
PHP

15 時間目

foreach文では、
連想配列を使用し、
各keyに名前を付ける事も可能です。

index.php(PHPファイル)

<?php

```
$a=array( "【A】" => "赤",  
         "【B】" => "青",  
         "【C】" => "緑");
```

```
foreach($a as $key => $value){  
    echo $key.$value;  
}
```

?>

【A】赤【B】青【C】緑

Keyと値がそれぞれ表示された。

連想配列でkeyに名前を付ける。

index.php(PHPファイル)

```
<?php  
  
$a=array( "日本"=>"東京",  
          "韓国"=>"ソウル",  
          "中国"=>"北京");  
  
foreach($a as $key => $value){  
    echo $key.$value;  
}  
  
?>
```

【日本】東京【韓国】ソウル【中国】北京

Keyと値がそれぞれ表示された。

連想配列でkeyに名前を付ける。

index.php(PHPファイル)

```
<?php  
  
$a=array( "山田"=>"20歳",  
          "佐藤"=>"21歳",  
          "鈴木"=>"19歳");  
foreach($a as $key =>$value){  
echo $key.":"".$value."<br>"  
}  
  
?>
```

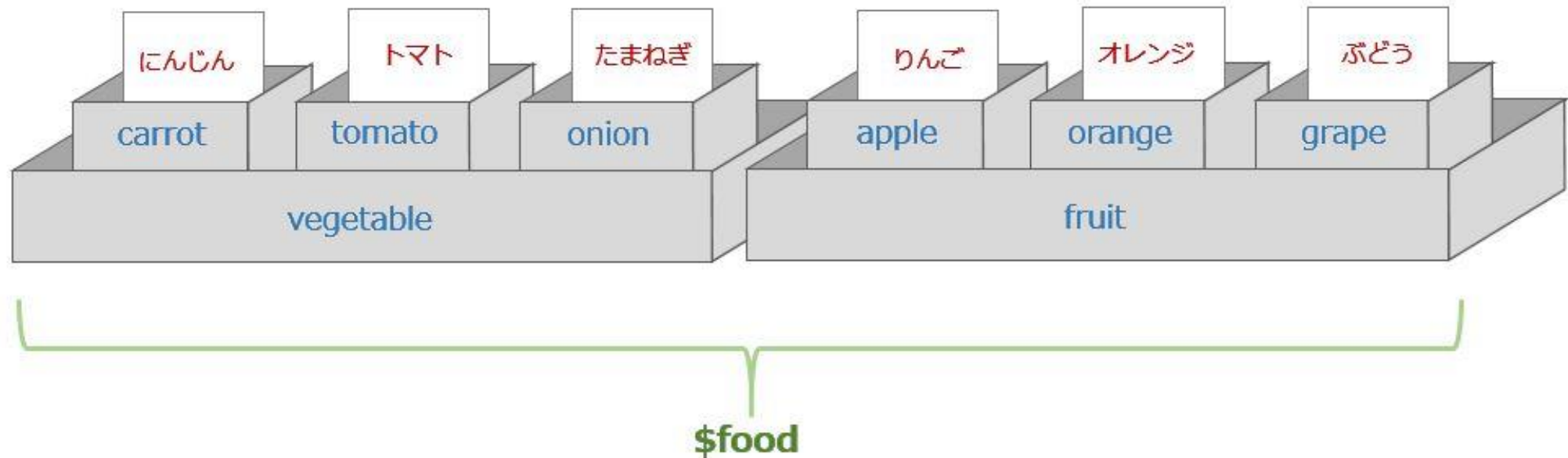
":"という文字列を、\$keyと\$valueの間に
入れ、\$key : \$value を連結する。
さらに、
を最後に入れる事で、
それぞれのkeyと値ごとに改行する。

山田:20歳
佐藤:21歳
鈴木:19歳

foreach文では、
多次元連想配列(ネスト) と
各keyに名前を付ける事も可能です。

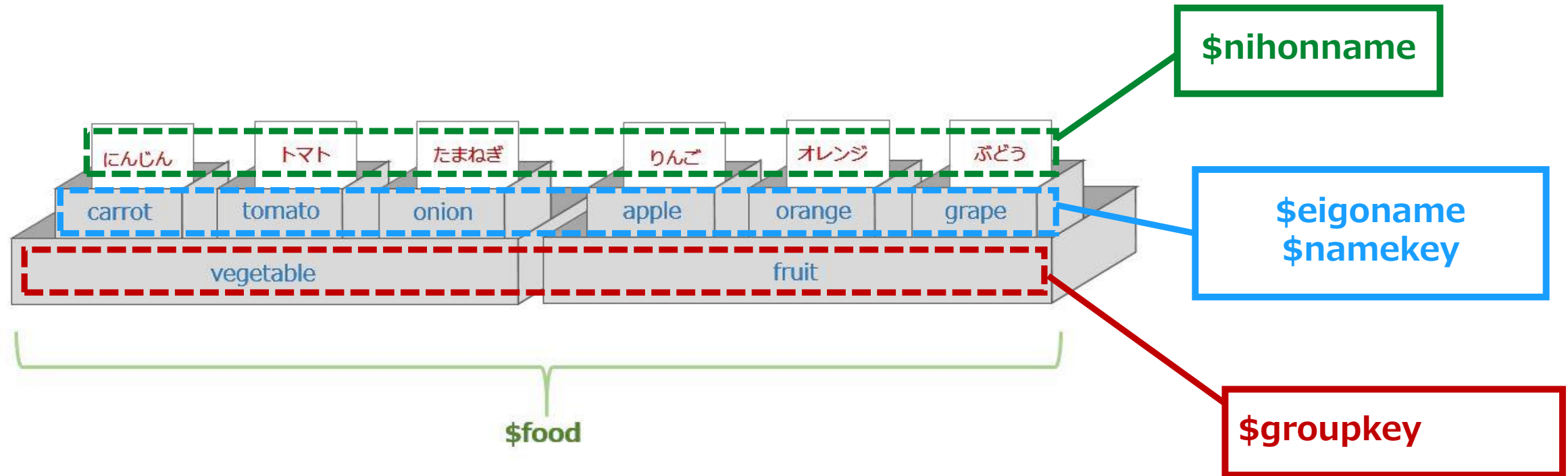
foreach文の多次元連想配列(ネスト)とkeyによる名前付け

下記のような多次元連想配列を
foreach文で処理をする



foreach文の多次元連想配列(ネスト)とkeyによる名前付け

foreach文で処理をする為に
keyと値部分に、右の名前を付ける



foreach文の多次元連想配列(ネスト)とkeyによる名前付け

index.php(PHPファイル)

```
<?php
$food = array(
    "vegetable" => array("carrot" => "にんじん",
                        "tomato" => "トマト",
                        "onion" => "たまねぎ"),
    "fruit" => array("apple" => "りんご",
                    "orange" => "オレンジ",
                    "grape" => "ぶどう")
);

foreach( $food as $groupkey => $namekey ){
    echo "■".$groupkey."<br>";
    foreach( $namekey as $eigename => $nihonname ){
        echo $eigename." : ".$nihonname."<br>";
    }
    echo "<br>";
}

?>
```

← → ↺ 🏠

```
■ vegetable
carrot : にんじん
tomato : トマト
onion : たまねぎ

■ fruit
apple : りんご
orange : オレンジ
grape : ぶどう
```

foreach文では、
格納した値をソート（値の大きい順/小さい順等）で
処理することもできる。

ソートなしの通常の場合

index.php(PHPファイル)

```
<?php  
  
$a=array("d", "c", "b", "a");  
  
foreach($a as $key => $value){  
    echo $value."<br>";  
}  
  
?>
```

d
c
b
a

格納した順番で表示される

ソートありの場合

index.php(PHPファイル)

```
<?php  
  
$a=array("d", "c", "b", "a");  
sort($a);  
  
foreach($a as $key => $value){  
    echo $value."<br>";}  
  
?>
```

a
b
c
d

アルファベット順に表示された！

ソートの種類

ソートにはいくつかの種類があります

関数の書き方	並べ替えの基準	意味
sort(\$変数)	値（バリュー）	配列\$変数の各値を昇順（小さい順に並び変わった状態）に並べ替える。 前ページで紹介したモノ
rsort(\$変数)	値（バリュー）	配列\$変数の各値を降順（大きい順に並び変わった状態）に並べ替える。
asort(\$変数)	値（バリュー）	配列\$変数の各値を昇順（小さい順に並び変わった状態）に並べ替える。 連想配列において各配列のキーと値との関係を維持しつつ配列をソートします。 主に実際の値の並び方が重要である連想配列をソートするために使われます。
arsort(\$変数)	値（バリュー）	配列\$変数の各値を降順（大きい順に並び変わった状態）に並べ替える。 連想配列において各配列のキーと値との関係を維持しつつ配列をソートします。 主に実際の値の並び方が重要である連想配列をソートするために使われます。
ksort(\$変数)	添字（キー/key）	配列\$変数の各値を昇順（小さい順に並び変わった状態）に並べ替える。 連想配列において各配列のキーと値との関係を維持しつつ配列をソートします。 主に実際の値の並び方が重要である連想配列をソートするために使われます。
krsort(\$変数)	添字（キー/key）	配列\$変数の各値を降順（大きい順に並び変わった状態）に並べ替える。 連想配列において各配列のキーと値との関係を維持しつつ配列をソートします。 主に実際の値の並び方が重要である連想配列をソートするために使われます。

rsort(\$変数)

配列\$aの各値を降順（大きい順に並び変わった状態）に並べ替える

index.php(PHPファイル)

```
<?php  
  
$a=array("d", "c", "b", "a");  
rsort($a);  
  
foreach($a as $key => $value){  
    echo $value."<br>";  
}  
  
?>
```

d
c
b
a

アルファベットの大きい順(z→a)
に表示された！

asort(\$変数)

連想配列で**各配列のキーと値を維持**した状態で、
配列\$aの各**値を昇順**（**小さい順**に並び変わった状態）に並べ替える

index.php(PHPファイル)

```
<?php  
  
$a=array("red"=>"d",  
         "blue"=>"c",  
         "white"=>"b",  
         "black"=>"a"  
        );  
asort($a);  
  
foreach($a as $key => $value){  
    echo $key.":".$value."<br>";  
}  
  
?>
```

← → ↺ 🏠

black:**a**
white:**b**
blue:**c**
red:**d**

値が昇順（小さい順=アルファベット順）に並んだ。

arsort(\$変数)

連想配列で**各配列のキーと値を維持**した状態で、
配列\$aの各**値を降順**（**大きい順**に並び変わった状態）に並べ替える

index.php(PHPファイル)

```
<?php  
  
$a=array("red"=>"d",  
         "blue"=>"c",  
         "white"=>"b",  
         "black"=>"a"  
        );  
arsort($a);  
  
foreach($a as $key => $value){  
    echo $key.":".$value."<br>";  
}  
  
?>
```

← → ↺ 🏠

red:**d**
blue:**c**
white:**b**
black:**a**

値が昇順（大きい順=z→aの順）
に並んだ。

ksort(\$変数)

配列\$aの**各key**を**昇順**（**小さい順**に並び変わった状態）に並べ替える

index.php(PHPファイル)

```
<?php  
  
$a=array("d", "c", "b", "a");  
ksort($a);  
  
foreach($a as $key => $value){  
    echo $value."<br>";  
}  
  
?>
```

black:a
blue:c
red:d
white:b

keyが昇順（小さい順）に表示された

ksort(\$変数)

配列\$aの**各key**を**降順**（**大きい順**に並び変わった状態）に並べ替える

index.php(PHPファイル)

```
<?php  
  
$a=array("d", "c", "b", "a");  
ksort($a);  
  
foreach($a as $key => $value){  
    echo $value."<br>";  
}  
  
?>
```

white:b
red:d
blue:c
black:a

keyが降順（大きい順）に表示された