

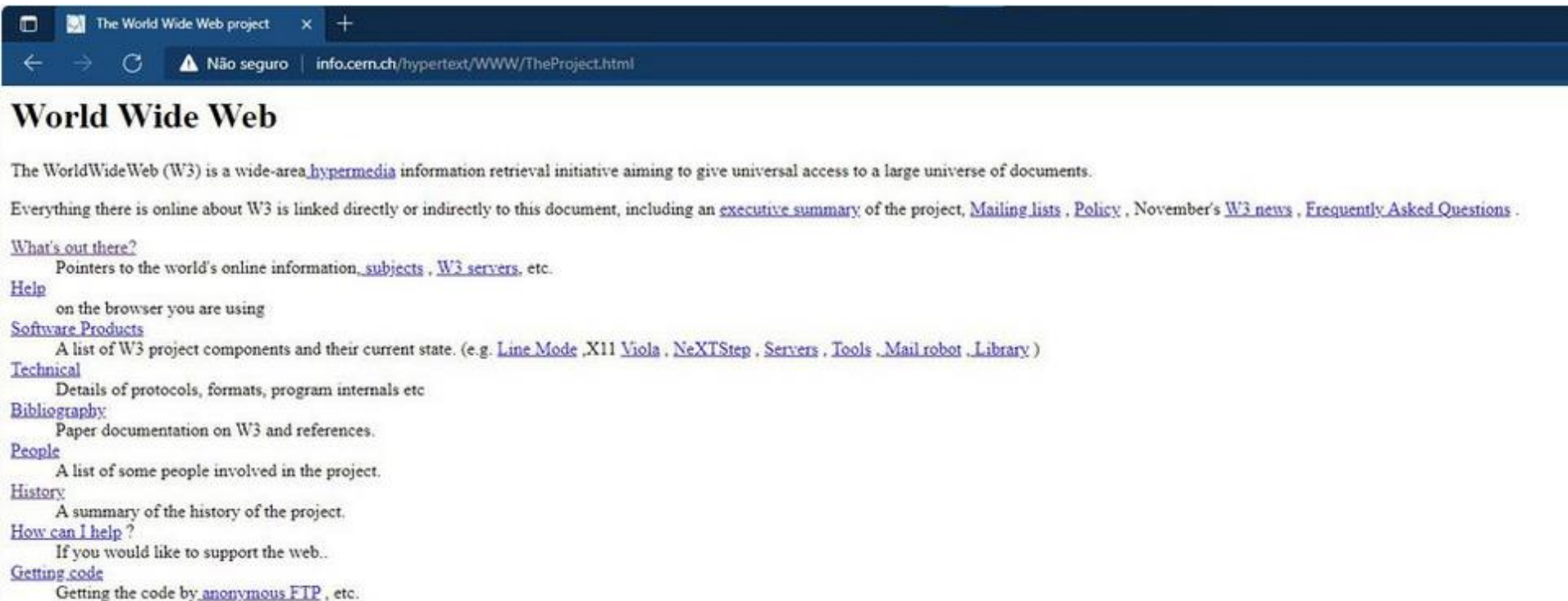
PHP Básico

- **Introdução ao PHP**
- **Sintaxe Básica**
- **Variáveis**
- **Operadores** (aritméticos)

PHP

- Criado por Rasmus Lerdorf in 1994
- Originalmente *Personal Home Page* hoje *PHP: Hypertext Preprocessor*
- Linguagem script (*server-side*)
- Tipagem dinâmica
- Multiparadigma (processualmente, orientada por objetos)
- Influenciada por Perl, C, C++, Java e Tcl.
- Cross-platform (Windows, Linux, Mac OSX...)

Páginas de Internet 1994



The screenshot shows a web browser window with a single tab titled "The World Wide Web project". The address bar displays "info.cern.ch/hypertext/WWW/TheProject.html" with a warning icon and the text "Não seguro". The main content area has the heading "World Wide Web" and a paragraph describing the project as a wide-area hypermedia information retrieval initiative. Below this, there are several sections with underlined links: "What's out there?", "Help", "Software Products", "Technical", "Bibliography", "People", "History", "How can I help?", and "Getting code". Each section provides a brief description of its content.

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)
Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)
on the browser you are using

[Software Products](#)
A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

[Technical](#)
Details of protocols, formats, program internals etc

[Bibliography](#)
Paper documentation on W3 and references.

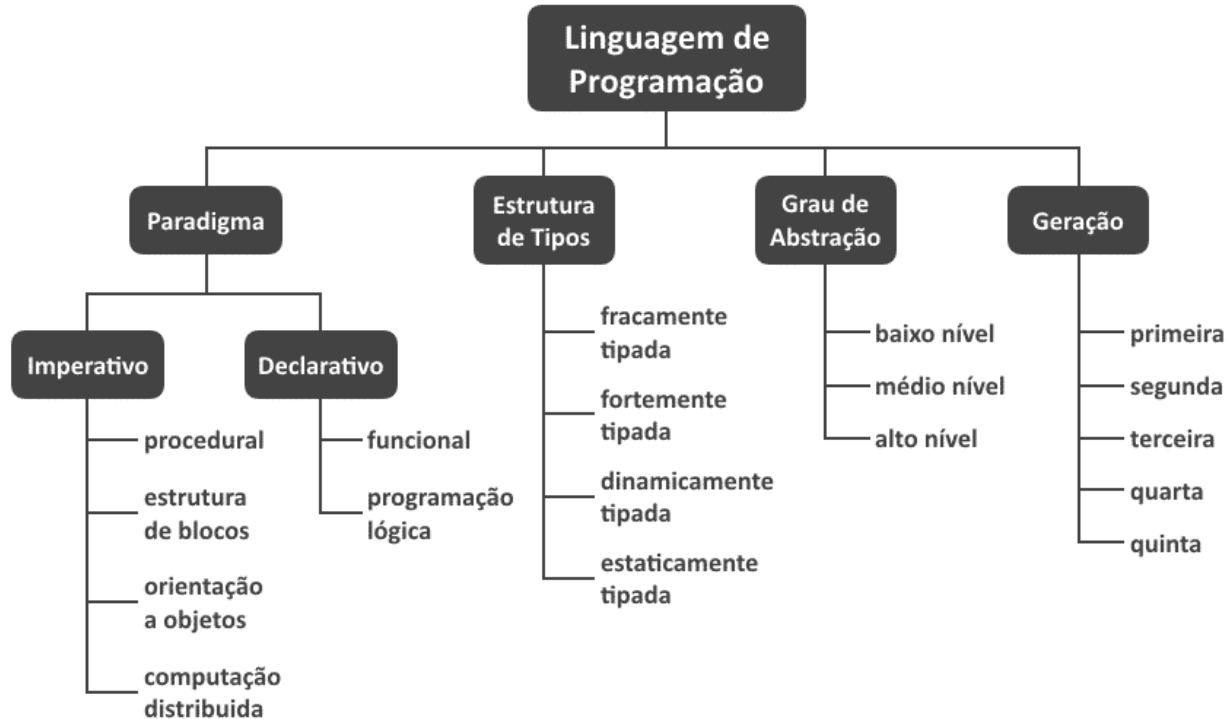
[People](#)
A list of some people involved in the project.

[History](#)
A summary of the history of the project.

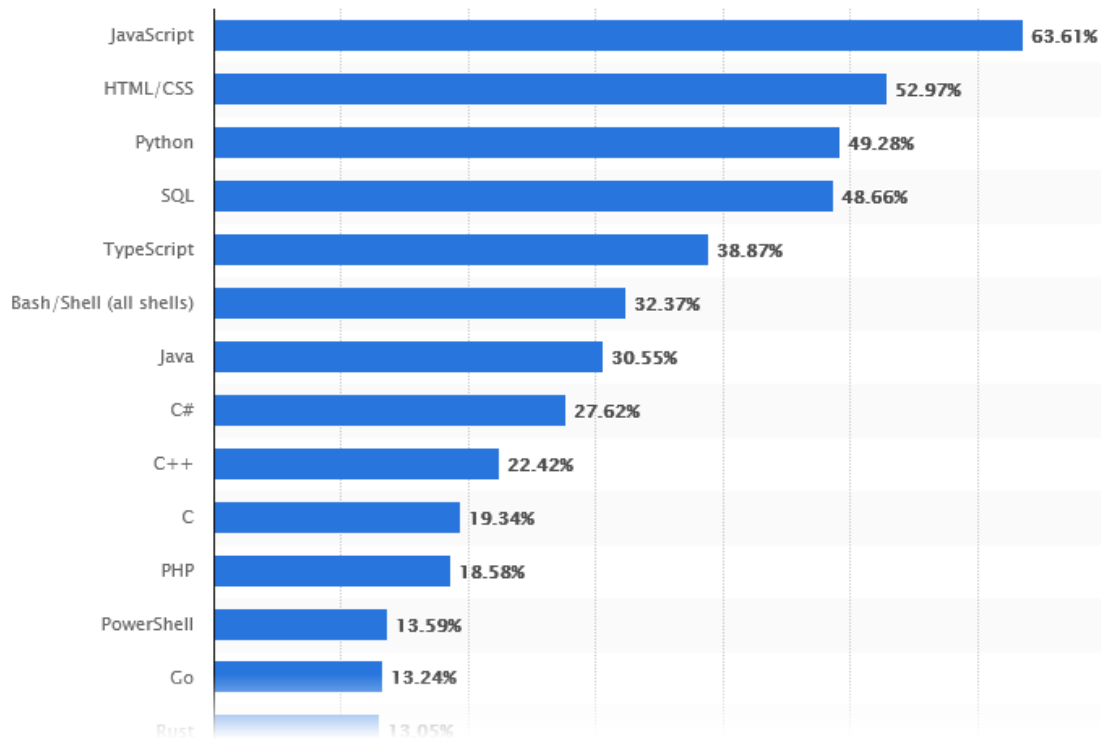
[How can I help?](#)
If you would like to support the web..

[Getting code](#)
Getting the code by [anonymous FTP](#), etc.

Entre as linguagens de programação mais usadas hoje estão:



2023 / 2024



⊕ Expand statistic

[Additional Information](#)

© Statista 2024

[Show source](#)

Linguagens de programação do lado do servidor

Linguagens de programação mais populares do lado do servidor

<small>? W3Techs.com</small>	Uso	mudança desde então 1 de Janeiro de 2024
1. O PHP	76,5%	-0.2%
2. ASP.NET (A)	6,4%	-0.2%
3. Ruby (tradução	5,7%	+0,1%
4. Java em Java	4,7%	
5. JavaScript	3,2%	+0,1%

Porcentagens de sites

Ferramentas (sugestões)

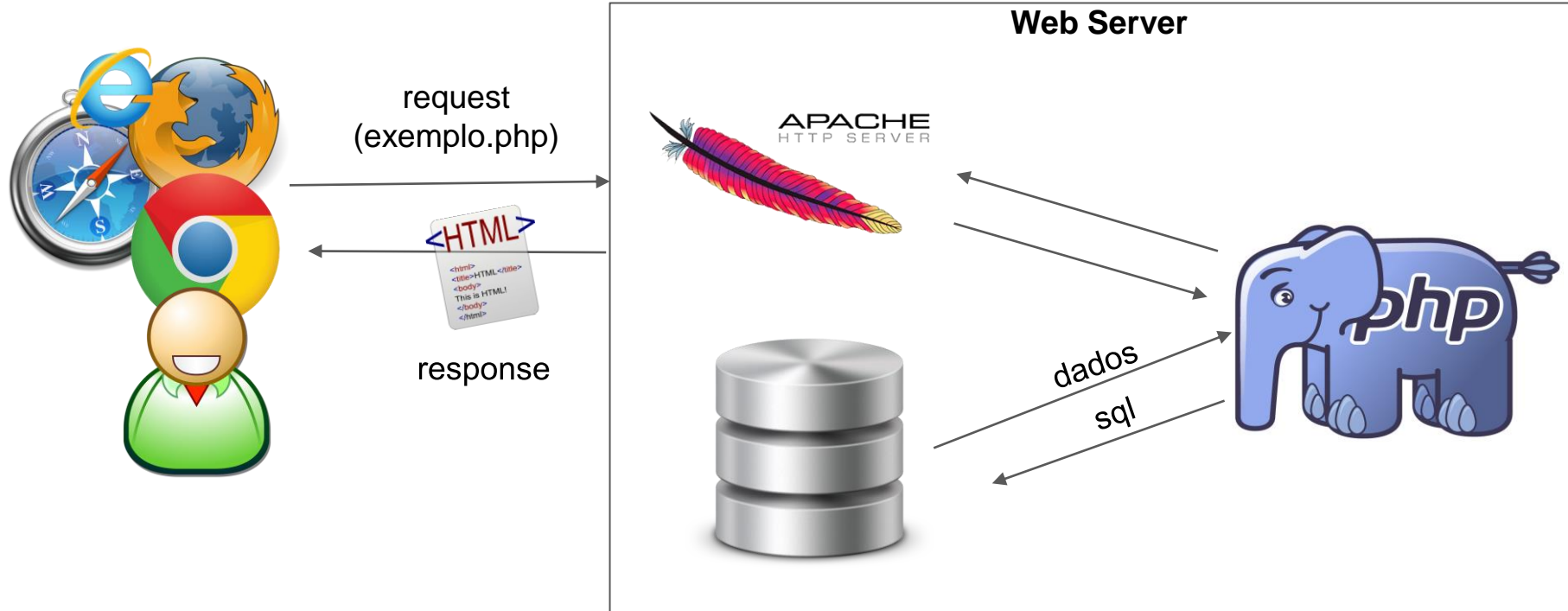
- Ferramentas Web:

- <http://ideone.com/>
- <https://3v4l.org/>
- <http://phpfiddle.org/>
- <http://phptester.net/>

- Ferramentas desktop:

- <http://www.wampserver.com/> → Servidor e banco de dados
 - <https://notepad-plus-plus.org/>
 - <https://netbeans.org/>
 - <https://code.visualstudio.com/>
 - https://www.apachefriends.org/pt_br/index.html → Servidor e banco de dados
- Diagram illustrating the mapping of desktop tools to their functions:
- The first tool (<http://www.wampserver.com/>) is mapped to "Servidor e banco de dados" via a right-pointing arrow.
 - The next three tools (<https://notepad-plus-plus.org/>, <https://netbeans.org/>, and <https://code.visualstudio.com/>) are grouped by a right-facing curly bracket and labeled "Editores".
 - The final tool (https://www.apachefriends.org/pt_br/index.html) is mapped to "Servidor e banco de dados" via a right-pointing arrow.

Exemplo de Arquitetura com PHP



Arquitetura PHP

- Executável que interpreta e executa scripts PHP
- Suporta extensões escritas em C
 - Projetos PECL (PHP Extension Community Library) outros
- Pode utilizar bibliotecas de terceiros escritas em PHP
 - Projeto PEAR (PHP Extension and Application Repository) e outros
- Precisa, normalmente, de um serviço HTTP para responder requisições Web

Sintaxe básica (HTML5 + PHP)

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>PHP Teste</title>
```

```
  </head>
```

```
  <body>
```

```
    <?php echo '<p>Hello World</p>'; ?>
```

```
  </body>
```

```
</html>
```

O objetivo final da maioria dos documentos web PHP será exibir dinamicamente algum conteúdo HTML, pois esta é a linguagem de marcação utilizada pelos navegadores para exibição e formatação dos elementos de uma página web.

Este é um exemplo de uma página web em HTML5 com um trecho de instruções para interpretação do PHP.

O cliente não terá acesso ao código PHP, apenas ao conteúdo final escrito pelo mesmo.

Separação de instruções (; ?>)

```
<?php
```

```
    echo 'Isto é um teste';
```

```
?>
```

Hello World

': ?>

```
<?php echo 'Isto é um teste' ?>
```

```
<?php echo 'Nós omitimos a última  
tag de fechamento';
```

Basicamente um código **PHP** começa com o símbolo de menor seguido de um sinal de interrogação e o termo **PHP**.

Cada instrução é separada pelo caractere ponto e vírgula.

Para fechar um bloco de códigos **PHP** utilizamos o símbolo de interrogação e o sinal de menor.

É válido salientar que a instrução de fechamento do **PHP** é opcional quando não existe nenhum outro tipo de código após o bloco de instruções

Comentários (//, /* */ , #)

```
<?php
```

```
    echo 'Isto é um teste'; // Estilo de  
comentário de uma linha em c++
```

```
/* Este é um comentário de múltiplas linhas  
    ainda outra linha de comentário */
```

```
echo 'Isto é ainda outro teste';
```

```
    echo 'Um teste final'; # Este é um comentário  
de uma linha no estilo shell
```

```
?>
```

Assim como outras linguagens, o PHP permite a escrita de comentários, que permitem ao desenvolvedor escrever textos que possam explicar a outros desenvolvedores o que significa aquele código ou até documentar de forma estruturada elementos da aplicação através do PHPdoc.



Require & Include (Require_Once & Include_Once)

- Funcionam com caminhos

- **Absolutos:**

- “D:/wamp64/www/PW II/aula 01/inclusao.php”

- **Relativos:**

- “inclusao.php”  Mesmo nível.
 - “php/inclusao.php”  Subir um nível.

Require & Include (Require_Once & Include_Once)

- A diferença entre include e require é a forma como um erro é tratado.
 - **Require** produz um erro E_COMPILE_ERROR, o que encerra a execução do script.
 - **Include** apenas produz um WARNING, o que permite continuar a execução do script.
- **Include_Once** tem a garantia que o arquivo não será incluído novamente se ele já foi incluído antes.
- **Require_Once** é análogo ao **Include_Once**.

Variáveis e Tipos “básicos”

```
<?php
```

```
$minha_var; // null
```

```
$texto = "Ola Mundo !"; // string
```

```
$x = 5; // integer
```

```
$y = 10.5; // float ou double
```

```
$verdadeiro = true; // boolean
```

```
$nulo = null; // null
```

O elemento básico de qualquer linguagem de programação é variável.

Variáveis são basicamente locais onde as informações são definidas ou referenciadas. No PHP as variáveis começam com o símbolo do cifrão (\$) seguido pelo nome da variável.

Se nenhum valor é definido para esta variável, ela começa automaticamente com o valor NULL ou nulo.

Para definir um valor basta utilizar o símbolo de igual seguido do valor que pode ser um destes tipos primitivos ou também um array ou uma referência a um objeto.

Como o PHP é uma linguagem de tipagem dinâmica, não é necessário que o desenvolvedor defina um tipo para a variável, pois a mesma poderá ter seu tipo alterado de acordo com o valores a serem definidos na execução do código.

Variáveis (diferença entre caixas)

```
<?php
```

```
//duas variáveis diferentes
```

```
$verdadeiro = true;
```

```
$Verdadeiro = false;
```

É válido salientar que os nomes de variáveis no PHP fazem distinção entre letras maiúsculas e minúsculas (caixas alta e baixa).

Portanto, neste exemplo, existem duas variáveis distintas pois uma possui uma letra em caixa diferente da outra.

Echo e Print

```
<?php
```

```
echo("Olá Mundo!");
```

```
echo "Olá Mundo !";
```

```
echo "Olá ", "Mundo!";
```

```
<?php
```

```
print("Olá Mundo!");
```

```
print "Olá Mundo!";
```

```
$ret = print "Hello World";
```

```
print("Olá mundo! ".$ret);
```

Outro recurso fundamental de qualquer ambiente de desenvolvimento é a **saída padrão**, isto é, uma forma para se exibir conteúdo para o usuário. No PHP podemos utilizar tanto o echo e como o print. Como nenhum deles é exatamente uma função PHP, o uso do parêntese é opcional.

As diferenças entre eles são que o echo é ligeiramente mais rápido e pode aceitar mais de um parâmetro enquanto o print tem valor de retorno 1 e pode ser usado como uma função.

Strings ('Single quoted' e "Double quoted")

```
<?php
```

```
$versao = 7;
```

```
echo 'PHP versão $versao';
```

```
//saída: PHP versão $versao
```

```
echo "PHP versão $versao";
```

```
//saída: PHP versão 7
```

Para definir uma *string* ou texto, o PHP permite utilizar dois diferentes tipos de caracteres delimitadores, o apóstrofo ou aspas.

Quando uma string é delimitada por apóstrofos, nomes de variáveis serão considerados partes integrantes do texto.

Já com o uso das aspas, nomes de variáveis serão interpretados e seus valores e não seus nomes é que serão considerados para integrar a *string*.

Operadores de String (concatenação “.” e “.=”)

```
<?php
```

```
$a = "Olá ";
```

```
$b = $a . "Mundo!";
```

```
$a = "Hello ";
```

```
$a .= "World!";
```

Um importante recurso para a manipulação de duas ou mais strings é o operador de concatenação.

O sinal de ponto no PHP é utilizado com este propósito: permitir a união de uma ou mais strings.

Ele também pode ser utilizado com o símbolo de igual para permitir que o PHP faça a concatenação do conteúdo já presente na variável à esquerda do ponto com o conteúdo a direita do sinal de igual.

unset()

```
<?php  
  
$a = 1;  
$b = $a;  
unset($a);  
echo $a; //Notice: Undefined variable: a  
echo $b;
```

O unset() destrói variáveis previamente especificadas.

O comportamento de unset() dentro de uma função pode variar dependendo do tipo de variável que você está tentando destruir.

Operadores Aritméticos

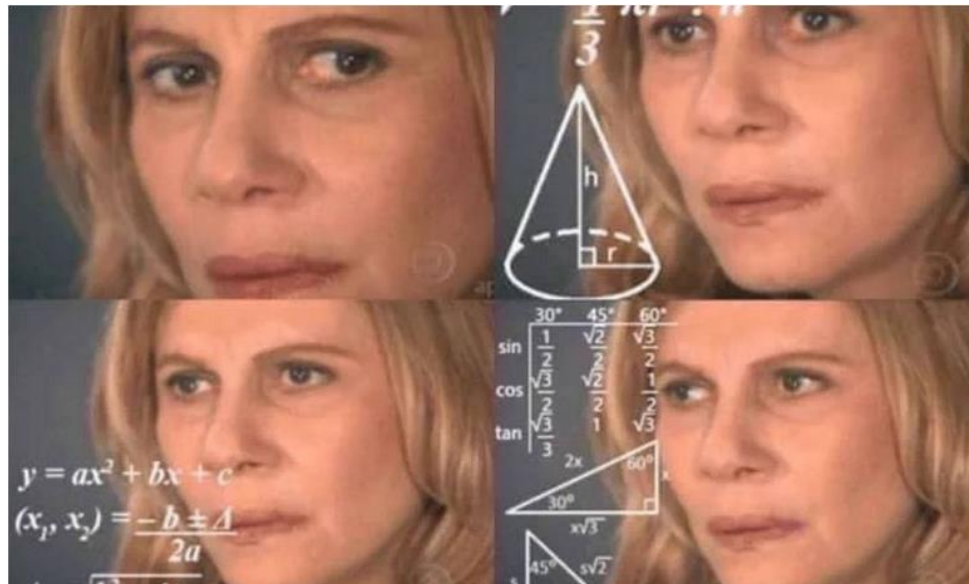
Operador	Nome	Exemplo	Resultado
+	Adição	<code>\$x = 2 + 2;</code>	4
-	Subtração/Negação	<code>\$x = 4 - 2;</code>	2
*	Multiplicação	<code>\$x = 2 * 10;</code>	20
/	Divisão	<code>\$x = 15 / 5</code>	3
%	Módulo (sobra da divisão)	<code>\$x = 5 % 2</code>	1
**	Exponencial	<code>\$x = 4 ** 4;</code>	256

Outro conjunto de operadores comumente disponível nas linguagens de programação são os operadores aritméticos.

Ressaltando que além das quatro operações básicas da aritmética, no PHP, existe também o operador de módulo, que permite obter o valor da sobra de uma divisão e o operador de exponencial.

Ordem de precedência

- Desafio:
 - $8/2(2+2)$



Ordem de precedência

PHP usa a ordem de precedência conhecida como BODMAS

B	RACKETS	() [] { }
O	RDER POWER OF	$\sqrt{\quad}$ $(\quad)^2$
D	IVIDE	/ ÷
M	ULTIPLY	* X
A	DDITION	+
S	UBTRACTION	—