**01 Projeto da aula anterior**

Caso queira, você pode [baixar o projeto do curso](https://caelum-online-public.s3.amazonaws.com/1241-php/01/1241-aula01.zip) no ponto em que paramos na aula anterior.

**02 O que são variáveis**

**Transcrição**

Na última aula nós conversamos bastante sobre o que é o PHP e como executar um código PHP, e acabamos exibindo uma mensagem para o usuário. Essa mensagem é um dado fixo, e não conseguimos, depois de exibi-la, manipulá-la de alguma forma. Então, como é que conseguimos lidar com dados que mudam...ou seja, dados variáveis?

Uma variável, em uma linguagem de programação, é um espaço no programa que fica salvo na memória RAM, espaço esse a qual podemos adicionar alguma informação. A partir do momento em que a informação é guardada, podemos realizar algumas operações, modificar o valor dessa variável e utilizá-la de outras formas posteriormente - por exemplo, exibindo-a para o usuário.

Esse conceito de variável é utilizado em qualquer linguagem de programação, e até mesmo na matemática (quando estudamos o conceito de funções e equações). Quando precisamos de um dado que não seja fixo e inalterável, utilizamos as variáveis. Mas como criamos variáveis no PHP?

No nosso projeto, criaremos um novo arquivo chamado variavel.php. Queremos armazenar a minha idade para realizarmos algumas operações com ela. Sabemos que o nome desse dado é idade, e que a minha idade (no caso eu, Vinicius) é 21 anos. Queremos informar ao PHP que ele deve armazenar na memória, em um espaço chamado idade, o valor 21.

No PHP, não precisamos informar que uma variável existe antes de utilizá-la. Ao invés disso, no momento da utilização, podemos informar que queremos que a variável idade tenha o valor 21. Para informarmos que determinado nome é uma variável, precisamos adicionar um símbolo de cifrão $ ao início dele. Para que a variável receba um valor, usamos o sinal de igual =.

$idade = 21

Dessa forma, o valor 21 será atribuído à variável $idade. A partir de agora, em qualquer lugar que quisermos utilizar esse valor, poderemos chamar a variável $idade. Por exemplo, podemos exibi-la na tela chamando echo $idade, o que nos retornará...21! Se mudarmos o valor para 22 e executarmos novamente, esse será o valor exibido na tela.

Nós ainda podemos, inclusive, alterar esse valor depois que ele é definido. Por exemplo:

$idade = 22;

$idade = 21;

echo $idade;

Dessa forma, o valor exibido na tela será 21. Com isso, aprendemos o conceito de variáveis: um valor que, mesmo depois de definido, pode ser modificado. Atualmente temos uma variável cujo valor é um número, e podemos fazer operações com números - por exemplo, descobrir quantos anos terei daqui a 10 anos. Como fazemos esse tipo de operação matemática com o PHP?

Entenderemos mais sobre isso no próximo vídeo.

# 03 Variáveis

**Ao desenvolver um programa, certamente precisaremos armazenar dados. Seja para realizar cálculos, processamentos ou somente para exibir para o usuário. Para armazenar estes dados, utilizamos variáveis.**

**Qual a melhor definição do termo variável?**

Top of Form

* Alternativa correta



Um arquivo temporário que o PHP lê sempre que é executado

Bottom of Form

  Alternativa correta



Um espaço na RAM onde através de um nome, podemos manipular e acessar algum valor.

 Alternativa correta! Além disso, como o próprio nome já diz, variáveis podem variar, ou seja, seus valores podem ser alterados durante a execução do programa.

 Alternativa correta



Um número que representa a idade da pessoa que está programando

# 04 Operações matemáticas

## Transcrição

No último vídeo estávamos pensando sobre como realizar operações matemáticas com as nossas variáveis. Para entendermos esse processo, criaremos um arquivo operacoes.php.

Se você estiver utilizando algum editor de texto ou IDE que não o PhpStorm, lembre-se sempre de adicionar a extensão .php ao final do arquivo!

No exemplo anterior, a ideia era calcularmos a idade que eu teria em 10 anos. Portanto, criaremos uma variável $idadeDaqui10Anos. Sabendo que hoje eu tenho 21 anos, podemos adicionar a esse valor + 10:

$idadeDaqui10Anos = 21 + 10;

Com isso, realizaremos uma soma. Dentro de operações matemáticas, podemos inclusive utilizar outras variáveis:

$idade = 21;

$idadeDaqui10Anos = $idade + 10;

Dessa forma, o PHP entenderá que o valor de idadeDaqui10Anos é o valor de $idade, que definimos como 21, mais o valor 10, que não foi armazenado em outra variável. Podemos finalizar com echo $idadeDaqui10Anos para verificarmos se o PHP realiza essa operação corretamente.

Executando php operacoes.php no terminal, receberemos o valor "31". Se mudarmos o valor de $idade para 25, receberemos "35". Assim, aprendemos que uma operação matemática é muito simples de ser executada com o PHP, e ainda podemos utilizar variáveis dentro de uma operação.

Além da soma, podemos fazer outras operações, como a subtração:

$idade = 21;

$idadeHa5Anos = $idade - 5;

echo $idadeHa5Anos;

Dessa forma, o valor "16" será exibido na tela. Antes de continuarmos com outras operações, vamos apagar todo o código que escrevemos até agora. Em seguida, informaremos que $soma recebe 2 + 2; que $subtracao recebe 2 - 2; que $multiplicacao recebe 2 \* 2 (o asterisco é utilizado para essa operação); e $divisao recebe 2 / 2.

É interessante não utilizar cedilhas, tils e outros acentos em nomes de variáveis, pois às vezes elas podem atrapalhar o funcionamento da aplicação!

$soma = 2 + 2;

$subtracao = 2 - 2;

$multiplicacao = 2 \* 2;

$divisao = 2 / 2;

Essa são as quatro operações matemáticas básicas a partir das quais conseguiremos chegar em outros resultados. Por exemplo, se quisermos a potência de um número, por exemplo, 2 elevado ao cubo, poderíamos fazer 2 \* 2 \* 2. Existe ainda o operador de potência, representado por dois asteriscos \*\*.

$doisAoCubo = 2 \*\* 3;

//ou

$doisAoCubo = 2 \* 2 \* 2;

Ambas as construções, obviamente, retornam o mesmo resultado (8). Outra operação que podemos buscar é o $restoDaDivisao. Por exemplo, se dividirmos 10 por 3 e quisermos o quanto sobrou dessa operação, utilizamos, ao invés de /, o %:

$restoDaDivisao = 10 % 3;

echo $restoDaDivisao;

Executando esse código, teremos como retorno o valor "1". Isso porque 10 dividido por 3 é 3, e o resto é 1 (não estamos contando, claro, divisões que aceitam números decimais).

Repare, então, que fizemos todas essas operações com números inteiros. Será que é possível trabalharmos com números decimais? Que outros tipos de dados o PHP consegue trabalhar em uma variável?

Conversaremos sobre isso no próximo vídeo.

# 05 Matemática com PHP

**É mais do que comum nós precisarmos realizar operações matemáticas durante a execução de nosso programa. Quais os operadores matemáticos disponíveis no PHP?**

Top of Form

* Alternativa correta



+ (soma), - (subtração), \* (multiplicação), / (divisão)

Bottom of Form

  Alternativa correta



+ (soma), - (subtração), x (multiplicação), / (divisão), ^ (exponenciação), % (módulo ou resto da divisão)

  Alternativa correta



+ (soma), - (subtração), \* (multiplicação), / (divisão), \*\* (exponenciação), % (módulo ou resto da divisão)

Top of Form

* Alternativa correta! Com estes operadores nós conseguimos realizar as operações matemáticas com PHP

Bottom of Form

Parabéns, você acertou!