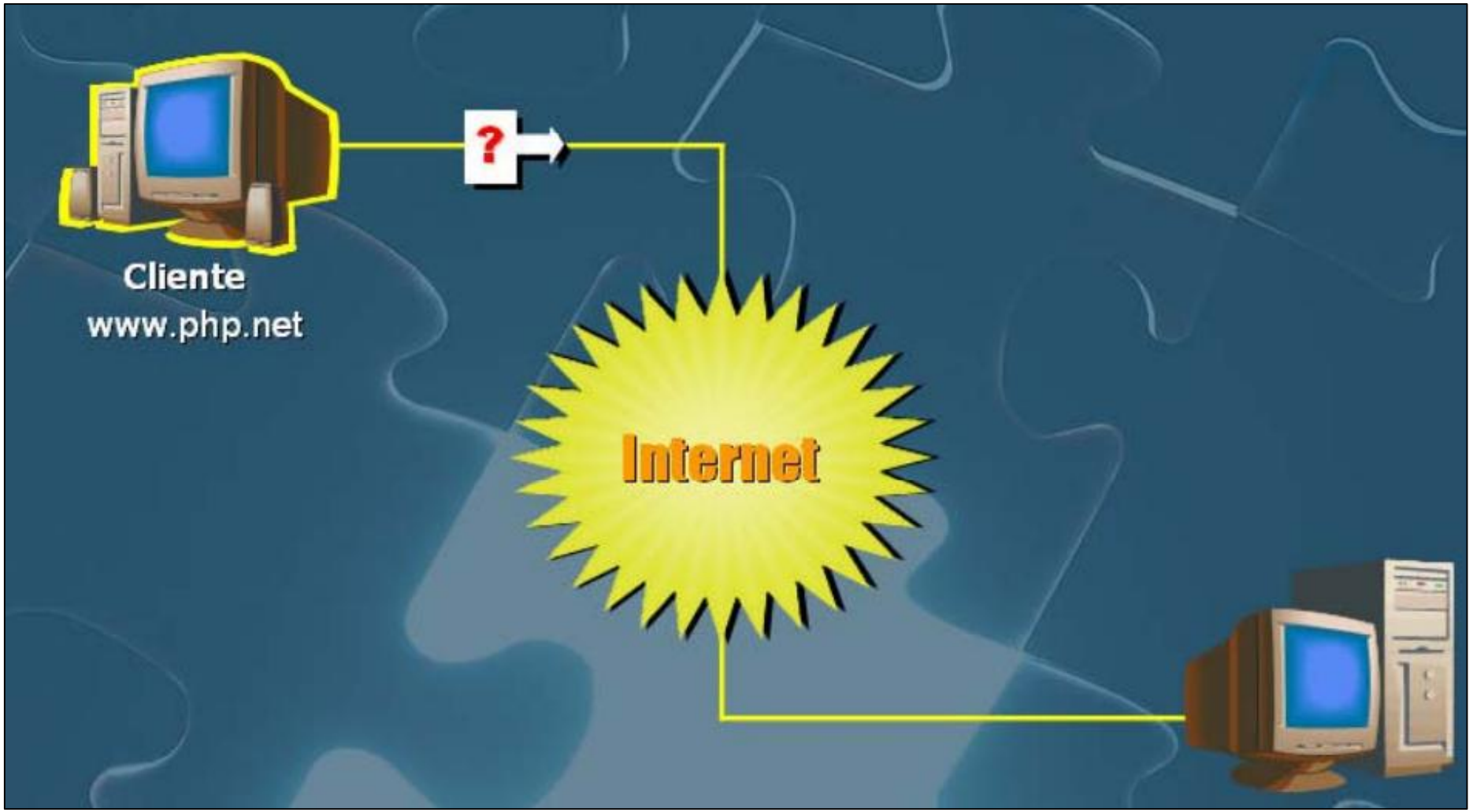
SW-I
SISTEMAS WEB I

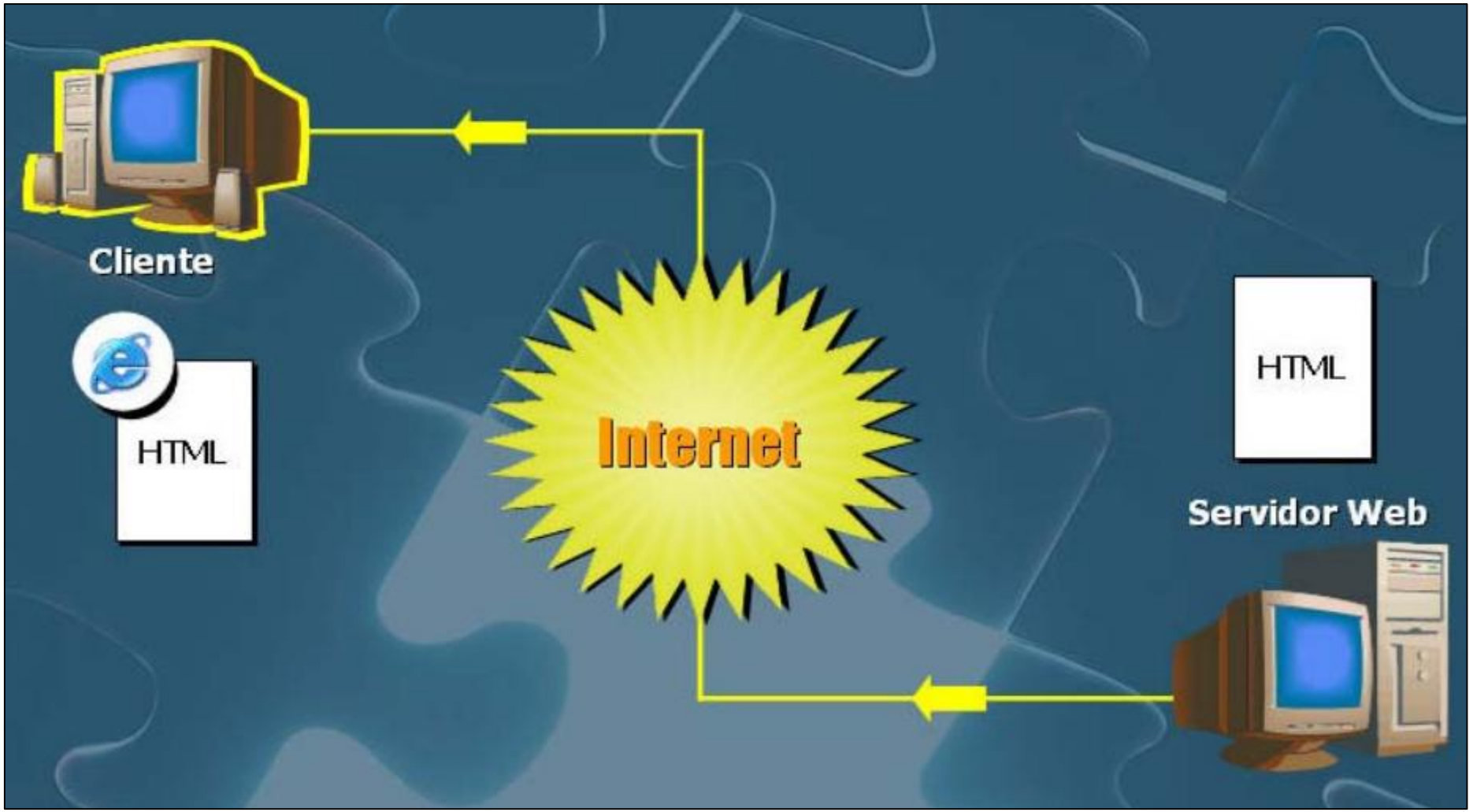
Prof. Anderson Vanin

Requisição HTML



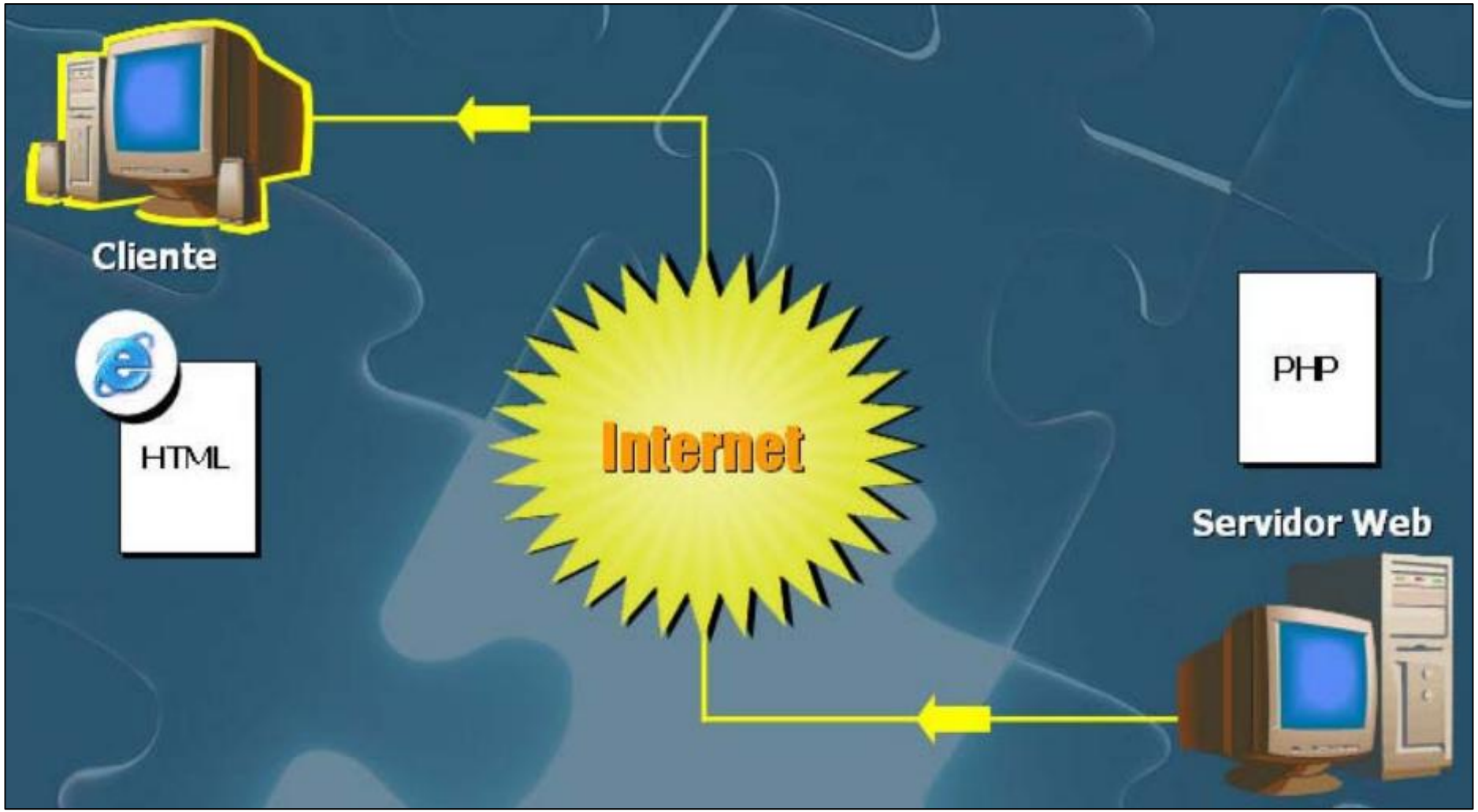
Requisição HTML

Requisição HTML



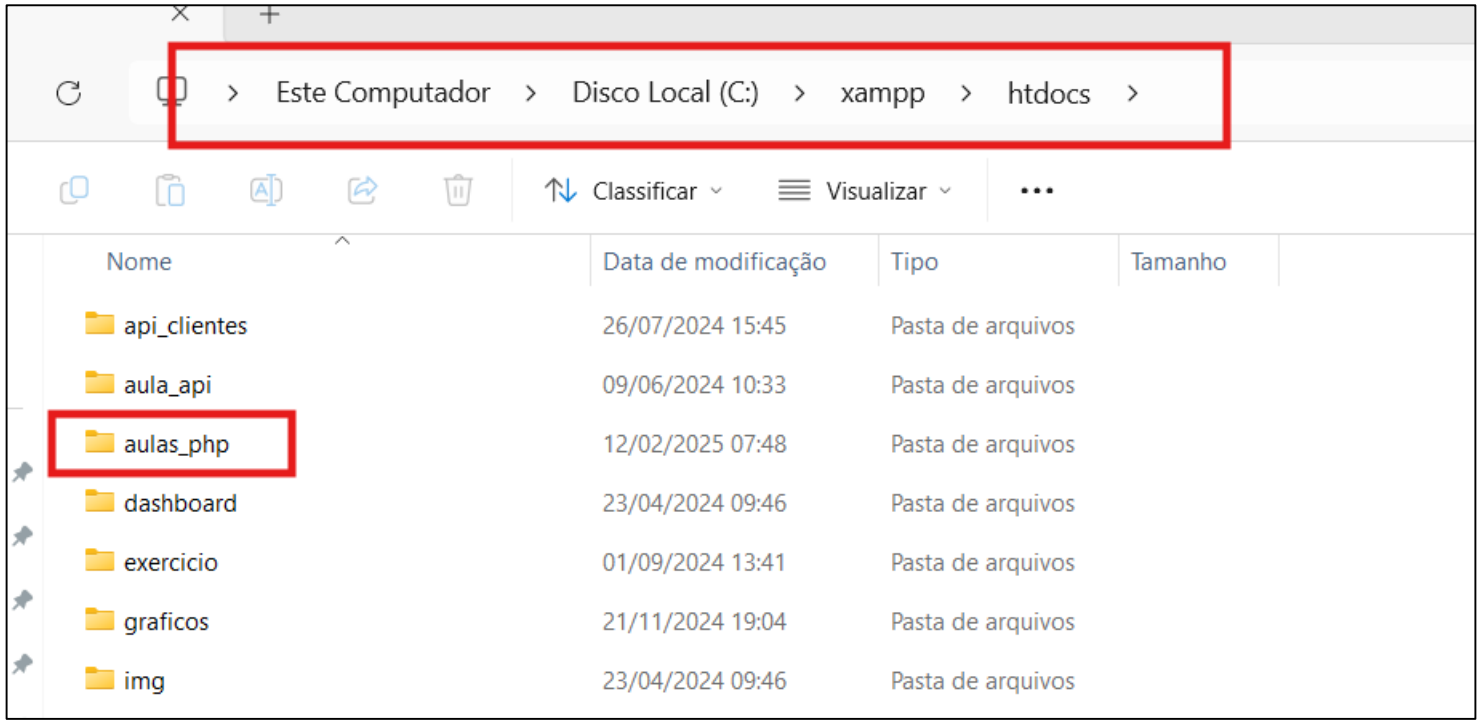
Requisição PHP

Requisição PHP

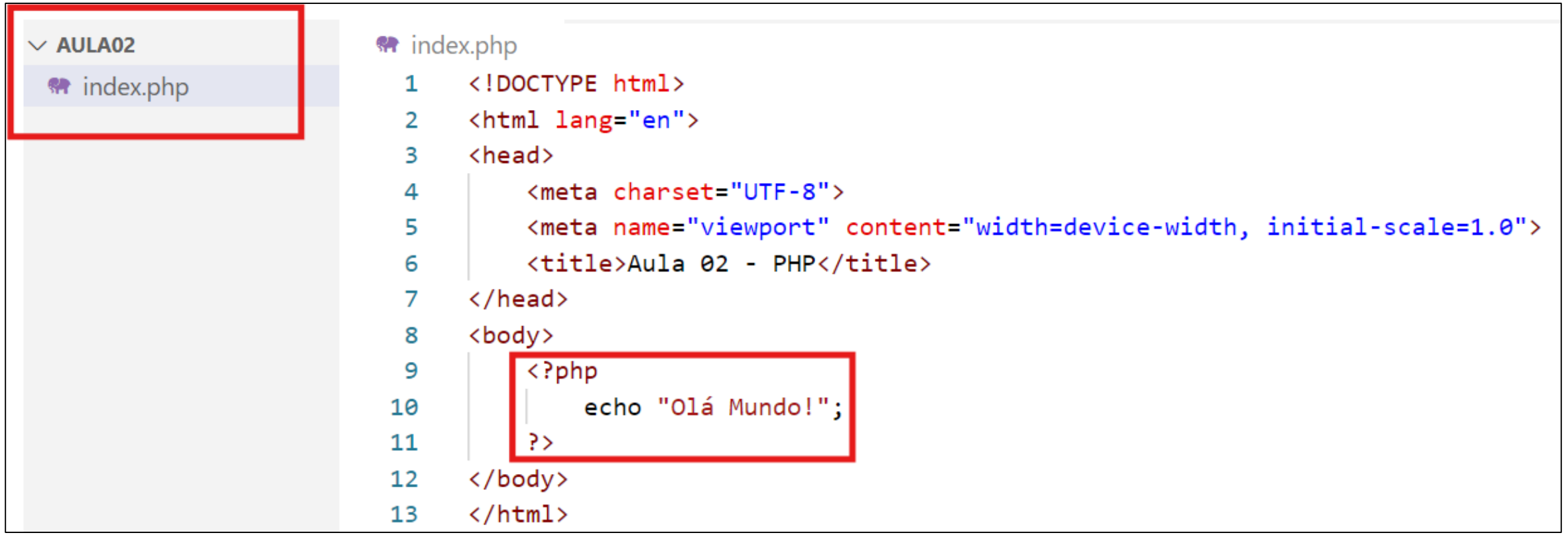


Olá Mundo

Para não fugir a tradição faremos o programa Para não fugir a tradição, faremos o programa **Olá Mundo!**



Olá Mundo


AULA02
index.php
index.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Aula 02 - PHP</title>
7 </head>
8 <body>
9 <?php
10 echo "Olá Mundo!";
11 ?>
12 </body>
13 </html>

Olá Mundo

Variáveis e Constantes

Os identificadores de variáveis em PHP devem seguir algumas regras básicas:

- Devem começar pelo símbolo de **$**
- O segundo caractere deve ser uma letra ou o caractere *underline* "_"
- Os demais caracteres podem ser letras, números ou *underline*;
- Não são aceitos símbolos como **!**, **@**, **&**, **)** e outros
- Não utilize caracteres acentuados, nem mesmo **ç**

Obs1: É muito comum o programador iniciante esquecer de colocar o **$** no início do identificador de variável. Com o tempo você se acostuma!

Obs2: O PHP é ***case sensitive***, por isso, **$nome**, **$Nome**, **$NOME** e **$NoMe** são variáveis totalmente distintas para a linguagem!

Testes

1) O identificador **$_nome** é:

a) Válido

b) Inválido

2) O identificador **$12anos** é:

a) Válido

b) Inválido

3) O identificador **$Sal_Líq** é:

a) Válido

b) Inválido

Testes

4) O identificador **$Aumento%** é:

a) Válido

b) Inválido

5) O identificador **$Casal20** é:

a) Válido

b) Inválido

6) O identificador **Cod_Produto** é:

a) Válido

b) Inválido

Variáveis e Atribuições

Em programas desenvolvidos em PHP, não precisamos declarar as variáveis, pois elas serão automaticamente criadas conforme elas forem utilizadas.

Para criarmos uma variável para armazenar um nome, basta que façamos uma atribuição inicial.

Por exemplo: `$nome = “Bruno Marton”;`

De maneira análoga, consideramos as atribuições:

```
$nota = 3.5;
$idade = 15;
```

Variáveis e Atribuições

A definição automática de tipos do PHP pode parecer um milagre mas ela pode confundir o programador desavisado. Por exemplo, veja as linhas a seguir:

```
$a = 4;
$b = “101 Dálmatas”;
$s = $a + $b;
echo $s;
```

Pergunto: Qual seria o resultado exibido na tela?

A resposta certa é **105**! Mas como?

O PHP interpretaria **$s** como a soma numérica entre **$a** e **$b**. Como **$b** é uma ***string*** que começa por um número, os caracteres serão ignorados e o valor somará com o valor de **$a**. Assim, o PHP calcularia **4 + 101 = 105**. Bem intuitivo, não acha?

Variáveis e Atribuições

O principal deslize foi considerar que o operador "**+**" faria a concatenação de ***strings***, mas essa tarefa é realizada pelo operador "**.**" (ponto). Assim, se substituirmos por:

```
$s = $a . $b;
```

Teremos o resultado **4101 Dálmatas**, concatenando as duas variáveis. As conversões necessárias seriam feitas automaticamente pelo PHP.

Variáveis – Tipos Primitivos

O PHP suporta alguns tipos primitivos de dados:

- Inteiro simples (***integer ou int***)
- Inteiro long (***long***)
- Real precisão simples (***float ou real***)
- Real precisão dupla (***double***)
- String (***string***)
- Matriz (***array***)
- Objeto (***object***)

Obs.: O tipo lógico (***boolean***), apesar de presente nas versões mais recentes é tratado como valores inteiros (***1 True*** ou ***vazio False***)

Conversões Explícitas

Podemos também forçar a conversão de um valor para um determinado tipo primitivo, indicando-o antes da expressão, entre parênteses.

```
$x = 3.5;
$y = 4.3;
$z = “9.9”;
$n1 = (int) $x + $y;
$n2 = (int) ($x + $y);
$n3 = (real) ($y + $z);
echo “Resultados: $n1, $n2, $n3”;
```

A mensagem exibida na tela seria:

```
Resultados: 7.3, 7, 14.2
```

Constantes

As constantes PHP não começam por **$**. O restante é totalmente baseado nas regras para variáveis. Para declarar uma constante, usamos a instrução **define**:

```
define (“pi", 3.1415926536);
```

Para utilizar as constantes em nosso código:

```
$circ = (real) 2 * pi * $raio;
```

Importante: Na hora de exibir o conteúdo de uma constante, devemos lembrar que o PHP não terá como diferenciar o identificador da constante do restante do código, portanto devemos usar o operador de concatenação (**.**)

```
echo “O valor de pi é ”.pi;
```

A instrução acima mostrará: **O valor de pi é 3.1415926536**

Exibição de Variáveis

Considere o seguinte código:

```
$n = “penta”;

echo “O Brasil é $ncampeão!”;
```

O que será exibido?

Acertou quem respondeu respondeu ERRO!

Na verdade, o PHP vai procurar por uma variável chamada **`$ncampeão`**.

Exibição de Variáveis

Este problema pode ser resolvido de duas maneiras:

```
echo “O Brasil é “.$n.” campeão!”;
```

ou ainda

```
echo “O Brasil é ${n}campeão!” {n}campeão!”;
```

Testes

7) Considerando as instruções abaixo:

```
$a = “Casal 20”;
$b = 10;
$c = $a + $b;
echo $c;
```

Qual será o resultado exibido na tela?

a) Casal 2010
b) 30
c) 10
d) ERRO!

8) Com a atribuição **`$pre = “hiper”;`** Qual das linhas abaixo mostrará a mensagem “hipertexto”?

a) `echo “$pre.texto”;`
b) `echo $pre”texto”;`
c) `echo “$pretexto”;`
d) `echo “${pre}texto”;`

Testes

9) Considerando as instruções abaixo:

```
$x = “P”;
$y = “H”;
$z = “$x$y$x”;
echo $z;
```

Qual será o resultado exibido na tela?

a) XYX
b) PHP
c) NADA
d) ERRO!

10) Qual das opções abaixo faria a declaração de uma constante?

a) `define inc=2;`
b) `define(“inc”,2);`
c) `define(“$inc”,2);`
d) `$inc = 2;`

OPERADORES

Operadores Aritméticos

Considerando:

```
$a = 5; $b = 2;
```

Op	Função	Exemplo	Resultado
+	Adição	$a+$b	7
-	Subtração	$a-$b	3
*	Multiplicação	$a*$b	10
/	Divisão	$a/$b	2.5
%	Resto da divisão	$a%$b	1

Operadores de Incremento/Decremento

```
$cont = $cont + 1;
$cont++;
$x = $x - 1;
$x--;
```

Operadores de Incremento/Decremento

```
$a = 3;
$b = 2;
$s = (++$a) + $b;
echo "$a, $b, $s";

Resultado: 4, 2, 6
```

```
$a = 3;
$b = 2;
$s = ($a++) + $b;
echo "$a, $b, $s";

Resultado: 4, 2, 5
```

Testes

11) Considerando as instruções abaixo:

```
$x = 5;
$y = 4;
$n1 = (--$x)+$y;
echo $n1;
```

Qual será o resultado exibido na tela?

a) 8
b) 9
c) 4
d) 5

Operadores Binários

Op	Função
~	NÃO binário
|	OU binário
&	E binário
^	XOR binário
>>	Deslocamento à direita
<<	Deslocamento à esquerda

Operadores Binários

Considerando:

```
$x = 14; //1110

$y = 12; //1100
```

Expressão	Base 2	Base 10
~$x	0001	1
$x&$y	1100	12
$x\|$y	1110	14
$x^$y	0010	2
$x>>2	0011	3
$x<<2	111000	56

Operadores de Atribuição

Op	Exemplo	Corresponde a
+=	$a += $b;	$a = $a + $b;
-=	$a -= $b;	$a = $a - $b;
*=	$a *= $b;	$a = $a * $b;
/=	$a /= $b;	$a = $a / $b;
%=	$a %= $b;	$a = $a % $b;
.=	$a .= $b;	$a = $a.$b;
&=	$a &= $b;	$a = $a & $b;
|=	$a |= $b;	$a = $a | $b;
^=	$a ^= $b;	$a = $a ^ $b;
>>=	$a >>= $b;	$a = $a>>$b;
<<=	$a <<= $b;	$a = $a<<$b;

Testes

12) Considerando as instruções abaixo:

```
$x = 5;
$y = 4;
$x += $y;
echo $x;
echo $y;
```

Quais serão os valores exibidos?

a) 5 e 4

b) 4 e 5

c) 9 e 9

d) 9 e 4

Operadores Relacionais

Op	Função
==	Igual a
===	Idêntico
>=	Maior ou igual a
<=	Menor ou igual a
>	Maior que
<	Menor que
!=	Diferente de
<>	Diferente de

Igual ou Idêntico?

Exemplo:

```
$a = 3;
$b = “3”;
if ($a == $b){
      echo “A é igual a B A é igual a B ;”
}else{
      echo “A não é igual a B”;
}
if ($a === $b){
      echo “A é idêntico a B A é idêntico a B ;”
}else{
      echo “A não é idêntico a B”;
}
```

Testes

13) Considerando as instruções abaixo:

```
$x = “100 vergonha”;
$y = 50;
$z = 2 * $y;
if ($x == $z){
        echo “São iguais”;
}else{
        echo “São diferentes”;
}
```

O que será exibido pelo trecho do código apresentado acima?

a) São iguais
b) São diferentes
c) Não será exibido nada
d) O PHP retornará ERRO

Operadores Lógicos

Op	Função
!	Negação Lógica
&&	E
||	OU
AND	E
OR	OU
XOR	OU exclusivo

Operadores Lógicos

! (NOT)	
x	!x
V	F
F	V

|| (OR)			
x	y	z	x || y || z
V	V	V	V
V	V	F	V
V	F	V	V
V	F	F	V
F	V	V	V
F	V	F	V
F	F	V	V
F	F	F	F

Operadores Lógicos

&& (AND)			
x	y	z	x && y && z
V	V	V	V
V	V	F	F
V	F	V	F
V	F	F	F
F	V	V	F
F	V	F	F
F	F	V	F
F	F	F	F

Operadores Lógicos

XOR			
x	y	z	x XOR y XOR z
V	V	V	F
V	V	F	V
V	F	V	V
V	F	F	V
F	V	V	V
F	V	F	V
F	F	V	V
F	F	F	F

Operadores Lógicos

Todos Operadores Lógicos					
x	y	z	||	&&	XOR
V	V	V	V	F	F
V	V	F	V	F	V
V	F	V	V	F	V
V	F	F	V	F	V
F	V	V	V	F	V
F	V	F	V	F	V
F	F	V	V	F	V
F	F	F	F	V	F

Testes

14) Considerando o trecho abaixo:

```
$a = 4;
$b = 8;
$r = (($a==$b) || ($a+$b==12));
```

O valor lógico de **r** é:

a) Verdadeiro

b) Falso

Testes

15) Considerando o trecho abaixo:

```
$a = 4;
$b = 8;
$r = (($a==$b) && ($a+$b==12));
```

O valor lógico de **r** é:

a) Verdadeiro

b) Falso

Testes

16) Considerando o trecho abaixo:

```
$a = 5;
$b = 10;
$r = (($a<=$b) XOR (2*$a==$b));
```

O valor lógico de **r** é:

a) Verdadeiro

b) Falso

Operador Ternário

Utilizaremos o **? :** para criar condições simples que realizam atribuições.

EXEMPLO:

```
if ($m > 3){
      $a = $a + 1;
}else{
      $a = $a – 1;
}


$a = ($m>3) ? ($a+1) : ($a-1);
```

Precedência dos Operadores

Ordem	Operadores	Descrição
1º	- ! ~ ++ --	negativo, não-lógico, inversão, incr. e decr.
2º	* / %	multiplicação, divisão, resto
3º	+ - .	adição, subtração, concatenação
4º	<< >>	deslocamentos binários
5º	> < >= <=	operadores relacionais
6º	== != <>	igual e diferente
7º	&	AND binário
8º	^	XOR binário
9º	|	OR binário
10º	&&	AND lógico
11º	||	OR lógico
12º	? :	operador ternário
13º	+= -= *= ...	operadores de atribuição
14º	AND	AND lógico (menor prioridade)
15º	XOR	XOR lógico
16º	OR	OR lógico (menor prioridade)

Exercícios

1. Declare três variáveis que conterão respectivamente: nome de uma pessoa, idade e um valor (pode ser de salário ou de mesada), serão valores atribuídos, não digitados. Verifique o que foi impresso na tela abaixo e reproduza isto em um programa utilizando os dados das variáveis criadas. Os itens que estão sublinhados são valores de variáveis.

```
Olá! Qual o seu nome?
Tyrone
Oi Tyrone! Quantos anos você tem?
18
Então você tem 18!!!
Quanto você ganha Tyrone?
100
100! Eu espero que seja por hora e não por ano!!!!! rsrsrs....
```

Exercícios

2. Faça um programa PHP que defina uma constante PI com valor 3.141592. Crie uma variável, atribua um valor que represente o raio de uma esfera. Calcule e apresente sua área e seu volume, dados pelas fórmulas:

$$AREA = 4\pi R^2$$

$$VOLUME = \frac{4}{3}\pi R^3$$

Exercícios

3. Faça um programa PHP que crie duas variáveis e atribua dois valores inteiros. Apresente todas as operações: adição, subtração, divisão, multiplicação e resto da divisão utilizando os operadores de atribuição.