



Enxurrada de Bits

# Entradas, botões e eventos

# <input>

- É um elemento de formulário
- Especifica um campo de entrada em que o usuário pode inserir dados
- É o elemento de formulário mais importante
- Pode ser exibido de diversas formas, dependendo do atributo “type”

# Atributos <input>

- type = "text" [padrão], "number", "password", etc
  - Determina qual será o tipo de input utilizado
- size = número
  - Estabelece o tamanho inicial do input, sendo este valor em caracteres
- maxlength = número
  - Especifica o número máximo de caracteres que o usuário pode inserir, sendo que se nada for especificado, o usuário pode inserir uma quantidade ilimitada
- placeholder = texto
  - Utilizado para indicar o valor esperado do elemento
- value = texto
  - Define o valor inicial do elemento

**Obs.:** Alguns atributos só se aplicam a tipos de input específicos

*Exemplo:* checked é apenas para tipo checkbox e radio (veja no próximo slide)

# Atributo type

- type = “text”

```
<input type="text" placeholder="Esse é um input de texto">
```

Esse é um input de texto

- type = “number”

```
<input type="number" value="10">
```

10

- type = “password”

```
<input type="password" value="123456">
```

•••••

# Atributo type

- type = “checkbox”

```
<input type="checkbox" id="exemplo">  
<label for="#exemplo">Esse é o label do exemplo</label>
```

☒ Esse é o label do exemplo

☐ Esse é o label do exemplo

- type = “file”

```
<input type="file">
```

Escolher arquivo

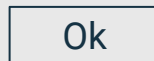
Nenhum arquivo selecionado

## <button>

- É um elemento de formulário
- Define um botão clicável

*Exemplo:*

`<button type="button"> Ok </button>`



- Sempre especifique o atributo type para o elemento (button, submit ou reset)
- O nome do botão que irá aparecer na página fica entre as tags <button>
- Também conseguimos fazer um botão utilizando <input type="button">

# Estilizando um botão

- Utilize a tag button para estilizar seu botão ou defina um id ou uma classe para isso, como no seguinte exemplo:

`<button type="button" id="botao"> Ok </button>`

```
#botao {  
  border: 2px solid #153243;  
  background-color: #e8e8e8;  
  color: #153243;  
  padding: 8px 16px;  
  transition-duration: 0.4s;  
}
```

Ok

```
#botao:hover {  
  border: none;  
  background-color: #153243;  
  color: #e8e8e8;  
  cursor: pointer;  
}
```

Ok

# Utilizando as entradas do usuário

Aprendemos como coletar entradas do usuário, mas como utilizá-las?

Usando JavaScript 😁



# Eventos

Os eventos HTML são "coisas" que acontecem aos elementos HTML.

Quando o JavaScript é usado em páginas HTML, o JavaScript pode "reagir" a esses eventos.

Um evento HTML pode ser algo que o navegador faz ou que um usuário faz.

Aqui estão alguns exemplos de eventos HTML:

- Uma página da web em HTML terminou de carregar
- Um campo de entrada HTML foi alterado
- Um botão HTML foi clicado

# Eventos

- Frequentemente, quando os eventos acontecem, você quer fazer algo.
- JavaScript permite que você execute código quando eventos são detectados e esse código normalmente está dentro de alguma função!
- Podemos atribuir funções a eventos tanto no HTML quanto no JavaScript:
  - HTML:  
`<elemento evento="funçãoJavaScript()">`
  - JavaScript:  
`elemento.addEventListener("evento", funçãoJavaScript);`

# Exemplo: evento onclick

- Executa uma função JavaScript quando um botão é clicado

No HTML:

```
<button onclick="minhaFuncao()">Clique em mim!</button>
```

No JavaScript:

```
let meu_botao = document.querySelector("#botao");  
meu_botao.addEventListener("click", minhaFuncao);
```

# Utilizando as entradas do usuário

- Como sabemos como pegar elementos HTML e colocar eventos neles, podemos trabalhar com os dados que o usuário inserir!


## HTML

```
<input type="text" id="nome" placeholder="Digite seu nome aqui">  
<button type="button" id="botao">Ok</button>
```

## JavaScript

```
function mostrarNome() {  
  let divNova = document.createElement("div");  
  let inputNome = document.querySelector("#nome");  
  let textoNovo = document.createTextNode(`Bem vindo ${inputNome.value}`);  
  
  divNova.appendChild(textoNovo);  
  document.body.appendChild(divNova);  
}  
  
let botaoOk = document.querySelector("#botao");  
botaoOk.onclick = mostrarNome;
```

## Resultado



Initial state:

Digite seu nome aqui

Ok

Final state:

amigo

Ok

Bem vindo amigo

# Referências

Material da disciplina de Programação Web do prof. Flávio Coutinho:

<https://fegemo.github.io/cefet-web/classes/js2/>

<https://fegemo.github.io/cefet-web/classes/js2/#23>

Documentação da Mozilla Developer Network:

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side\\_web\\_APIs/Manipulating\\_documents](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Manipulating_documents)

Documentação da W3Schools:

[https://www.w3schools.com/js/js\\_htmlDOM\\_html.asp](https://www.w3schools.com/js/js_htmlDOM_html.asp)

[https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)

# Leiam

Lista de todos os eventos com explicação:

[https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp)

Dúvidas?

# Colocando em Prática

- Fazer um site que dá as boas vindas ao usuário
  - Adicione dois inputs, sendo um para pegar o nome e o outro para pegar a idade
  - Adicione um botão de “OK”
  - Quando o usuário apertar o botão de “OK”, deve aparecer um h1 dando as boas vindas (precisa usar o nome e a idade coletados do input)
  - Estilize bem para que o usuário se sinta feliz com a recepção =)
- Fazer uma mini calculadora
  - Adicione dois inputs para receber números
  - Adicione um input para receber a operação e um botão para calcular o resultado ou adicione quatro botões, sendo um para cada operação
  - Após o cálculo faça aparecer no site a operação feita e o resultado
  - Estilize bem para que o usuário se sinta feliz com a recepção =)