

# Validações

# Obter um formulário

O primeiro objetivo será obter o formulário e adicionar um comportamento quando desejamos capturar o evento.

```
{}
```

```
window.addEventListener("load", function(){  
    let formulario = document.querySelector("form.cadastro")  
    //Agregamos o evento de submit ao formulário  
    formulario.addEventListener("submit", function(event){  
        // Depois que o form foi enviado :)  
    });  
});
```

# Validando um campo vazio

Em primeiro lugar, dentro do evento, iremos interromper o envio do formulário com `event.preventDefault()`.

Em seguida, obteremos nosso input com `querySelector` para finalmente perguntarmos se o campo está vazio.

```
{ }
```

```
event.preventDefault();  
let campoNome = document.querySelector("input.nome");  
if(campoNome.value == ""){  
    alert("O campo não pode ser vazio!");  
}
```

# Array de erros

Nosso próximo passo é criar uma array para acumular esses erros e alterar nossa lógica, ou seja, se a matriz não estiver vazia, evitamos o envio do formulário, caso contrário, o formulário será enviado.

{ }

```
let errors = [];  
let campoNome = document.querySelector("input.nome");  
if(campoNome.value == ""){  
    errors.push("O campo nome está vazio.");  
}  
if(errors.length > 0){  
    event.preventDefault();  
}
```

# Mostrando erros com HTML

Nosso próximo passo é criar uma tag `<div>` dentro do nosso html, onde mostraremos a lista de erros. Para fazer isso, colocaremos acima de nosso formulário, adicionando uma tag `<ul>` para indicar o início de uma lista.

html

```
<div class="errors">
  <ul>
    <!--Aqui vão os erros-->
  </ul>
</div>
```

# Mostrando erros com JavaScript

Finalmente, em nosso arquivo JavaScript, passaremos por uma array de erros para que sejam mostrados dentro do <div> <ul>.

```
{}
```

```
let ulErrors = document.querySelector("div.errors ul");  
for (let i = 0; i < errors.length; i++) {  
    ulErrors.innerHTML += "<li>" + errors[i] + "</li>";  
}
```

DigitalHouse>  
Coding School