
PHP

14 時間目

foreach文は
配列に格納した値を全て
順番に処理します

foreach文のイメージ

東京、千葉、埼玉、神奈川を
配列に格納した順番に表示したい場合

index.php(PHPファイル)

```
<?php  
  
$a = array("東京","千葉","埼玉","神奈川");  
echo $a[0];  
echo $a[1];  
echo $a[2];  
echo $a[3];  
  
?>
```

普通に、一つひとつ記述すると
面倒で時間がかかる。

index.php(PHPファイル)

```
<?php  
  
$a = array("東京","千葉","埼玉","神奈川");  
foreach($a as $value){  
    echo $value;  
}  
  
?>
```

foreach文だと3行だけで良く
簡単に記述できる！

foreach文の書き方

解説

下記は、foreach文の書き方です。
いろいろと解説がありますが、ここでは、「こんなモノがあるんだ・・・」という程度の理解で問題ありません。

```
$○ = array(値0,値1,値2・・・);
```

通常の配列と同じ書き方

```
foreach($○ as $×){
```

foreach文の宣言

\$○の配列で格納した全ての値は、
\$×の変数で処理できるという意味

```
処理内容;}
```

処理内容

補足

\$○、\$×の変数名は、どれも任意の名前（好きな名前）を付ける事ができますが、教科書等では、値の\$X部分を『\$value』としているのが一般的です。

index.php(PHPファイル)

<?php

\$a=array("山田","佐藤","鈴木");foreach(**\$a** as **\$value**){echo **\$value**;

}

?>

← → ↺ 🏠

山田佐藤鈴木

教科書的には、\$valueとすることが多いが、
\$valueの部分は、任意の名前（好きな名前）でOK。
\$bでも、\$abcでも、\$xyzでもOK。

index.php(PHPファイル)

<?php

\$b=array("山田","佐藤","鈴木");foreach(**\$b** as **\$z**){echo **\$z**;

}

?>

前ページの\$valueの部分を\$zに変更しても、
処理される内容は変わらない。

< > ↺ 🏠

山田佐藤鈴木

index.php(PHPファイル)

```
<?php  
  
$a=array("山田","佐藤","鈴木");  
foreach($a as $value){  
    echo $value  
    echo "<br>";  
}  
  
?>
```

この場所に、
を入れると、
山田、
で改行、佐藤、
で改行、鈴木、
で改行
という順番で処理される。

← → ↺ 🏠

山田
佐藤
鈴木

index.php(PHPファイル)

<?php

\$a=array(**100,200,300**);foreach(\$a as \$value){
echo \$value;
}

?>

数字の場合は、クォテーションで
囲む必要はない。

< > ↺ 🏠

100200300

index.php(PHPファイル)

<?php

```
$a=array("sky","sea","lake");  
foreach($a as $value){  
    echo $value;  
}
```

?>

アルファベットは、文字列なので
クォテーションで囲む必要あり。

< > ↺ 🏠

skysealake

index.php(PHPファイル)

<?php

\$a=5;**\$b=3;****\$c=7;**\$a=array(**\$a,\$b,\$c**);foreach(\$a as \$value){
echo \$value;
}

?>

変数を、配列の中に入れることも出来る。

< > ↺ 🏠

537

keyを表示することも可能

解説

foreachでは、配列同様に、value(値) とkey (添字) の両方を処理することが出来ます。

foreach文では、配列同様に、
value(値) とkey (添字) の両方を
処理することが出来ます。

keyを表示することも可能

解説

下記は、foreach文でkeyを表示する書き方です。
いろいろと解説がありますが、ここでは、「こんなモノがあるんだ・・・」という程度の理解で問題ありません。

```
$○ = array(値0,値1,値2・・・);
```

通常の配列と同じ書き方

```
foreach($○ as $△ => $X){
```

foreach文の宣言

```
echo $△.$X;}
```

記述する処理は何でも良いが、
今回は、キーと値をそれぞれ表示
するという処理を記述。

\$○の配列で格納した全ての値は、\$△という
key(添字)の箱に入り、\$Xの変数で処理でき
るという意味。

補足

\$○、\$△、\$xの変数名は、どれも任意の名前（好きな名前）を付ける事ができますが、教科書等では、
keyの\$△部分を『\$key』、値の\$X部分を『\$value』としているのが一般的です。

index.php(PHPファイル)

```
<?php  
$a=array("山田","佐藤","鈴木");  
foreach($a as $b => $c){  
    echo $b.$c;  
}  
?>
```

Keyと値をそれぞれ表示。

Key部分を、\$b、
値部分を、\$cとしている。

0山田1佐藤2鈴木

Keyと値がそれぞれ表示された。

index.php(PHPファイル)

```
<?php  
  
$a=array("山田","佐藤","鈴木");  
foreach($a as $b => $c){  
echo $b."の箱は".$c."<br>";  
}  
  
?>
```

。（ドット/ピリオド）は、文字列や変数を連結する役割。

また、最後に、
を記述することで、
「key “の箱は” 値」を1セットで改行で区切って
表示している。

**0の箱は山田
1の箱は佐藤
2の箱は鈴木**