

Unidade 1

Enunciado:

1. Crie uma função em JavaScript que receba como argumento três números. Essa função deverá verificar qual é o maior número entre os três e exibir esse número no console. Para resolver este exercício você deverá fazer uso de variáveis e estrutura de decisão IF.

Solução javascript

```
function encontrarMaiorNumero(num1, num2, num3) {  
    let maior;  
  
    if (num1 > num2 && num1 > num3) {  
        maior = num1;  
    } else if (num2 > num1 && num2 > num3) {  
        maior = num2;  
    } else {  
        maior = num3;  
    }  
  
    console.log("O maior número é: " + maior);  
}  
  
// Exemplo de uso da função  
  
encontrarMaiorNumero(10, 20, 15); // Deve exibir: O maior número é: 20
```

Enunciado:

2. Crie um array e inicialize-o com os valores 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Crie uma função que receba como argumento um número qualquer. Utilizando a estrutura de repetição for, verifique se o valor passado como parâmetro está armazenado no array. Se estiver, apresente no console a posição do array onde o número está armazenado. Caso contrário, exiba no console uma mensagem informando que o número não foi encontrado.

Solução javascript

```
// Inicializando o array com os valores de 1 a 10
```

```
let numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
function verificarNumero(numero) {  
  let encontrado = false
```

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

```
    if (numeros[i] === numero) {
```

```
      console.log("O número " + numero + " está na posição " + i + " do array.");
```

```
      encontrado = true;
```

```
      break;
```

```
    }
```

```
  }
```

```
  if (!encontrado) {
```

```
    console.log("O número " + numero + " não foi encontrado no array.");
```

```
  }
```

```
}
```

```
// Exemplo de uso da função
```

```
verificarNumero(5); // Deve exibir: O número 5 está na posição 4 do array.
```

```
verificarNumero(11); // Deve exibir: O número 11 não foi encontrado no array.
```

Enunciado:

3. Crie um array de 10 posições e armazene nesse array 10 números quaisquer. Use a função `map()` para gerar um novo array com o dobro de cada valor armazenado.

Código JavaScript:

```
// Inicializando o array com 10 números quaisquer
```

```
let numeros = [3, 7, 1, 9, 4, 6, 8, 2, 5, 10];

// Usando a função map() para gerar um novo array com o dobro de cada valor
let numerosDobrados = numeros.map(function(numero) {
    return numero * 2;
});

// Exibindo o array original e o array com os valores dobrados
console.log("Array original:", numeros);

console.log("Array com valores dobrados:", numerosDobrados);
```

Enunciado:

4. Crie uma função que receba dois números como parâmetros e execute a soma desses dois números. Dentro dessa função, verifique se o resultado dessa soma é um número par. Se for, exiba no console o número resultante e a mensagem “O número é par”. Caso contrário, exiba a mensagem “O número não é par”.

Solução:

```
function verificarSomaPar(num1, num2) {
    let soma = num1 + num2;

    if (soma % 2 === 0) {
        console.log(soma + " - O número é par");
    } else {
        console.log(soma + " - O número não é par");
    }
}

// Exemplo de uso da função
verificarSomaPar(4, 6); // Deve exibir: 10 - O número é par
verificarSomaPar(5, 7); // Deve exibir: 12 - O número é par
verificarSomaPar(5, 8); // Deve exibir: 13 - O número não é par
```

Unidade 2

Enunciado

1. Crie uma página HTML com um botão que tenha o título “Calcular Tabuada”. Diante disso, em um arquivo externo, manipule o evento de click desse botão da seguinte forma:

- capture o elemento botão;
- crie uma função que calcule a tabuada do número 8 e atribua essa função a uma variável. Utilize a estrutura de repetição for para calcular a tabuada e o console para exibir os valores calculados;
- usando escutadores de eventos, adicione ao evento de click do botão a função criada no tópico anterior.

Solução:

Arquivo HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Calcular Tabuada</title>
</head>
<body>
<button id="button1">Calcular Tabuada</button>

<script src="script.js"></script>

</body>
</html>
```

Arquivo script.js

```
const btn1 = document.querySelector("#button1");
```

```
const calcTabuada = function () {  
  
    let n = 8;  
  
    for( let i = 0; i < 11; i++){  
        console.log( `${n} x ${i} = ${n*i}`);  
    }  
  
}  
  
btn1.addEventListener('click', calcTabuada );
```

2 Alterando o HTML do exercício anterior, adicione um novo botão com o título “Calcular tabuada do numero 5”. A partir disso, altere o código Javascript do exercício anterior do seguinte modo:

- capture o elemento botão adicionado;
- adicione um escutador para o evento de click. Quando o botão for clicado, um código Javascript deverá ser executado removendo o evento do botão que calcula a tabuada e incluindo uma nova função que calcule agora a tabuada do número 5.

Arquivo Index.html

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
<title>Calcular Tabuada</title>  
</head>  
<body>  
<button id="button1">Calcular Tabuada</button>
```

```
<br><br>
```

```
<button id="button2">Calcular tabuada do número 5</button>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

Arquivo Script.js

```
const btn1 = document.querySelector("#button1");
```

```
const btn2 = document.querySelector("#button2");
```

```
const calcTabuada = function () {
```

```
  let n = 8;
```

```
  for( let i = 0; i < 11; i++){
```

```
    log( `${n} x ${i} = ${n*i}` );
```

```
  }
```

```
}
```

```
btn1.addEventListener('click', calcTabuada );
```

```
btn2.addEventListener('click', function(){
```

```
  console.log("Trocando o evento do botão");
```

```
  removeEventListener('click', calcTabuada);
```

```
  const calcTabuada5 = function(){
```

```
    let n = 5;
```

```
for( let i = 0; i < 11; i++){  
  log( `${n} x ${i} = ${n*i}`);  
}  
}
```

```
addEventListener('click', calcTabuada5);  
});
```

3 - Crie um formulário de contato em HTML com os seguintes campos:

- Nome.
- E-mail para contato.
- Tipo de contato contendo uma lista com as opções: contato, sugestão ou reclamação.
- Mensagem.

Em um arquivo Javascript externo, valide se o usuário preencheu todos os campos. Para isso, utilize o evento onBlur, para verificar se as informações dos inputs texts foram preenchidas, e o onChange, para verificar se o usuário selecionou algum item. Caso o campo esteja em branco, emita um alerta.

Index.html

```
<!DOCTYPE html>  
  
<html lang="pt-br">  
  
<head>  
  
<title>Formulario de Contato</title>  
  
</head>  
  
<body>
```

```
<form action="#" method="GET">
```

Nome:

```
<input type="text" id="nome" />
```

```
<br><br>
```

E-mail:

```
<input type="email" id="email" />
```

```
<br><br>
```

Tipo do contato:

```
<select id="tipo">
```

```
<option value="">Selecione um tipo</option>
```

```
<option value="contato">Contato</option>
```

```
<option value="sugestao">Sugestão</option>
```

```
<option value="Reclamação">Reclamação</option>
```

```
</select>
```

```
<br><br>
```

```
<textarea id="mensagem"></textarea>
```

```
<br><br>
```

```
<button id="enviar">Enviar contato</button>
```

```
</form>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

Arquivo Script.JS


```
const campoNome = document.querySelector('#nome');

const campoEmail = document.querySelector('#email');

const campoTipo = document.querySelector('#tipo');

const campoMensagem = document.querySelector('#mensagem');
```

```
addEventListener('blur', function(){

if (campoNome.value === ""){

alert("O campo nome não pode ficar em branco");

}

});
```

```
campoEmail.addEventListener('cblur', function(){

if (campoEmail.value === ""){

alert("O campo e-mail não pode ficar em branco");

}

});
```

```
campoTipo.addEventListener('change', function(){

if (campoTipo.value === ""){

alert("O campo tipo de mensagem não pode ficar em branco");

}

});
```

```
campoMensagem.addEventListener('blur', function(){

if (campoMensagem.value === ""){
```

```
alert("O campo mensagem não pode ficar em branco");  
  
}  
  
});
```

Unidade 3

1. Crie um HTML com um formulário de contato básico solicitando ao usuário que digite o seu nome, e-mail, assunto e mensagem (para isso use um textarea). Em seguida, invente um arquivo script.js e implemente um script que:

- a) Selecione os campos utilizando o método getElementById(<id>) e atribua-os a constantes.
- b) Verifique se o nome, e-mail e mensagem foram informados. Caso não tenham sido informados quaisquer um desses campos, emita um alerta avisando o usuário.

Para isso, use o objeto form.

Arquivo HTML

```
<!DOCTYPE html>  
<html lang="pt-br">  
<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>Validando o formulário</title>  
</head>  
<body>  
<h1>Formulário de contato</h1>  
  
<form id="form" action="#" method="GET">
```

```
Nome: <input type="text" id="nome" />

<br><br>

e-mail: <input type="email" id="email" />

<br><br>

Assunto: <input type="text" id="assunto" />

<br><br>

Mensagem: <textarea id="mensagem"></textarea>

<br><br><br>

<input type="submit" id="enviar" value="Enviar">

</form>


<script src="script.js"></script>

</body>

</html>
```

Arquivo script.js

```
const form = document.getElementById("form");

function recebeEventoForm(evento){

    evento.preventDefault();

    const nome = form.querySelector("#nome");
    const email = form.querySelector("#email");
    const mensagem = form.querySelector("#mensagem");

    if ( nome.value == "" ){
        alert("nome é obrigatório");
    }

    if ( email.value == "" ){
```

```
alert("email é obrigatório");  
}
```

```
if ( mensagem.value == ""){  
    alert("mensagem é obrigatório");  
}  
}
```

```
form.addEventListener('submit', recebeEventoForm);
```

2. Crie um formulário de login, solicitando que o usuário digite o e-mail como login e uma senha. Além disso, invente um arquivo script.js e programe a validação dos dados de login digitados. Para realizar a validação, faça uma constante com o login e outra com a senha e compare com os dados digitados. Se os dados não forem iguais, crie um elemento parágrafo log abaixo do campo menu, defina o texto com a mensagem “Login ou senha inválido!”. Caso os dados estejam corretos, emita um alerta informando “login correto”.

Arquivo HTML

```
<!DOCTYPE html>  
<html lang="pt-br">  
<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>Login</title>  
</head>  
<body>  
<h1>Formulário de Login</h1>
```

```
<form id="form" action="#" method="GET">
<div id="campos">
<p>
Login: <input type="text" id="login" placeholder="e-mail" />
</p>
<p>
Senha: <input type="password" id="senha" />
</p>
</div>
<input type="submit" id="login" value="Login">
</form>
<script src="script.js"></script>
</body>
</html>
```

Arquivo script.js

```
const login = "fullture@fullture.com.br"
const senha = "sucesso";

const form = document.getElementById("form");

function recebeEventoForm(evento){

evento.preventDefault();

const input_login = form.querySelector("#login");
const input_senha = form.querySelector("#senha");

if ( ( input_login.value != login) || (input_senha.value != senha) ){
const p = document.createElement('p');
```

```
innerHTML = "Login ou senha invalido!";  
  
style.color = 'red';  
  
const div_campos = document.getElementById('campos');  
append(p);  
} else {  
alert("Login correto!");  
}  
  
}  
  
form.addEventListener('submit', recebeEventoForm);
```

3. Crie uma página HTML, CSS e um semáforo por meio do CSS e mais dois botões com o título PARAR e SEGUIR. Quando o estudante clicar no botão SEGUIR, o semáforo troca a luz para verde; e, quando clicar no botão para PARAR, o semáforo troca a luz para vermelho.

Arquivo HTML

```
<!DOCTYPE html>  
  
<html lang="pt-br">  
  
<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>Login</title>  
  
  
<style>  
  
#semaforo{  
  
border:1px solid black;  
  
padding: 10px;
```

```
width:70px;
}
#vermelho{
background-color:rgb(214, 93, 93);
width:70px;
height: 70px;
}
#Amarelo{
background-color:rgba(253, 253, 75, 0.842);
width:70px;
height: 70px;
}
#verde{
background-color:rgb(77, 184, 77);
width:70px;
height: 70px;
}
#botoes{
padding:10px;
}
</style>
</head>
<body>
```

```
<div id="semaforo">
<div id="vermelho"></div>
<div id="Amarelo"></div>
<div id="verde"></div>
</div>
<div id="botoes">
<button id="parar">PARAR</button>
```

```
<button id="seguir">SEGUIR</button>
</div>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

Arquivo script.js

```
const btnParar = document.getElementById('parar');
const btnSeguir = document.getElementById('seguir');
const vermelho = document.getElementById('vermelho');
const verde = document.getElementById('verde');
```

```
btnParar.addEventListener('click', function(){
  style.backgroundColor = 'rgb(77, 184, 77)';
  style.backgroundColor = 'red';
});
```

```
btn.addEventListner('click', function(){
  style.backgroundColor = 'green';
  style.backgroundColor = 'rgb(214, 93, 93)';
});
```

4 - Desafio: utilizando o código HTML do exemplo código 6 (transcrito a seguir) e o JavaScript do exemplo código 2 (transcrito a seguir), implemente um novo botão que quando acionado o evento de clique, crie um novo curso na página. Lembre-se de observar a estrutura de um novo curso e adicione todos os elementos desta estrutura.

Exemplo do código 6

```
<!DOCTYPE html>

<html lang="pt-br">

<head>

<title>Testando os métodos de acesso ao DOM</title>

<meta charset="utf-8" />

</head>

<body>

<header>

<div id="logo">



</div>

<nav class="primary-nav">

<ul>

<li class="menu-item"><a href="#">Home</a></li>

<li class="menu-item"><a href="#">FullturePro</a></li>

<li class="menu-item"><a href="#">FulltureFalst</a></li>

<li class="menu-item"><a href="#">Sobre nós</a></li>

</ul>

</nav>

</header>

<main>

<h1>Bem vindo a Fullture! Torne sua Carreira à Prova de Futuro!</h1>

<div class="container">

<section class="section-a">

<h2>DEV FULL STACK</h2>

<p>Já pensou que em menos de um ano, você estará pronto para construir um aplicativo do zero? Com nosso curso, você terá uma visão ampla de todo projeto para criação de uma aplicação e poderá definir o seu caminho profissional, apto para atuar no Front-end, Back-end e Mobile em uma das maiores empresas do Brasil e do mundo.</p>
```

```
<button id="button1">Leia Mais</button>

</section>

<section class="section-b">

<h2>DATA SCIENCE</h2>

<p>O FullturePro Data Science é um curso completo e de alto nível para você construir uma
carreira à prova de futuro. Abrangendo todos os tópicos relevantes da área - desde às bases
matemáticas e estatísticas ao aprendizado da linguagem Python</p>

<button id="button2">Leia Mais</button>

</section>

<section class="section-c">

<h2>Product Manager</h2>

<p>O Curso de Product Management da Fullture é um curso completo de alto nível para
aprender como atuar na carreira de Product
Management em empresas modernas digitais e de tecnologia.</p>

<button id="button3">Leia Mais</button>

</section>

</div>

</main>

<script src="script.js"></script>

</body>

</html>
```

Exemplo do código 2

```
// selecionando os elementos HTML acessando o DOM
```

```
const paragrafo = querySelector("p");
console.log(paragrafo);

const menu_item = querySelector(".menu-item");
console.log(menu_item);

const div_logo = querySelector("#logo");
console.log(div_logo);
```

```
const btn1 = getElementById("button1");
```

```
console.log(btn1);
```

```
const menu_itens = querySelectorAll(".menu-item");
```

```
console.log(menu_itens);
```

HTML do exemplo 1 alterado com a adição de um novo botão

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
<title>Testando os métodos de acesso ao DOM</title>
```

```
<meta charset="utf-8" />
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<div id="logo">
```

```

```

```
</div>
```

```
<nav class="primary-nav">
```

```
<ul>
```

```
<li class="menu-item"><a href="#">Home</a></li>
```

```
<li class="menu-item"><a href="#">FullturePro</a></li>
```

```
<li class="menu-item"><a href="#">FulltureFalst</a></li>
```

```
<li class="menu-item"><a href="#">Sobre nós</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<main>
```

```
<h1>Bem vindo a Fullture! Torne sua Carreira à Prova de Futuro!</h1>
```

```
<button id="criar-curso">Criar Novo Curso</button>
```

```
<div class="container">
```

```
<section class="section-a">
```

```
<h2>DEV FULL STACK</h2>
```

```
<p>Já pensou que em menos de um ano, você estará pronto para construir um aplicativo do zero? Com nosso curso, você terá uma visão ampla de todo projeto para criação
```

```
de uma aplicação e poderá definir o seu caminho profissional, apto para atuar no Front-end, Back-end e Mobile em uma das maiores empresas do Brasil e do mundo.</p>
```

```
<button id="button1">Leia Mais</button>
```

```
</section>
```

```
<section class="section-b">
```

```
<h2>DATA SCIENCE</h2>
```

```
<p>O FullturePro Data Science é um curso completo e de alto nível para você construir uma carreira à prova de futuro. Abrangendo todos os tópicos relevantes da área - desde às bases matemáticas e estatísticas ao aprendizado da linguagem Python</p>
```

```
<button id="button2">Leia Mais</button>
```

```
</section>
```

```
<section class="section-c">
```

```
<h2>Product Manager</h2>
```

```
<p>O Curso de Product Management da Fullture é um curso completo de alto nível para aprender como atuar na carreira de Product
```

```
Management em empresas modernas digitais e de tecnologia.</p>
```

```
<button id="button3">Leia Mais</button>
```

```
</section>
```

```
</div>
```

```
</main>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
</html>
```

Código JavaScript que adiciona um novo curso a página. Este código deve ser adicionado ao final do código 2.

```
const btnCriar = document.getElementById("criar-curso");
```

```
addEventListener('click', function(){

    let par = document.createElement('p');

    innerHTML = 'Este é o novo curso adicionado dinamicamente na página de exemplo da Fullture.';

    let titulo = document.createElement('h2');

    innerHTML = 'Novo Curso Adicionado';

    let botao = document.createElement('button');

    innerHTML = 'Saiba mais';

    id = 'button4';

    let s = document.createElement('section');

    s.className = 'section-d';

    s.appendChild(titulo);

    s.appendChild(par);

    s.appendChild(botao);

    const cont = document.querySelector('.container');

    cont.appendChild(s);

});
```

Unidade 4

1 Crie uma página html contendo um formulário e três botões. No formulário, o usuário deverá preencher um campo informando seu e-mail. Os botões que você deverá criar são: GRAVAR, que irá gravar o e-mail digitado na sessão; RECUPERAR, que irá recuperar a informação armazenada na sessão e exibir no console; e LIMPAR, que irá excluir a informação da sessão.

Arquivo HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<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>Testando sessionStorage</title>
</head>
<body>

<p>
e-mail<br>
<input type="text" id="email" />
</p>
<p>
<button id="gravar">Gravar</button>
<button id="recuperar">Recuperar</button>
<button id="limpar">Limpar</button>
</p>

<script src="script.js"></script>

</body>

</html>
```

Script.js

```
const myStorage = window.sessionStorage;
```

```
const email = document.getElementById('email');  
const gravar = document.getElementById('gravar');  
const recuperar = document.getElementById('recuperar');  
const limpar = document.getElementById('limpar');
```

```
gravar.addEventListener('click', function(){  
  log('Gravando');  
  setItem('email', email.value)  
});
```

```
recuperar.addEventListener('click', function(){  
  log('Recuperando');  
  let s_email = myStorage.getItem('email');  
  alert(s_email);  
});
```

```
limpar.addEventListener('click', function(){  
  console.log('Removendo');  
  removeItem('email');  
});
```

2. Altere o código Javascript do exercício gravando, recuperando e excluindo a informação no localStorage.

Código JS utilizando webstorage

```
const email = document.getElementById('email');  
const gravar = document.getElementById('gravar');  
const recuperar = document.getElementById('recuperar');  
const limpar = document.getElementById('limpar');
```

```
addEventListener('click', function(){
```

```
console.log('Gravando');

const emailJSON = JSON.stringify(email.value);
localStorage.setItem('email', emailJSON);

});

addEventListener('click', function(){
console.log('Recuperando');

const email_text = localStorage.getItem('email');
const s_email = JSON.parse(email_text);

alert(s_email);
});

addEventListener('click', function(){
log('Removendo');
removeItem('email');

});
```

3. Crie um formulário que irá solicitar ao usuário que informe seu nome, a data de nascimento e o seu CPF. Quando o formulário for submetido você deverá armazenar os dados do formulário em um banco de dados simulado por um array e esse array deverá ser armazenado no localStorage. Cada informação cadastrada deverá ser salva em um objeto e este objeto atribuído ao array. Você deverá exibir os dados já cadastrados em uma tabela. Quando você abrir a página novamente, ou atualizar a página, não se esqueça de recuperar as informações.

Código JavaScript da Solução do exercício

```
const dados = [];  
  
function recuperaDados() {  
  
  const registros = localStorage.getItem('dados');  
  
  const lista = JSON.parse(registros);  
  
  if ( lista !== null){  
  
    for(let item of lista) {  
  
      adicionaItemTable(item);  
  
    }  
  
  }  
  
  } recuperaDados();  
  
  
  const form = document.getElementById('form');  
  
  const nome = form.querySelector('#nome');  
  
  const cpf = form.querySelector('#cpf');  
  
  const data = form.querySelector('#data');  
  
  
  form.addEventListener('submit', function(event){  
  
  
  
  
    event.preventDefault();  
  
  
  
  
    const obj = adicionaDados(nome, cpf, data);  
  
    adicionaItemTable(obj);
```

```
nome.value = "";
```

```
cpf.value = "";
```

```
data.value = "";
```

```
});
```

```
const btnSalvar = document.getElementById('salvar');
```

```
btnSalvar.addEventListener('click', function(){
```

```
const dadosJSON = JSON.stringify(dados);
```

```
setItem('dados', dadosJSON);
```

```
alert('Dados salvo com sucesso!');
```

```
});
```

```
function adicionaDados(nome, cpf, data){
```

```
let obj = {
```

```
nome:nome.value,
```

```
cpf:cpf.value,
```

```
data:data.value
```

```
}
```

```
dados.push(obj);
```

```
return obj;
```

```
}
```

```
function adicionalItemTable(obj) {
```

```
    const tbody = document.querySelector('tbody');
```

```
    let td = document.createElement('td');
```

```
    innerHTML = obj.nome;
```

```
    let td2 = document.createElement('td');
```

```
    innerHTML = obj.data;
```

```
    let td3 = document.createElement('td');
```

```
    innerHTML = obj.cpf;
```

```
    let tr = document.createElement('tr');
```

```
    tr.appendChild(td);
```

```
    tr.appendChild(td2);
```

```
    tr.appendChild(td3);
```

```
    tbody.appendChild(tr);
```

```
}
```

