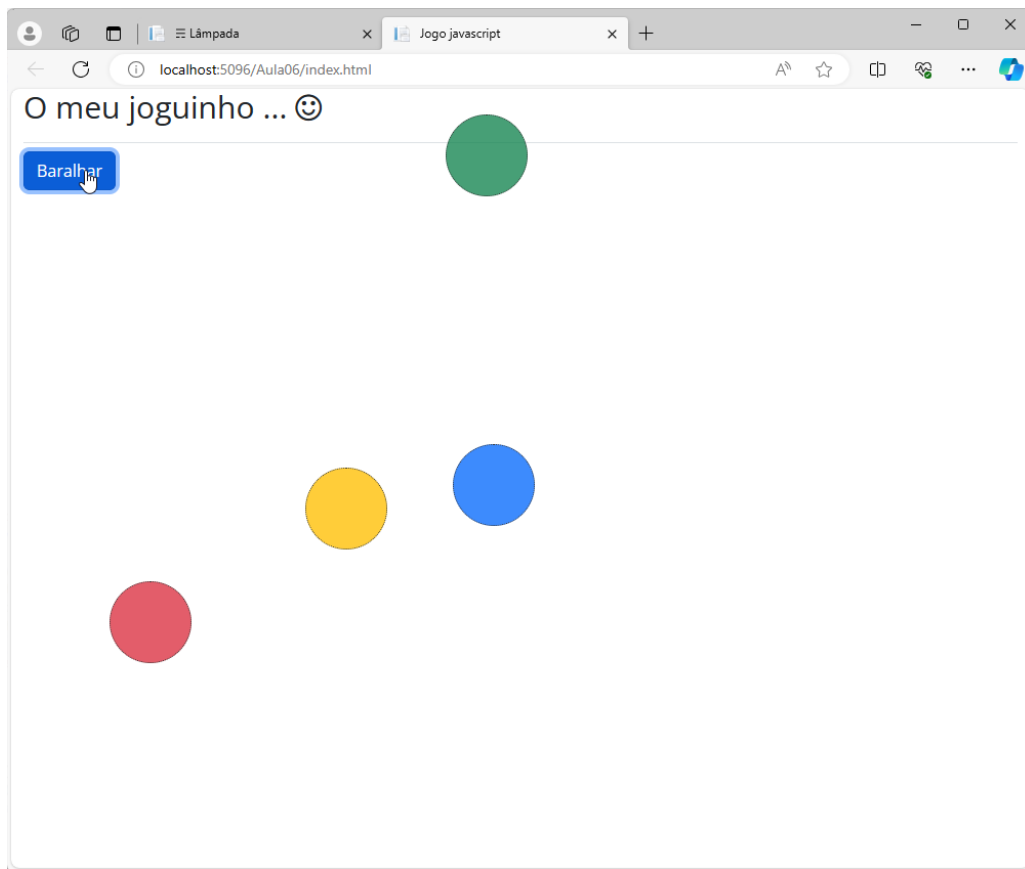
Ajudas técnicas:

Para determinar o tamanho de um objeto: `object.width`, `object.height`

Para determinar o tamanho da janela: `window.innerWidth`, `window.innerHeight`

Exercício 5

Possível resultado...



Exercise 1

General tests

1. Build a web page using the content of **slides 35 to 37**.
To re-execute the code you must reload the page in the browser.
2. Insert the code/script in a separate file (named `myScript.js`, for example) verify if the file is correctly open by the browser and execute the code.
3. Try to change `console.log` with `document.write` or `alert`, to test different approaches on user information presentation.

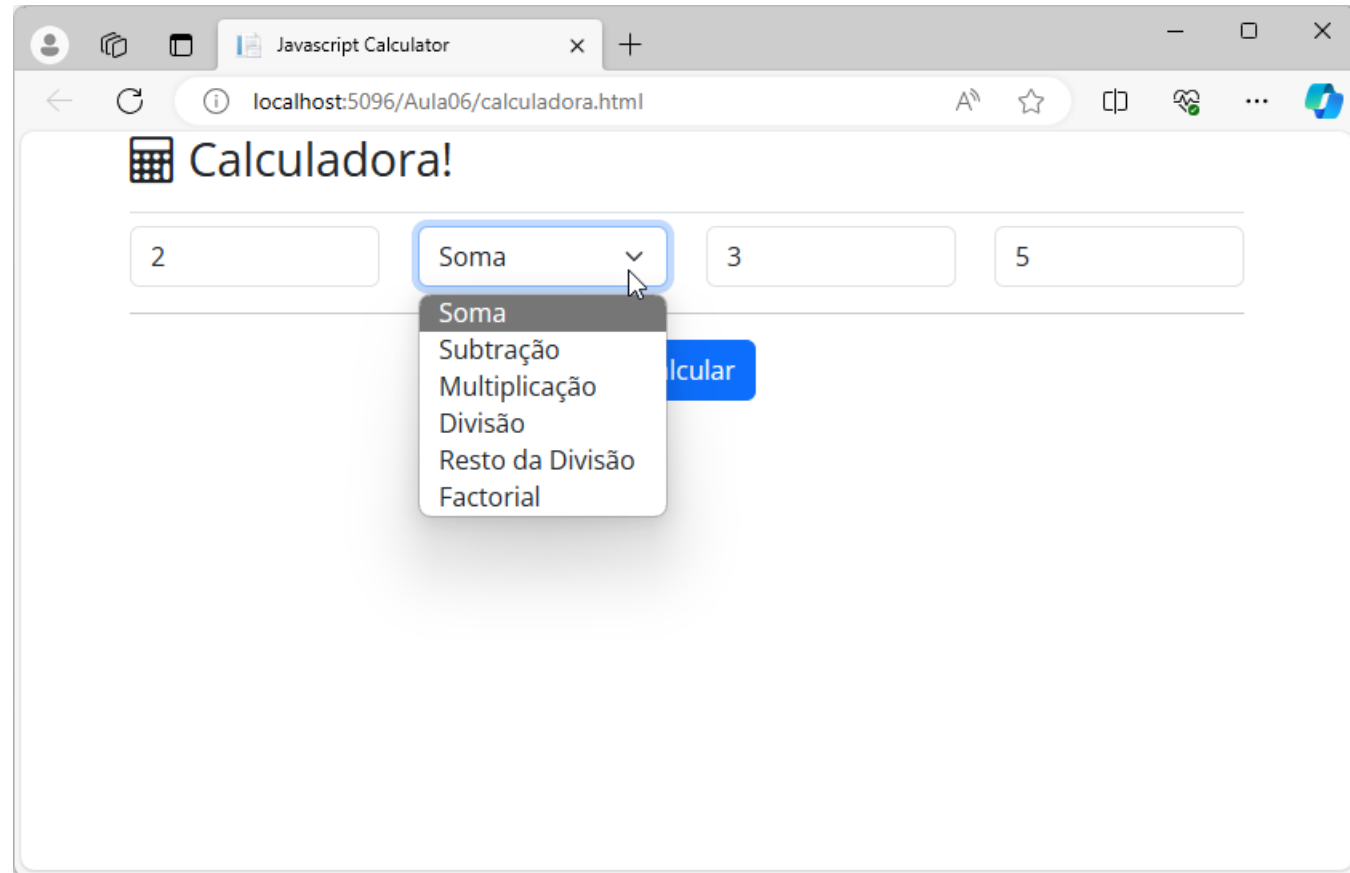
Exercise 2

Calculator - review of theoretical class

1. Complete the exercise of **slide 55** from the tutorial lesson by implementing the function `calcula()` to apply to different operations, accordingly with the value selected on the variable `op`.
2. Add support for more operations, such as multiplication, division, remainder of entire division, factorial, ... taking care of the values used in the variables, such as division by zero or factorial of negative numbers.

Exercise 2

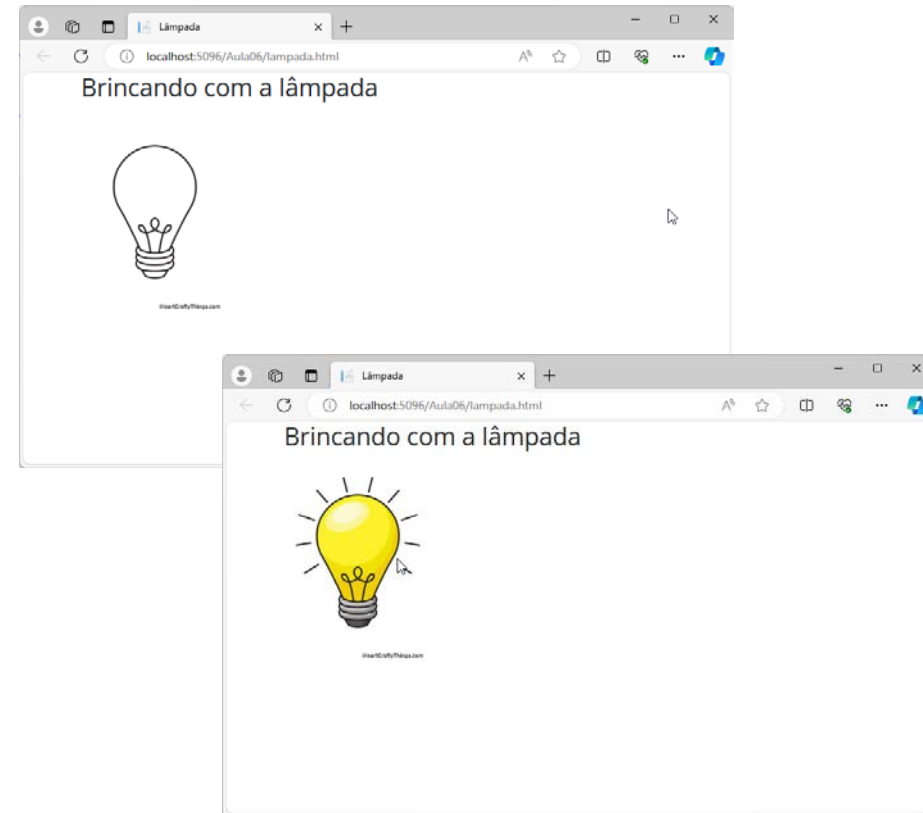
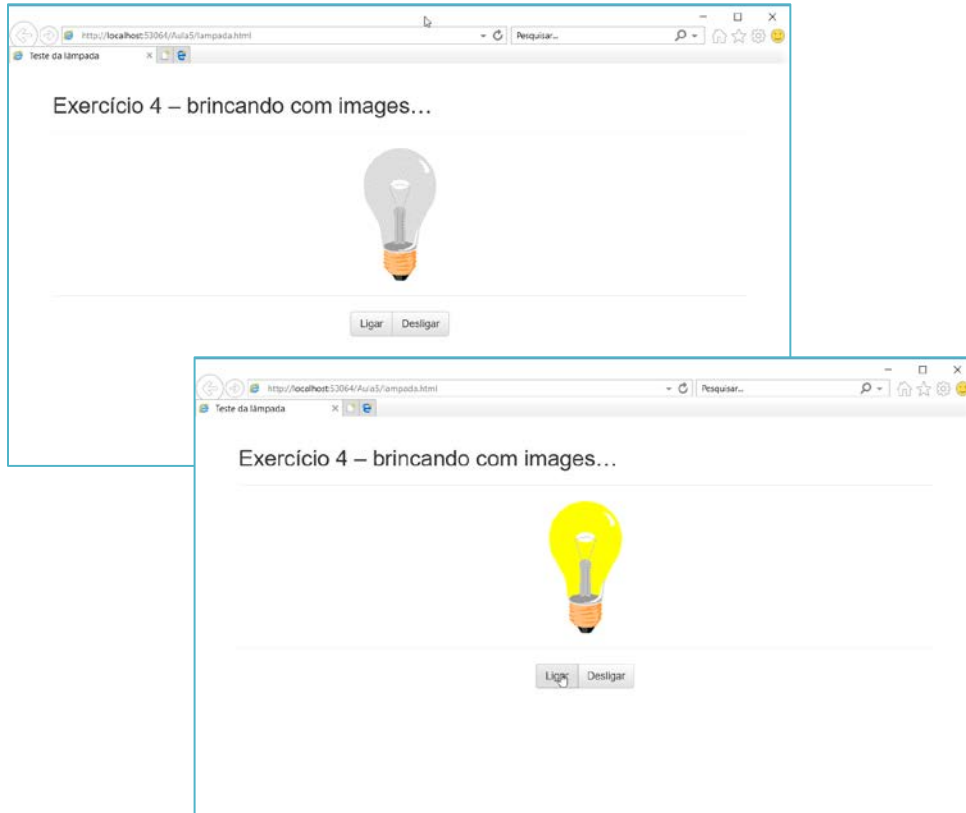
Calculator – expected result



Exercise 3

Playing with pictures

Turn the light on and off by activating the correct switch ... Or moving the mouse over the bulb



Exercise 4

Tiny "game" implementation

Implement a game with four rectangles/circles of different colors (red, blue, yellow and green).

The rectangles are shuffled whenever the "baralhar/shuffle" button is pressed.

Whenever the mouse passes over one of the rectangles, should be given information about its name and position.

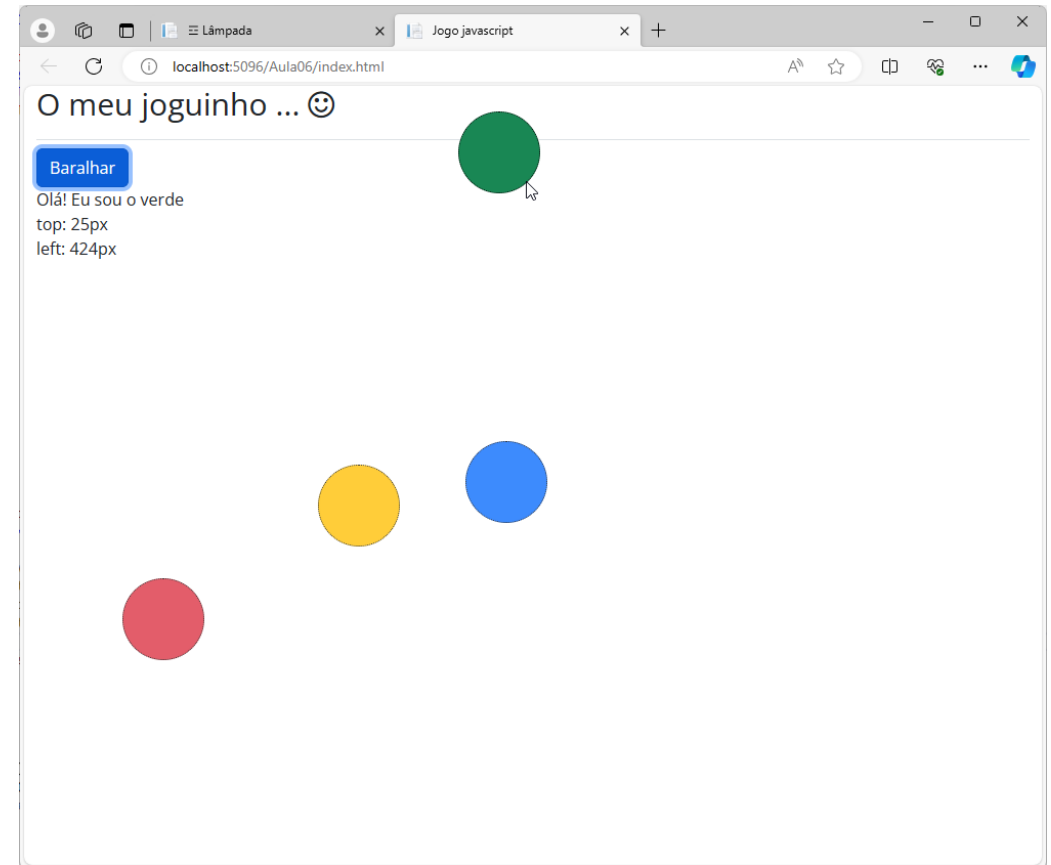
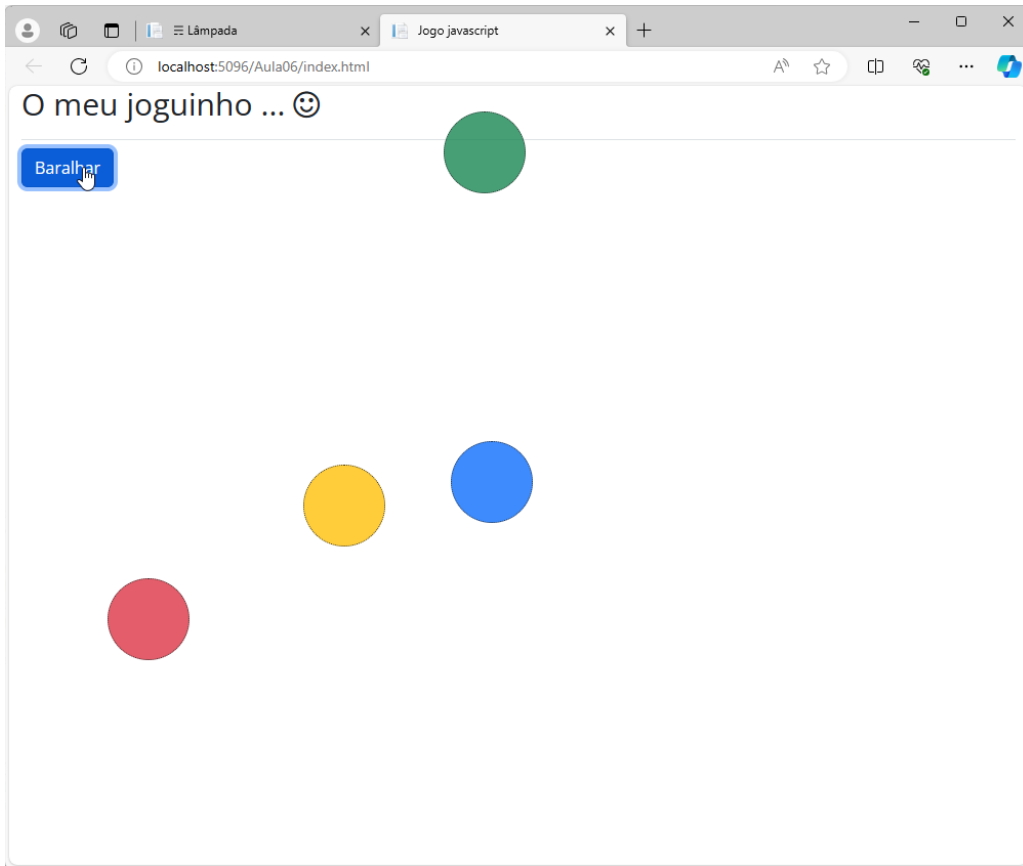
Tech aids:

To evaluate each object size: `object.width`, `object.height`

To evaluate the window size `window.innerWidth`, `window.innerHeight`

Exercise 5

Expected representation ...



Código HTML / HTML Code

TODO: the javascript ...

```
<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <title>Jogo javascript</title>
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Open+Sans&display=swap" rel="stylesheet">
  <link href="..lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" />
  <link href="..lib/font-awesome/dist/css/font-awesome.min.css" rel="stylesheet" />
  <style>
    body {
      font-family: 'Open Sans';
    }

    piece {
      width: 80px; height: 80px;
      position: absolute; border-radius: 40px; border: dotted 1px black;
      opacity: .8;

      piece:hover {
        opacity: 1;
      }
    }
  </style>
</head>
<body>
  <div class="container-fluid">
    <div class="border-bottom pb-2 mb-2"><h2>0 meu joguinho ... <i class="fa fa-smile-o"></i></h2></div>
    <button onclick="scramble()" class="btn btn-primary">Baralhar</button>
    <span id="currentInfo"></span>
    <piece id="azul" class="bg-primary" onmouseover="setCurrentDiv()" onmouseout="clearSelected()"></piece>
    <piece id="vermelho" class="bg-danger" onmouseover="setCurrentDiv()" onmouseout="clearSelected()"></piece>
    <piece id="verde" class="bg-success" onmouseover="setCurrentDiv()" onmouseout="clearSelected()"></piece>
    <piece id="amarelo" class="bg-warning" onmouseover="setCurrentDiv()" onmouseout="clearSelected()"></piece>
  </div>
  <script src="..lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
```

```
<script>
  /* Baralha as peças */
  function scramble() {
    var largura = window.innerWidth;
    var altura = window.innerHeight;
    /* Azul */
    azul.style.top = Math.floor(Math.random() * (altura - 80)).toString() + "px";
    azul.style.left = Math.floor(Math.random() * (largura - 80)).toString() + "px";
    console.log("Azul: " + azul.style.top + " / " + azul.style.left);
    /* Vermelho */
    vermelho.style.top = Math.floor(Math.random() * (altura - 80)).toString() + "px";
    vermelho.style.left = Math.floor(Math.random() * (largura - 80)).toString() + "px";
    console.log("Vermelho: " + vermelho.style.top + " / " + vermelho.style.left);
    /* Verde */
    verde.style.top = Math.floor(Math.random() * (altura - 80)).toString() + "px";
    verde.style.left = Math.floor(Math.random() * (largura - 80)).toString() + "px";
    console.log("Verde: " + verde.style.top + " / " + verde.style.left);
    /* Amarelo */
    amarelo.style.top = Math.floor(Math.random() * (altura - 80)).toString() + "px";
    amarelo.style.left = Math.floor(Math.random() * (largura - 80)).toString() + "px";
    console.log("Amarelo: " + amarelo.style.top + " / " + amarelo.style.left);
  }
  /* Deteta qual a peça que foi "tocada" */
  function setCurrentDiv() {
    //var x = event.target.tagName;
    document.getElementById("currentInfo").innerHTML = "<p>Olá! Eu sou o " + event.target.id + "<br />" +
      "top: " + event.target.style.top + "<br>left: " + event.target.style.left + "</p>";
  }
  function clearSelected() {
    document.getElementById("currentInfo").innerText = "";
  }
  /* Quando o jogo começa, baralha de imediato as peças */
  window.onload = scramble;
</script>
</body>
</html>
```