

Faça o que eu fiz na aula

Na última aula, aprendemos sobre arrays e como podemos utilizá-los para simplificar o nosso código. Aqui temos um array que é uma lista de saldos de contas correntes em um banco:

```
$saldos = [  
    2500,  
    3000,  
    4400,  
    1000,  
    8700,  
    9000,  
];
```

Se quisermos imprimir cada um destes saldos na tela do usuário, vimos anteriormente que podemos utilizar a função `for` e uma variável de incremento para acessarmos as posições do array.

Mas existe uma maneira mais fácil, para iterarmos em um array, existe também a função `foreach`, que vem do inglês “para cada” que faz exatamente isso, nos permite tratar com cada uma das posições do vetor.

```
foreach ($saldos as $saldo) {  
    echo "<p>$saldo</p>";  
}
```

E se entrarmos no navegador de internet, vamos ver que os saldos foram impressos, seguindo a posição em que foram armazenados no array.

Para pegarmos o menor saldo do array, usamos a função `sort` para ordenar de maneira crescente os saldos

É importante lembrar que o `sort` aceita um array que será passado por referência, ou seja, não será criada uma cópia no escopo interno da função, ela será modificada na íntegra.

Então podemos imprimir o menor saldo assim:

```
echo "Menor saldo: {$saldos[0]}";
```

E acessando no navegador, conseguimos ver o menor saldo impresso na tela.

Temos um novo array, com vários nomes separados por vírgula.

```
$nomes = "Giovanni, João, Maria, Pedro";
```

Para iterarmos por ele, precisamos transformá-lo em um array para que o loop `foreach` possa iterar sobre ele, fazemos isso com a função `explode`:

```
$arrayNomes = explode(",", $nomes);  
  
foreach ($nomes as $nome) {  
    echo "<p>Olá $nome</p>";  
}
```

Agora para fazermos o processo reverso, de transformar um array em uma string, usamos a função `implode`

```
$nomesJuntos = implode(",", $arrayNomes);
```