

# Programação Web I

**Eventos** 

Prof. Pablo Werlang pablowerlang@ifsul.edu.br

**Eventos** 

Eventos são atividades desempenhadas pelo JS quando um determinado gatilho ocorre.

- Precisa determinar um ou mais elementos como alvo, que é quem irá engatilhar o evento.
- Escolher uma atividade que irá engatilhar o evento:
  - click, mouseenter, mouseleave.
  - Input, change, keypress, keyup, keydown.
- Função que será executada quando o evento disparar.

```
HMTL
```

```
<body>
     <button>Clique-me</button>
</body>
```

```
const button = document.querySelector('button');
function func() {
    button.innerHTML = 'Fui clicado';
};
button.addEventListener('click', func);
```

**Eventos** 

#### Pequena observação sobre funções em JS:

```
// modo clássico
function myFunction(a,b) {
    return a + b;
}

// atribui a função anônima à variável
const myFunction = function(a,b) {
    return a + b;
}

// usando arrow function
const myFunction = (a,b) => {
    return a + b;
}
```

Estas maneiras de escrever uma função em JS são válidas. Se quiser saber mais sobre os detalhes que diferem as duas versões, <u>leia este link</u>.

```
const button = document.querySelector('button');
button.addEventListener('click', function() {
    button.innerHTML = 'Fui clicado';
});
```

```
const button = document.querySelector('button');
button.addEventListener('click', () => {
    button.innerHTML = 'Fui clicado';
});
```

**Eventos** 

click - Dispara quando o elemento é clicado.

#### **CSS**

```
button {
    padding: 5px 10px;
    background: #202020;
    color: white;
    border-radius: 5px;
    margin: 30px;
    text-align: center;
    width: 250px;
```

#### **HMTL**

```
<body>
     <button>Clique-me</button>
</body>
```

Clique-me

JS

```
let state = 0;
const button = document.querySelector('button');
button.addEventListener('click', () => {
    if (state == 0){
        button.innerHTML = 'Fui clicado';
    else if (state == 1){
        button.innerHTML = 'Já chega';
    else {
        button.innerHTML = `Perdeu a graça. já clicou ${ state + 1 } vezes`;
    state++;
});
```

**Eventos** 

```
input {
    width: 30px;
}

#result {
    font-size: 1.5rem;
    margin: 0;
}
```

#### **HMTL**

```
input - Dispara quando
digitamos no elemento
(input).
```

```
125 x 15 = 1875
```

```
JS
const n1 = document.querySelector('#n1');
const n2 = document.querySelector('#n2');
const result = document.querySelector('#result');
function multiplica() {
    const vn1 = parseInt(n1.value);
    const vn2 = parseInt(n2.value);
   const mult = vn1 * vn2;
   result.innerHTML = mult;
n1.addEventListener('input', multiplica);
n2.addEventListener('input', multiplica);
```

**Eventos** 

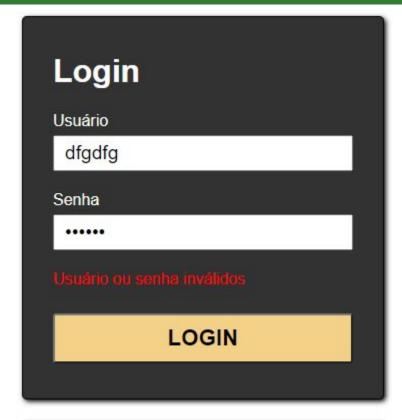
#### **Outros eventos:**

- mouseenter: Dispara quando o mouse entra na área do elemento ao passar por cima dele.
- mouseleave: Dispara quando o mouse sai da área do elemento ao passar por cima dele.
- change: Dispara quando o elemento perde o foco (ex. Trocamos o cursor para outro campo de texto) e somente se o conteúdo foi alterado.
- **keypress**: Dispara quando pressionamos a uma tecla enquanto o elemento possui o foco. Este evento segue disparando enquanto não tirarmos o dedo da tecla.
- **keyup**: Dispara quando deixa de detectar o pressionar de uma tecla se o elemento possui foco.
- **keydown**: Dispara quando detectado o pressionar de uma tecla se o elemento possui foco.

**Eventos** 

#### **Exercício:**

- Tela de login (bonitinha).
  - Caixas de texto, botão, caixa de login, mensagem.
- Mostra na caixa uma mensagem de erro caso o usuário não tenha conseguido se logar. Usuários cadastrados:
  - Usuário = aluno, senha = underere123
  - Usuário = professor, senha = admin789
- Em caso de sucesso, remova a caixa e mostre uma outra mensagem de sucesso.
- Siga o exemplo ao lado (não precisar ter o css igual, é só um modelo).
- Bônus: Permitir também que o usuário faça o login pressionando ENTER em qualquer uma das caixas. Instituto Federal Sul-rio-grandense | Campus Charquea



professor logado com sucesso.