PHP - Arrays, Strings, Função e Web

- Arrays são dinâmicos
- Array associativo => Da pra mudar o índice do array para qual voce quer, exemplo:

```
$conta1 = [
    'titular' => 'Julio',
    'idade' => 22
];
$contas = [12421 => $conta1];
//Dessa forma, a $conta1 está no índice 12421 de $contas
```

- Em arrays simples, com índice numérico, basta usar [] para adicionar um elemento no final (por exemplo, $lista[] = 12;)
  - nesse caso o PHP automaticamente incrementa o índice
- Chaves de arrays associativos só podem ser
  - Inteiro
  - String
  - Se você tentar definir uma chave com outro tipo que não seja inteiro ou string, o PHP tentará converter para um desses tipos.
- a associação entre chave e valor é feito com =>
- um array associativo também é chamado de mapa ou dicionário
- O array no PHP não é um array de verdade, como conhecemos nas demais linguagens. Internamente, os arrays são armazenados como HashTables (tabelas de espalhamento), e por isso eles são tão poderosos. Têm tamanho dinâmico, podem ter strings como seus índices e podem ser manipulados de diversas formas.
- Uma função retorna um valor, ou seja, a chamada da função representa um valor após sua execução. Uma subrotina apenas executa um código isolado.

- include e require => mesma coisa, mas o require para o programa se não encontra o arquivo.
- require_once => não permite que o php tente incluir o mesmo arquivo 2 vezes.

```
include 'funcoes.php';
include ('funcoes.php');
require 'funcoes.php';
require ('funcoes.php');
```

- para acessar um valor de um array associativo dentro de string devemos omitir as aspas da chave, por exemplo "$conta[titular]"
- e alternativamente podemos usar chaves em volta do array, por exemplo: "{$conta['titular']}"
- A função strtoupper não colocaria letras com acento em maiúsculo, enquanto a mb_strtoupper consegue fazer isso sem problemas.

- A seguir, 2 formas diferentes de fazer a mesma coisa:

```
foreach ($contasCorrentes as $cpf => $conta) {
    exibeMensagem("$cpf $conta[titular] $conta[saldo]");
}
```

```
foreach ($contasCorrentes as $cpf => $conta) {
    exibeMensagem("$cpf {$conta['titular']} {$conta['saldo']}");
}
```

- List serve para passar os valores de um array para outro.
- List de 2 formas diferentes:

```
$info = array('Café', 'marrom', 'cafeína');
[$bebida, $cor, $substancia] = $info;
echo "$bebida é $cor e $substancia o faz especial.\n";
```

```
$info = array('Café', 'marrom', 'cafeína');
list($bebida, $cor, $substancia) = $info;
echo "$bebida é $cor e $substancia o faz especial.\n";
```

- unset() => remove um item de uma lista:

```
unset($contasCorrentes['123.123.123-44']);
```

-
 => quebra de linha em html
- - PHP já tem um servidor embutido, o comando para rodar o servidor é: php -S localhost:8080

- Podemos misturar código PHP com código HTML, assim podemos criar HTML dinamicamente.

- Para tal é preciso demarcar o código PHP, abrindo e fechando a tag PHP: `<? php ... ?>`
- para imprimir algum valor podemos usar `<?= $valor ?>`
- Exemplo de código PHP com HTML:

```
<dl>
    <?php foreach($contasCorrentes as $cpf => $conta) { ?>
    <dt>
        <h3><?= $conta['titular']; ?> - <?= $cpf; ?></h3>
    </dt>
    <dd>
        Saldo: <?= $conta['saldo']; ?>
    </dd>
    <?php } ?>
</dl>
```

Diferença entre == e ===

- AVISO: a explicação a seguir supõe que a função strpos() retorna null quando não encontra nada, mas na verdade retorna false.
- == compara o valor, === compara o valor e o tipo
- exemplo na verificação de e-mail, se rodarmos esse código, as duas mensagens serão exibidas!

```
public function validaEmail($email) {
    $posicao = strpos($email,@) ;
    if($posicao == 0) { //exibe uma menssagem de erro dizendo que faltou a conta//
    }
    if($posicao == null) { //exibe uma menssagem de erro dizendo que aquele campo é
especifico para email//
    }
}
```

- Isso acontece porque, na **tabela de comparação de tipos do php** o valor 0 é igual a null!
- Então, nossa validação de e-mail deve ser escrita um pouco diferente:

```
public function validaEmail($email) {
    $posicao = strpos($email,@) ;
    if($posicao === 0) { //exibe uma menssagem de erro dizendo que faltou a conta//
    }
    if($posicao === null) { //exibe uma menssagem de erro dizendo que aquele campo é
especifico para email//
    }
}
```

- Agora sim somente uma mensagem será exibida. Se a posição for zero, mostramos a mensagem dizendo que faltou a conta. Se não houver @ mostramos a mensagem dizendo que o campo é especifico para e-mail.

Dúvidas e assuntos a pesquisar

- SOLID e Design Patterns