

## Laboratório 8

Neste laboratório vamos apresentar o JavaScript, a linguagem de programação que é o verdadeiro cérebro por trás de toda a interatividade que vemos na web. Enquanto o HTML se encarrega da estrutura de uma página e o CSS da sua aparência, é o JavaScript que dá vida a tudo, permitindo que as páginas reajam às ações do usuário, validem formulários, exibam animações, atualizem conteúdo dinamicamente e muito mais. Pensem nele como a linguagem que transforma uma página estática em uma experiência rica e interativa, a peça fundamental que completa o trio de tecnologias essenciais para o desenvolvimento web moderno.

### Página HTML:

1 – Crie uma pasta chamada lab08. Nessa pasta, crie um arquivo chamado **main** com a extensão **.html (main.html)**. Neste arquivo será criada nossa página HTML. Conforme a figura abaixo:

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <title>Página Principal</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

### Usando JavaScript:

1 – Vamos introduzir, no *head*, uma pequena instrução do JavaScript para usar a função `alert()`.

```
<html lang="pt-BR">
  <head>
    <title>Página Principal</title>

    <script>
      alert('Hello World!');
    </script>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

2 – Crie na pasta lab07 um arquivo chamado **script** com a extensão **.js (script.js)**. Nesse arquivo, vamos criar a instância de uma variável com o texto “Olá mundo!”, conforme abaixo:

```
let msg = 'Olá mundo!';  
alert(msg);
```

3 – Vamos realizar o link do JavaScript no *head* da nossa página HTML.

```
<html lang="pt-BR">  
  <head>  
    <title>Página Principal</title>  
  
    <script>  
      alert('Hello World!');  
    </script>  
  
    <script src="script.js"></script>  
  </head>  
  <body>  
    <p>Hello World!</p>  
  </body>  
</html>
```

## Simulando um login:

1 – Vamos agora simular um pequeno mecanismo de login. Para isso, vamos criar na pasta lab08 os arquivos **login.html** e **login.js**.

2 – No arquivo **login.html**, vamos adicionar a estrutura abaixo:

```
<html lang="pt-BR">  
  <head>  
    <title>Página de Login</title>  
  
    <script src="login.js"></script>  
  </head>  
  <body>  
    <div id="mensagem"></div>  
  </body>  
</html>
```

3 – No arquivo **login.js**, vamos adicionar as instruções:

```
window.onload = function() {  
  var login = prompt("Digite seu login: ");  
  var password = prompt("Digite sua senha: ");  
  
  if (login === "admin" && password === "admin") {  
    document.getElementById("mensagem").innerHTML = "<h2>Olá, " + login + "!</h2>";  
  } else {  
    document.getElementById("mensagem").innerHTML = "<h2>Login ou senha inválidos para " + login  
      + "!</h2>";  
  }  
};
```

Abra o arquivo **login.html** e valide a funcionalidade de login implementada no arquivo JavaScript.

4 – Vamos criar a mesma simulação, mas agora utilizando arrays. Para isso, vamos criar os arquivos **login-array.html** e **login-array.js**.

5 – O conteúdo do arquivo **login-array.html** é o mesmo do arquivo **login.html**. Só modificar o link do arquivo JavaScript para [login-array.js](#).

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <title>Página de Login</title>

    <script src="login-array.js"></script>
  </head>
  <body>
    <div id="mensagem"></div>
  </body>
</html>
```

6 – Abaixo temos o conteúdo do arquivo **login-array.js**:

```
window.onload = function() {
  let logins = [];

  logins["admin"] = "admin";
  logins["user"] = "123456";
  logins["root"] = "senha01";

  let login = prompt("Digite seu login: ");
  let password = prompt("Digite sua senha: ");

  if (logins[login] === password) {
    document.getElementById("mensagem").innerHTML = "<h2>Olá, " + login + "!</h2>";
  } else {
    document.getElementById("mensagem").innerHTML = "<h2>Login ou senha inválidos para " + login + "!</h2>";
  }
};
```

Abra o arquivo **login-array.html** e valide a funcionalidade de login implementada no arquivo JavaScript.