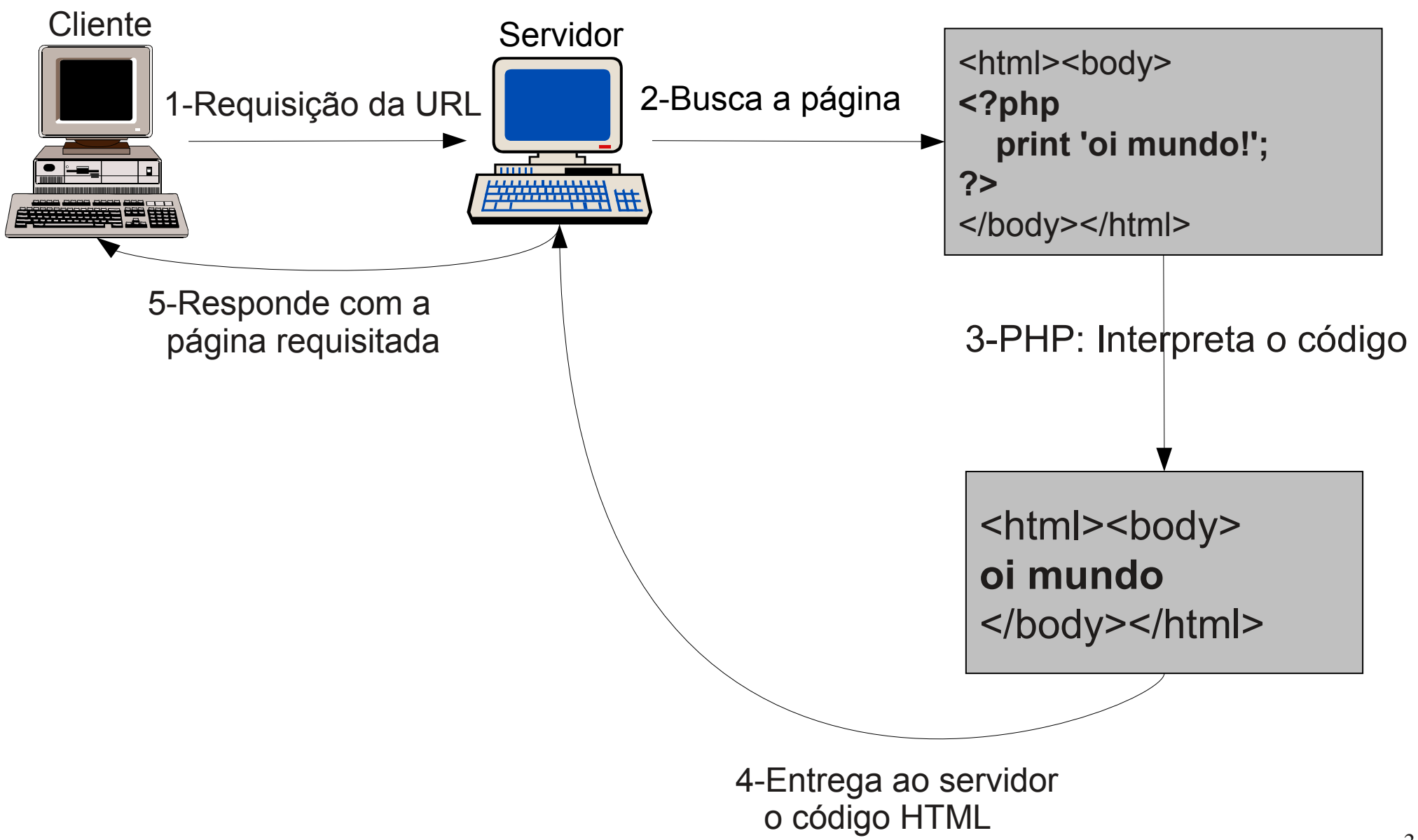




Introdução

- **PHP Hypertext Preprocessor**
 - Outros nomes: Personal Home Page, Professional Home Page.
- Linguagem de programação *server-side*;
- Fornece suporte a vários banco de dados;
- Trabalha na geração de páginas HTML;
- HTML gera uma página web com texto e imagens estáticos.
- Já o PHP trabalha com páginas dinâmicas.
 - O conteúdo da mesma página pode ser diferenciado para cada usuário que acessar o site.

Como funciona a comunicação?



Por que utilizar PHP?

- PHP envolve:
 - Simplicidade com script;
 - Independente de plataforma;
 - Foi criado inicialmente para aplicações Web;
 - Otimizado para melhorar o tempo de resposta das aplicações web;
 - É Software Livre!
- Seu site:
 - <http://www.php.net>

Declaração padrão

```
<?php
    // código em php (comentário igual da ling. 'C')
    echo 'Oi mundo!';
    /* Muito legal esse comentário também */
?>
```

- Para utilizar a tag “<?php” sem o “php”, ou seja, somente “<?”, deve-se ativar a opção:

```
short_open_tag = On
```

Dicas úteis para iniciantes....

1. Seja amigo do manual e da documentação do PHP (http://php.net/manual/pt_BR/index.php);
2. Dê atenção aos erros do PHP;
3. Utilize um IDE e um GUI (ou seja ninja... ;-]);
4. Utilize um Framework;
5. Aprenda algumas práticas:
 - DRY: Don't Repeat Yourself (Não Se Repita)
 - OAOO: Once And Only Once (Uma Vez e Somente Uma Vez)
 - KISS: Keep It Simple, Stupid (Mantenha Isso Simples, Estúpido)
 - YAGNI: You Aren't Gonna Need It (Você Não Irá Precisar Disso)
6. Organize o seu código (Identação e espaços);
7. Organize seus arquivos;
8. Sempre use “<?php” ... “?>”;
9. Use nomes consistentes e lógicos para variáveis, funções, métodos e classes;
10. Não se preocupe em perguntar, mas nunca tenha medo de tentar.

Tipos de dados em PHP

- Três tipos básicos de dados:
 - Integer;
 - Double;
 - String;
- Mais tipos:
 - Array;
 - Object;
- PHP é de tipagem dinâmica;
- As variáveis podem modificar até mesmo o tipo durante a execução do código.

Constantes em PHP

- ..valores que nunca mudam
- Constantes são definidas com a função *define()*
- Exemplo:
 - `define("DSW", "Desenvolvimento de Sistemas WEB")`
- Para saber se a constante existe, pode-se utilizar a função *defined()*.

Que retorna Verdadeiro ou Falso.

Variáveis em PHP

- As variáveis em PHP são declaradas adicionado o símbolo \$ (moeda) no início do identificador. Exemplo:

```
$nome = 'Jacson';
```

```
$sobrenome = 'Rodrigues';
```

```
echo "$nome $sobrenome";
```

- O tipo da variável é determinado assim que o valor é associado a mesma.
- Permite “Type Casting” para modificar o tipo de dados de forma explícita. Exemplo:

```
$foo = 10; // $foo é um inteiro
```

```
$bar = (boolean) $foo; // $bar é um booleano.
```

Tipos podem mudar ou ser mesclados

- Exemplo:

```
$foo = "0"; // $foo é string (ASCII 48)
```

```
$foo += 2; // $foo é agora um inteiro (2)
```

```
$foo = $foo + 1.3; // $foo é agora um float (3.3)
```

```
$foo = 5 + "10 pequenos porcos"; // $foo é inteiro (15)
```

```
$foo = 5 + "10 minúsculos porcos"; // $foo é inteiro (15)
```

Operadores em PHP

- Todos os operadores (aritméticos, de atribuição, de comparação e os lógicos) são similares aos operadores em C e C++.
- Porém em PHP a concatenação de strings é feita com o operador “.”. Exemplo:
 - `$nome = “Meu nome é”.$meunome;`

Estruturas em PHP

- Estrutura do IF:

```
if ( <condição> ) {  
    // código php  
} else {  
    // código php  
}
```

- Sintaxe Alternativa

```
if( <condição> ):  
    // código php  
else:  
    // código php  
endif;
```

Estruturas em PHP

- Repetição For

```
for( $i=0 ; $i < 10 ; $++i )  
{  
    echo ( "O valor é:" . $i ); //algo diferente ao utilizar o echo?  
}
```

- Syntaxe Alternativa

```
for ( $i=0 ; $i < 10 ; $++i ):  
    // código php  
endfor;
```

Estruturas em PHP

- Repetição ForEach:

```
$vetor = array(1,2,3,4,'jeiks');
```

```
foreach ($vetor as $item) {  
    echo "$item<br/>";  
}
```

```
foreach ($vetor as $indice => $item) {  
    echo "vetor[$indice] = $item<br/>";  
}
```

Estruturas em PHP

- Repetição While:

```
$i = 1;  
while ( $i <= 10 ) {  
    echo $i++;  
}
```

- Syntaxe Alternativa:

```
$i = 1;  
while ($i <= 10):  
    echo $i;  
    $i++;  
endwhile;
```

Estruturas em PHP

- Repetição Do-While:

```
$i = 0;
```

```
do {
```

```
    echo $i++;
```

```
} while ($i > 0);
```


Estruturas em PHP

- Case:

```
switch (expressão) {  
    case valor1:  
        // código em php  
        break;  
    case valor2:  
        // código em php  
        break;  
    default:  
        // código em php  
}
```

Arrays em PHP

- Exemplo:

```
$vetor = array(1,2,3,4,5);
```

```
$nomes = array('professor', 'aluno');
```

```
$arr = array( "foo" => "bar" , 12 => true );
```

```
echo $arr["foo"]; // bar
```

```
echo $arr[12];    // 1
```

```
$arr = array("somearray" =>
```

```
    array(6 => 5, 13 => 9, "a" => 42));
```

```
echo $arr["somearray"][6];    // 5
```

```
echo $arr["somearray"][13];   // 9
```

```
echo $arr["somearray"]["a"];  // 42
```

Operadores em PHP

- Operadores de atribuição:
 - = → Atribui
 - += → Atribui e soma
 - = → Atribui e decrementa
 - *= → Atribui e multiplica
 - /= → Atribui e divide
 - .= → Atribui e concatena
 - %= → Atribui com módulo

Operadores em PHP

- Operadores aritméticos:
 - + → Adição
 - → Subtração
 - * → Multiplicação
 - / → Divisão
 - % → Módulo (resto da divisão)
 - ++ → Incremento
 - → Decremento

Operadores em PHP

- Operadores relacionais:
 - == → igual a
 - != → diferente de
 - > → maior que
 - < → menor que
 - >= → maior ou igual
 - <= → menor ou igual

Operadores em PHP

- Operadores lógicos:
 - && → E (and)
 - || → OU (or)
 - ! → Negação (not)

Funções em PHP

- Exemplos:

```
function vaca() {  
    echo 'muuuu!!!';  
}
```

```
function quadrado($numero) {  
    echo "O quadrado de $numero é: " . ($numero * $numero);  
}
```

Funções em PHP

- Exemplo:

```
$total = 0;
```

```
function kmParaMilhas($quilometros) {  
    global $total;  
    $total += $quilometros;  
    return $quilometros * 0.6  
}
```

```
echo 'percorreu ' . kmParaMilhas(100) . 'milhas </br>';  
echo 'percorreu ' . kmParaMilhas(200) . 'milhas </br>';  
echo 'percorreu ' . $total . 'quilometros';
```


Classes em PHP

- Exemplo:

```
class Computador {  
    var $cpu;  
    function ligar() {  
        echo "Ligando computador a {$this->cpu}";  
    }  
}
```

```
$obj = new Computador();
```

```
$obj->cpu = '500Mhz';
```

```
$obj->ligar();
```

```
Computador::ligar();
```