



Para saber mais: HOF e Currying

Muitos conceitos já foram aprendidos até aqui, mas me arrisco dizer que os mais importantes são "first-class functions", "high-order functions" e funções puras.

Funções puras já sabemos o que são.

"First-class functions" também. É ter funções como um tipo, podendo passar funções por parâmetros, retornar funções, etc.

"High-order functions" são funções que operam sobre outras funções, ou seja, funções que recebem outras por parâmetro, ou que retornam novas funções. HOF são bastante comuns em programação funcional.

Agora um outro conceito, o de curried functions: Uma função "curried", ao invés de receber vários parâmetros, recebe um parâmetro por vez, retornando outra função que recebe um parâmetro também, e assim em diante.

Por exemplo:

Ao invés de:

```
function soma(int $a, int $b): int
{
    return $a + $b;
}
```

```
echo soma(2, 2); // 4
```

COPIAR CÓDIGO

Teríamos:

```
function soma(int $a): callable
{
    return function (int $b) use ($a): int {
        return $a + $b;
    };
}

echo soma(2)(2); // 4
```

[COPIAR CÓDIGO](#)

Através de curried functions podemos aplicar conceitos como aplicações parciais que vimos neste vídeo. O código parece bem mais complexo, mas o PHP 7.4 trouxe uma novidade que facilita bastante. Vamos ver ainda neste capítulo.