

Programa de Formação Web Developers PHP

Agosto/2021



CONSTRUSITE BRASIL
O SEU SUCESSO ONLINE

GALAXPAY 
SEU NEGÓCIO RECORRENTE



01

Conhecendo o HTML

Você conhece a sintaxe básica do HTML?

02

Formulários

Elementos de um formulário, tipos de campos, a importância da validação dos dados.

03

Formulários + PHP

Recebendo e validando novamente os dados do formulário no backend.

04

Operadores

Incríveis dicas, e utilidades do inspecionar elemento





Conhecendo o HTML



O que é HTML?

HTML é um acrônimo para **Hypertext Markup Language**. É a linguagem de marcação padrão para criar páginas web atualmente.

A estrutura de qualquer página web é escrita em HTML, antes de começar a acrescentar a estilização.

A extensão do arquivo normalmente é `.html`, mas quando trabalhamos com PHP vamos salvar todos os arquivos com a extensão `.php`.



Programa de Formação Web Developers PHP 2021!

Bora juntos compartilhar conhecimentos?

A screenshot of a code editor window titled "/media/html-inovacao/webdevs/aula04/index.html - Sublime Text (UNREGISTERED)". The editor shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Formação Webdevs 2021</title>
5   </head>
6   <body>
7     <h1>Programa de Formação Web Developers PHP 2021!</h1>
8     <p>Bora juntos compartilhar conhecimentos?</p>
9   </body>
10 </html>
```

A estrutura do HTML

A declaração `<!DOCTYPE html>` indica que aquele é um documento HTML5.

O elemento `<html>` é a raiz do documento. Toda o restante da estrutura deve ficar dentro dele.

Dentro de `<head>` você deve colocar as informações que não são visíveis no corpo da página, mas são importantes para o navegador e os buscadores.

É dentro da tag `<head>` que devem ficar o título, descrição, scripts e estilos da página. Essas informações não ficam visíveis no navegador.

Dentro do `<body>` é onde você vai colocar a estrutura que aparece no navegador ao abrir a página.

Atividades

- 1) Em um arquivo novo (.php) escreva um bloco html, com as tags corretas, e dentro dele, exibir um `<h1> Hello world! </h1>`
- 2) Abaixo deste título, crie um parágrafo com a data de hoje, e seu nome. Ambos utilizando as tags corretas.



Construindo um formulário



HTML Forms

Você está pesquisando por **sapatos**

/media/html-inovacao/webdevs/aula04/busca.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

formulario.php x busca.php x cadastro.php x

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Exemplo de formulário</title>
5   </head>
6   <body>
7
8     <h2>HTML Forms</h2>
9     <?php
10       $search = filter_input(INPUT_GET, 'search');
11
12       if (!empty($search)) {
13         echo '<p>Você está pesquisando por <b>' . $search . '</b></p>';
14       }
15     ?>
16
17     <form method="GET" action="busca.php">
18       <input type="text" id="search" name="search" value="<?= !empty($search) ? $
19         search : '' ?>" placeholder="0 que deseja buscar?" />
20       <br><br>
21       <input type="submit" value="Submit" />
22     </form>
23
24   </body>
25 </html>
26
```


webdevs > aula04 > formulario.php > html > body

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Exemplo de formulário</title>
5   </head>
6   <body>
7
8     <h2>HTML Forms</h2>
9
10    <form method="POST" action="cadastro.php" id="myCustomForm">
11      <label for="name">Nome:</label><br />
12      <input type="text" id="name" name="name" />
13      <br>
14
15      <label for="phone">Telefone:</label><br />
16      <input type="text" id="phone" name="phone" />
17      <br><br>
18      <input type="submit" value="Submit" />
19    </form>
20
21  </body>
22 </html>
```

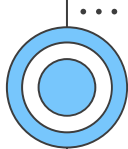
← → 🔄 ⚠ Não seguro | 192.168.2.100/webdevs/aula04/formulario.php

📱 Apps ➡ Intranet Const... 📄 PHP Sandbox, ... 🖼 Gaddafipsum 🔄 h

HTML Forms

Nome:

Telefone:



webdevs > aula04 > cadastro.php > ...

```
1 <?php
2
3 $nome = filter_input(INPUT_POST, 'name');
4 $phone = filter_input(INPUT_POST, 'phone');
5
6 echo 'Dados informados: ' . $nome . ' | ' . $phone;
7 |
```



⚠ Não seguro | 192.168.2.100/webdevs/aula04/cadastro.php

Apps Intranet Const... PHP Sandbox,... Gaddafipsum

Dados informados: construsite | 32323232



O atributo `method`

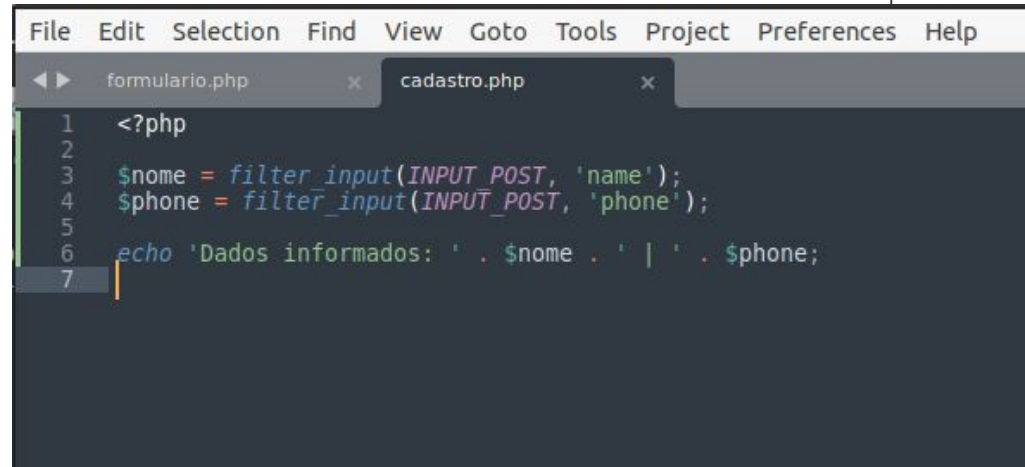
O atributo `method` do form pode ser POST ou GET. A diferença entre eles está na forma como o navegador trata os dados que o usuário preencheu quando enviar o formulário.

- **GET:** os dados são colocados como parâmetros na URL. Muito usado em campos de busca.
- **POST:** os dados ficam escondidos após o envio. Todos os outros tipos de formulário (cadastro, login, etc) usam esse padrão.

O atributo `action`

O `action` deve ser a página que vai receber os dados do formulário após o envio.

Se estiver vazio, vai enviar os dados para a própria página onde está o formulário.



```
File Edit Selection Find View Goto Tools Project Preferences Help
formulario.php x cadastro.php x
1 <?php
2
3 $nome = filter_input(INPUT_POST, 'name');
4 $phone = filter_input(INPUT_POST, 'phone');
5
6 echo 'Dados informados: ' . $nome . ' | ' . $phone;
7
```

Os atributos `name` e `value`

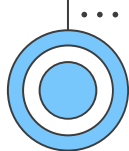
Observe que todos os campos do formulário tem um atributo `name`.

Isso é **obrigatório** em todos os campos, pois é assim que o PHP vai identificar os dados que o usuário preencheu no formulário.

Não é permitido repetir o `name` em campos diferentes, exceto no caso de inputs do tipo radio e checkbox.

O `value` é o valor que virá preenchido por padrão no campo. Se não for informado, o campo virá vazio.

Se você estiver construindo um campo de busca, é útil preencher automaticamente esse campo com o que foi pesquisado pelo usuário.



Tipos de campos e outros atributos

Existem diversos tipos de campos que podem ser usados em um formulário.

Alguns dos mais comuns são: `text`, `password`, `radio`, `checkbox`, `select`, `textarea`, `email`.

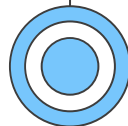
O padrão HTML5 também define vários atributos úteis para trabalhar com formulário (como `required`).

Você vai trabalhar com todos eles ao longo de sua carreira como pessoa desenvolvedora.

Veja a documentação com mais exemplos no site da W3C:

https://www.w3schools.com/html/html_form_input_types.asp

https://www.w3schools.com/html/html_form_attributes.asp





Programando um formulário com PHP




Acessando os dados no backend

O PHP possui a função `filter_input` para acessar os dados que vieram do formulário. O que você passa no primeiro parâmetro depende do `method` do formulário.

- `INPUT_GET`: quando o `method` for GET
- `INPUT_POST`: quando o `method` for POST

No segundo parâmetro, você deve identificar qual campo quer acessar.

 **Lembre-se:** a única informação que vai chegar ao PHP sobre os campos é o atributo `name` que você definiu.

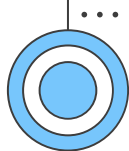
```
<?php
$nome = filter_input(INPUT_POST, 'name');
$phone = filter_input(INPUT_POST, 'phone');
```


Atividades

- 1) Escreva um formulário, que irá receber 3 campos, Nome , Telefone e E-mail. Este formulário, ao ser enviado, deve cair em outra página, onde irá exibir:
Usuário Logado: Nome
Segue dados abaixo:
Telefone: Telefone
E-mail: Email

*Importante


- 1 - Utilizar as quebras de linha, conforme exemplo acima.
- 2- Caso algum campo chegue vazio no PHP, retornar esse erro e voltar a tela.



Validação Frontend

Para validação no frontend,
utilizamos a biblioteca jQuery
Validate:

<https://jqueryvalidation.org/>

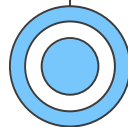
 **Lembre-se:** você precisa incluir
o arquivo base do jQuery no HTML da
página para tudo funcionar certinho:
<http://code.jquery.com/>

Evite trabalho à toa!

Os scripts que você precisa colocar
no <head> da página estão na
documentação. Não precisa ficar
decorando isso, só ir lá e copiar.

```
<head>
  <meta charset="utf-8" />
  <title>Validação de formulário</title>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+30JU5yExlq6GSYGSHK
  <script src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery.validate.min.js"></s
</head>
```



```
formcomvalidacao.php x JS script.js
webdevs > aula04 > formcomvalidacao.php > html > head > script
4 <head>
5   <meta charset="utf-8" />
6   <title>Validação de formulário</title>
7
8   <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/x
9   <script src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.3/dist/jquery
10  <script src="js/script.js"></script>
11 </head>
12
13 <body>
14
15   <h2>Validate!</h2>
16
17   <form method="POST" action="cadastro.php" id="myCustomForm">
18     <label for="customerName">Nome:</label><br />
19     <input type="text" id="customerName" name="customerName" />
20     <br>
21
22     <label for="customerEmail">Email:</label><br />
23     <input type="email" id="customerEmail" name="customerEmail" />
24     <br><br>
25
26     <input type="submit" value="Submit" />
27   </form>
```

```
formcomvalidacao.php JS script.js x
webdevs > aula04 > js > JS script.js > ($) callback > rules
1 $(function() {
2   $('#myCustomForm').validate({
3     rules: {
4       customerName: {
5         required: true,
6         minlength: 5,
7       },
8       customerEmail: {
9         required: true,
10        email: true,
11      }
12    }
13  });
14 }
15 }
```

Validate!

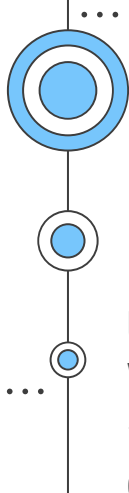
Nome:

This field is required.

Email:

Please enter a valid email address.

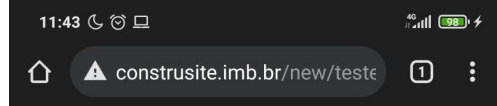
Submit



jQuery Validate + HTML5

Utilizamos o plugin do jQuery Validate junto com o atributo `type="email"` do HTML5 para oferecer uma experiência completa ao usuário.

O layout do teclado em smartphones se ajusta ao type do input graças ao HTML5, enquanto o jQuery Validate possui uma validação mais completa e exibe mensagens de erro melhores que o padrão do HTML5.

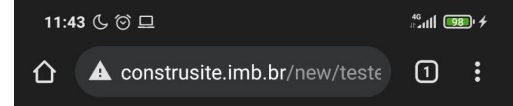


Validate!

Nome:

Email:

Submit



Validate!

Nome:

Email:



Riempimento automatico con LastPass

Submit



Sanitização dos dados no backend

No PHP, utilizamos a função `filter_input()` para receber os dados que o usuário preencheu no formulário:

https://www.php.net/manual/pt_BR/function.filter-input.php

No terceiro parâmetro, informe as flags de acordo com o tipo do dado que você espera receber naquele campo:

<https://www.php.net/manual/en/filter.filters.sanitize.php>

Isso serve para **sanitizar** o dado recebido, reduzindo os riscos à segurança da aplicação.

Se o usuário tentar realizar um ataque XSS, por exemplo, isso garante que não vai funcionar.

```
vs > aula04 > cadastro.php > ...
<?php

$nome = filter_input(INPUT_POST, 'customerName', FILTER_SANITIZE_STRING);
$phone = filter_input(INPUT_POST, 'customerEmail', FILTER_SANITIZE_EMAIL);

echo 'Dados informados: ' . $nome . ' | ' . $phone;
|
```

Validação backend

Para realmente **validar** a informação, utilizamos a função `filter_var`:

https://www.php.net/manual/pt_BR/function.filter-var.php

Com ela, se o formato do dado não se encaixar no que é esperado, o PHP limpa a variável.

⚠ **Lembre-se:** já existem várias bibliotecas que implementam essas validações, não é preciso ficar fazendo na mão todas as vezes que for programar.

Mas você precisa conhecer qual o jeito correto para dar manutenção em códigos antigos, que não usam bibliotecas.

```
2
3 $email = filter_input(INPUT_POST, 'customerEmail', FILTER_SANITIZE_EMAIL);
4 $email = filter_var($email, FILTER_VALIDATE_EMAIL);
5
```

Validação backend completa

```
webdevs > aula04 > cadastro.php > ...
```

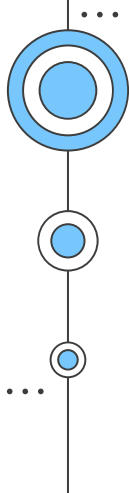
```
1  <?php
2
3  $nome = filter_input(INPUT_POST, 'customerName', FILTER_SANITIZE_STRING);
4  $email = filter_input(INPUT_POST, 'customerEmail', FILTER_SANITIZE_EMAIL);
5  $email = filter_var($email, FILTER_VALIDATE_EMAIL);
6  |
7  echo 'Dados informados: ' . $nome . ' | ' . $email;
8
```

Hora de colocar a mão na massa

1- Você foi contratado para criar uma página de Trabalhe Conosco para o site do seu cliente. Ele precisa que o usuário informe os seguintes dados:

- Nome
- Telefone
- Email
- Cidade
- Estado (campo de seleção com as opções MG, SP, RJ, ES)
- Resumo profissional
- Fala outros idiomas?
 - Deve ter as opções Inglês e Espanhol. A pessoa pode marcar um, ambos ou nenhum.
- Disponível para trabalho presencial?
 - Obrigatório, com as opções Sim e Não

Continua >>



Hora de colocar a mão na massa

Depois de enviar, deve ser exibido para o usuário uma mensagem de sucesso e todos os dados que ele preencheu no formulário.

Implemente validação no front e no backend. Se não souber quais métodos ou flags utilizar, pesquise na documentação no jQuery Validate e do PHP.

