

## 課題 ユーザ定義関数⑨(ジェネレーター)

フォルダ名 : Q20

ファイル名 : index.php, function.php

### ファイル構成

```
└─ Q20
   └─ index.php (メインプログラム)
      └─ function.php (checkSales 関数)
```

配布した index.php, function.php をもとに、演習を行いなさい。

### index.php (未完成)

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>ジェネレーター</title>
</head>
<?php
//売上データ
$data = [
    //商品名          単価          販売数量
    ['name' => 'りんご', 'price' => 150, 'num' => 8],
    ['name' => 'ブドウ', 'price' => 690, 'num' => 2],
    ['name' => 'キウイ', 'price' => 98, 'num' => 5],
    ['name' => 'すいか', 'price' => 500, 'num' => 2],
    ['name' => 'バナナ', 'price' => 198, 'num' => 6]
];
//基準額
$base = 500;
?>
<body>
  <?php
  ?>
</body>
</html>
```

function.php (未完成)

```
<?php
```

実行結果

(基準額として「500」を指定して実行した場合)

## 売上金額500円超の品目

- 1 : りんご
- 2 : ブドウ
- 3 : すいか
- 4 : バナナ

(基準額として「1000」を指定して実行した場合)

## 売上金額1000円超の品目

- 1 : りんご
- 2 : ブドウ
- 3 : バナナ

処理手順<index.php>

1. 基準額とともに<h2>タグで見出しを表示
2. 売上データ二次元配列と基準額を引数に指定して checkSales 関数を呼び出す
3. checkSales 関数の戻り値より、商品名を連番とともに表示する。
4. 2、3.を繰り返す

※checkSales 関数の戻り値がない場合(yield での戻りではない場合)は繰り返しを終了する

<ユーザ定義関数の仕様>

売上チェック関数	
関数名	checkSales
引数	売上データ(二次元配列)
	基準額(整数型)
戻り値	基準額を超えた売上データ 1 件分(一次元配列)
処理内容	① 引数で受け取った売上データ二次元配列より 1 件(1 行分)取り出す ② 取り出した売上データの売上金額(単価×販売数量)が、引数で受け取った基準額以上であれば、処理対象としているデータ(配列 1 行分)を yield で返す。 ③ ①②を繰り返す