

CC1620 - Desenvolvimento Web
CCP120 - Programação Full Stack



Não esqueçam de
marcar presença!

JavaScript





JavaScript

- JavaScript pode interagir com o HTML !!!
 - O seu código pode alterar os elementos do HTML

JavaScript

- JavaScript pode interagir com o HTML!!!
- Um meio de fazer isso é usando o *getElementById*

HTML:

```
<p id="element"> Texto </p>
```

script:

```
<script type="text/javascript">  
... document.getElementById("element").innerHTML="texto_novo";  
</script>
```

Pega o elemento com id = "element"

Altera o texto para o novo valor atribuído



Exemplo 1

Fazer uma página que solicita o *nome* e a *idade* do usuário, assim como o *ano atual* assim que a página for carregada e calcula o ano de nascimento do usuário.

A página deve exibir o nome do usuário e seu ano de nascimento como parte de seu conteúdo.



Funções

Funções

- Funções podem ser interpretadas como mini programas
- Declarando uma função sem parâmetros:

Palavra-chave
function

```
function exibeMensagem() {  
    ... alert('Mensagem de teste!');  
}
```

Sempre temos ()
depois do nome da
função

- Chamando / invocando / utilizando a função:

```
exibeMensagem();
```


Funções

- Declarando uma função com parâmetros:

Palavra-chave
function

```
function exibeMensagem(texto){  
    ... alert(texto);  
}
```

Variável que recebe
o valor enviado na
chamada da função

- Chamando / invocando a função:

```
exibeMensagem("Mensagem de teste!");
```

Chamada da função com
argumento

Funções

- Declarando uma função com parâmetros e retorno:

Palavra-chave
return

```
function soma(num1, num2) {  
    return num1 + num2;  
}
```

- Chamando a função:

```
var resultado = soma(5, 3);  
alert(resultado);
```

soma(5, 3) será
“substituída” pelo
valor retornado

JavaScript



- JavaScript pode interagir com o HTML!!!
 - Isso pode acontecer com o uso dos Eventos do HTML:
 - *HTML Events*
 - Eventos são “ações” detectadas pelos elementos HTML

JavaScript



- Exemplos de eventos:
 - Uma mudança em um elemento HTML (*onchange*)
 - Um clique do mouse (*onclick*)
 - Passar o mouse sobre um elemento HTML (*onmouseover*)
 - Pressionar uma tecla do teclado (*onkeydown*)
 - Quando a página acabar de carregar (*onload*)

HTML - tag *<input>*

- Caixas de Texto:

- `<input type = "text" id = "numero1">`

Nome:

- Botões

- `<input type = "button" value = "Enviar" onclick = "soma();">`

Enviar



Exemplo 2

Fazer uma página que solicita um número X para o usuário, em uma **caixa de entrada**, e apresenta incrementos de um desse número X vezes.



Exemplo 3

Fazer uma página com duas caixas de texto que permite a entrada de dois números, que serão inseridos pelo usuário, e, assim, que o botão *soma* for pressionado, exibe o resultado na própria página.



Exemplo 4

Fazer uma página que solicita dois números e realiza a soma deles se pelo menos um for negativo. Caso contrário, realiza a multiplicação. O cálculo deve acontecer quando o mouse passar por cima de uma *div*.




Exercícios

Exercícios


Em todos os exercícios crie páginas usando a tag **input** com os atributos **type = "text"** ou **type = "button"**.

Nome:

Também aplique estilo nas páginas com CSS posicionando os elementos na página.




Exercício 1 - Crie um botão em uma página html que, ao ser clicado, vai fazer aparecer um número aleatório entre 0 e 50 na própria página (conteúdo da página).



Exercício 2 - Crie uma caixa de texto em uma página html. Crie um botão no mesmo documento que chama uma função em JavaScript e, dependendo do valor introduzido, mostra uma mensagem diferente na própria página:

- Entre 0 e 10: mostra “Insuficiente”
- Entre 10 e 15: mostra “Bom”
- Maior que 15, mostra “Muito Bom”



Exercício 3 - Faça uma página simplificada de cadastro. A página deve ter três entradas: *Nome*, *Login* e *Senha*. A página deve ter também 3 botões: *Cadastrar*, *Limpar* e *Mostrar Dados*.


Você só pode fazer o cadastro de uma pessoa de cada vez. Assim, você não pode cadastrar uma pessoa se as variáveis já estiverem ocupadas. Nesse caso, exiba um alerta. Se você quiser cadastrar uma pessoa nova, utilize o botão limpar antes.



Exercício 3 - continuação

Você só pode cadastrar uma pessoa se todos os campos estiverem preenchidos. A entrada "Senha" precisa ser do tipo *password*.

O botão *Mostrar Dados* exibe todos os dados da pessoa como conteúdo da página.



Exercício 4 - Continue a página do exercício 3, fazendo agora uma área de Login. Você precisa ter duas entradas (além das 3 que já existem): *Login* e *senha* e um botão de *Login*. O login só pode ser concluído com sucesso se os campos de Login estiverem iguais aos utilizados durante o cadastro. Se o login for correto, mostre um alerta com o nome do usuário, se algum dado estiver incorreto, mostre um alerta dizendo qual é o dado incorreto.



Vá além!

Leia mais sobre funções em JavaScript

https://www.w3schools.com/js/js_functions.asp

https://www.w3schools.com/js/js_function_definition.asp



Não esqueçam de
marcar presença!



Material desenvolvido por:

- Danilo Périco
- Isaac de Jesus
- Fagner Pimentel

Fontes:

- <https://www.w3schools.com/>