

JS Front Aplicando Eventos

JS Front onload

O evento **onload** é um evento que permite que todo o script seja executado quando o objeto de **document** dentro do objeto de window estiver totalmente carregado.

```
{} window.onload = function(){  
    console.log('o documento está pronto');  
}
```



O código js geralmente é escrito dentro desta função para **evitar erros** que podem ocorrer se o documento não estiver totalmente carregado quando o script for executado.

JS Front

onclick

O evento **onclick** nos permite executar uma ação quando o elemento ao qual estamos aplicando a propriedade é clicado.

```
{ } btn.onclick = function(){  
    console.log('Você clicou!');  
}
```

JS Front

preventDefault()

O **preventDefault ()** nos permite evitar que o evento padrão ou nativo do elemento ao qual estamos aplicando seja executado.

Podemos usá-lo, por exemplo, para evitar que uma tag "a" se comporte nativamente e faça outra coisa.



Sempre temos que ter selecionado o elemento ao qual queremos aplicar o `preventDefault()` por meio dos seletores.

JS Front

preventDefault()

```
{  
  let hiperlink = document.querySelector('a');  
  
  hiperlink.addEventListener('click', function(event){  
    console.log('clizou');  
    event.preventDefault();  
  });  
}
```

JS Front

preventDefault()

```
let hiperlink = document.querySelector('a');
```

```
{}
```

```
hipervinculo.addEventListener('click', function(event){
```

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

```
  event.preventDefault();
```

```
});
```

Prevenimos a ação nativa

Escutamos o evento

Guardamos o elemento

JS Front

Eventos mais usados

- Onclick → Quando o usuário clica
- Ondblclick → Quando o usuário faz double click
- Onmouseover → Quando o mouse se move sobre o elemento
- Onmousemove → Quando o mouse se move
- Onscroll → Quando roda o scroll do mouse
- Onkeydown → Quando aperta uma tecla
- Onload → Quando a página carrega
- Onsubmit → Quando enviamos um formulário

DigitalHouse>
Coding School