

## 4. データのグループ：配列の操作

1. 配列の定義として正しいものを選択してください。(1つ選択)

- A. 配列の作成時に各要素のキーを指定しなかった場合、キーは1から始まる
- B. 配列\$numbersと変数\$numbersは同時に使用できる
- C. 配列の各要素のキーには文字列を利用することはできない
- D. 配列の各要素の値には文字列を使用することができる

2. 次のコードを実行した結果として正しいものを選択してください。(1つ選択)

```
$numbers = array(10, 20, 30);  
$numbers[] = 40;  
$numbers[] = 50;  
print $numbers[4];
```

- A. 10
- B. 20
- C. 30
- D. 40
- E. 50
- F. パースエラーになる

3. 次のコードは2つの配列を作成しています。コードが正しく実行されるとき、空欄①、②に入る演算子の組み合わせとして適切なものを選択してください。(1つ選択)

```
$colors['sea'] ① 'blue';  
$colors['leaf'] ① 'green';  
$colors['night'] ① 'black';  
  
$scores = array( '国語' ② 80,  
                 '数学' ② 90,  
                 '英語' ② 75);
```

- A. ① = ② =
- B. ① == ② =
- C. ① = ② ==
- D. ① = ② =>
- E. ① => ② =>
- F. ① == ② ==

4. PHPの配列の説明として誤っているものを選択してください。(1つ選択)

- A. 数値以外のキーを指定する連想配列では、配列に要素を追加した場合の順序は保持されないため、foreach文で配列から要素を取り出す順番は不定である
- B. 添字を省略して要素を追加することができる
- C. 配列をコピーした場合、参照数が増えるだけで実際にデータはコピーされないため、要素数が多い配列をコピーしてもメモリの消費量はあまり増えない
- D. 配列に要素を追加する場合は、最後の要素の後ろに加えるだけでなく、要素の途中に挿入することもできる

5. 次のコードを実行した結果として正しいものを選択してください。(1つ選択)

```
$colors[0] = 'red';  
$colors[1] = 'yellow';  
$colors[3] = 'orange';  
$colors[2] = 'blue';  
  
foreach ($colors as $color) {  
    print "$color ";  
}
```

- A. red yellow blue orange
- B. red yellow orange blue
- C. 0 1 3 2
- D. 0 1 2 3

6. 次のコードを実行した結果として正しいものを選択してください。(1つ選択)

```
$colors['sea'] = 'blue';  
$colors['leaf'] = 'green';  
$colors['night'] = 'black';  
$colors['sun'] = 'red';  
  
asort($colors);  
foreach($colors as $key => $value) {  
    print " $key \t: $value\n <br>";  
}
```

- A.    0    : black  
      1    : blue  
      2    : green  
      3    : red
- B.    leaf : black  
      night : blue  
      sea   : green  
      sun   : red
- C.    leaf : green  
      night : black  
      sea   : blue  
      sun   : red
- D.    night : black  
      sea   : blue  
      leaf   : green  
      sun   : red

7. 次のコードのforeach文をfor文で書き直した場合、同じ結果になるものを選択してください。。(1つ選択)

```
$fruits = array(1 => 'apple',  
               2 => 'banana',  
               3 => 'lemon',  
               4 => 'orange');  
foreach ($fruits as $key => $value) {  
    print "{$key} {$value}";  
}
```

- A. 

```
$n = count($fruits);  
for($i = 0; $i <= $n ; $i++){  
    print "{$i} {$fruits[$i]}";  
}
```
- B. 

```
$n = count($fruits);  
for($i = 1; $i <= $n ; $i++){  
    print "{$i} {$fruits[$i]}";  
}
```
- C. 

```
$n = count($fruits);  
for($i = 0; $i < $n ; $i++){  
    print "{$i} {$fruits[$i]}";  
}
```
- D. 

```
$n = count($fruits);  
for($i = 1; $i < $n ; $i++){  
    print "{$i} {$fruits[$i]}";  
}
```

8. 配列から文字列を作成する関数を選択してください。(1つ選択)

- A. unset()
- B. implode()
- C. explode()
- D. in\_array()

9. var\_dump()関数を実行すると次のように表示される2次元配列があります。文字列Tokyoを表示するための記述として正しいものを選択してください。(1つ選択)

```
array(3) {  
  [0]=>  
    array(3) {  
      ["name"]=>  
        string(6) "Yamada"  
      ["age"]=>  
        int(24)  
      ["country"]=>  
        string(5) "Osaka"  
    }  
  [1]=>  
    array(3) {  
      ["name"]=>  
        string(6) "Tanaka"  
      ["age"]=>  
        int(32)  
      ["country"]=>  
        string(5) "Tokyo"  
    }  
  [2]=>  
    array(3) {  
      ["name"]=>  
        string(5) "Ikeda"  
      ["age"]=>  
        int(27)  
      ["country"]=>  
        string(5) "Kyoto"  
    }  
}
```

- A. print \$profiles['country'];
- B. print \$profiles[2][3];
- C. print \$profiles[1][2];
- D. print \$profiles[1]['country'];
- E. print \$profiles[0]['country'];
- F. print \$profiles[3]['Tokyo'];

10. var\_dump()関数を実行すると次のように表示される2次元配列があります。この2次元配列を作成するための記述として正しいものを選択してください。(2つ選択)

```
array(2) {  
  ["apple"]=>  
    array(2) {  
      ["color"]=>  
        string(3) "red"  
      ["stock"]=>  
        int(1000)  
    }  
  ["banana"]=>  
    array(2) {  
      ["color"]=>  
        string(6) "yellow"  
      ["stock"]=>  
        int(2000)  
    }  
}
```

- A. `$arrays = array(  
 array('color' => 'red', 'stock' => 1000 ),  
 array('color' => 'yellow', 'stock' => 2000 )  
);`
- B. `$arrays = array(  
 ('color' => 'red', 'stock' => 1000 ),  
 ('color' => 'yellow', 'stock' => 2000 )  
);`
- C. `$arrays[[]] = 'apple' => array('color' => 'red', 'stock' => 1000 ),  
 'banana' => array('color' => 'yellow', 'stock' => 2000 );`
- D. `$arrays['apple']['color'] = 'red';  
$arrays['apple']['stock'] = 1000;  
$arrays['banana']['color'] = 'yellow';  
$arrays['banana']['stock'] = 2000;`
- E. `$arrays = array(  
 'apple' => array('color' => 'red', 'stock' => 1000 ),  
 'banana' => array('color' => 'yellow', 'stock' => 2000 )  
);`