PHP

**O que é o PHP?**

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

**Para o que serve o PHP?**

Qualquer coisa. O PHP é focado principalmente nos scripts do lado do servidor, portanto, você pode fazer qualquer coisa que outro programa CGI pode fazer, como coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais.

* **Scripts no lado do servidor (server-side).** Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para isto funcionar: **o interpretador do PHP (CGI ou módulo do servidor), um servidor web e um navegador web.** Você precisa rodar o servidor web conectado a uma instalação do PHP. Você pode acessar os resultados de seu programa PHP com um navegador web, visualizando a página PHP através do servidor web. Tudo isso pode rodar na sua máquina pessoal se você estiver apenas experimentando programar com o PHP. Veja a seção das [instruções de instalação](https://www.php.net/manual/pt_BR/install.php) para mais informações.
* **Scripts de linha de comando.** Você pode fazer um script PHP para executá-lo sem um servidor ou navegador. A única coisa necessária é o interpretador PHP. Esse tipo de uso é ideal para script executados usando o cron (Unix, Linux) ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto simples. Veja a seção [Utilizando o PHP na linha de comando](https://www.php.net/manual/pt_BR/features.commandline.php) para mais informações.
* **Escrever aplicações desktop.** O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas se você conhece bem o PHP, e gostaria de usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, você pode usar o PHP-GTK para escrever programas assim. Você também tem a possibilidade de escrever aplicações multiplataformas desse jeito.

**Primeiros passos para utilizar o PHP**

Para iniciar seus estudos no PHP, você deve instalar o arquivo mais recente do PHP no próprio site da linguagem.

link:<https://windows.php.net/download/>

Depois de instalar o PHP na sua máquina, você deve ir à pasta onde está o arquivo e copiar o endereço dele, feito isso você entrará em **variáveis de ambiente**, em seguida você deve procurar a pasta **path**, depois de entrar nela, você deverá adicionar o endereço que copiou do arquivo PHP lá atrás, depois de ter feito todos os passos sua máquina estará pronta para programar em PHP.

Sempre devemos criar um ambiente adequado para desenvolver em PHP.

Logo em seguida devemos escolher uma IDE (Ambiente de desenvolvimento integrado) para podermos programar em PHP, temos várias no mercado, as mais indicadas são o PhpStorm que é **pago** e o Eclipse que é um **opensource** que também é bem interessante.

Porém eu estou usando o Vscode, pois já estou habituado a suas ferramentas.

Agora, iremos criar nosso primeiro arquivo com o código PHP.

Para isso precisamos começar com o comando <?php.

obs:Lembre-se que o espaço (ou nova linha) após <?php é obrigatório. Esta informa para o interpretador do PHP que o código que virá deve ser lido e interpretado.

O que são variáveis?

Um espaço na RAM onde através de um nome, podemos manipular e acessar algum valor. Além disso, como o próprio nome já diz, variáveis podem variar, ou seja, seus valores podem ser alterados durante a execução do programa.

E para declarar uma variável em PHP e muito simples, basta seguir o exemplo:

Ex:  **$idade = 22**

Seguindo o exemplo, temos 4 elementos;

$ - O cifrão serve para indicar ao PHP que aquele termo junto dele é uma variável.

idade - Idade seria o nome da variável.

= - O sinal de igual, passa o sentido de atribuição.

22 - 22 seria o valor da variável.

**Operações matemáticas**

Agora que já aprendemos sobre variáveis, podemos começar a fazer operações matemáticas com essas variáveis. Como por exemplo somar, subtrair, multiplicar e dividir.

**Mas como fazer com que o PHP entenda que ele deve fazer essas operações?**

Muito simples, você deve usar os sinais que irei apresentar para vocês;

**+ (soma)**

**- (subtração)**

**\* (multiplicação)**

**/ (divisão)**

**\*\* (exponenciação)**

**% (módulo ou resto da divisão)**

Esses são os principais sinais que você vai utilizar no seu dia a dia, porém existem muitos outros. Agora irei exemplificar como podemos aplicar o que aprendemos agora em um código PHP;

**$soma = 2 + 2;**

**$subtracao = 2 - 2;**

**$multiplicacao = 2 \* 2;**

**$divisao = 2 / 2;**

**$doisAoCubo = 2 \*\* 3;**

**//ou**

**$doisAoCubo = 2 \* 2 \* 2;**

**$restoDaDivisao = 10 % 3;**

**Tipos de dados**

O PHP utiliza checagem de tipos dinâmica. Em outras palavras, uma variável pode conter valores de diferentes tipos em diferentes momentos da execução do script.

Os tipos de dados suportados por esta linguagem são divididos em dois grupos:

**Escalares**

* Inteiro
* Ponto flutuante (chamado de float ou Double)
* String
* Booleanos

**Compostos**

* Array
* Objeto
* Iterable
* NULL
* resource

Os tipos de dados compostos callable, iterable, resource e NULL serão abordados num outro momento.

Para saber mais sobre o primeiro grupo **(Escalares)**, disponibilizarei um link para consulta.

link: <https://www.devmedia.com.br/tipos-de-dados-do-php/25566>

**Tipos de String**

Dando continuidade ao tópico anterior.

O tipo de dados STRING é uma cadeia de caracteres alfanuméricos (letras, números e caracteres especiais). O tipo STRING pode ser utilizado de duas maneiras:

Utilizando aspas simples o valor da variável será exatamente o texto contido entre as aspas, com exceção de: \\ e \’.

ex: **echo ‘Eu tenho’. $idade. ‘anos’;**

Utilizando aspas duplas qualquer variável ou caractere de escape será expandido antes de ser atribuído.

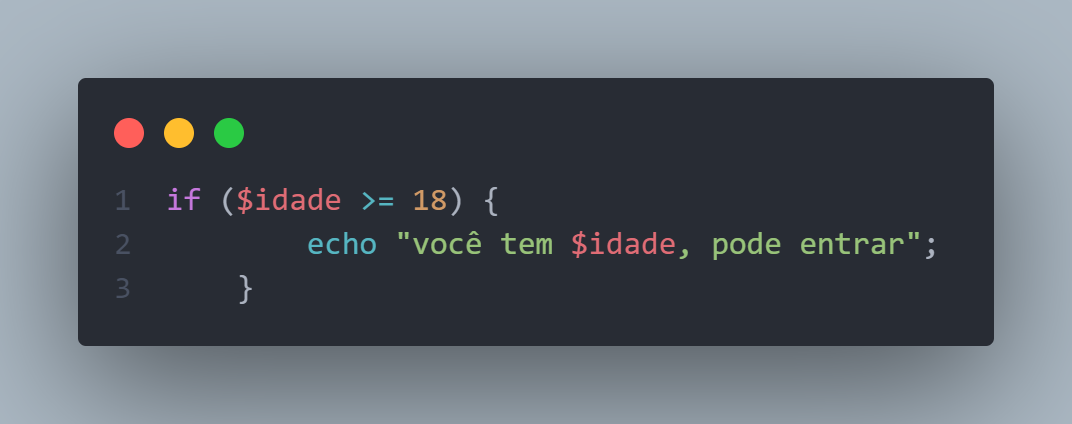
ex: **echo “Eu tenho $idade anos”;**

**Sequência de escape:** Caractere de escape é um termo usado em computação para identificar um único caractere numa cadeia de caracteres que altera o significado de seu sucessor.

**Verificação em PHP**

Como o PHP é uma linguagem dinâmica, temos a possibilidade de fazer verificações na linguagem usando um simples comando, O **if,** ele serve para criarmos essa verificação.

sintax:



Saber os operadores lógicos é muito importante para utilizar essa estrutura de verificação.

**op (operadores lógicos)**

> - maior que

> = - maior ou igual

< - menor que

< = - menor ou igual

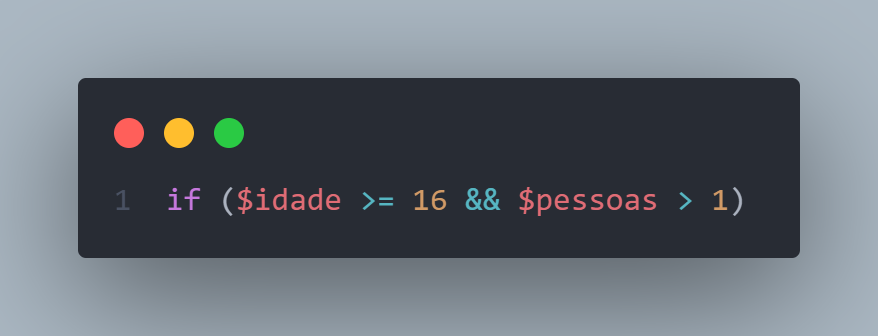
== - igual

= - recebe

&& - e

|| - ou

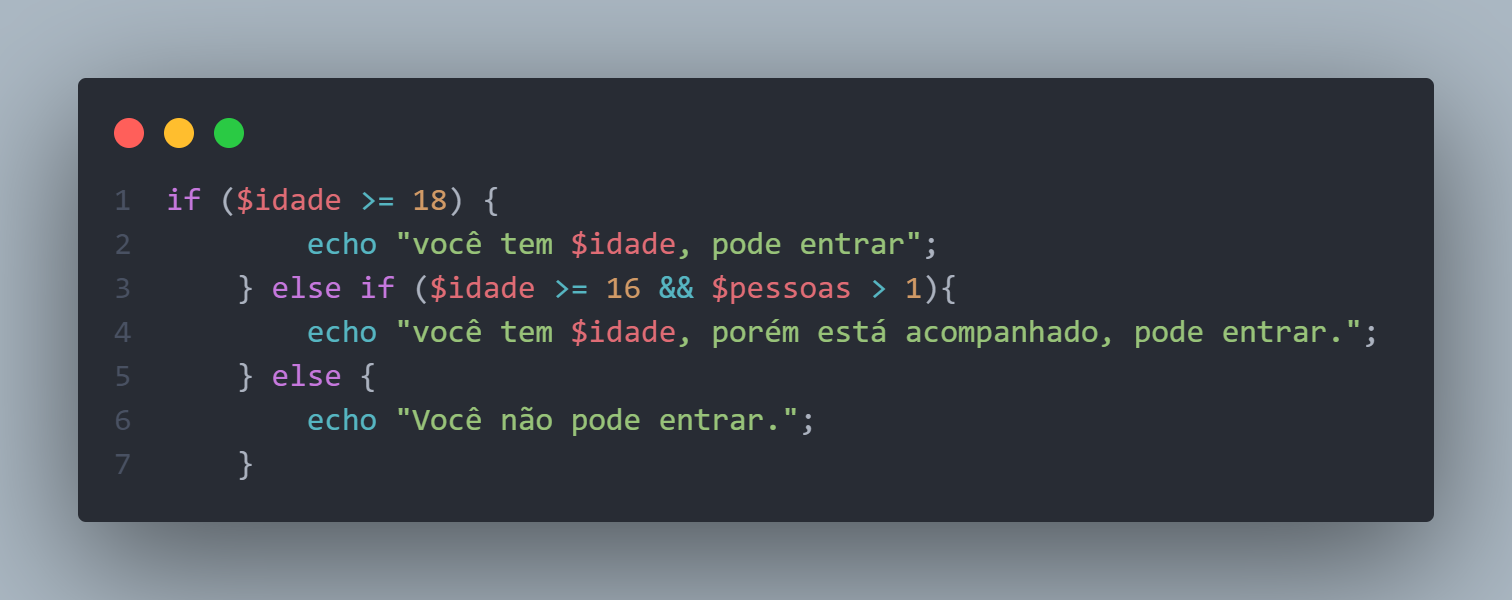
exemplo:



Também podemos conhecer o comando **else**, que serve para não termos que fazer diversas verificações, ele é utilizado depois da fechando do bloco de comando e executa o próximo um comando desse do seu próprio bloco.

Esse **else** também pode ser acompanhado de um **if** quando quiser criar uma condição.

segue o exemplo abaixo;

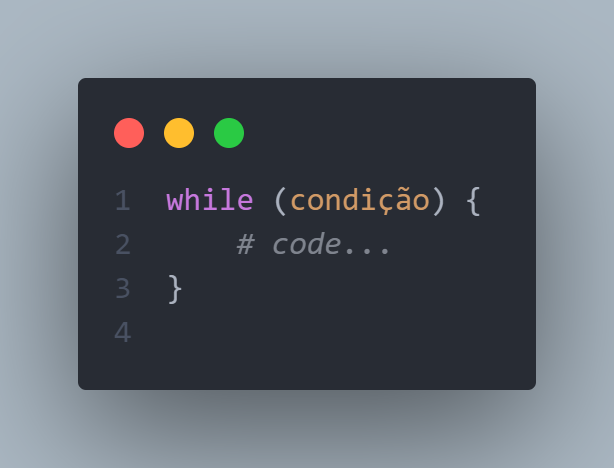


**Estrutura de repetição no PHP**

Estruturas de repetição são recursos que permitem executar trechos de código mais de uma vez de acordo com uma condição. O PHP possui quatro estruturas de repetição: for, foreach, while e do-while.

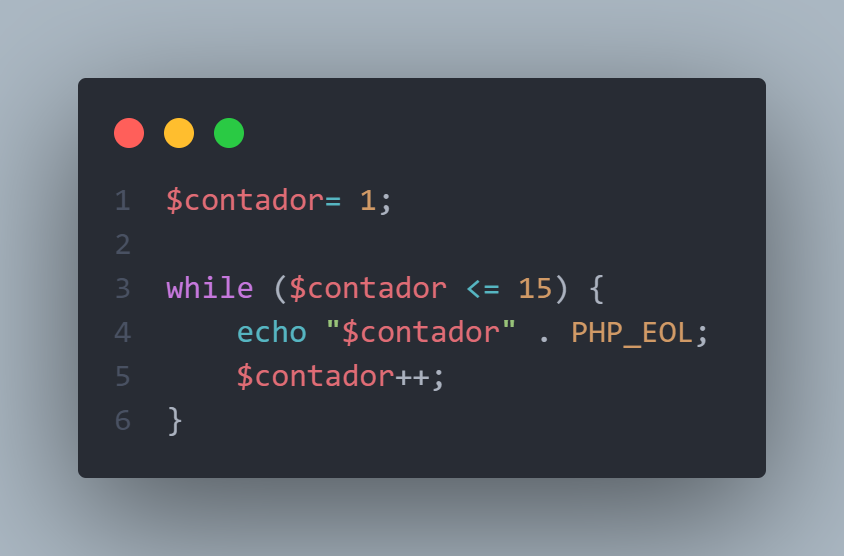
**While**

**Sintaxe do while:**

****

A sintaxe é composta por uma expressão, validada antes de cada iteração do loop, e pelo bloco de código a ser executado, declarado entre chaves ´ {} ´. Se a expressão retornar falso, o loop não é executado.

**Exemplo de uso:**

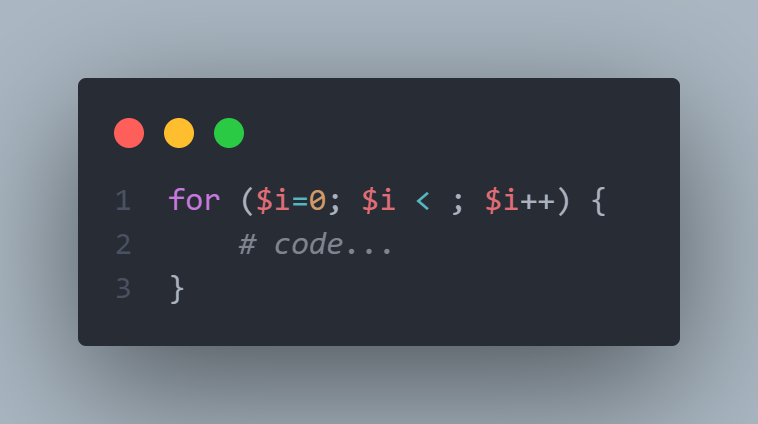
****

Ao executar este código serão impressos os números de 1 a 15.

**For**

O for é a estrutura de repetição do PHP que utilizamos quando sabemos a quantidade de repetições que devem ser executadas.

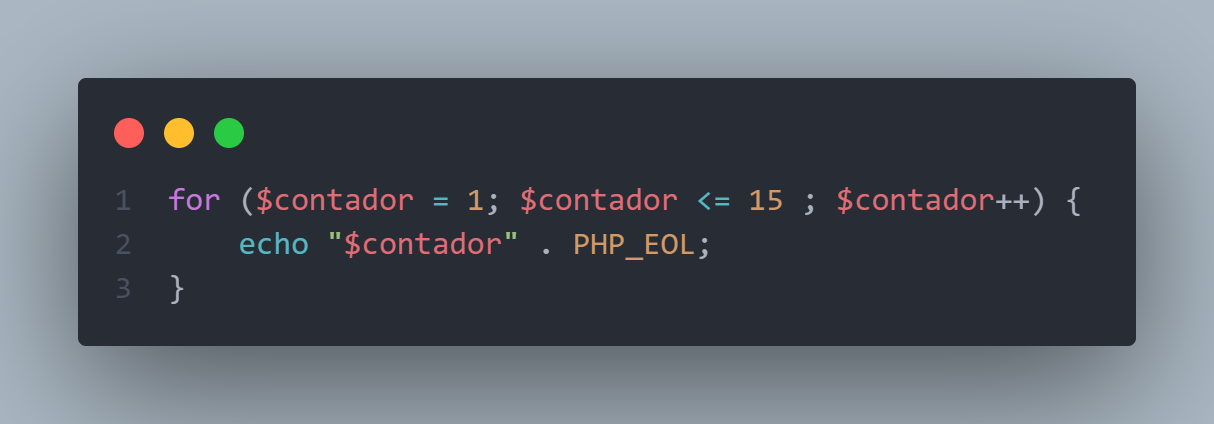
**Sintaxe do for:**



A sintaxe é composta por três expressões separadas por ponto e vírgula, cada uma delas podendo ter zero, uma ou mais declarações separadas por vírgula:

* **Expressão 1**: Executada somente uma vez, ao iniciar o loop. Normalmente a utilizamos para declarar e inicializar as variáveis que faremos uso para controlar o número de iterações do loop;
* **Expressão 2**: Expressão booleana, validada antes de cada iteração do loop. Se a expressão contiver múltiplas expressões, todas serão executadas, mas somente o resultado da última será considerado. Se a expressão for vazia, ela será interpretada como verdadeira. O loop somente será executado enquanto essa expressão retornar **true**;
* **Expressão 3**: Executada ao final de cada iteração, normalmente a utilizamos para declarar a forma de atualização do valor da variável avaliada na expressão 2.

Exemplo de uso:



**PHP - O Servidor**

Todo site ou serviço da internet, está hospedado em algum computador. Em um tipo especial de computador: um servidor.

Quando você entra em um site, como o da Globo e vê uma foto, esta foto está armazenada em algum servidor. Então, seu navegador vai atuar por você:

Cliente - “Ei servidor, quero acessar sua página, manda os dados pra mim"

Servidor - "OK, toma aí o HTML, o CSS, as imagens, vídeos..."

Cliente - "Ok! Agora quero os dados dessa outra página?"

Servidor - "Só um minuto, vou procurar. Pronto. Enviando."

E fica nessa eterna 'conversa'.

E como explicamos no tutorial anterior, PHP é uma linguagem de programação voltada para o lado do servidor. Ou seja: vamos precisar de um servidor para estudar PHP.

Na verdade, não é obrigatório, você pode comprar um servidor (o que é **muito** caro) ou pagar um serviço de hospedagem, mas isso tem vários inconvenientes, como uma possível lentidão.

Por isso, vamos criar um servidor de PHP em nossa própria máquina!

Além da questão da velocidade (se rodar cliente e servidor numa mesma máquina, o resultado é praticamente instantâneo, sem demora), a conexão nunca vai cair, seu site nunca ficará fora do ar e não tem perigo de alguém invadir seus sistemas enquanto você está em processo de aprendizagem, pois nesse estágio deixamos várias falhas.

<https://www.phpprogressivo.net/2019/01/Necessario-para-Comecar-Programar-PHP-Instalando-Ambiente.html>

**Montando um servidor Web: *Apache*, *MySQL* e *PHP***

Para começar a programar em PHP, vamos precisar de três coisas instaladas em seu computador:

* Apache - Servidor HTTP Apache, programa que vai simular um servidor e as comunicações entre servidor e cliente, tudo na sua máquina, numa espécie de 'rede interna'
* MySQL - Sistema de gerenciador de banco de dados, para que possamos armazenar, consultar, alterar e fazer de tudo com informações (dados do usuário, login, senha etc.)
* PHP - Interpretador PHP, um programa que vai interpretar o código PHP que vamos escrever durante nosso curso

## **WAMP, MAMP, LAMP e o XAMPP**

Calma, não se assuste com essa sopa de letrinhas! São apenas abreviações:

WAMP - Windows, Apache, MySQL e PHP

MAMP - Mac, Apache, MySQL e PHP

LAMP - Linux, Apache, MySQL e PHP

XAMPP - Cross(x) Apache, MySQLDB, PHP e Perl

**O que raios é isso?**

Simples: seu ambiente de programação. Em vez de ter que sair instalando e configurando vários programas e funcionalidades (Apache, MySQL, PHP, Perl etc.), você instala uma única dessas opções, e elas já contém tudo. Literalmente você vai ter tudo pronto, instalado e configurado, baixando apenas uma coisa.

Como estou com uma máquina em Linux e outras pessoas em Windows, vamos instalar e usar o XAMPP, pois ele é mais completo e *cross-plataform*, ou seja, funciona em tudo que é sistema operacional.

Você pode escolher qualquer uma das opções lá de cima. O PHP é o mesmo para todos e serve em todos esses servidores.