第２回　配列・連想配列

課題２－１

★第1段階（ファイル名：kadai02\_1.php）

配列 $fruits を作成し、下記の実行結果が出るようにしましょう。

テーブル

自動的に生成された説明

**■実行結果（sample02\_1.php　第1段階）**

**※赤字の部分は配列の中身が表示されるようにしましょう。（文字列「苺」の直打ちはNG）**

配列fruitsの3番目の値は「苺」です。

★第2段階（ファイル名：kadai 02\_1.php）

配列 $fruits に「パイナップル」を追加しましょう。配列の表示にはprint\_rを用いること。

**■実行結果（sample02\_1.php　第２段階）**

**※赤字の部分が今回追加された実行結果です。**

配列fruitsの3番目の値は「苺」です。

Array

(

[0] => りんご

[1] => バナナ

[2] => 苺

[3] => ぶどう

[4] => キウイ

[5] => パイナップル

)

★第３段階（ファイル名：kadai02\_1.php）

配列 $fruits の1番目を「スイカ」に上書きし、内容の確認表示をしましょう。

※課題２－１　第2段階　のコードはそのまま残しておくこと。

その後に、今回の第3段階のコードを追記。

最終的には、配列1番目の値の上書き前と、上書き後の実行結果が正しく表示されていること。

※配列の要素の始まる番号（インデックス）に注意してください。

1番目＝１ではありません。

**■実行結果（kadai02\_1.php　第３段階）**

**※赤字の部分が今回追加された実行結果です。**

配列fruitsの3番目の値は「苺」です。

Array

(

[0] => りんご

[1] => バナナ

[2] => 苺

[3] => ぶどう

[4] => キウイ

[5] => パイナップル

)

配列1番目を上書き

Array

(

[0] => スイカ

[1] => バナナ

[2] => 苺

[3] => ぶどう

[4] => キウイ

[5] => パイナップル

)

課題２－２

★第1段階（ファイル名：kadai02\_2.php）

課題２－１で作成した配列$fruitsは1次元配列でした。

ここでは、$fruits の他に、一次元配列の$vegetableを作成しましょう。



配列 $fruits と$vegetableをvar\_dump関数用いて表示してください。

**実行結果（kadai02\_2.php　第１段階）**

array(5) {

[0]=>

string(9) "りんご"

[1]=>

string(9) "バナナ"

[2]=>

string(3) "苺"

[3]=>

string(9) "ぶどう"

[4]=>

string(9) "キウイ"

}

array(5) {

[0]=>

string(12) "キャベツ"

[1]=>

string(6) "人参"

[2]=>

string(12) "ピーマン"

[3]=>

string(6) "茄子"

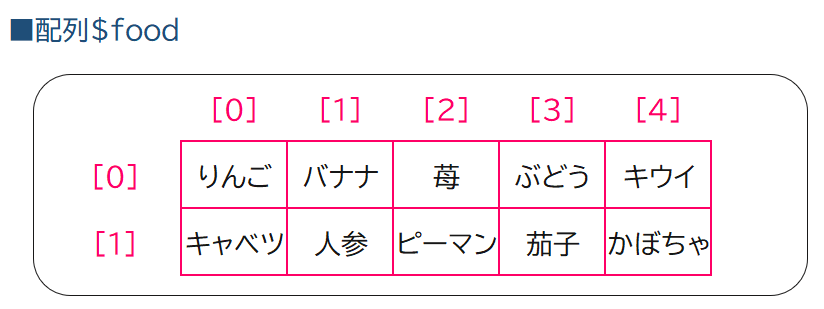
[4]=>

string(12) "かぼちゃ"

}

★第2段階（ファイル名：kadai02\_2.php）

配列 $fruitsと $vegetableを多次元配列 $foodに格納し、2次元配列の内容をブラウザで表示してください。



**実行結果（kadai02\_2.php　第２段階）**

【※この例では、第1段階の実行結果部分は省略】

array(2) {

[0]=>

array(5) {

[0]=>

string(9) "りんご"

[1]=>

string(9) "バナナ"

[2]=>

string(3) "苺"

[3]=>

string(9) "ぶどう"

[4]=>

string(9) "キウイ"

}

[1]=>

array(5) {

[0]=>

string(12) "キャベツ"

[1]=>

string(6) "人参"

[2]=>

string(12) "ピーマン"

[3]=>

string(6) "茄子"

[4]=>

string(12) "かぼちゃ"

}

}

★第3段階（ファイル名：kadai02\_2.php）

多次元配列2行3列目の内容を表示しましょう。

**実行結果（kadai02\_2.php　第３段階）**

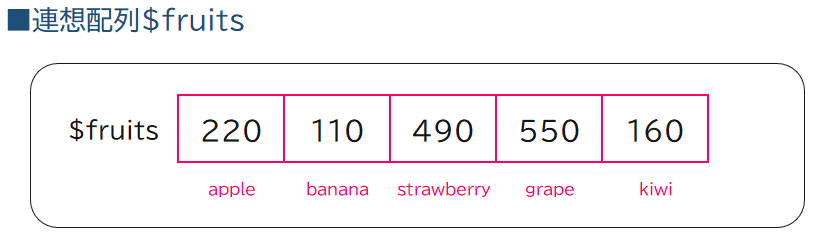
【※この実行結果例では、第２段階までの実行結果部分は省略】

$food 2行3列目は、ピーマンです。

課題２－３

★第1段階（kadai02\_3.php）

下図のような連想配列を宣言し、配列の内容を表示（var\_dump）しましょう。



実行結果のように、きちんとキーと価格（価格はint型）が表示されていることを確認してください。

**実行結果（kadai02\_3.php　第1段階）**

array(5) {

["apple"]=>

int(220)

["banana"]=>

int(110)

["strawberry"]=>

int(490)

["grape"]=>

int(550)

["kiwi"]=>

int(160)

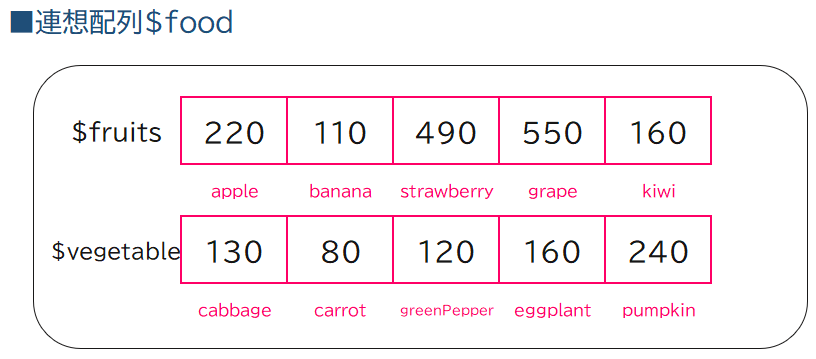
}

★第2段階（kadai02\_3.php）

下図のような連想配列を作成しましょう。

多次元配列の構文と連想配列の構文を組み合わせ、時間が掛かってもいいので、作成してみてください。

結果はvar\_dumpで表示しましょう。



実行結果のように表示されることを確認してください。

**実行結果（kadai02\_3.php　第２段階）**

foodの内容を表示

array(2) {

["fruits"]=>

array(5) {

["apple"]=>

int(220)

["banana"]=>

int(110)

["strawberry"]=>

int(490)

["grape"]=>

int(550)

["kiwi"]=>

int(160)

}

["vegetable"]=>

array(5) {

["cabbage"]=>

int(130)

["carrot"]=>

int(80)

["greenPepper"]=>

int(120)

["eggplant"]=>

int(160)

["pumpkin"]=>

int(240)

}

}

★第3段階（kadai02\_3.php）

配列$fruitsのキーと値を、foreachを用いて、表示（echo）してみましょう。

**実行結果（kadai02\_3.php　第３段階）**

【第3段階】配列$fruitsの中身をforeachで順番に表示します。  
apple：220円  
banana：110円  
strawberry：490円  
grape：550円  
kiwi：160円  
------------------------------------

★第4段階（kadai02\_3.php）

配列$foodのキーと値を、種別ごとにforeachを用いて、表示（echo）してみましょう。

ヒント：2重ループです。

まず、種別（fruitsとvegetable）でループを回しましょう。

続いて、キーと値、すなわち商品名と価格が表示されるように中のループを作成しましょう。

　　　 表示形式は実行結果のように、「商品名」や「円」など分かり易いように文言を追加してください。

※実行結果は次のページです。

**実行結果（kadai02\_3.php　第4段階）**

【第4段階】配列$foodの中身をforeachで種別ごとに、順番に表示します。

■種別：fruits

　商品名：apple　／　　価格：220円

　商品名：banana　／　　価格：110円

　商品名：strawberry　／　　価格：490円

　商品名：grape　／　　価格：550円

　商品名：kiwi　／　　価格：160円

------------------------------------

■種別：vegetable

　商品名：cabbage　／　　価格：130円

　商品名：carrot　／　　価格：80円

　商品名：greenPepper　／　　価格：120円

　商品名：eggplant　／　　価格：160円

　商品名：pumpkin　／　　価格：240円

------------------------------------