

Treinamento Fundamentos de PHP

Turma 2023/01 – Aula 3/8



1

Semana 1



Aula 3 – Funções, escopo e web

- Funções definidas pelo usuário
- Argumentos de funções
- Escopo
- Funções anônimas
- Arrow function
- Inclusão de arquivo
- PHP na web

2

Recapitulando...



Na última aula vimos sobre:

- Operadores
 - Aritméticos
 - de comparação
 - Lógicos
- Estrutura de controle
 - If/else,
 - Switch
- Estrutura de repetição
 - While
 - Do-while
 - For
 - Foreach
- Atividade 1: Calculadora CLI
- Desafio 1: Mega-Sena CLI
- Prazo para entrega: 22/06 (qui)

Copyright© 2023 Accenture All Rights Reserved

3

3

Funções



Recurso que permite criar blocos de código com uma determinada tarefa e que podem ser reutilizadas em diferentes partes do código.

Sintaxe:

```
function nomeFuncao($arg1, $arg2, ..., $argN) {
    //comandos
    return <expressão>; // opcional
}
```

https://www.php.net/manual/pt_BR/functions.user-defined.php

Copyright© 2023 Accenture All Rights Reserved

4

4

Funções



Principais características:

- Uma vez definida a função, ela pode ser chamada em qualquer parte do código.
- As funções no PHP podem ser aninhadas, o que significa que é possível definir uma função dentro de outra função. Isso é também conhecido como função interna.
- É possível definir funções condicionais, que são criadas somente se uma determinada condição for atendida
- As funções também podem ser passadas como parâmetros para outras funções ou armazenadas em variáveis, o que é conhecido como funções de primeira classe.
- O PHP não suporta sobrecarga de funções, e também não é possível cancelar ou alterar a definição de funções previamente declaradas.

Copyright© 2023 Accenture All Rights Reserved

5

5

Funções



Dica: Você pode consultar as funções internas do PHP em:

https://www.php.net/manual/pt_BR/indexes.functions.php

Copyright© 2023 Accenture All Rights Reserved

6

6

Funções



Parâmetros (ou argumentos)

- Usado para passar um valor para o escopo da função
- O PHP suporta a passagem de parâmetros por valor (o padrão), passagem por referência e valores padrões de argumentos. Lista de argumentos de número **variável** e argumentos nomeados também são suportadas.
- Uma função pode ter ou não parâmetros. Em caso de não ter, os parênteses são declarados vazio
- Quando declarado, por padrão, é obrigatória a passagem do parâmetro na chamada da função
- Para definir valores padrão para argumentos usa-se a sintaxe similar a atribuição de uma variável. **O padrão é usado apenas quando o parâmetro não é especificado.**

- Nas versões mais recentes, a declaração estrita de tipos de argumento é suportada
`declare(strict_types=1);`
https://www.php.net/manual/pt_BR/language.types.declarations.php#language.types.declarations.strict

Copyright© 2023 Accenture All Rights Reserved

7

7

Funções



Escopo

Define onde uma variável pode ser usada e acessada em um script PHP.

- Escopo global – definida fora de qualquer função e pode ser acessada de qualquer lugar do código PHP
- Escopo local – definida dentro de uma função e só pode ser acessada dentro dessa função

```

1  <?php
2
3  // escopo global
4  $numero1 = 10;
5
6  function somaNumeros() {
7      // acessando escopo global com a palavra chave global
8      global $numero1;
9
10     $numero2 = 20; // escopo local
11
12     return $numero1 + $numero2;
13 }
14
15 echo somaNumeros();
16

```

Copyright© 2023 Accenture All Rights Reserved

8

8

Funções



Superglobais

Variáveis nativas disponíveis em todos os escopos:

- **\$GLOBALS**: Referencia todas variáveis disponíveis no escopo global
- **\$_SERVER**: Informação do servidor e ambiente de execução
- **\$_GET**: Variáveis HTTP GET
- **\$_POST**: Variáveis HTTP POST
- **\$_FILES**: Variáveis de Upload de Arquivos HTTP
- **\$_COOKIE**: Cookies HTTP
- **\$_SESSION**: Variáveis de sessão
- **\$_REQUEST**: Variáveis de requisição HTTP
- **\$_ENV**: Variáveis de ambiente

https://www.php.net/manual/pt_BR/language.variables.superglobals.php

Copyright© 2023 Accenture All Rights Reserved

9

9

Funções



Passagem de parâmetro por valor x referência

- A passagem de parâmetro **por valor** é a forma padrão de passar parâmetros em PHP. Isso significa que uma **cópia do valor** do parâmetro é passada para a função, **deixando o valor original inalterado**. Qualquer modificação feita dentro da função na cópia do valor não afetará a variável original.
- Já na passagem de parâmetro **por referência**, a referência da variável é passada para a função, em vez do valor propriamente dito. Isso significa que, **qualquer alteração feita na variável dentro da função afetará diretamente a variável original**. Para passar um parâmetro por referência, é necessário colocar um **'&'** (e comercial) antes do nome do parâmetro na definição da função

Copyright© 2023 Accenture All Rights Reserved

10

10

Funções



Funções anônimas (Closure)

- São funções que **não possuem um nome definido** e podem ser utilizadas em situações onde é necessário passar uma função como parâmetro para outra função, ou para declarar uma função "inline", sem a necessidade de criar uma função nomeada separada.

```
1 <?php
2
3 // Função anônima que recebe um parâmetro e retorna o dobro do valor
4 $dobro = function($valor) {
5     return $valor * 2;
6 };
7
8 // Chamando a função anônima
9 echo $dobro(5); // Saída: 10
10
```

```
1 <?php
2
3 // Array de números
4 $numeros = [1, 2, 3, 4, 5];
5
6 // Utilizando a função anônima com array_map para obter o dobro de cada número
7 $numeros_dobrados = array_map(function($valor) {
8     return $valor * 2;
9 }, $numeros);
10
```

https://www.php.net/manual/pt_BR/functions.anonymous.php

Copyright© 2023 Accenture All Rights Reserved

11

11

Funções



Função nomeada x Função anônima

- A primeira é uma função anônima e é armazenada em uma variável **\$funcao_anonima**, que imprime a mensagem armazenada na variável **\$mensagem** usando a palavra-chave **use**.
- A segunda é uma função nomeada **funcaoNomeada** que recebe uma mensagem como argumento e a imprime.

```
aula03 > funcao_anonima4.php > ...
1 <?php
2
3 $mensagem = 'Olá, mundo!';
4
5 $funcao_anonima = function() use ($mensagem) {
6     echo $mensagem . PHP_EOL;
7 };
8
9 function funcaoNomeada($msg) {
10     echo $msg . PHP_EOL;
11 }
12
13 $funcao_anonima();
14 funcaoNomeada($mensagem);
15
16 function executaFuncao($funcao) {
17     $funcao();
18 }
19
20 executaFuncao($funcao_anonima, 'Oi');
21 executaFuncao(funcaoNomeada, 'Oi');
22
```

Copyright© 2023 Accenture All Rights Reserved

12

12

Funções



Função nomeada x Função anônima

- Ambas têm o mesmo resultado: imprimir a mensagem "Olá, mundo!". No entanto, elas são definidas de maneira diferente.
- A diferença principal é que a função anônima é criada e atribuída a uma variável, enquanto a função nomeada é definida com um nome específico.

```

aula03 > funcao_anonima4.php > ...
1  <?php
2
3  $mensagem = 'Olá, mundo!';
4
5  $funcao_anonima = function() use ($mensagem) {
6      echo $mensagem . PHP_EOL;
7  };
8
9  function funcaoNomeada($msg) {
10     echo $msg . PHP_EOL;
11 }
12
13 $funcao_anonima();
14 funcaoNomeada($mensagem);
15
16 function executaFuncao($funcao) {
17     $funcao();
18 }
19
20 executaFuncao($funcao_anonima, 'Oi');
21 executaFuncao(funcaoNomeada, 'Oi');
22
  
```

Copyright© 2023 Accenture All Rights Reserved

13

13

Funções



Arrow Function (Função seta)

- A função de seta é uma forma abreviada de escrever funções anônimas.
- Elas possuem algumas limitações em comparação com as funções anônimas tradicionais, como, por exemplo, não permitir o uso da palavra-chave **use** para importar variáveis externas

Nota: Após o fechamento das chaves nas funções anônimas, é necessário o ponto e vírgula para sinalizar o encerramento (ver linha 8).

```

aula03 > funcao_anonima3.php > ...
1  <?php
2
3  $a = [1, 2, 3, 4, 5];
4
5  // Função anônima
6  $dobro = function($valor) {
7      return $valor * 2;
8  };
9
10 // Função anônima abreviada (a partir do PHP 7.4)
11 $cubo = fn($n) => $n * $n * $n;
12
13
14 $b = array_map($cubo, $a);
15
16 print_r($b);
  
```

Copyright© 2023 Accenture All Rights Reserved

14

14

Inclusão de arquivo



- A inclusão de arquivos é comumente usada para evitar a duplicação de código em diferentes partes de um script e para organizar melhor o código em arquivos separados.
- As formas de incluir arquivos em um script PHP são:
 - **include** – Inclui um arquivo e se o arquivo não for encontrado, um aviso é emitido, mas o script continua sendo executado;
 - **require** – Inclui um arquivo e se o arquivo não for encontrado, um erro fatal é gerado e o script é interrompido;
 - **include_once**: Igual ao include, mas verifica se o arquivo já foi incluído antes. Caso positivo, ele não inclui novamente;
 - **require_once**: Igual ao require, mas verifica se o arquivo já foi incluído antes para não incluir novamente.

<https://www.php.net/manual/en/function.include.php>

Copyright© 2023 Accenture All Rights Reserved

15

15

Inclusão de arquivo



Exemplo básico de include

```
vars.php
<?php

$color = 'green';
$fruit = 'apple';

?>

test.php
<?php

echo "A $color $fruit"; // A

include 'vars.php';

echo "A $color $fruit"; // A green apple

?>
```

Copyright© 2023 Accenture All Rights Reserved

16

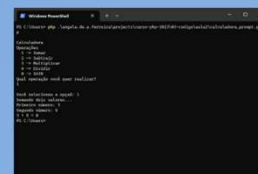
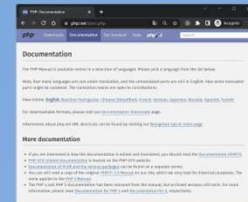
16

PHP na Web



A diferença entre o PHP no console e na web está na forma como ele é usado.

- Na web, o PHP é usado para gerar conteúdo dinâmico em uma página da web, processar formulários, interagir com bancos de dados, etc.
- No console, o PHP é usado para criar aplicativos de linha de comando (CLI) que podem fazer coisas como processar arquivos em lote, realizar tarefas agendadas, executar scripts de manutenção, entre outras.



Copyright© 2023 Accenture All Rights Reserved

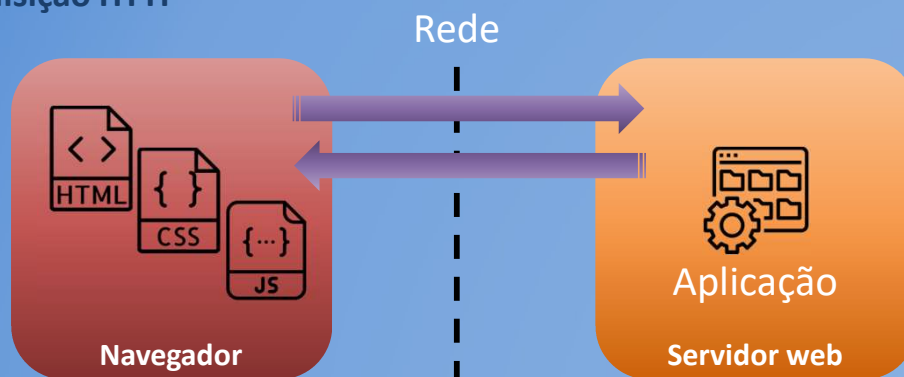
17

17

PHP na Web



Requisição HTTP



Copyright© 2023 Accenture All Rights Reserved

18

18

HTML



File | C:/Users/angela.de.a.ferreira/projects/curso-php-2023/turma_2023.1/aula3/pagina-web.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Título da página</title>
5 </head>
6 <body>
7   <!-- Conteúdo da página -->
8   <h1>Conteúdo da aula 3</h1>
9   <ol>
10    <li>Funções definidas pelo usuário</li>
11    <li>Argumentos de funções</li>
12    <li>Escopo</li>
13    <li>Funções anônimas (lambda)</li>
14    <li>Inclusão de arquivos de código</li>
15    <li>PHP na Web</li>
16  </ol>
17 </body>
18 </html>

```



file:///C:/Users/angela.de.a.ferreira/projects/curso-php-2023/turma_2023.1/aula3/pagina-web.html

Copyright© 2023 Accenture All Rights Reserved

19

19

HTML

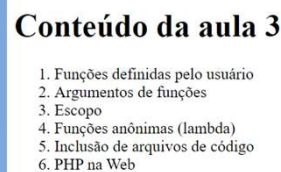


Título da página
localhost:8080/pagina-web.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Título da página</title>
5 </head>
6 <body>
7   <!-- Conteúdo da página -->
8   <h1>Conteúdo da aula 3</h1>
9   <ol>
10    <li>Funções definidas pelo usuário</li>
11    <li>Argumentos de funções</li>
12    <li>Escopo</li>
13    <li>Funções anônimas (lambda)</li>
14    <li>Inclusão de arquivos de código</li>
15    <li>PHP na Web</li>
16  </ol>
17 </body>
18 </html>

```



<http://localhost:8080/pagina-web.html>

Copyright© 2023 Accenture All Rights Reserved

20

20

PHP com HTML

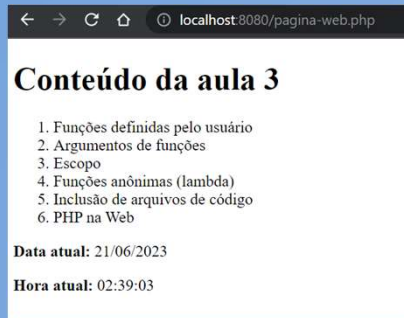


```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Título da página</title>
5 </head>
6 <body>
7   <!-- Conteúdo da página -->
8   <h1>Conteúdo da aula 3</h1>
9   <ol>
10    <li>Funções definidas pelo usuário</li>
11    <li>Argumentos de funções</li>
12    <li>Escopo</li>
13    <li>Funções anônimas (lambda)</li>
14    <li>Inclusão de arquivos de código</li>
15    <li>PHP na Web</li>
16  </ol>
17
18  <p>
19    <b>Data atual:</b>
20    <?php echo date('d/m/Y')?>
21  </p>
22  <p>
23    <b>Hora atual:</b>
24    <?= ' ' . date('h:i:s')?>
25  </p>
26
27 </body>
28 </html>

```

<http://localhost:8080/pagina-web.php>



Copyright© 2023 Accenture All Rights Reserved

21

21

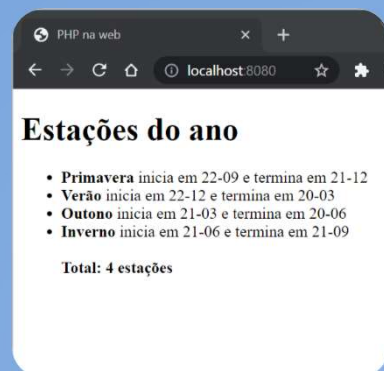
PHP na Web



```

1 <?php
2 $estacoes = [
3   "Primavera" => ["inicio" => "22-09", "fim" => "21-12"],
4   "Verão" => ["inicio" => "22-12", "fim" => "20-03"],
5   "Outono" => ["inicio" => "21-03", "fim" => "20-06"],
6   "Inverno" => ["inicio" => "21-06", "fim" => "21-09"]
7 ];
8 ?>
9
10 <!DOCTYPE html>
11 <html lang="pt-br">
12 <head>
13   <meta charset="UTF-8">
14   <meta http-equiv="X-UA-Compatible" content="IE=edge">
15   <meta name="viewport" content="width=device-width, initial-scale=1.0">
16   <title>PHP na web</title>
17 </head>
18 <body>
19   <h1>Estações do ano</h1>
20   <ul>
21     <?php foreach ($estacoes as $estacao => $info) {
22       echo "<li><b>{$estacao}</b> inicia em {$info['inicio']} e termina em {$info['fim']}</li>";
23     }
24     echo "<br><b>Total: ' . count($estacoes) . ' estações'";
25   <?>
26   </ul>
27 </body>
28 </html>

```



Copyright© 2023 Accenture All Rights Reserved

22

22

PHP na Web



- Em resumo, o que precisamos para rodar o PHP na Web é ter um servidor web, além do PHP, é claro.
- Neste treinamento vamos utilizar o próprio servidor embutido do PHP ("PHP Development Server"). Ele **não** deve ser usado para produção.
- Para iniciar o servidor web, basta digitar na linha de comando (dentro da pasta onde está o arquivo de script PHP):
`php -S localhost:8080`

Copyright© 2023 Accenture All Rights Reserved

23

23

Atividade



Jogo TicTacToe (Jogo da Velha)

- Alterar o jogo TicTacToe para que, durante uma partida, o jogador possa reiniciar o jogo
- Adicionar o histórico das jogadas



Copyright© 2023 Accenture All Rights Reserved

24

24