

Programação para WEB

PHP Básico

Prof. Bruno Camargo Ribeiro

Recuperando Dados

Assuntos que serão abordados:

1 Formulário - Recuperando dados	2
1.1 Uso do <code>\$_POST[]</code>	2
1.1.1 Elemento input – text e password	2
1.1.2 Elemento <i>checkbox</i>	3
1.1.3 Elemento <i>radio</i>	6
1.1.4 Elemento <i>oculto</i>	7
1.1.5 Elemento <i>select</i>	8
1.1.6 Elemento <i>textarea</i>	9
1.2 Uso do <code>\$_GET[]</code>	10
1.3 Uso do <code>\$_REQUEST[]</code>	10
2 Uso do <code>\$_GET[]</code> com URL	12
2.1 Uso de variáveis para compor a URL	13

1 Formulário - Recuperando dados

Existem diferentes maneiras de recuperar os dados pelo PHP, os quais deverão ser aplicados de acordo com a origem dos dados. Vamos trabalhar com a coleção de dados **`$_POST[]`**, **`$_GET[]`** e **`$_REQUEST[]`**.

1.1 Uso do `$_POST[]`

O método ***post*** (formulário), passa suas variáveis codificadas dentro da própria submissão, desta forma não são visíveis pelo usuário, apenas o servidor e o navegador tem acesso a estas informações.

1.1.1 Elemento input – text e password

Para recuperar os dados do formulário o processo é bem simples, mas devemos levar em consideração as indicações da propriedade ***name*** dos elementos do HTML.

```
exemplo01.html x exemplo01.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo01.php">
7     Login:
8     <input type="text" name="nome"/> <br><br>
9     Senha:
10    <input type="password" name="senha"/> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>
```

```
exemplo01.html x exemplo01.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Nome: " . $_POST['nome'];
4   echo "<br>";
5   echo "Senha: " . $_POST['senha'];
6   ?>
```

Usando variáveis para receber os dados

```
exemplo02.html x exemplo02.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo02.php">
7     Login:
8     <input type="text" name="nome"/> <br><br>
9     Senha:
10    <input type="password" name="email"/> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>

exemplo02.html x exemplo02.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   $nome = $_POST['nome'];
4   $senha = $_POST['email'];
5   echo "<b>Nome: </b>". $nome;
6   echo "<br>";
7   echo "<b>Senha: </b>". $senha;
8   ?>
```

Este recurso pode ser utilizado nos demais elementos.

1.1.2 Elemento *checkbox*

Devemos tomar cuidado ao recuperar dados do elemento *checkbox*, pois somente serão considerados os elementos com valores atribuídos.

```
exemplo03.html x exemplo03.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo03.php">
7     Qual linguagem de programação você conhece?<br>
8     <input type="checkbox" name="asp"> ASP <br>
9     <input type="checkbox" name="php"> PHP <br>
10    <input type="checkbox" name="java"> JAVA <br>
11    <br><input type="submit" value="Enviar">
12  </form>
13 </body>
14 </html>
```

```
exemplo03.html x exemplo03.php x
1 <?php
2     echo "<h3> Recuperando os dados </h3>";
3     echo $_POST['asp'];
4     echo "<br>";
5     echo $_POST['php'];
6     echo "<br>";
7     echo $_POST['java'];
8     echo "<br>";
9     ?>
```

Caso não tenha nenhum valor selecionado, teremos uma mensagem de erro, pois o PHP entenderá que a variável não existe.

Recuperando os dados

Notice: Undefined index: asp in C:\xampp\htdocs\CPS - PHP\aula11\exemplo03.php on line 3

Notice: Undefined index: php in C:\xampp\htdocs\CPS - PHP\aula11\exemplo03.php on line 5

Notice: Undefined index: java in C:\xampp\htdocs\CPS - PHP\aula11\exemplo03.php on line 7

Para resolver este problema, a solução é utilizar a função **isset()**. Esta por sua vez verifica a existência da variável.

```
exemplo04.html x exemplo04.php x
1 <?php
2     echo "<h3> Recuperando os dados </h3>";
3
4     if (isset($_POST['asp'])) {
5         echo $_POST['asp'];
6         echo "<br>";
7     }
8     if (isset($_POST['php'])) {
9         echo $_POST['php'];
10        echo "<br>";
11    }
12    if (isset($_POST['java'])) {
13        echo $_POST['java'];
14        echo "<br>";
15    }
16    ?>
```

Etec "Sales Gomes" – 101 – Tatuí

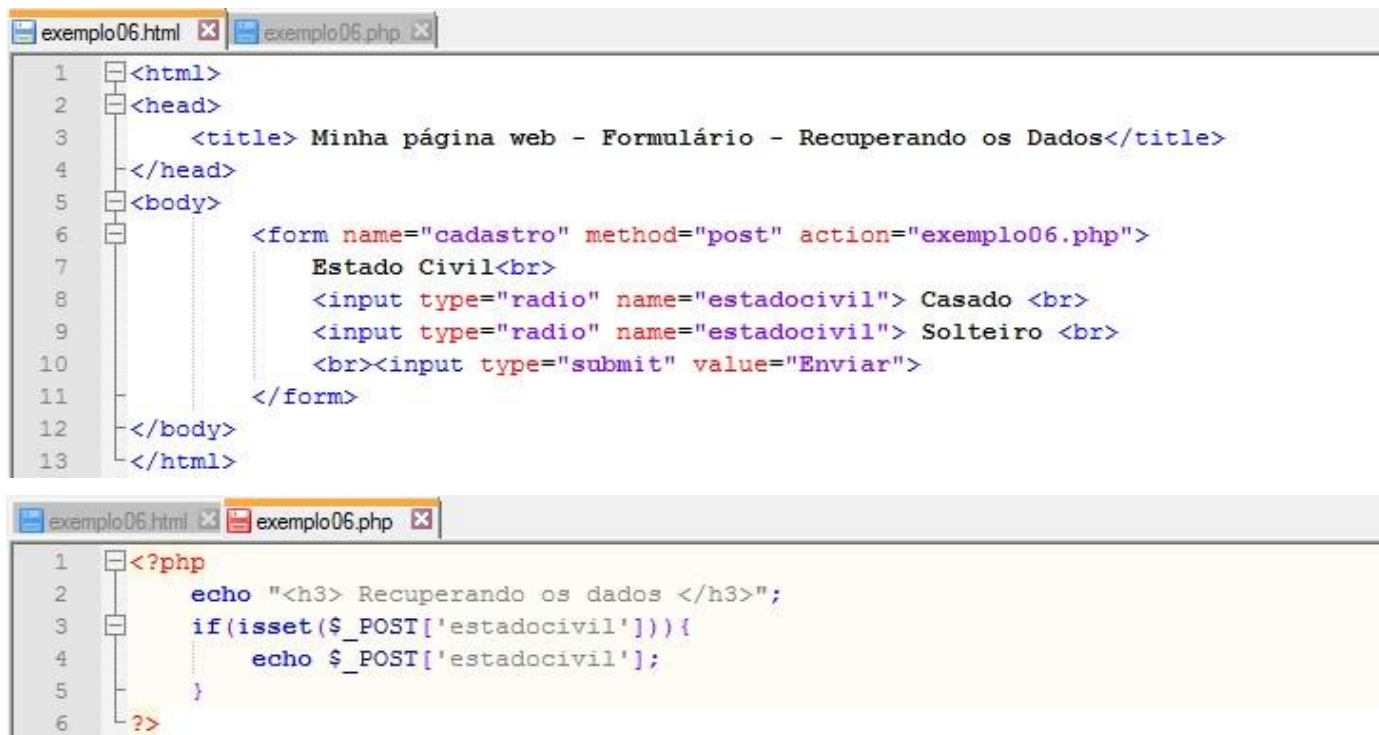
Verifique que a resposta apresenta somente a expressão **on**, informando que a opção foi selecionada. Para melhorar a recuperação dos dados utilize o *value* no elemento *checkbox* do seu formulário.

```
1 <html>
2 <head>
3     <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6     <form name="cadastro" method="post" action="exemplo05.php">
7         Qual linguagem de programação você conhece?<br>
8         <input type="checkbox" name="asp" value="Sim, você escolheu ASP"> ASP <br>
9         <input type="checkbox" name="php" value="Selecionado"> PHP <br>
10        <input type="checkbox" name="java" value="Java"> JAVA <br>
11        <br><input type="submit" value="Enviar">
12    </form>
13 </body>
14 </html>
```

```
1 <?php
2     echo "<h3> Recuperando os dados </h3>";
3
4     if (isset($_POST['asp'])) {
5         echo "ASP: " . $_POST['asp'];
6         echo "<br>";
7     }
8     if (isset($_POST['php'])) {
9         echo "PHP: " . $_POST['php'];
10        echo "<br>";
11    }
12    if (isset($_POST['java'])) {
13        echo "JAVA: " . $_POST['java'];
14        echo "<br>";
15    }
16    ?>
```


1.1.3 Elemento *radio*

Semelhante à estrutura de recuperação do *checkbox*, utilize o **isset()** para evitar erros de definição.



```
1 <html>
2 <head>
3     <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6     <form name="cadastro" method="post" action="exemplo06.php">
7         Estado Civil<br>
8         <input type="radio" name="estadocivil"> Casado <br>
9         <input type="radio" name="estadocivil"> Solteiro <br>
10        <br><input type="submit" value="Enviar">
11    </form>
12 </body>
13 </html>
```

```
1 <?php
2     echo "<h3> Recuperando os dados </h3>";
3     if(isset($_POST['estadocivil'])) {
4         echo $_POST['estadocivil'];
5     }
6 >
```

Note que novamente a resposta é a expressão **on**, para qualquer uma das opções selecionadas, desta forma, utilize o *value* no elemento *radio* do seu formulário para atribuir um valor.



```
1 <html>
2 <head>
3     <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6     <form name="cadastro" method="post" action="exemplo07.php">
7         Estado Civil<br>
8         <input type="radio" name="estadocivil" value="Casado"> Casado <br>
9         <input type="radio" name="estadocivil" value="Solteiro"> Solteiro <br>
10        <br><input type="submit" value="Enviar">
11    </form>
12 </body>
13 </html>
```

```
exemplo07.html x exemplo07.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      if(isset($_POST['estadocivil'])){
4          echo $_POST['estadocivil'];
5      }
6  ?>
```

1.1.4 Elemento *oculto*

Para recuperar elementos ocultos (*hidden*) o processo é o mesmo que o elemento do tipo *text*.

```
exemplo08.html x exemplo08.php x
1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo08.php">
7          Enviando dados ocultos.
8          <input type="hidden" name="user" value="admin">
9          <input type="hidden" name="nivel" value="2">
10         <input type="submit" value="OK">
11     </form>
12 </body>
13 </html>

exemplo08.html x exemplo08.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3
4      echo "Olá, usuário <b>".$_POST['user']."</b> você possui o "
5      .$_POST['nivel']."º nível de acesso."
6
7  ?>
```

1.1.5 Elemento *select*

Indiferente da quantidade de opções (*option*) o valor de recuperação, estará indicado no atributo *name*.

```
exemplo09.html x exemplo09.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo09.php">
7     Estados:
8     <select name="estados">
9       <option> São Paulo </option>
10      <option> Rio de Janeiro </option>
11      <option> Minas Gerais </option>
12      <option> Bahia </option>
13    </select><br><br>
14    <input type="submit" value="Enviar">
15  </form>
16 </body>
17 </html>
```

```
exemplo09.html x exemplo09.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "<b>Estado: </b>".$_POST['estados'];
4 >?>
```

Melhorando o resultado da recuperação com *value*.

```
exemplo10.html x exemplo10.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo09.php">
7     Estados:
8     <select name="estados">
9       <option value="SP"> São Paulo </option>
10      <option value="RJ"> Rio de Janeiro </option>
11      <option value="MG"> Minas Gerais </option>
12      <option value="BA"> Bahia </option>
13    </select><br><br>
14    <input type="submit" value="Enviar">
15  </form>
16 </body>
17 </html>
```



```
exemplo10.html x exemplo10.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "<b>Estado (sigla): </b>".$_POST['estados'];
4  ?>
```

1.1.6 Elemento *textarea*

```
exemplo11.html x exemplo11.php x
1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <form name="cadastro" method="post" action="exemplo11.php">
7          Mensagem:<br>
8          <textarea name="mensagem" cols="80" rows="7"></textarea>
9          <br><br>
10         <input type="submit" value="Enviar">
11     </form>
12 </body>
13 </html>
```

```
exemplo11.html x exemplo11.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "<b>Mensagem: </b><br>".$_POST['mensagem'];
4  ?>
```

Recuperação semelhante ao *text*.

1.2 Uso do \$_GET[]

O método **get** (formulário), passa suas variáveis via URL deixando as informações visíveis para futura recuperação. Cuidado ao expor as informações e principalmente com o limite de caracteres de uma URL.

```
exemplo01.html exemplo01.php
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="get" action="exemplo01.php">
7     Login:
8     <input type="text" name="nome"/> <br><br>
9     E-mail:
10    <input type="password" name="email"/> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>
```

```
exemplo01.html exemplo01.php
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Nome: " . $_GET['nome'];
4   echo "<br>";
5   echo "E-mail: " . $_GET['email'];
6 >
```

1.3 Uso do \$_REQUEST[]

Semelhante ao \$_POST[] e \$_GET[], o \$_REQUEST é uma variável de coleção genérica, sendo assim não há diferença se o valor passado for pelo método **get** ou **post**. Geralmente a busca segue uma ordem: **get**, **post**, **cookies** e **sessão** (ver *variables_order* do pnp.ini).

Usando método **get**.

Etec "Sales Gomes" – 101 – Tatuí

```
exemplo02.html x exemplo02.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="get" action="exemplo02.php">
7     Login:
8     <input type="text" name="nome"/> <br><br>
9     E-mail:
10    <input type="password" name="email"/> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>
```

```
exemplo02.html x exemplo02.php x
1 <?php
2   echo "<h3> Recuperando os dados </h3>";
3   echo "Nome: " . $_REQUEST['nome'];
4   echo "<br>";
5   echo "E-mail: " . $_REQUEST['email'];
6 ?>
```

Usando método **post**.

```
exemplo03.html x exemplo03.php x
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <form name="cadastro" method="post" action="exemplo03.php">
7     Login:
8     <input type="text" name="nome"/> <br><br>
9     E-mail:
10    <input type="password" name="email"/> <br><br>
11    <input type="submit" value="Enviar" />
12  </form>
13 </body>
14 </html>
```

```
exemplo03.html x exemplo03.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Nome: " . $_REQUEST['nome'];
4      echo "<br>";
5      echo "E-mail: " . $_REQUEST['email'];
6  ?>
```

2 Uso do \$_GET[] com URL

Muitas vezes utilizamos os links ou formamos links dinâmicos que permitem a passagem de dados via URL, desta forma podemos utilizar o \$_GET[] ou \$_REQUEST[], para recuperar estes dados.

As informações (**variável e conteúdo**), devem ser passadas depois no nome do arquivo, precedido do sinal de "?".

```
exemplo04.html x exemplo04.php x
1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6
7      <a href="exemplo04.php?id=10"> Passando dados via URL </a>
8
9  </body>
10 </html>

exemplo04.html x exemplo04.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Id: " . $_GET['id'];
4  ?>
```

Para passar mais de um valor via URL, utilize o símbolo "&" para separar os elementos.

```
exemplo05.html x exemplo05.php x
1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6
7      <a href="exemplo05.php?id=10&user=admin"> Passando dados via URL </a>
8
9  </body>
10 </html>
```



```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Id: " . $_GET['id'];
4      echo "<br>";
5      echo "User: " . $_GET['user'];
6  ?>

```

Usando o mesmo exemplo com `$_REQUEST[]`.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6
7      <a href="exemplo06.php?id=10&user=admin"> Passando dados via URL </a>
8
9  </body>
10 </html>

```

```

1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Id: " . $_REQUEST['id'];
4      echo "<br>";
5      echo "User: " . $_REQUEST['user'];
6  ?>

```

2.1 Uso de variáveis para compor a URL

Para criar links dinâmicos, podemos estruturá-los a partir de variáveis do PHP.

```

1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6
7      <?php
8          $idBanco = 10;
9          $nomeUser = "admin"
10      ?>
11
12      <a href="exemplo07_dados.php?id=<?php echo $idBanco;?>&user=<?php echo $nomeUser;?>">
13          Passando dados via URL </a>
14
15  </body>
16 </html>

```

Etec "Sales Gomes" – 101 – Tatuí

```
exemplo07.php x exemplo07_dados.php x
1  <?php
2      echo "<h3> Recuperando os dados </h3>";
3      echo "Id: " . $_REQUEST['id'];
4      echo "<br>";
5      echo "User: " . $_REQUEST['user'];
6  ?>
```

Ou podemos indicar do seguinte formato.

```
exemplo08.php x exemplo08_dados.php x
1  <html>
2  <head>
3      <title> Minha página web - Formulário - Recuperando os Dados</title>
4  </head>
5  <body>
6      <?php
7          $idBanco = 10;
8          $nomeUser = "admin";
9
10         echo "<a href='exemplo08_dados.php?id=";
11         echo $idBanco;
12         echo "&user=";
13         echo $nomeUser;
14         echo "'>Passando dados via URL </a>";
15     ?>
16 </body>
17 </html>
```

Simplificando.

Etec "Sales Gomes" – 101 – Tatuí

```
exemplo09.php exemplo09_dados.php
1 <html>
2 <head>
3   <title> Minha página web - Formulário - Recuperando os Dados</title>
4 </head>
5 <body>
6   <?php
7     $idBanco = 10;
8     $nomeUser = "admin";
9     echo "<a href='exemplo09_dados.php?id=".$idBanco."&user=".$nomeUser."'>Passando dados via URL </a>";
10  ?>
11 </body>
12 </html>
```

Ou ainda:

```
<?php
$idBanco = 10;
$nomeUser = "admin";
echo "<a href='exemplo10_dados.php?id=$idBanco&user=$nomeUser'>Passando dados via URL </a>";
?>
```