

# 解答・解説集

- ▶この解答・解説集は、3～8章の各章末の練習問題の解答です。
- ▶薄く糊付けしてありますが、本書より取り外して使用することもできます。

## CHAPTER 4 練習問題

P.58

### 問題 1

答え ②

解説

Webサイトを公開するには、Webサーバの所定のフォルダ以下に、サイトを構成するファイル（HTMLファイルやPHPファイルなど）を配置しますが、そのフォルダの最上位階層をドキュメントルートといいます。

以降に、①、③、④について説明します。

#### ① ローカルホスト

ローカルホスト（localhost）は、自分自身のコンピュータをあらわす名称です。WebサーバとWebクライアント（Webブラウザ）が同じコンピュータ上で動作するときには、Webブラウザから以下のアドレスを指定することにより、自身のコンピュータのWebサーバにアクセスできます。

```
http://localhost/
```

必要に応じて、ここにHTMLやPHPのファイル名を指定します。

#### ③ ディレクトリ

ディレクトリは、フォルダの別名です。LinuxなどのUNIX系OSでは、フォルダの代わりにディレクトリという言葉が使われています。

#### ④ Webコンテンツ

Webサイトを構成する中身を意味します。Webサイトは、文章、画像、音声などのさまざまなデータからできています。それらのデータそのものや、データをもとに構成されるサービスをひっくるめてWebコンテンツと呼びます。

## 問題 2

答え ① <br>

解説

Webブラウザ画面上に文章を改行して表示するには、改行させたい位置に<br>を出力します。<br>はHTML言語のタグの1つです。<BR>と書いても問題ありません。また、改行させるには、<p>タグによって段落を作成するなどの方法もあります。

## CHAPTER 5 練習問題

P.81~82

### 問題 1

答え ②

解説

変数名に使える文字は、英字、数字、アンダーバーです。ただし、数字は先頭の文字とすることはできないので、例えば、\$2hensuという名称は使えません。

以降に、①から④について説明します。

- ① … OK：英字のみで構成されていますので問題ありません。
- ② … NG：-（ハイフン）は変数名としては使えない文字です。
- ③ … OK：数字を先頭以外で使っているため問題ありません。
- ④ … OK：変数名にはアンダーバーを含めることができます。

また、変数名に英字を使う場合、大文字と小文字は区別されますので注意しましょう。例えば、次に示す変数は、すべて別々の変数として扱われます。

```
$hensu
$Hensu
$hEnSu
$HENSU
```

### 問題 2

答え ③

解説

"(ダブルクォーテーション)で囲まれた文字列の中に変数を書くと、変数に入っている値が表示されます。これを「変数の展開」といいます。リストAの8行目では、変数\$nameが展開されて③のように表示されます。

変数の展開については、問題4の解説でも説明していますので参照してください。

### 問題 3

答え ① NAME\_THEME

#### 解説

定数を定義する書き方についての問題です。定数は、ある値に別名をつけたもの、と捉えるとわかりやすいでしょう。定数は、いちど値を定義したら、その値を変えることはできません。

リストBの8行目では、const キーワードを使って定数を定義しています。9行目と15行目では、その定数が使われています。

定数を定義するには、次の2つの方法があります。

1. define 関数を使う
2. const キーワードを使う

define 関数と const キーワードの構文は、次のようになります。

```
define ( 定数名, 定数の値 );  
const 定数名 = 定数の値;
```

define 関数で指定する定数名は、'(シングルクォーテーション)または"(ダブルクォーテーション)で囲みます。

対して、const キーワードで指定する定数名は'や"で囲まないのが、この点についても注意してください。

### 問題 4

答え

```
1: <?php  
2:     header('Content-type: text/html; charset=UTF-8');  
3: ?>  
4: <html>  
5: <body>  
6: <?php  
7:     $name = 'よしこ';  
8:     echo 'こんにちは、', $name, 'さん。';  
9: ?>  
10: </body>  
11: </html>
```

#### 解説

変数に文字列を入れて、その値を Web ブラウザ画面上に表示させるプログラムです。上記の解答は、\$name に「よしこ」という名前を入れた例になります。

文字列は、'(シングルクォーテーション)または"(ダブルクォーテーション)で囲みますので、7行目は次のように記述することもできます。

```
$name = "よしこ";
```

8行目では、変数に入った値を表示する処理を行っています。Webブラウザに出力するには、echo文のほかにprint文が使えるので、次のような書き方も可能です。

```
print 'こんにちは、' . $name . 'さん。';
```

ピリオド

ピリオドは文字列演算子といい、複数の文字列をつなぐことができます。

また、変数を展開させる方法が使えるので、次のいずれかの書き方も可能です。

```
echo "こんにちは、$name さん。";
```

変数名の後ろにスペースを1つ入れています。

```
print "こんにちは、$name さん。";
```

変数名の後ろにスペースを1つ入れています。

```
print "こんにちは、{$name}さん。";
```

変数名を{ }で囲むと直後にスペースを入れずに済みます。

## CHAPTER 6 練習問題

P.94

### 問題 1

答え ④

解説

array関数で配列を作成するときにキーを省略した場合、作成された配列のキーは0からの連番になります。リストAで作成された配列の、各要素のキーは次表のようになります。

要素	キー
'タマ'	0
'ポチ'	1
'トラ'	2
'タロ'	3

したがって、'タロ'の文字列が入った要素を指すキーは3になります。

### 問題 2

答え ① 'height' または "height"

② 155

③ 'weight' または "weight"

④ 45

解説

array関数を使って連想配列を作成するプログラムです。連想配列のキーに文字列を使う場合は、キーの文字列を'(シングルクォーテーション)または"(ダブルクォーテーション)で囲みます。したがって、①と③はシングルクォーテーションの代わりにダブルクォーテーションで囲むこともできます。

array関数を使って連想配列を作る場合は、キーと値の間に=>(イコールと大なり記号)を書きます。

10行目と11行目では、作成した連想配列\$dataを使っています。連想配列を使うときのキーの文字列についても、'(シングルクォーテーション)または"(ダブルクォーテーション)で囲むのを忘れないようにしてください。

## CHAPTER 7 練習問題

P.111~112

### 問題 1

答え ①

解説

HTMLフォームデータをPHPで受け取る場合には、次の2点がポイントになります。

1. フォームの送信方法 (GET または POST)
2. フォーム項目の名称 (部品名)

フォームの送信方法には、GETメソッドとPOSTメソッドがあります。どちらのメソッドを使うかは、HTML文書内の<form>タグのmethod属性で指定します。GETメソッドの場合get、POSTメソッドの場合postを指定します。

リストAの6行目では、<form>タグを次のように記述しています。

```
<form method="post" action="form.php">
```

method=postと指定されていますので、フォームデータをPOSTメソッドで送信することがわかります。送信先は、action属性で指定されたform.phpです。

PHPプログラムでフォームデータを受け取るには、POSTメソッドの場合\$\_POSTの連想配列を、GETメソッドの場合\$\_GETの連想配列を参照することにより可能です。

どのフォーム項目かを特定するには、フォーム項目の名称(部品名)を知る必要があります。フォーム項目の部品名は、HTML文書内の各フォーム項目のname属性で指定します。

リストAでは、ラジオボタンにbtypeという部品名が付けられています。POSTメソッドですので、\$\_POSTの連想配列を参照し、フォーム項目の部品名をキーとして指定します。キーは文字列なので、'(シングルクォーテーション)または"(ダブルクォーテーション)で囲んでください。したがって、①のような書き方により、選択されたラジオボタンを取得できます。

## 問題 2

### 答え formName.php

```
1: <?php
2:     header('Content-type: text/html; charset=UTF-8');
3: ?>
4: <html>
5: <body>
6: <?php
7:     $name = $_POST['name'];
8:     $name = htmlspecialchars($name, ENT_QUOTES, 'UTF-8');
9:     echo $name, 'さん、こんにちは!';
10: ?>
11: </body>
12: </html>
```

### 解説

リストBの6行目では、<form>タグを次のように記述しています。

```
6: <form method="post" action="formName.php">
```

この<form>タグでは、method="post"と書かれていますので、HTMLフォームデータの送信方法はPOSTメソッドであることがわかります。また、action="formName.php"と書かれていますので、フォームデータを受け取るプログラムはformName.phpとなります。

リストBの7行目を見ると、テキストボックスの部品名がname属性によって"name"と指定されています。したがって、formName.phpのプログラムで、\$\_POST['name']という記述を行うと、入力されたテキストデータを受け取ることができます。

受け取った文字をそのままWebブラウザに表示することはセキュリティ上危険なため、次の8行目でhtmlspecialchars関数を呼び、HTML言語の特殊文字(小なり記号など)をエスケープさせてから表示しています。

HTML出力のエスケープ処理については、7.2.2を参照してください。

## CHAPTER 8 練習問題

P.127~128

### 問題 1

#### 答え ①

### 解説

インクリメント演算子を使った計算についての問題です。リストAの2行目のように、インクリメント演算子による加算と、他の変数への代入が組み合わせられている場合、インクリメント演算子を変数の前と後ろのどちらに置くかで、演算されるタイミングが異なります。デクリメント演算子についても同じことがいえます。

リストAを以下に示します。

```
$aaa = 1;  
$bbb = $aaa ++;
```

2行目の右辺を見ると、\$aaaの後ろに++(インクリメント演算子)が指定されています。このように\$aaaの後ろに++を書いた場合、先に\$aaaの値が\$bbbに代入され、その後で\$aaaに1が足されます。したがって、2行目の処理が終わった時点で、\$bbbは1になります。

次に、\$aaaの前に++を指定した場合を比較して見てみましょう。

```
$aaa = 1;  
$bbb = ++ $aaa;
```

このコードの2行目では、\$aaaの前に++が指定されています。このように\$aaaの前に++を書いた場合、先に\$aaaに1が足され、その後で\$aaaの値が\$bbbに代入されます。したがって、2行目の処理が終わった時点で、\$bbbは2になります。

## 問題 2

答え ④

解説

変数を使って計算する問題です。リストBの2行目が実行された直後には、\$aaaには100が、\$bbbには200が入っています。3行目では、\$aaaに\$bbbを足した結果が\$bbbに入れられます。\$aaaは計算に使うために参照されるだけですので、その値は変化しません。よって、\$aaaは100のまま変わりません。

リストBの3行目では、\$bbbが左辺と右辺の両方で使われているので、ちょっと戸惑ってしまうかもしれません。=(イコール)が出てきたときには、右辺の値を左辺に入れる処理が行われますので、まず右辺側に着目しましょう。

リストBの3行目では、\$aaa + \$bbb の計算が先に行われ、その計算結果が\$bbbに入ります。結果として、\$bbbの値は300になります。

## 問題 3

答え ① ( \$aaa + \$bbb + \$ccc ) / 3

解説

変数を使って計算する問題です。平均値を求めるには、\$aaaと\$bbbと\$cccを足した結果を3で割ります。割り算を行うための演算子は、/(スラッシュ)です。

複数の種類の演算子を使う場合は、演算子の優先度に注意する必要があります。足し算と割り算では、割り算の方が優先度が高くなっています。優先度にかかわらず特定の計算を先に行うには、その計算部分を()で囲んでください。

演算子は数多くあるので、すべての優先度を覚えておくのは大変です。優先度にかかわらず、先に計算したい部分を明示的に()で囲むようにすると、間違いを防ぐことができ、コードも見やすくなります。

## 問題 1

答え ②

## 解説

比較演算子を使ったif文の条件についての問題です。

if文のカッコ内の記述( `$val > 5` )では、`>`の比較演算子を使っていますが、これは「`$val`は5より大きいか」という条件をあらわしています。`$val`が5の場合、この条件を満たさないのでelseブロック内の処理が実行されます。

もし「`$val`は5以上か」という条件を作りたい場合は、( `$val >= 5` )のように、`>=`の演算子を使います。

## 問題 2

答え イ

## 解説

変数の値が奇数かどうかをチェックする条件を作る問題です。値が奇数かどうかをチェックするには、2で割った余りを判定することにより可能です。2で割った余りが1の場合は奇数、2で割った余りが0の場合は偶数と判断できます。割った余りを求める演算子は`%`です。

以下に示すif文

```
if( $aaa % 2 )
```

は、次のように書いても同じ意味です。

```
if( ($aaa % 2) == 1 )
```

このif文の条件は、「`$aaa`を2で割った余りが1の場合」と読み取れるので、こちらの方がわかりやすいかもしれません。

`if( $aaa % 2 )`のように、if文の条件として計算式を書くと、計算した結果の値に対して、真偽の判定が行われます。計算結果が0以外の場合は「真」、0の場合は「偽」と判定されます。

( `$aaa % 2` )の計算結果は、必ず0か1のどちらかの値になるため、これを利用して真偽の判定を行っているのです。

## 問題 3

答え ① ==

② &gt;=

## 解説

if文の条件を作る問題です。問題文に書かれている仕様とリストCを見比べて、どのような条件を作ればよいかを考えてみましょう。

リストCの8行目と11行目では、以下に示す条件を作る必要があります。



8行目 … \$tensuが100に等しいかどうか

11行目 … \$tensuが60以上かどうか

条件が作成できたら、該当する比較演算子を当てはめていきます。

変数が、ある値に等しいかどうかを判定する演算子は==です。また、変数が、ある値以上かどうかを判定する演算子は>=となります。

リストCでは、\$tensuの値が60未満の場合、elseブロックの中の処理(15行目)が実行されます。

## 問題 4

- 答え** ① ||  
② &&

### 解説

西暦年について、次に示す3つの条件をすべて満たしている場合、うるう年と判定できます。

#### ●うるう年の条件

- (ア) 4で割り切れる年。
- (イ) ただし、100で割り切れる年はうるう年でない。
- (ウ) しかし、400で割り切れる年はうるう年となる。

これらの条件を見ると、400で割り切れる年は必ずうるう年であることがわかります。

8行目のif文では、上記の3つの条件を組み合わせた記述を行い判定しています。まずは、if文の最初の条件が( \$year % 400 == 0 )となっていることに注目してください。これは上記(ウ)の条件をあらわしています。

400で割り切れる年の場合は必ずうるう年になりますので、①には「または」の意味を持つ論理演算子が入ります。つまり、論理演算子の||が入ります。

さらに100で割り切れなくて、かつ、4で割り切れる年がうるう年の条件になりますので、②には、「かつ」の意味を持つ論理演算子&&が入ります。

余談になりますが、PHPでうるう年を簡単に判定できる方法を紹介します。次のようにdate関数とmktime関数を組み合わせて使うと、指定した西暦年がうるう年かどうかを調べられます。

```
$ret = date( "L", mktime( 0, 0, 0, 1, 1, 2016 ) );
```

うるう年かどうかを調べたい西暦年を指定します。

date関数の戻り値(上記の例では\$ret)が、1の場合うるう年です。0の場合うるう年ではありません。

コンピュータの環境によっては、指定できる西暦年の範囲が1901年から2038年までの間に制限される場合がありますので注意してください。

## 問題 1

答え ウ

## 解説

8行目において、\$cntが100以下の間whileループを繰り返すように条件を指定していますが、繰り返しを途中で止めるには、break文を記述します。

## 問題 2

答え ① &lt;

## 解説

for文の繰り返し条件を作成する問題です。問題文によると10回の繰り返しを行う必要があることがわかります。変数\$cntは、繰り返し回数を数えるために使われるカウンタです。\$cntは0から始まり、繰り返しのたびに1ずつ加算されますので、\$cntが10未満の間処理を繰り返すと、10回の繰り返し処理になります。

したがって、「\$cntが10未満」という条件をあらわすには、\$cnt < 10と記述します。

また、本問題の答えとしては、<(小なり記号)が正解となりますが、別の書き方として、繰り返しを続けるための条件を「\$cntが9以下」とすることも可能です。その場合は、次のような記述になります。

## ● 変数\$cntが9以下の間繰り返す

```
for( $cnt = 0; $cnt <= 9; $cnt ++){  
    echo '楽しいな。', "<br>¥n";  
}
```

## 問題 1

答え ウ

## 解説

関数の呼び出し方についての問題です。関数は定義しただけでは実行されません。実行させるには、関数を呼び出す必要があります。

関数に引き渡す引数がなく、戻り値も受けとらない場合は、次の構文を使います。

## ● 関数の呼び出し(引数と戻り値がない場合)

```
関数名();
```

## 問題 1

答え

- ① name
- ② price

※①と②を入れ替えた答えも正解です。

解説

図Aの表示結果を見ると、shohinテーブルから商品名と価格を取得する必要があることがわかります。SQLのSELECT文では取得したい列の名前を指定できます。表Aを見ると、商品名の列名はname、価格の列名はpriceなので、SELECT文にそれらの列名を指定します。