

EFEITOS NO DOM

Forms

Forms

É comum utilizarmos inputs de formulários para criarmos uma interface entre funções de JavaScript e o usuário final do site. Para isso precisamos aprender como pegar os valores dos formulários.

```
<form name="contato" id="contato">
  <label for="nome">Nome</label>
  <input type="text" name="nome" id="nome">
  <label for="email">Email</label>
  <input type="email" name="email" id="email">
  <label for="mensagem">Mensagem</label>
  <textarea name="mensagem" id="mensagem"></textarea>
</form>
```

```
document.forms; // lista com os formulários
document.forms.contato; // form com nome contato
document.forms.contato.elements; // elementos
document.forms[0].elements[0].value; // valor do primeiro
```

Values

A propriedade `value` retorna o valor do elemento no formulário. Se adicionarmos um callback ao `keyup` (tecla levantar), podemos ficar de olho no evento e puxar o valor sempre que ele mudar. `change` dispara quando houver mudanças.

```
const form = document.getElementById('contato');  
function handleKeyUp(event) {  
  console.log(event.target.value);  
}  
form.addEventListener('keyup', handleKeyUp);
```

Validação

O método `checkValidity` verifica se um input com o atributo `required`, é válido ou não. A propriedade `validationMessage` possui a mensagem padrão de erro do browser. É possível modificar com `setCustomValidity('')`

```
<input type="email" name="email" id="email" required>  
<span class="erro"></span>
```

```
const form = document.getElementById('contato');  
function handleChange(event) {  
  const target = event.target;  
  if(!target.checkValidity()) {  
    target.classList.add('invalido');  
    target.nextElementSibling.innerText =  
target.validationMessage;  
  } else {  
    target.classList.remove('invalido');  
  }  
}  
form.addEventListener('change', handleChange);
```


Select

```
<select name="cores" id="cores">
  <option value="black">Preto</option>
  <option value="white">Branco</option>
  <option value="blue">Azul</option>
</select>
<input type="color">
```

```
const form = document.getElementById('contato');
function handleChange(event) {
  document.body.style.backgroundColor = event.target.value;
}
form.addEventListener('change', handleChange);
```

Diferentes Inputs

```
<input type="color">
<input type="date">
<input type="number">
<input type="range">
<input type="password">
```

```
const form = document.getElementById('contato');
function handleChange(event) {
  console.log(event.target.value)
}
form.addEventListener('change', handleChange);
```

Checkbox

Retorna o valor de value, se estiver checada ou não. `checked` retorna true ou false.

```
<label for="identidade">Possui identidade?</label>
<input type="checkbox" value="identidade" id="identidade">
<label for="casado">Casado?</label>
<input type="checkbox" value="casado" id="casado">
```

```
const form = document.getElementById('contato');
function handleChange(event) {
  if(event.target.checked)
    console.log(event.target.value);
}
form.addEventListener('change', handleChange);
```


Radio

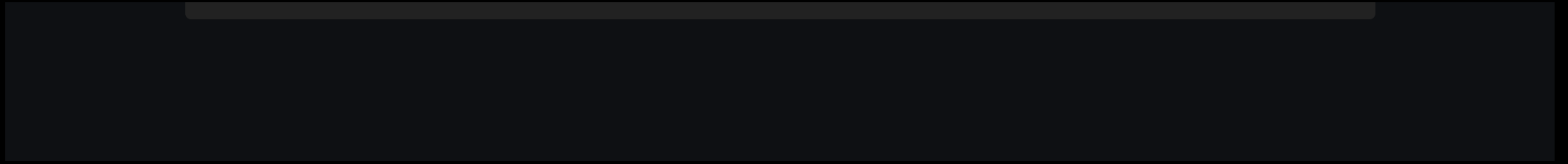
A diferença entre Radio e Checkbox é que radio aceita apenas uma seleção por grupo. Radio é agrupado pelo atributo name.

```
<input type="radio" id="guitarra" value="guitarra"
name="instrumento" />
<label for="guitarra">Guitarra</label>

<input type="radio" id="violao" value="violao"
name="instrumento" />
<label for="violao">Violão</label>

<input type="radio" id="baixo" value="baixo"
name="instrumento" />
<label for="baixo">Baixo</label>
```

```
const form = document.getElementById('contato');
function handleChange(event) {
  if(event.target.checked)
    console.log(event.target.value);
}
```



Pegando todos os valores

Ao invés de selecionarmos elemento por elemento, podemos utilizar um objeto para colocarmos todos os dados que o usuário colocar no formulário.

```
<form name="contato" id="contato">
  <label for="nome">Nome</label>
  <input type="text" name="nome" id="nome">
  <label for="email">Email</label>
  <input type="email" name="email" id="email">
  <label for="mensagem">Mensagem</label>
  <textarea name="mensagem" id="mensagem"></textarea>
</form>
```

```
const form = document.getElementById('contato');
const dados = {};
function handleChange(event) {
  dados[event.target.name] = event.target.value;
}
form.addEventListener('change', handleChange);
```