

Desenvolvimento Web II

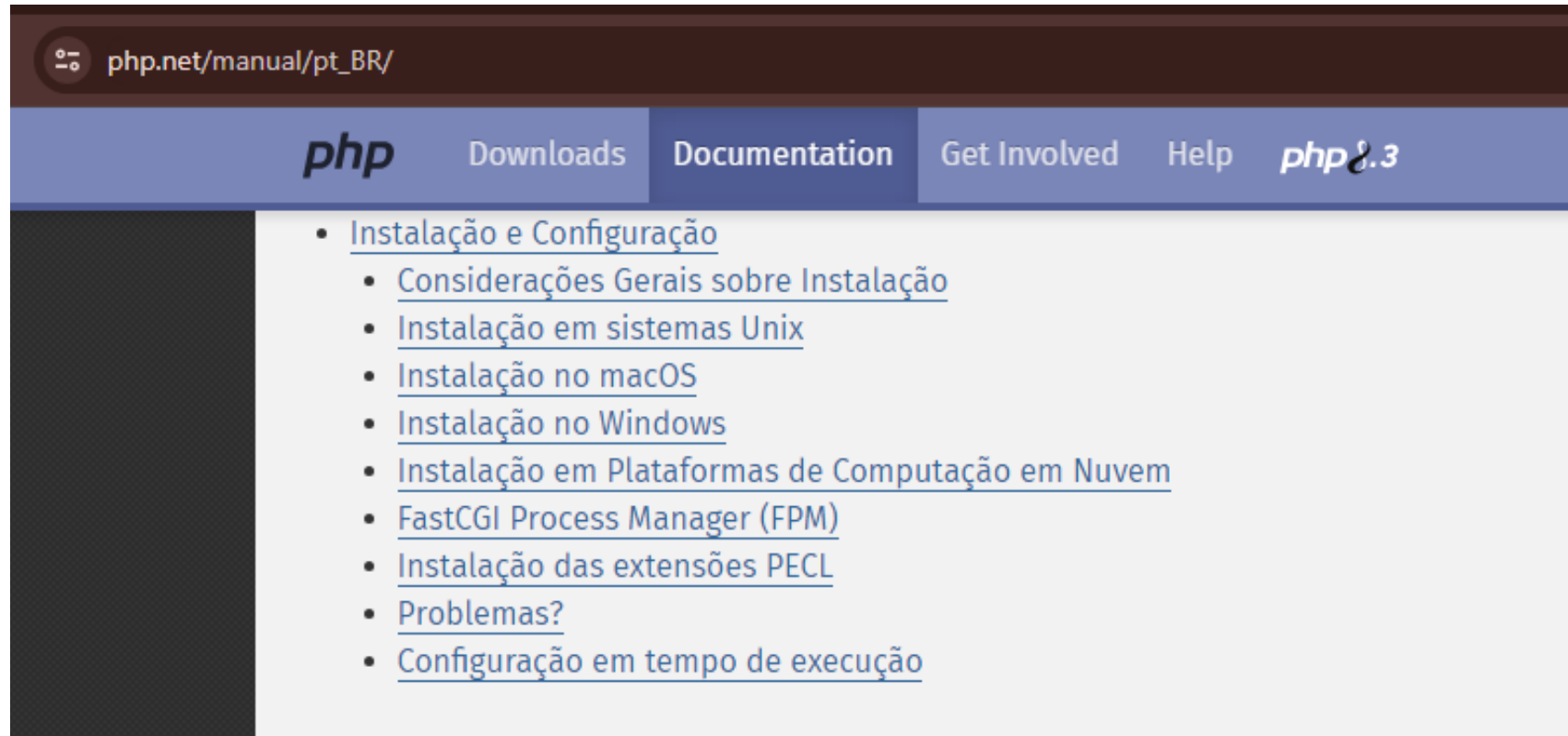
PHP Básico

Introdução ao PHP

- PHP: abreviação para Hypertext Preprocessor (Pré-Processador de Hipertexto)
- 1ª versão criada no outono de 1994 por Rasmus Lerdorf. Atualmente estamos na versão 8.2.15.
- É utilizado por 79,2% dos sites da internet
- PHP é simples e fácil de aprender, pois é similar a outras existentes no mercado.
- Possui código aberto, ou seja, é gratuita para qualquer usuário.

Introdução ao PHP

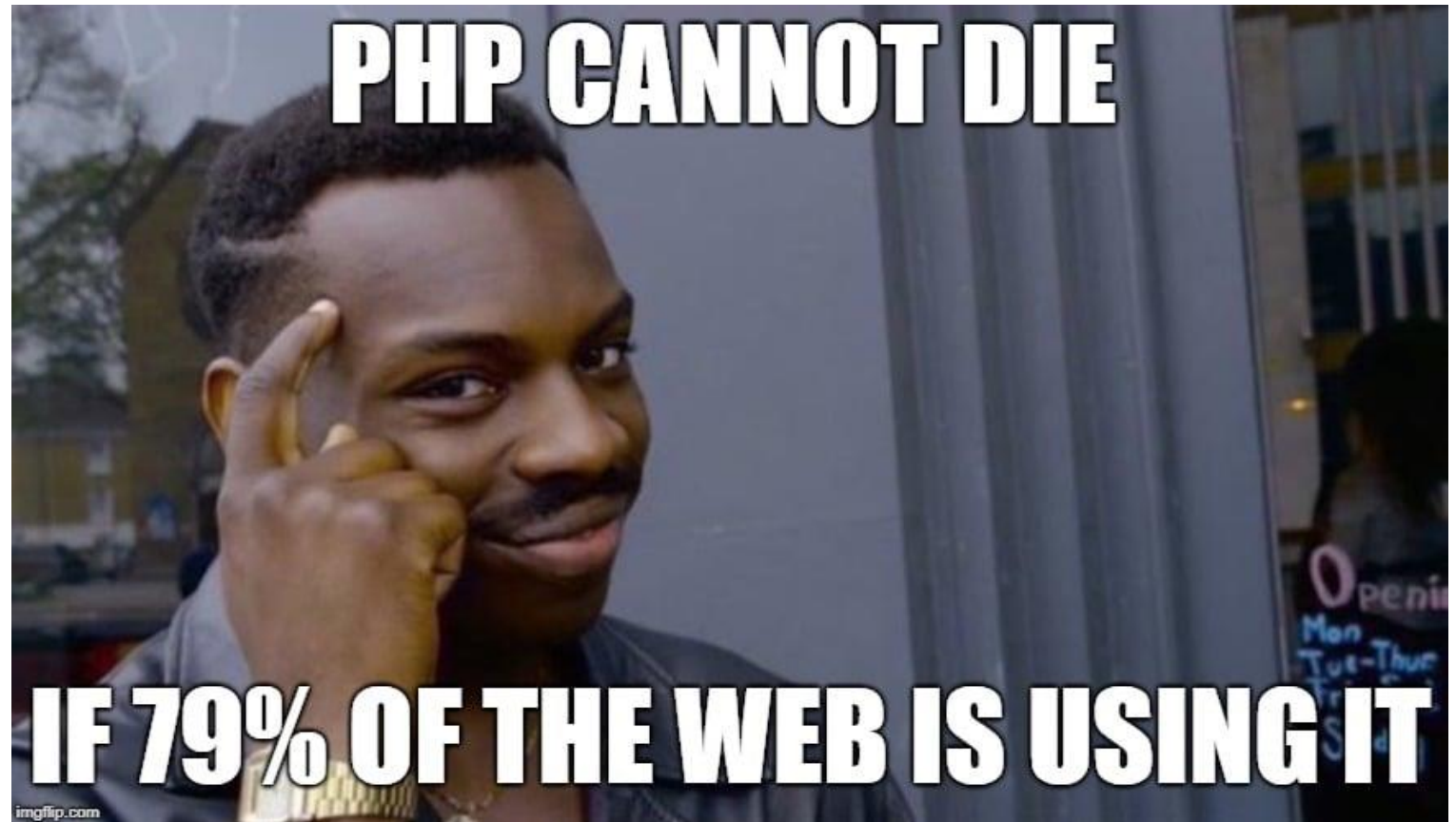
- Multiplataforma, ou seja, pode ser instalado em diversos sistemas operacionais.



Introdução ao PHP

- Alto desempenho: consegue executar muitas funções e consumir muitos dados com a utilização de poucos recursos do servidor
- A linguagem PHP é compatível com os principais bancos de dados, como MySQL, SQLite, Firebird, Interbase e Oracle.
- Possui diversos frameworks e bibliotecas facilitam muito a vida de um desenvolvedor. Dentre os frameworks mais conhecidos encontra-se Laravel, o CodeIgniter, o Zend e o Symfony.
- É utilizado por:
 - Facebook
 - Wikipedia
 - Wordpress

Não está
ficando
obsoleto?



Projetos de migração de solução.

Arquivo php

- Todo o trecho de programação PHP deve estar entre as tags `<?php` e `?>`, para que o servidor Web saiba que esse trecho deve ser processado.

Elemento	Descrição
<code><?php</code>	Informa que aqui começa um programa PHP.
<code>//</code>	Representam uma linha de comentário. Tudo que vem após estas barras na mesma linha é ignorado pelo PHP. Os comentários são muito úteis para uma boa documentação do seu programa. As duas barras servem para transformar uma única linha em comentário, mas você pode usar o <code>/*</code> para iniciar uma sequência de comentários e depois finalizar os comentários com o <code>*/</code> .
<code>echo</code>	É um dos comandos mais utilizados em PHP. Serve para escrever alguma coisa na tela.
<code>?></code>	Informa que aqui termina o programa PHP.

Arquivo php

Arquivos PHP pode conter

- Texto
 - Tags HTML
 - Scripts PHP
-
- Entretanto, os comandos HTML devem aparecer fora das tags `<?php` e `?>`, pois essas limitam um trecho de programa PHP.
 - Mas e se você tentar ver o Código fonte da página acessada?

Exemplo

```
<html>
  <body>
    <?php
      $data_de_hoje = date ("d/M/Y",time()) ; ?>
      <p align="center"> Hoje é dia <?php echo $data_de_hoje; ?>
    </p>
  </body>
</html>
```


Arquivos php

- Toda essa mistura entre o HTML e o PHP é muito útil, pois:
 - Utilizamos o PHP para gerar os dados dinamicamente
 - Enquanto o HTML é usado para formatar e exibir esses dados nas páginas mostradas no browser.

Manipulando os dados em PHP

Diferente de outras linguagens, no PHP **não é necessário fazer declaração de variáveis**. Basta atribuir diretamente um valor a ela, e a partir desse momento já está criada.

As variáveis no PHP são **representadas por um cifrão (\$) seguido pelo nome da variável. Os nomes de variável são case-sensitive (ou seja, há distinção entre maiúsculas e minúsculas)**.

Um **nome de variável válido inicia-se com uma letra ou sublinhado**, seguido de qualquer número de letras, números ou sublinhados.

Manipulando os dados em PHP

```
<?php
$var = 'Bob';
$Var = 'Joe';
echo "$var, $Var";      // exibe "Bob, Joe"

$4site = 'not yet';     // inválido; começa com um número
$_4site = 'not yet';    // válido; começa com um sublinhado
$täyte = 'mansikka';    // válido; 'ä' é um caracter ASCII (extendido) 228
?>
```

Manipulando os dados em PHP

- Recomendação: Não é necessário inicializar variáveis no PHP no início do programa, contudo é uma ótima prática.
- `isset($teste)`: função utilizada para verificar se a variável já foi inicializada.

12 variáveis predefinidas em php 8

1. `$GLOBALS`

2. `$_SERVER`

3. `$_REQUEST`

4. `$_FILES`

5. `$_ENV`

6. `$_SESSION`

7. `$_COOKIE`

8. `$_GET`

9. `$_POST`

10. `$http_response_header`

11. `$argc`

12. `$argv`

Principais tipos de dados suportados

- tipo bool
- tipo string
- tipo array
- tipo object
- tipo void

Manipulando os dados em PHP

- Dados numéricos suportados:

Dados	Descrição
5	Valor inteiro na base decimal.
4.012	Valor real (ou ponto flutuante) com três casas decimais.
.14	Valor real com duas casas decimais. É o mesmo que 0.14. Quando o número começa com um ponto o 0 é considerado como algarismo inicial.
033	Valor inteiro na base octal. Todo valor iniciado com 0 é tratado como base 8 (na base 8 são utilizados apenas os algarismos 0, 1, 2, 3, 4, 5, 6 e 7).
0xBC	Valor inteiro na base hexadecimal. Todo valor iniciado com 0x é tratado como base 16 (na base 16 são utilizados os 10 algarismos do sistema decimal e mais as letras A, B, C, D, E e F, que representam os valores de 10 a 15).
43000000	É um número real grande, que pode ser expresso utilizando a notação científica: 4.3E+7.

Manipulando os dados em PHP

- Dados alfanuméricos (textos):
 - Também conhecidos como Strings.
 - São sequências de caracteres, que podem ser delimitadas por aspas simples ('), aspas duplas (") ou aspas invertidas (`), dependendo da utilização desejada.

- Uso de aspas simples:
 - não permite interpolação de variáveis
 - O caractere de aspas simples no meio do texto deve ser acompanhado pelo caractere de controle/escape (\). Porém permite o caractere “ sem a necessidade de sequencia de escape.

```
<?php
echo 'isto é uma string comum';
// Imprime: isto é uma string comum

echo 'Arnold disse uma vez: "I\'ll be back"';
// Imprime: Arnold disse uma vez: "I'll be back"

echo 'Você tem certeza em apagar C:\\*.*?';
// Imprime: Você tem certeza em apagar C:\\*.*?

echo 'Você tem certeza em apagar C:\\*.*?';
// Imprime: Você tem certeza em apagar C:\\*.*?

echo 'Isto não será substituído: \\n uma nova linha';
// Imprime: Isto não será substituído: \\n uma nova linha

echo 'Variáveis $também não $expandem';
// Imprime: Variáveis $também não $expandem
```


Manipulando os dados em PHP

- Aspas duplas:
- Permite interpolação de variáveis
- Permite o caractere de aspas simples no meio do texto sem caractere de controle (\).

```
<?php
    $palavra= "teste";
    $frase= "Isto é uma
    $palavra";
    echo $frase;
?>
```

Sequências de escape	
Sequências	Significado
\n	Nova linha (LF ou 0x0A (10) em ASCII)
\r	Retorno de carro (CR ou 0x0D (13) em ASCII)
\t	Tabulação horizontal (HT ou 0x09 (9) em ASCII)
\v	Tabulação vertical (VT ou 0x0B (11) em ASCII)
\e	Escape (ESC or 0x1B (27) em ASCII)
\f	Form feed (FF ou 0x0C (12) em ASCII)
\\	contrabarra ou barra invertida
\\$	Sinal de cifrão
\"	aspas duplas

Manipulando os dados em PHP

- O uso de aspas ao contrário (``) executar comandos do sistema operacional por meio de um programa PHP.
- Exemplo:

```
<html>  
<body>  
<?php  
echo `dir`;  
?>
```

Manipulando os dados em PHP

- Constantes: são definidas através do comando define. Exemplo:

```
<html>
<body>
<?php
define ("turma", "Desenvolvimento Web 2");
echo "O nome da turma é";
echo "<br>";
echo turma;
?>
</body>
</html>
```

Manipulando os dados em PHP

- Principais constantes predefinidas:

Constante	Descrição
TRUE	Valor verdadeiro (utilizado para comparação).
FALSE	Valor falso.
_ _FILE_ _	Contém o nome do script que está sendo executado.
_ _LINE_ _	Contém o número da linha do script que está sendo executado.
PHP_VERSION	Contém a versão corrente do PHP.
PHP_OS	Nome do sistema operacional no qual o PHP está rodando.
E_ERROR	Exibe um erro ocorrido em um script. A execução é interrompida.
E_WARNING	Exibe uma mensagem de aviso do PHP. A execução não para.
E_PARSE	Exibe um erro de sintaxe. A execução é interrompida.
E_NOTICE	Mostra que ocorreu algo, mas não necessariamente um erro. A execução não para.

Manipulando os dados em PHP

- Conversão de variáveis

```
<?php
$x = 50;
$y = 2.55;
$soma = (int) $y + $x;
echo $soma;
?>
```

Conversor	Descrição
(int), (integer)	Converte para inteiro.
(real), (float), (double)	Converte para ponto flutuante.
(string)	Converte em string.
(array)	Converte em array (vetor).
(object)	Converte em objeto.

Operadores - Aritméticos

Exemplo	Nome	Resultado
$+\$a$	Identidade	Conversão de $\$a$ para int ou float conforme apropriado.
$-\$a$	Negação	Oposto de $\$a$.
$\$a + \b	Adição	Soma de $\$a$ e $\$b$.
$\$a - \b	Subtração	Diferença entre $\$a$ e $\$b$.
$\$a * \b	Multiplicação	Produto de $\$a$ e $\$b$.
$\$a / \b	Divisão	Quociente de $\$a$ e $\$b$.
$\$a \% \b	Resto da divisão	Resto de $\$a$ dividido por $\$b$.
$\$a ** \b	Exponencial	Resultado de $\$a$ elevado a $\$b$.

Operadores – Incremento/Decremento

Exemplo	Nome	Efeito
<code>++\$a</code>	Pré-incremento	Incrementa <i>\$a</i> em um, e então retorna <i>\$a</i> .
<code>\$a++</code>	Pós-incremento	Retorna <i>\$a</i> , e então incrementa <i>\$a</i> em um.
<code>--\$a</code>	Pré-decremento	Diminuiu <i>\$a</i> em um, e então retorna <i>\$a</i> .
<code>\$a--</code>	Pós-decremento	Retorna <i>\$a</i> , e então diminui <i>\$a</i> em um.

Exemplos

```
$a = 8
```

```
$b = 1;
```

```
$r = (++$a) + $b;
```

```
echo $r;
```

```
$r = $r + $b++;
```

```
echo $r;
```


Exemplos

```
$a = 6;
```

```
$b = 2;
```

```
$r = (--$a) + $b;
```

```
echo $r;
```

Operadores – Atribuição

Exemplo	Equivalente	Operação
\$a += \$b	\$a = \$a + \$b	Adição
\$a -= \$b	\$a = \$a - \$b	Subtração
\$a *= \$b	\$a = \$a * \$b	Multiplicação
\$a /= \$b	\$a = \$a / \$b	Divisão
\$a %= \$b	\$a = \$a % \$b	Módulo
\$a **= \$b	\$a = \$a ** \$b	Exponentiation

Exemplo

```
$a = 5;
```

```
$b = 4;
```

```
$a+=$b;
```

```
echo $a;
```

```
echo $b;
```

Operadores – Atribuição

Exemplo	Equivalente	Operação
\$a &= \$b	\$a = \$a & \$b	Bitwise E
\$a = \$b	\$a = \$a \$b	Bitwise Ou
\$a ^= \$b	\$a = \$a ^ \$b	Bitwise Xor
\$a <<= \$b	\$a = \$a << \$b	Desloca para esquerda
\$a >>= \$b	\$a = \$a >> \$b	Desloca para direita

Se dados binários não são suportados, faz sentido termos estes operadores?

Exemplo

\$ip1 = 192;	\$m1 = 255;
\$ip2 = 168;	\$m2 = 255;
\$ip3 = 15;	\$m3 = 255;
\$ip4 = 23;	\$m4 = 0;

Qual é o endereço de rede do ip informado?

Operadores – Atribuição

Exemplo	Equivalente	Operação
<code>\$a .= \$b</code>	<code>\$a = \$a . \$b</code>	Concatenação de strings
<code>\$a ??= \$b</code>	<code>\$a = \$a ?? \$b</code>	Aglutinação de nulls (null coalesce)

`$a = null;`

`$b = 0;`

`$a = $a ?? $b; //se $a for nulo, atribui o valor de $b`

`//$a passa a ter o valor 0`

Estruturas de controle - if

```
<?php
if ($a > $b)
    echo "a is bigger than b";
?>
```

```
<?php
if ($a > $b) {
    echo "a is bigger than b";
    $b = $a;
}
?>
```

Estruturas de controle – if/else


```
<?php
if ($a > $b) {
    echo "a is greater than b";
} else {
    echo "a is NOT greater than b";
}
?>
```


Estruturas de controle – if/elseif/else

```
<?php
if ($a > $b) {
    echo "a é maior que b";
} elseif ($a == $b) {
    echo "a é igual a b";
} else {
    echo "a é menor que b";
}
?>
```

Estruturas de controle – if/elseif/else (sintaxe alternativa)

```
<?php
if ($a == 5):
    echo "a equals 5";
    echo "...";
elseif ($a == 6):
    echo "a equals 6";
    echo "!!!";
else:
    echo "a is neither 5 nor 6";
endif;
?>
```



Estruturas de controle – while

```
<?php
/* example 1 */
$i = 1;
while ($i <= 10) {
    echo $i++;
}
/* example 2 */
$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
?>
```

Recapitulação da última aula

- Pegar o resultado inteiro de uma divisão:
 - Não há um operador para este objetivo, porém a seguinte função atinge a este objetivo:
 - `intdiv(dividendo, divisor)`

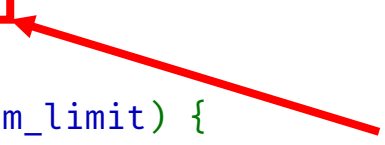
Estruturas de controle – do..while

```
<HTML>
<BODY>
<?php
$i = 0;
do {
    echo "<p>$i</p>";
} while ($i > 0);
?>
</BODY>
</HTML>
```

Destaque para o HTML

Estruturas de controle – do..while

```
$minimum_limit = 10;
$factor = 2;
$i = 5;
<?php
do {
    if ($i < 5) {
        echo "$i is not big enough";
        break;
    }
    $i *= $factor;
    if ($i < $minimum_limit) {
        break;
    }
    echo "$i is ok";
} while (0);
?>
```



Estruturas de controle – for

```
<?php
```

```
/* exemplo 1 */
```

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

Estruturas de controle – for

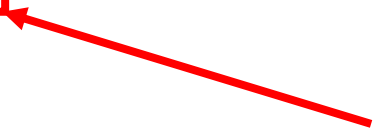
```
/* exemplo 2 */
```

```
for ($i = 1; ; $i++) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
}
```


Estruturas de controle – for

```
/* exemplo 3 */
```

```
$i = 1;  
for ( ; ; ) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
    $i++;  
}
```



Estruturas de controle – for

/* exemplo 4 */

```
for ($i = 1, $j = 0; $i <= 10; $j += $i, print  
$i, $i++);  
?>
```

Estruturas de controle – for

```
<?php
```

```
/* exemplo 5 */
```

```
for ($i = 1; $i <= 10; $i++):
```

```
    echo $i;
```

```
endfor;
```

Estruturas de controle – switch

```
<?php
// Esta declaração switch:

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;
}
```

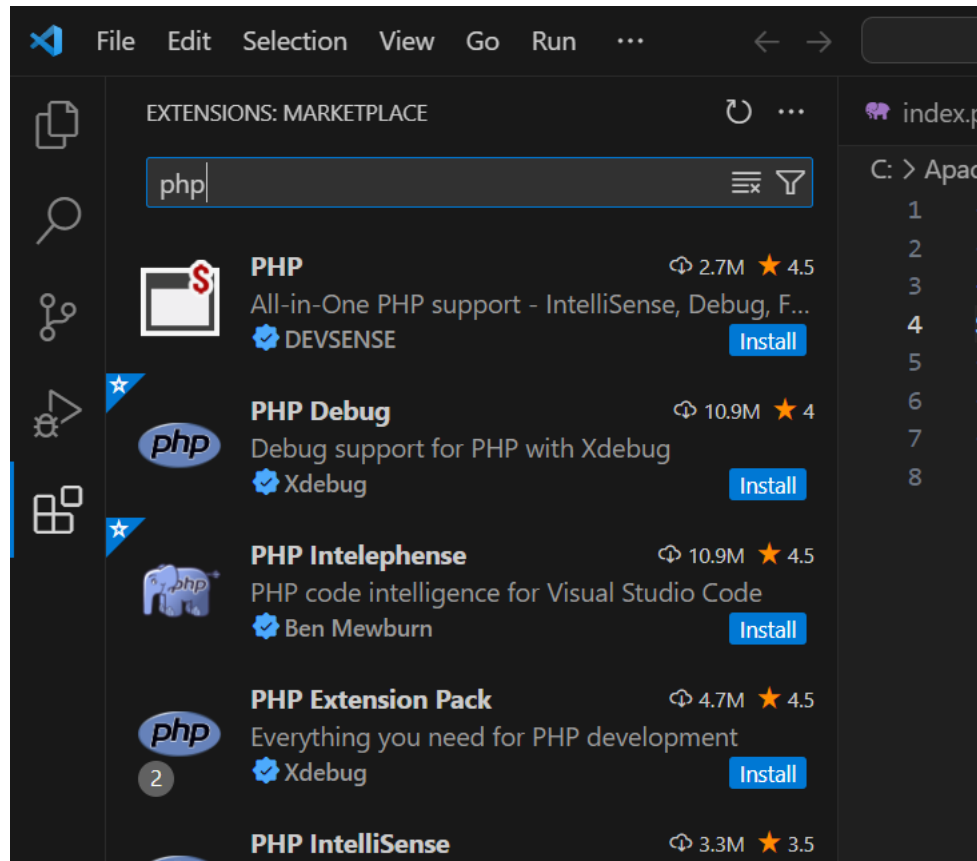
Estruturas de controle – match

```
<?php
$comida = 'bolo';

$valor_de_retorno = match ($comida) {
    'apple' => 'Essa comida é uma maçã',
    'banana' => 'Essa comida é um banana',
    'bolo', 'cake' => 'Essa comida é um bolo',
};

echo $valor_de_retorno;
?>
```

Extensão php intelephense



Referências

- <https://www.alura.com.br/artigos/php-uma-introducao-linguagem>
- [https://www.php.net/manual/pt BR/](https://www.php.net/manual/pt_BR/)