

# Validação do lado do cliente

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Captura do formulário

O primeiro objetivo será obter o formulário. Para isso temos duas opções:

JS

```
let formulario = document.querySelector("form.reservation")
```

JS

```
let formulario = document.forms["reservation"]
```

# Eventos do formulário

O evento submit é aquele que é executado quando enviamos os dados.

JS

```
formulario.addEventListener("submit", function(event){});
```

JS

```
formulario.onsubmit = (event) => {}
```

# Validando os campos

Podemos obter nosso **input** com **querySelector** para que finalmente validarmos se o valor do campo está vazio.

JS

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



Para interromper o envio do formulário,  
usamos: **event.preventDefault()**

# Armazenar os erros

Criamos um **array** para acumular esses erros e mudar nossa lógica. Ou seja, se o array não estiver vazio, impedimos o envio do formulário, caso contrário o formulário será enviado.

JS

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

# Apresentando os **erros**



```
<section class="erros">
  <ul>
    ...
  </ul>
</section>
```

# No javascript



html

css

js

```
if(erros.length > 0){  
    event.preventDefault();  
    let ulErros = document.querySelector(".erros ul");  
    erros.forEach(error => {  
        ulErros.innerHTML += `<li>${error}</li>`  
    });  
}
```

DigitalHouse>  
Coding School