**01 Introdução**

**Transcrição**

Olá pessoal, sejam bem vindos à Alura! Meu nome é Vinicius Dias e vou guiá-los nesse treinamento de PHP - Avançando com a Linguagem.

Ao longo do treinamento, aprenderemos a trabalhar com listas de dados e informações com tipos complexos, trabalharemos com funções e separaremos nosso programa em arquivos diferentes, manipularemos listas (seja retornando valores e substituindo-os na lista original ou manipulando por referência) e conheceremos os tipos de erro que o PHP oferece.

lém disso, discutiremos o papel do PHP na Web, qual o seu comportamento e o que ele precisa para funcionar nesse ambiente. Ao final, teremos acesso a uma lista de correntistas, criaremos diversas contas e faremos uma série de operações nelas, como depósitos, saques, remoções e alterações de valores.

Claro que os estudos não acabam por aqui, mas esse será um ótimo início. Espero que você tire proveito e ganhe ainda mais interesse em aprender essa linguagem deprogramação. Bons estudos e até o próximo vídeo!

# 02 Recapitulando

## Transcrição

Nesse vídeo vamos recapitular rapidamente os conteúdos que já aprendemos com PHP, de modo que possamos continuar o nosso aprendizado. Primeiramente, precisamos ter o PHP instalado em nossa máquina. Essa instalação foi feita na primeira parte do curso, PHP - Primeiros Passos.

Além dessa instalação, escrevemos alguns códigos que basicamente sintetizam tudo que aprendemos durante o treinamento. Executamos O PHP diretamente do terminal, fazendo alguns testes simples, e depois criamos os nossos arquivos PHP, nos quais trabalharemos variáveis, operações matemáticas, tipos de dados, estruturas de decisão (if e else) e estruturas de repetição (while e for).

Em um desafio, construímos um programa que exibe todos os números ímpares entre 1 e 100. Para isso, criamos um laço de repetição - uma parte do código que exibe ou não, dependendo de uma condição, um número que vai de 1 a 100. No caso, a condição era de que o número só seria exibido se fosse ímpar - ou seja, se o resto de sua divisão (cujo operador no PHP é %) fosse igual a 1.

O desafio seguinte era a exibição de uma tabuada. Nesse código, selecionávamos em uma variável o valor do multiplicador do qual exibiríamos a tabuada. Mais uma vez criamos um laço de repetições, pois queríamos realizar a multiplicação do número 1 até o 10, exibindo o resultado na tela. Nesse processo, utilizamos operações matemáticas, variáveis, a concatenação de strings, quebra d linha, etc.

Por último, fizemos um programa que calcula o IMC, com base nas variáveis $peso e $altura, e exibe se a pessoa está abaixo, acima ou dentro do peso recomendado.

Basicamente, aprendemos o que é o PHP, como instalá-lo e como ele funciona. Aqui, continuaremos esse aprendizado - claro, partindo do princípio de que todos os conceitos do curso anterior estão bem fixados. Caso tenha alguma dúvida, busque o fórum da Alura, onde temos uma comunidade de alunos e instrutores sempre dispostos a ajudar. Então, poderemos partir para novos conteúdos.

Nos exercícios de repetição, exibidos uma listagem de números para a lista de chamada de uma turma. Já quando aprendemos sobre variáveis, trabalhamos com idades. E se quiséssemos, agora, armazenar outros dados dos alunos dessa turma em uma variável - por exemplo, a idade? É possível armazenar mais de um dado em uma variável? Conversaremos sobre isso no próximo vídeo!

# 03 Lista de dados

## Transcrição

Terminamos o vídeo anterior comentando sobre a possibilidade de trabalharmos com vários dados em uma mesma variável. Por exemplo, se quisermos armazenar as idades de uma turma inteira em uma úncia variável.

Para explorarmos esse assunto, abriremos no nosso projeto uma nova pasta "avancando", na qual armazenaremos os novos arquivos - dentre eles o listas.php, que criaremos agora. Aqui, poderíamos ter diversas variáveis diferentes representando idades:

<?php

$idade1 = 21;

$idade2 = 23;

$idade3 = 19;

$idade4 = 21;

$idade5 = 30;

$idade6 = 25;

Dessa forma é possível armazenar todos os valores que precisamos. Entretanto, como saberemos quantas variáveis contendo valores de idades nós temos? Além disso, se quisermos exibí-las, não será trivial passarmos por todas essas variáveis - quantos mais idades, mais linhas, mais espaço de memória, mais nomes diferentes de variáveis (o que pode acarretar em erros), etc.

Seria mais interessante termos uma variável $idades (ou $idadeList) na qual pudéssemos passar uma lista de valores:

<?php

$idadeList = 21, 23, 19, 25, 30, 41, 18;

Em programação, um conjunto de dados dessa forma, exposto como uma lista, é chamado de **array** (ou "vetor"). Um array nada mais é do que uma sequência de informações, normalmente (mas não sempre) do mesmo tipo - por exemplo, uma lista de inteiros. Existem duas formas de informarmos o PHP de que esses valores devem ser lidos como um array. A primeira delas é com array():

<?php

$idadeList = array(21, 23, 19, 25, 30, 41, 18);

Nas versões mais recentes do PHP, não precisamos mais escrever a palavra array, bastando utilizarmos colchetes para rodear o nosso vetor:

<?php

$idadeList = [21, 23, 19, 25, 30, 41, 18];

Mas e se quisermos acessar os itens dessa lista? A principal característica de um vetor é que, além de seus valores, ele possui chaves/índices para acessá-los. Por padrão, se não definirmos esses índices, eles começam com o número 0. Por exemplo, se quisermos a $primeiraIdade, podemos passar o índice 0 entre colchetes:

<?php

$idadeList = [21, 23, 19, 25, 30, 41, 18];

$primeiraIdade = $idadeList[0];

Assim, estamos informando que queremos pegar o item da posição 0 desse array, que é o primeiro item. Se quiséssemos o terceiro item, pediríamos a segunda posição ($idadeList[2]), e assim por diante. Assim, poderemos exibir na tela uma variável que receba o valor de uma das posições desse array:

<?php

$idadeList = [21, 23, 19, 25, 30, 41, 18];

$umaIdade = $idadeList[2];

echo "$umaIdade";

Executando listas.php, teremos como retorno "19", que é o item da posição 2 de $idadeList. Levando em consideração tudo o que já aprendemos, como é que poderemos exibir todas as idades dessa lista?

Conversaremos sobre isso no próximo vídeo!

# 04 Arrays

**Arrays em PHP são um assunto muito importante, então utilizaremos bastante neste treinamento, mas já adianto que nem sempre é a melhor solução.**

**Tendo entendido para que servem arrays e como utilizá-los, qual a sintaxe correta para criar um array com os nomes dos 3 últimos filmes da Marvel?**

Top of Form

* Alternativa correta



$filmes = new array('Avengers: Endgame', 'Captain Marvel', 'Homem-Formiga e a Vespa');

Bottom of Form

  Alternativa correta



$filmes = ['Avengers: Endgame', 'Captain Marvel', 'Homem-Formiga e a Vespa'];

 Alternativa correta! Desta forma conseguimos criar uma lista (array) contendo os 3 últimos filmes lançados pela Marvel.

 Alternativa correta



$filmes = list('Avengers: Endgame', 'Captain Marvel', 'Homem-Formiga e a Vespa');

  Alternativa correta



$filmes = array('Avengers: Endgame', 'Captain Marvel', 'Homem-Formiga e a Vespa');

Alternativa correta! Desta forma conseguimos criar uma lista (array) contendo os 3 últimos filmes lançados pela Marvel.