

  
INSTITUTO  
**METRÓPOLE**  
DIGITAL


---

# Universidade Federal do Rio Grande do Norte - UFRN

## Instituto Metrópole Digital – IMD

### FORM

Nelson Ion de Oliveira

  
INSTITUTO  
**METRÓPOLE**  
DIGITAL

## Agenda

- Tag Form
- Construindo um formulário
- Validação de formulário com JavaScript

## Forms

- A tag `<form></form>` é o recurso HTML para obter informações do usuário.
- Essa tag sozinha não realiza nenhuma ação, para isso, precisamos informar:
  - Quais campos deverão compor o FORM;
  - A ação do form e
  - O tipo de submissão.

## Forms

- Formulários em HTML são usados para transmitir dados para o servidor.
- Um formulário HTML pode conter elementos como:
  - campos de entrada de texto (text & textarea);
  - caixas de seleção (checkboxes, radio-buttons, select, datalist...);
  - Elementos de agrupamento e legenda (fieldset, legend, label...).


## Forms

- Formulários são compostos de elementos que permitem criar interfaces objetivas de acordo com o tipo de informação que se deseja.
- Provavelmente o elemento mais utilizados em forms é o **<input>**.

## Forms


- O código abaixo apresenta um formulário simples com apenas um campo de entrada (input).
- Esse código encontra-se incompleto, falta o botão enviar!

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Nome Completo</title>
6     <style>
7     </style>
8   </head>
9   <body>
10
11   <form>
12     Nome: <input type="text">
13   </form>
14
15   </body>
16 </html>
```



# Forms

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Nome Completo</title>
6     <style>
7     </style>
8   </head>
9   <body>
10
11   <form>
12     Nome: <input type="text">
13     <input type="button" value="Enviar">
14   </form>
15
16 </body>
17 </html>
```



# Métodos Get e Post

- Após uma “**ação de submit**” as informações do formulário são enviadas ao servidor.
- Essas informações são tratadas no servidor no momento de sua chegada.
- O protocolo HTTP foi desenhado para permitir a comunicação entre cliente e servidor.
- Há dois métodos de requisições HTTP
  - GET e
  - POST.
- Um desses dois métodos deve ser usados quando um formulário é submetido para o servidor.
  - No exemplo de hoje **não haverá envio para o servidor.**

## Get e Post

- O método GET - envia dados através da URL
- Quando você usa o método GET as informações do formulário são visíveis na URL da requisição.
- Exemplo:  
`www.site.com.br/pagina.html?nome=nelson&sobrenome=ion`

## Get e Post

- Algumas observações sobre o GET:
  - Podem ser armazenadas em cache do navegador;
  - Permanecem no histórico do navegador;
  - Podem ser “favoritadas”;
  - Esse método nunca deve ser usado quando se lida com dados sensíveis;
  - Têm restrições de comprimento da URL;
  - Deve ser usado, preferencialmente, apenas para recuperar dados.

## Get e Post

- POST – envia dados através do corpo da mensagem HTTP.
- Quando se usa o POST as informações não ficam visíveis na URL da página.

- Exemplo: 

```
POST /test/demo_form.asp HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2
```

## Get e Post

- Algumas observações sobre o POST
  - As requisições nunca são armazenadas em cache;
  - Não permanecem no histórico do navegador;
  - Não podem ser “favoritadas”;
  - As requisições “não têm restrições” quanto ao comprimento dos dados.

## Forms

- Além do botão de enviar e da seleção do método de envio de um formulário, devemos, em cada campo em que o usuário digita informações, definir a propriedade **name** para que possamos recuperar os dados posteriormente.
- É uma boa prática também adicionar a propriedade **name** ao form.

## Forms

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Nome Completo</title>
6     <style>
7     </style>
8   </head>
9   <body>
10
11     <form name="cadastroCompleto">
12       Nome: <input type="text" name="nomeCompleto">
13       <input type="button" value="Enviar">
14     </form>
15
16   </body>
17 </html>
```

## Forms

- Lembra-se do DOM?
- Para obter o valor do campo **nomeCompleto** do form **cadastroCompleto** digite no console javascript

`document.forms.cadastroCompleto.nomeCompleto.value`

- Agora você já pode realizar cadastros com forms, ao invés de prompts em JavaScript.

## Forms

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Nome Completo</title>
6     <style>
7     </style>
8     <script>
9       function cadastrar() {
10         var lista = document.getElementById("lista");
11         lista.innerHTML += document.forms.cadastroCompleto.nomeCompleto.value + "<br />";
12       }
13     </script>
14   </head>
15   <body>
16     <form name="cadastroCompleto">
17       Nome: <input type="text" name="nomeCompleto">
18       <input type="button" value="Enviar" onclick="cadastrar();">
19     </form>
20
21     <div id="lista"></div>
22
23   </body>
24 </html>

```



## Forms

- O elemento **<input>** pode variar, dependendo do valor informado na propriedade **type**.
- Alguns dos tipos mais comuns:
  - text – Texto
  - password – Campo de texto para senhas
  - radio – Permite selecionar uma única opção do grupo.
  - checkbox – Permite múltiplas seleções no grupo.

## Forms

```

15 <body>
16 <form name="cadastroCompleto">
17   Nome: <input type="text" name="nomeCompleto"><br />
18   <br />
19   Sexo:<br />
20   <input type="radio" name="sexo" value="M">Masculino<br />
21   <input type="radio" name="sexo" value="F">Feminino<br />
22   <br />
23   Disciplinas que estou cursando<br />
24   <input type="checkbox" name="disciplina" value="IMD0404">Desenvolvimento Web I<br />
25   <input type="checkbox" name="disciplina" value="IMD0405">Desenvolvimento Web II<br />
26   <br />
27   <input type="button" value="Enviar" onclick="cadastrar();" />
28 </form>
29
30 <div id="lista"></div>
31
32 </body>

```



## Validação de Formulários

- Ao criar um formulário, a validação dos dados é um passo importante da implementação.
- Através da validação dos dados evita-se, por exemplo, a inserção de registros vazios em seu banco de dados.
- É possível efetuar a validação dos dados através de JavaScript.



## Validação com JavaScript

- Validando um campo de texto em JavaScript:
  - Declare uma função para validação;
  - Essa função poderá ser chamada ao clicarmos no botão Enviar do formulário, por exemplo.

```
<!-- Botão enviar chamando a função validarFormulario() ao ser clicado. -->  
<input type="button" onclick="validarFormulario()" value="Enviar">
```


## Validação com JavaScript

- Validando um checkbox ou radiobutton:


```
// Validando se a opção sexo foi marcada
if(document.forms.cadastroCompleto.sexo[0].checked == false &&
    document.forms.cadastroCompleto.sexo[1].checked == false){
    alert("O Campo 'Sexo' deve ser preenchido");
}
```


## Validação com JavaScript

- Além da validação em JavaScript, recomenda-se a validação a nível de aplicação/servidor.
- Por motivos óbvios, a validação em JavaScript não é segura, tendo em vista que pode ser desabilitada no navegador do usuário.
- Principais Vantagens:
  - Ganhos de desempenho.
  - Redundância na proteção da consistência dos dados.



## D vidas?





## Exerc cio

- Implemente um cadastro de usu rios com:
  - Nome;
  - E-mail.
- Utilize orienta  o a objetos para armazenar os cadastros realizados.
- O cadastro dever  permitir inclus es e remo  es dinamicamente, sem realizar o "refresh" (F5) da p gina.

## Referências

- **MILANI, André.** Construindo Aplicações Web com PHP e MySQL, Novatec Editora: São Paulo/SP, 2010.
- **Textos e Exemplos retirados dos sites:**
  - [www.w3schools.com/js/](http://www.w3schools.com/js/)
  - [www.codecademy.com](http://www.codecademy.com)
  - [www.w3.org/DOM/](http://www.w3.org/DOM/)