

PHP

Aula 4



Conteúdo da aula

Nesta aula iremos discutir mais um pouco sobre o tratamento de requisições em PHP, abordando o método GET, depois veremos os operadores básicos, matemáticos, condicionais e lógicos e por fim as estruturas condicionais.

Relembrando...



Relembrando

Na aula anterior fizemos um exemplo e um exercício usando o método POST, que utiliza um formulário e uma página de destino.



Relembrando

No exemplo informamos um nome e um e-mail, que são “pegos” na página de destino e mostrados na tela.

Utilizamos ainda o nome para alterar a cor do plano de fundo da página, mesclando PHP com o CSS.



```
13 <h1>Formulário</h1>
14 <hr>
15
16 <form action="destino-sb.php" method="POST">
17     <label for="nome">Nome:</label>
18     <input type="text" name="nome" id="nome" required>
19     <br><br>
20
21     <label for="email">E-mail</label>
22     <input type="email" name="email" id="email"
23     placeholder="Digite seu email" required>
24     <br><br>
25
26     <button type="submit">Cadastrar</button>
27     <button type="reset">Limpar</button>
28 </form>
```



Relembrando

```
9  <body>
10  <h1>Destino</h1>
11  <hr>
12  <?php
13      $nome = filter_input(INPUT_POST, "nome", FILTER_SANITIZE_SPECIAL_CHARS);
14      $email = filter_input(INPUT_POST, "email", FILTER_SANITIZE_EMAIL);
15
16      echo "<p>Nome informado: $nome<br>Email: $email</p>";
17  ?>
18  <style>
19      body{
20          background-color: <?= $nome; ?>;
21      }
22  </style>
```

Relembrando

Agora continuando...

Tratamento de requisições

Tratamento de requisições

Agora vamos fazer um exemplo usando o método GET.

Utilizaremos como base a página de destino do exemplo anterior (a que tem nome e e-mail só).

Esta:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Destino</title>
8  </head>
9  <body>
10     <h1>Destino</h1>
11     <hr>
12     <?php
13         $nome = filter_input(INPUT_POST, "nome", FILTER_SANITIZE_SPECIAL_CHARS);
14         $email = filter_input(INPUT_POST, "email", FILTER_SANITIZE_EMAIL);
15
16         echo "<p>Nome informado: $nome<br>Email: $email</p>";
17     ?>
18     <style>
19         body{
20             background-color: <?= $nome; ?>;
21         }
22     </style>
23 </body>
24 </html>
```



Tratamento de requisições

Desta forma, crie uma **nova página PHP** e copie o conteúdo desta página de destino e salve-a como “**destino-get.php**” por exemplo.



Tratamento de requisições

Agora vamos alterar o título da página e também o tipo de método utilizado (no *filter_input*)



Tratamento de requisições

```
12 <body>
13     <h1>Destino GET</h1>
14     <hr>
15     <?php
16         $nome = filter_input(INPUT_GET, "nome", FILTER_SANITIZE_SPECIAL_CHARS);
17         $email = filter_input(INPUT_GET, "email", FILTER_SANITIZE_EMAIL);
18
19         echo "<p>Nome informado: $nome<br>Email: $email</p>";
20     ?>
21     <style>
22     body{
23         background-color: <?= $nome; ?>;
24     }
25 </style>
```



Tratamento de requisições

Salve esta página e copie-a no diretório raiz do apache (C:\xampp\htdocs).

Esta etapa é sempre necessária caso você não esteja trabalhando (salvando) diretamente os arquivos dentro do diretório htdocs...



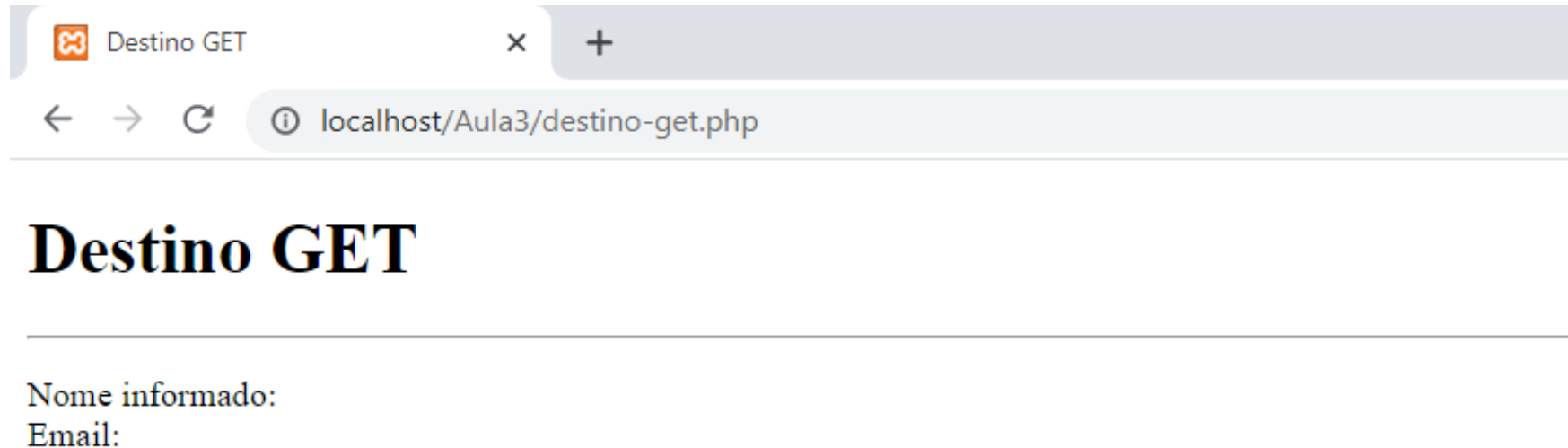
Tratamento de requisições

Agora abra a página que você acabou de criar no navegador.

<http://localhost/destino-get.php>

Ou algo parecido..

Tratamento de requisições





INSTITUTO
FEDERAL
São Paulo

Tratamento de requisições

A página não deve estar mostrando nenhuma informação,
pois não enviamos nenhum parâmetro ainda.



Tratamento de requisições

Como já foi dito, o método GET utiliza a URL da página para enviar os dados para a página de destino.

Podemos colocar as informações diretamente na URL.



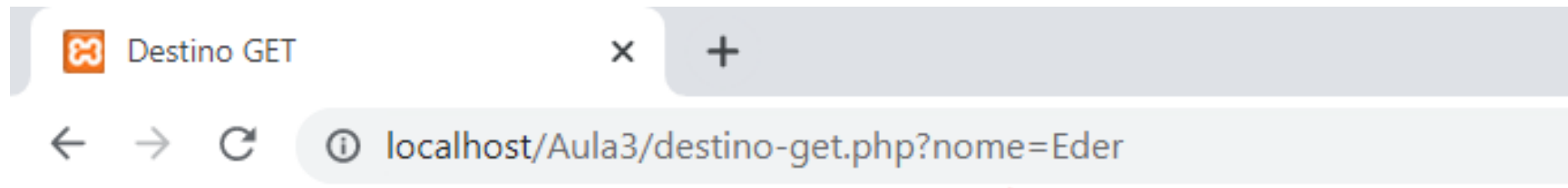
Tratamento de requisições

A sintaxe do método GET consiste em colocar na URL, logo após o nome da página, um símbolo de ? (interrogação).

Depois da interrogação vai o nome do campo, um sinal de igual “=” e o valor do mesmo.

Tratamento de requisições

Veja como fica => **destino-get.php?nome=Eder**



Destino GET

Nome informado: Eder
Email:

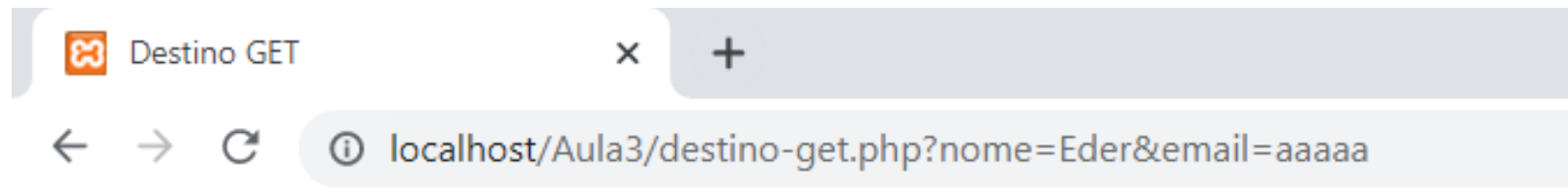


Tratamento de requisições

Para se colocar mais de um campo utiliza-se o símbolo “e-comercial” (&) seguido do nome do campo, o sinal de igual e seu valor.

Tratamento de requisições

Assim => **destino-get.php?nome=Eder&email=aaa**



Destino GET

Nome informado: Eder
Email: aaaaa



Tratamento de requisições

No entanto, esta forma de informar os parâmetros e valores diretamente na URL não é prática, nem a única forma de se utilizar o método.

Podemos utilizar um formulário com o atributo *method* igual a GET.



Tratamento de requisições

Ou podemos ainda criar um link (*a href*) com os parâmetros e valores na URL do link.

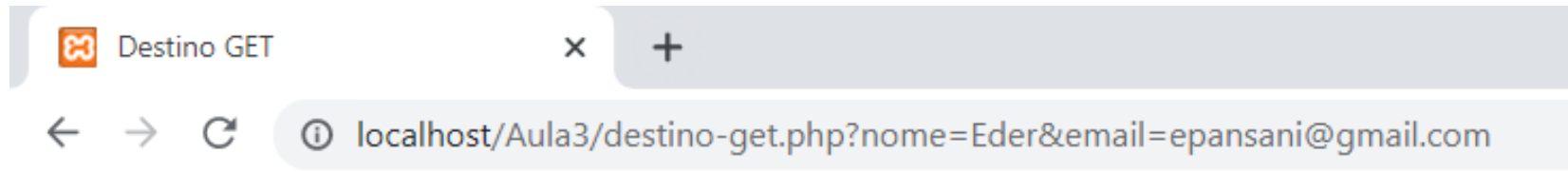
Vamos ver um exemplo com isso.



```
12 <body>
13     <h1>Destino GET</h1>
14     <hr>
15     <?php
16         $nome = filter_input(INPUT_GET, "nome", FILTER_SANITIZE_SPECIAL_CHARS);
17         $email = filter_input(INPUT_GET, "email", FILTER_SANITIZE_EMAIL);
18
19         echo "<p>Nome informado: $nome<br>Email: $email</p>";
20     ?>
21     <p>
22         <a href="destino-get.php?nome=Eder&email=epansani@gmail.com">
23             Nome: Eder Pansani Email: epansani@gmail.com
24         </a>
25     </p>
26     <style>
27         body{
28             background-color: <?= $nome; ?>;
29     }
```



Tratamento de requisições



Destino GET

Nome informado: Eder
Email: epansani@gmail.com

[Nome: Eder Pansani Email: epansani@gmail.com](#)



Tratamento de requisições

Agora vamos criar um segundo link, passando parâmetros diferentes:

```
17 <p>
18     <a href="destino-get.php?nome=José da Silva&email=josesilva29@gmail.com">
19         Outro link
20     </a>
21 </p>
```



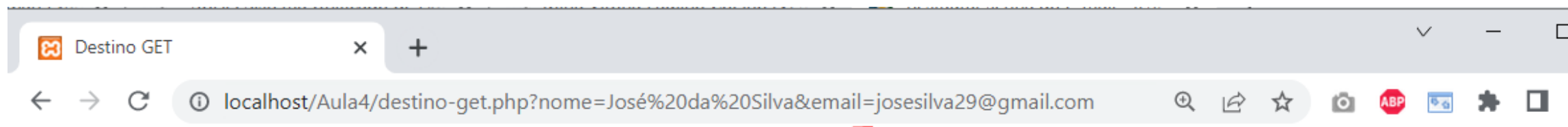
Tratamento de requisições

Note neste segundo link que o texto que aparece no link não tem relação com os parâmetros enviados.



INSTITUTO
FEDERAL
São Paulo

Tratamento de requisições



Destino GET

Nome informado: José da Silva
Email: josesilva29@gmail.com

Nome: Eder Pansani Email: epansani@gmail.com

Outro link



Operadores



Operadores

O PHP divide os operadores nos seguintes grupos:

- Operadores aritméticos
- Operadores de atribuição
- Operadores de comparação
- Operadores de incremento/decremento



Operadores

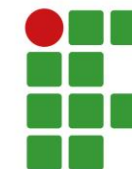
- Operadores lógicos
- Operadores de *string*
- Operadores de matriz
- Operadores de atribuição condicional



Operadores

Operadores aritméticos

Os operadores aritméticos do PHP são usados com valores numéricos para realizar operações aritméticas comuns, como adição, subtração, multiplicação etc.



Operadores

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power



Operadores

Operadores de atribuição

Os operadores de atribuição PHP são usados para definir o valor de uma variável. O operador básico de atribuição é “=”.



Operadores

Assignment	Same as...	Description
$x = y$	$x = y$	The left operand gets set to the value of the expression on the right
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x \% = y$	$x = x \% y$	Modulus



Operadores

Operadores de comparação

Os operadores de comparação do PHP são usados para comparar dois valores (números ou *strings*).

Operator	Name	Example	Result	St
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y	0
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type	L
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y	
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y	
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type	
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y	
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y	
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y	
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y	
<=>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7.	



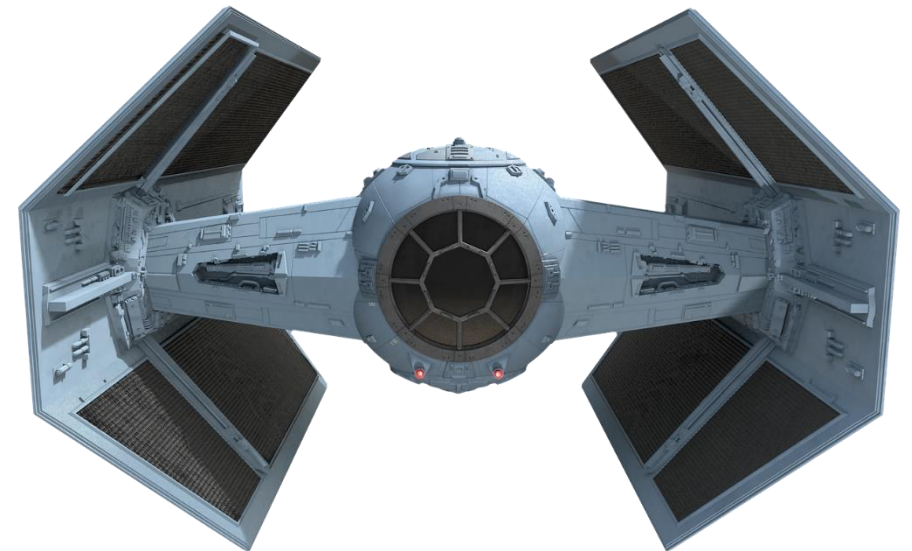
INSTITUTO
FEDERAL
São Paulo

Operadores

Operadores de comparação

Operador "nave espacial" (*spaceship*)

$\$x \leq \y





Operadores

Operadores de comparação

Operador "nave espacial" (*spaceship*)

O operador nave espacial é utilizado para comparar duas expressões. Ele retorna -1, 0 ou 1 quando \$x for respectivamente menor que, igual a ou maior que \$y.



Operadores

Operadores de Incremento/Decremento

Os operadores de incremento são usados para incrementar o valor de uma variável enquanto os operadores de decremento são usados para decrementar o valor de uma variável.



Operadores

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x++</code>	Post-increment	Returns <code>\$x</code> , then increments <code>\$x</code> by one
<code>--\$x</code>	Pre-decrement	Decrements <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x--</code>	Post-decrement	Returns <code>\$x</code> , then decrements <code>\$x</code> by one



Operadores

Operadores de Incremento/Decremento

https://www.w3schools.com/php/phptryit.asp?filename=tryphp_oper_pre_incr



Operadores

Operadores lógicos

Os operadores lógicos do PHP são usados para combinar instruções condicionais.



Operadores

Operator	Name	Example	Result
and	And	\$x and \$y	True if both \$x and \$y are true
or	Or	\$x or \$y	True if either \$x or \$y is true
xor	Xor	\$x xor \$y	True if either \$x or \$y is true, but not both
&&	And	\$x && \$y	True if both \$x and \$y are true
	Or	\$x \$y	True if either \$x or \$y is true
!	Not	!\$x	True if \$x is not true



Operadores

Operadores de concatenação de *strings*

O PHP tem dois operadores que são especialmente projetados para *strings*.

Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1



Operadores

Operadores de concatenação de *strings*

Importante! O PHP realiza automaticamente a conversão de tipos no caso de uma concatenação de uma *string* com um número.



Operadores

Isto é, este código é válido:

```
<?php
```

```
$fruta = "Maça";
```

```
$preco = 4.90;
```

```
echo "O preço da " . $maca . " é de R$ " . $preco . " o Kg.";
```

```
?>
```



Operadores

Os operadores de *Arrays* e de atribuição condicional em PHP não serão abordados, mas podem ser vistos na página do W3Schools:

https://www.w3schools.com/php/php_operators.asp

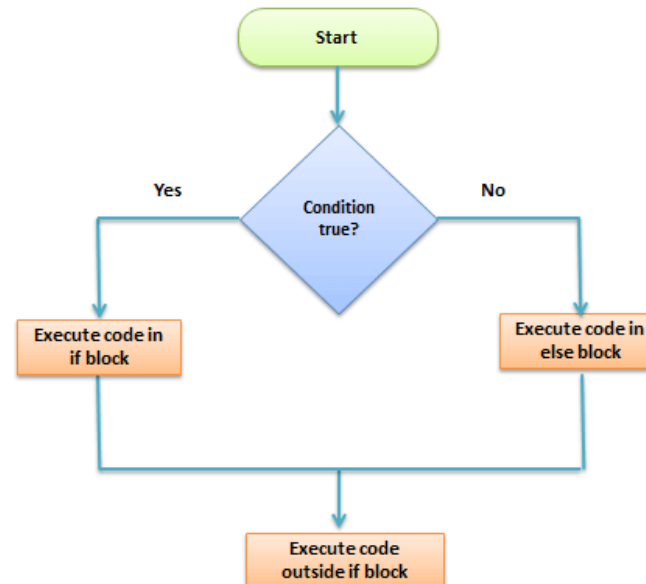
Estruturas condicionais

IF, ELSE e ELSEIF



Estruturas condicionais

As instruções condicionais são usadas para executar diferentes ações com base em diferentes condições.





Estruturas condicionais

PHP suporta as seguintes declarações condicionais:

- ***if*** - executa algum código se uma condição for verdadeira
- ***if...else*** - executa algum código se uma condição for verdadeira e outro código se essa condição for falsa



Estruturas condicionais

- ***if...elseif...else*** - executa códigos diferentes para mais de duas condições
- ***switch*** - seleciona um dos muitos blocos de código a serem executados



Estruturas condicionais

A instrução **if** executa algum código se uma condição for verdadeira.

```
if (condition) {  
    code to be executed if condition is true;  
}
```



Estruturas condicionais

A instrução ***if...else*** executa algum código se uma condição for verdadeira e outro código se essa condição for falsa.

```
if (condition) {  
    code to be executed if condition is true;  
} else {  
    code to be executed if condition is false;  
}
```




Estruturas condicionais

```
<?php
$d = date("N");

if ($d == "5") {
    echo "Bom dia. Sextou!!!!!!!!";
}else{
    echo "Bom dia...";
}
?>
```



Estruturas condicionais

Função *date* PHP:

https://www.php.net/manual/pt_BR/function.date.php

https://www.php.net/manual/pt_BR/datetime.format.php



Estruturas condicionais

A instrução ***if...elseif...else*** executa códigos diferentes para mais de duas condições.

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    code to be executed if first condition is false and this condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```



Estruturas condicionais

Exemplo,

```
<?php
$t = date("H");

if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```



Estruturas condicionais

Quando não explicitamos o operador lógico em testes por meio do IF, o comportamento padrão do PHP é retornar TRUE sempre que a variável tiver conteúdo válido.

Observe o exemplo,



Estruturas condicionais

```
<?php
$d = date("N");
if($d){
    echo 'A variável $d está preenchida.';
}
if($e){
    echo 'A variável $e está preenchida.';
}
?>
```



Estruturas condicionais



localhost/Aula3/test.php



A variável \$d está preenchida.

Warning: Undefined variable \$e in **C:\xampp\htdocs\Aula3\test.php** on line **38**



Estruturas condicionais

A instrução ***switch*** também é usada para executar diferentes ações com base em diferentes condições. Seu diferencial é selecionar um dos **muitos blocos de código** a serem executados .



E

```
switch (n) {  
    case label1:  
        code to be executed if n=label1;  
        break;  
    case label2:  
        code to be executed if n=label2;  
        break;  
    case label3:  
        code to be executed if n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is different from all labels;  
}
```



Estruturas condicionais

A declaração ***switch*** é similar a uma série de declarações ***IF*** na mesma expressão. Ela serve para comparar as **mesmas** variáveis (ou expressões), com diferentes valores, e executar pedaços diferentes de código dependendo de qual valor ela é igual.



Estruturas condicionais

```
<?php
```

```
if ($i == 0) {  
    echo "i é igual a 0";  
} elseif ($i == 1) {  
    echo "i é igual a 1";  
} elseif ($i == 2) {  
    echo "i é igual a 2";  
}
```

```
?>
```



Estruturas condicionais

```
<?php
switch ($i) {
    case 0:
        echo "i é igual a 0";
        break;
    case 1:
        echo "i é igual a 1";
        break;
    case 2:
        echo "i é igual a 2";
        break;
}
?>
```



Estruturas condicionais

Existe ainda a instrução especial **default**. Ela é um tipo de *case* que corresponde a tudo que não foi correspondido pelos outros cases.

Veja o exemplo,



Estruturas condicionais

```
<?php
switch ($i) {
    case 0:
    case 1:
    case 2:
        echo "i é menor que 3 mas não é negativo";
        break;
    case 3:
        echo "i é 3";
    default:
        echo "i é maior que 3";
}
?>
```



Estruturas condicionais

Importante!

Quando usado com *strings* o comando ***switch*** ele executa por padrão uma **comparação do tipo “==”** (igual, não idêntico)

Praticando



**INSTITUTO
FEDERAL**
São Paulo

Praticando

Vamos agora praticar um pouco com um exercício.





Praticando

Praticando 1 – Página “destino-get.php” aprimorada

A página de exemplo “destino-get.php” que fizemos em aula possui dois parâmetros: nome e e-mail, porém o nome também serve para mudar a cor de plano de fundo da página.



Praticando

Crie um novo parâmetro para a cor, separado do nome.

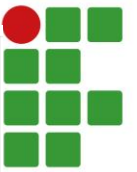
Assim será possível enviar 3 parâmetros por GET e alterar a página dinamicamente.



Praticando

Dica, você deverá:

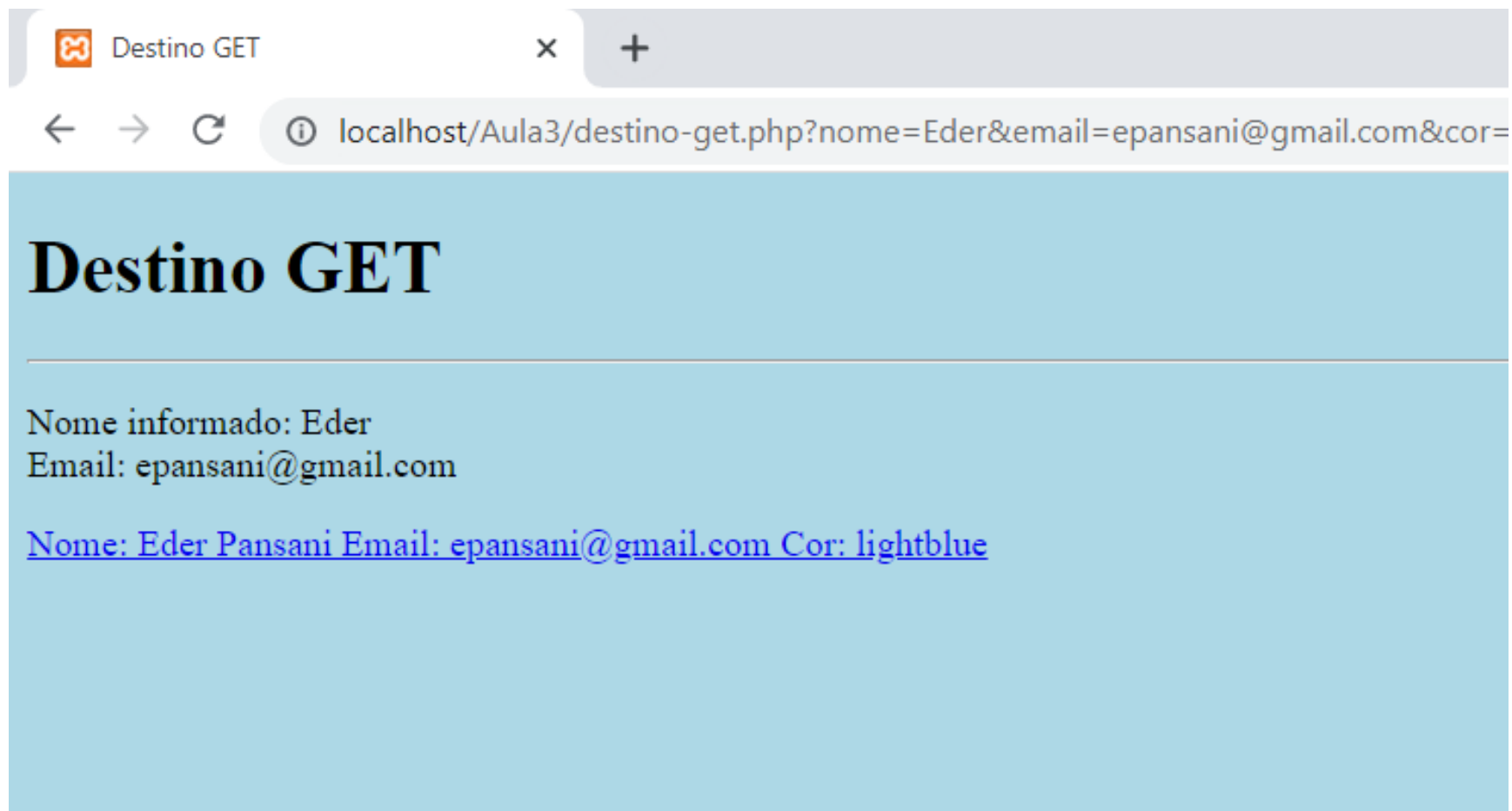
1. Pegar o novo parâmetro (cor)
2. Alterar o nome da variável no CSS
3. Colocar o novo parâmetro e seu valor da URL do link



```
12 <body>
13     <h1>Destino GET</h1>
14     <hr>
15     <?php
16         $nome = filter_input(INPUT_GET, "nome", FILTER_SANITIZE_SPECIAL_CHARS);
17         $email = filter_input(INPUT_GET, "email", FILTER_SANITIZE_EMAIL);
18         $cor = filter_input(INPUT_GET, "cor", FILTER_SANITIZE_EMAIL);
19
20         echo "<p>Nome informado: $nome<br>Email: $email</p>";
21     ?>
22     <p>
23         <a href="destino-get.php?nome=Eder&email=epansani@gmail.com&cor=lightblue">
24             Nome: Eder Pansani Email: epansani@gmail.com
25         </a>
26     </p>
27     <p>
28         <a href="destino-get.php?nome=José da Silva&email=josesilva29@gmail.com">
29             Outro link
30         </a>
31     </p>
32     <style>
33         body{
34             background-color: <?= $cor; ?>;
35         }
36     </style>
```



Praticando





Praticando

Agora separe os parâmetros nome e e-mail em um link e as cores em outros links. Para que seja possível alterar (usando o método GET) o nome e o e-mail e também as cores de plano de fundo clicando em diferentes links.



Praticando

Por exemplo crie 5 links: dois com nomes e e-mails diferentes e três com cores diferentes. Eles funcionariam de forma independente, porém sempre mantendo os dados/cores da página.



Destino GET

Nome informado:

Email:

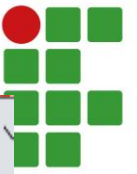
Nome: Eder Pansani Email: epansani@gmail.com


Outro link (do José).

Cor: azul claro (lightblue).

Cor: verde claro (lightgreen).

Cor: vermelho claro (lightcoral).



 Destino GET

localhost/Aula4/praticando-get.php?nome=Eder&email=epansani@gmail.com&cor=lightblue

Destino GET

Nome informado: Eder
Email: epansani@gmail.com

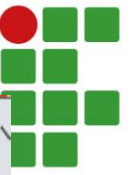
[Nome: Eder Pansani Email: epansani@gmail.com](#)


[Outro link \(do José\)](#)

[Cor: azul claro \(lightblue\)](#)

[Cor: verde claro \(lightgreen\)](#)

[Cor: vermelho claro \(lightcoral\)](#)



 Destino GET

localhost/Aula4/praticando-get.php?nome=Eder&email=epansani@gmail.com&cor=lightgreen

Destino GET

Nome informado: Eder
Email: epansani@gmail.com

[Nome: Eder Pansani Email: epansani@gmail.com](#)


[Outro link \(do José\)](#)

[Cor: azul claro \(lightblue\)](#)

[Cor: verde claro \(lightgreen\)](#)

[Cor: vermelho claro \(lightcoral\)](#)



 Destino GET

localhost/Aula4/praticando-get.php?nome=José%20da%20Silva&email=josesilva29@gmail.com...

Destino GET

Nome informado: José da Silva
Email: josesilva29@gmail.com

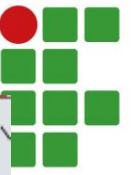
Nome: Eder Pansani Email: epansani@gmail.com


Outro link (do José)

Cor: azul claro (lightblue)

Cor: verde claro (lightgreen)

Cor: vermelho claro (lightcoral)



 Destino GET

localhost/Aula4/praticando-get.php?nome=José%20da%20Silva&email=josesilva29@gmail.com...

Destino GET

Nome informado: José da Silva
Email: josesilva29@gmail.com

[Nome: Eder Pansani Email: epansani@gmail.com](#)

[Outro link \(do José\)](#)

[Cor: azul claro \(lightblue\)](#)

[Cor: verde claro \(lightgreen\)](#)

[Cor: vermelho claro \(lightcoral\)](#)



Praticando

Praticando 2 - Calculadora média

Crie um formulário que tenha três campos, Nota 1, Nota 2 e Nota 3. Depois crie uma página destino para esse formulário que receba esses valores (usando POST) e calcule a média entre eles.



Praticando

Depois mostre a média e uma mensagem de acordo com as seguintes regras:

- Média menor que 4, **reprovado**
- Média maior que 4 e menor que 6, **de recuperação**
- Média maior ou igual a 6, **aprovado**



Praticando

Obs. Use um campo do tipo número, com valor mínimo igual a 0 e um passo (*step*) de 0,5.

Exemplo,



**INSTITUTO
FEDERAL**
São Paulo

Praticando

Praticando - Calculadora média

Nota 1:

Nota 2:

Nota 3:

Calcular média

Limpar



**INSTITUTO
FEDERAL**
São Paulo

Praticando

Praticando - Calculadora média

Um aluno com as notas **1**, **2** e **4** tem uma média igual a **2.33333333333333**

Com essa média o aluno está **REPROVADO**



INSTITUTO
FEDERAL
São Paulo

Praticando

Praticando - Calculadora média

Um aluno com as notas **5.5**, **3** e **4** tem uma média igual a **4.16666666666667**

Com essa média o aluno está **DE RECUPERAÇÃO**



**INSTITUTO
FEDERAL**
São Paulo

Praticando

Praticando - Calculadora média

Um aluno com as notas **7**, **8** e **4** tem uma média igual a **6.33333333333333**

Com essa média o aluno está **APROVADO**



Praticando

Praticando 3 - Animais

Crie uma página com quatro imagens de animais a sua escolha. Elas deverão estar posicionadas lado a lado horizontalmente.

Veja um exemplo:



**INSTITUTO
FEDERAL**
São Paulo

Praticando

← → ↻ ⓘ localhost/Aula3/praticando_bichos.php 🔍 ↗ ☆ 📷 ABP 📅 ⚙️ 🖱️ 👤 ⋮

Praticando - Animais





Praticando

Ao clicar em alguma das imagens uma nova página se abrirá (página de destino) mostrando:

- Qual o nome do animal que você clicou
- Algumas informações (simples) sobre o animal
- A imagem do animal



Praticando

Obs. Deverá existir apenas uma página de destino, que mostrará diferentes valores dependendo do que foi enviado (usando método GET) pela página de origem.



Praticando - Animais

Você clicou no **Gato**.

O Gato é um mamífero da família dos felídeos.





Praticando - Animais

Você clicou no **Cachorro**.

O Cachorro é um mamífero da família dos canídeos.



Praticando - Animais

Você clicou no **Hamster**.

O Hamster é um mamífero roedor da subfamília dos Cricetinae.





**INSTITUTO
FEDERAL**
São Paulo

Fim!



**INSTITUTO
FEDERAL**
São Paulo

Fim!

Dúvidas?

Perguntas?

Sugestões?