

中間（終了）総合理解度テスト

実施日： 年 月 日（ 曜日）

資料作成者：神田 IT スクール コンテンツ開発部

開始時間： 時 分 終了時間： 時 分

座席番号：

氏名：

会社名：

<<テストの流れ>>

下記の4つのテストを行ないます。制限時間は5時間20分です。

- ① PHP 入門～Web アプリ選択式問題【目安時間：60分】
- ② PHP 入門理解度テスト（プログラム作成）【目安時間：1時間30分】
- ③ PHP 関数理解度テスト（プログラム作成）【目安時間：1時間40分】
- ④ Web アプリ理解度テスト（プログラム作成）【目安時間：2時間10分】

※制限時間は厳守してください。

※自宅学習の方は本 PDF ファイルを「Microsoft Edge」で開いてください。

【各テストのスケジュール】

例）9時からテストを開始し、目安時間通りに進めた場合

09:00～10:00 ①PHP 入門～Web アプリ択式問題

10:00～11:30 ②PHP 入門理解度テスト

11:30～14:10 ③PHP 関数理解度テスト（お昼休み：11:50～12:50を挟んで）

14:10～16:20 ④Web アプリ理解度テスト

Step1 ■上記で記入する項目を記入したかどうか確認し、以下にチェックを入れてください。

- ☐ 「実施日」を記入しました。 ※通学時のみ
- ☐ 「開始時間」を記入しました。 ※通学時のみ
- ☐ 「座席番号」を記入しました。 ※通学時のみ
- ☐ 「氏名」を記入しました。
- ☐ 「会社名」を記入しました。

Step2 ■配布されたテスト資料が43ページで構成されているか確認します。

Step3 ■Eclipse のプロジェクトエクスプローラーのファイルは全てバックアップを取り、削除します。

Step4 ■制限時間は厳守してください。

※ただし、残り数行で書き終わる等、目途が立っている場合は最後まで仕上げていただいて構いません。
処理の組み立て方について検討もつかず、分からない時は、潔く諦めて提出するようにしましょう。

Step5 ■プログラム提出用フォルダを「総合理解度テスト_名前」として作成します。

①～④のテスト問題を解き、プログラムを完成させます。

Step6 ■すべてのプログラムが完成したら、終了時間を記入します。②と④で作成したプロジェクトを「プログラム提出用フォルダ」に格納し、zipで圧縮した後、提出フォームから提出してください。

Step7 ■提出が完了したら、問題用紙を講師に提出してください。

※自宅学習の方は、上記のプログラム作成問題の提出物と一緒に
解答を記入して上書き保存した本ファイルを提出してください。

中間・終了理解度テスト①選択式問題

<<テストの流れ>>

※テキストは使ってよい。ただし、PC は使ってはいけません。

Step ■問題文を読み、答えをア～エまたは○・×から 1 つ選び、回答欄に記入してください。

- 9. 以下の条件式で、「FALSE」判定されるものを選びなさい。

```
$a = 0;
```

ア	if(\$a == null)
イ	if(\$a == '0.0')
ウ	if(\$a === '0')
エ	if(\$a === 0)

解答

- 10. 以下プログラムの for 文が終了した時、変数\$j の値はいくつになっているか選びなさい。

```
$j = 0;
for ($i=0; $i<9; $i++){
    if ( $i % 2 != 0){
        $j++;
    }
}
```

ア	3
イ	4
ウ	5
エ	6

解答

- 11. 変数の値を出力するのに便利なショートタグはどれか選びなさい。

ア	<?= ?>
イ	<?php ?>
ウ	<% %>
エ	<%= %>

解答

- 12. 以下の説明について○か×で答えなさい。

変数\$num1 の値を使ってキャストを行うと変数\$num1 の値も変更される。

```
$num1 = 3.14;
```

```
$num2 = (int)$num1;
```

解答

- 13. if 文に関しての説明で間違っているものを選びなさい。

ア	各ブロック内に条件分岐文を書くこと if 文のネストと言う。
イ	elseif ブロックは任意の数だけ増やすことができる。
ウ	if 文は必ず if ブロックから始める必要はない。
エ	else ブロックは省略することができる。

解答

- 14. 以下の説明について○か×で答えなさい。
『switch 文で記述出来る内容は if 文でも記述できる。』

解答

- 15. 繰り返し構文で無限ループさせたい時の記述方法で間違っているものを選びなさい。

ア	for(; ;)
イ	while(false)
ウ	while(true)
エ	do{ ... }while(true);

解答

- 16. 配列の中身の状態全てを画面に表示できる関数はどれか選びなさい。

ア	echo
イ	print
ウ	array
エ	print_r

解答

②入門理解度テスト

<<テストの流れ>>

条件 ■eclipse を使用し、プログラムを作成する。
テキストは使ってよい。インターネットは使ってはいけません。

Step ■プロジェクトを「intro_test_名字のローマ字」として作成します。
問題を解き、プログラムを完成させます。

※ファイルの作成方法は以前の入門理解度テストの方法に沿って作成を行ってください。

問1 以下の仕様と実行結果から **calcTotal.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : calcTotal.php |
| ③アクセス URL | : http://localhost/intro_test_名字のローマ字/calcTotal.php |

概要

配列の要素の合計値を求めて画面に出力するプログラム。

作成仕様

@main メソッド

- ・ 3, 8, 1, 12, 5 の 5 つの要素値を持つ配列\$num を作成する。
- ・ 作成した配列の要素を 1 つずつ変数\$total に加算する。
- ・ 実行結果と同じ表示になるように、合計値を表示する。

実行結果

合計は 29 です。

問2 以下の仕様と実行結果から **echoRandomMessage.php** を作成しなさい。

ソースコード

①ソースフォルダ	: intro_test_名字のローマ字
②ファイル名	: echoRandomMessage.php
③アクセス URL	: http://localhost/intro_test_名字のローマ字/echoRandomMessage.php

概要

ランダムに生成された数値に応じて文字列を表示するプログラム。

作成仕様

- ・ 1～5 のランダムな数値を生成する。
- ・ 生成された数値に応じて、実行結果と同じ表示になるように下記のメッセージを表示する。
 - 5 → 「5」が出た！大当たり！！
 - 4 → 「4」が出た！当たり！！
 - 上記以外 → 「*」が出た！また来てね！ ※「*」には生成された数値が表示される。

ヒント：次のように記述すると、変数 **n** には 0,1,2,3,4 のいずれかの値がランダムに代入される。

```
$n = mt_rand(0, 4);
```

実行結果

- ・ 実行する度にランダムに表示されるようにしてください。

「5」が出た！大当たり！！

問3 以下の仕様と実行結果から **echoCountDown.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : echoCountDown.php |
| ③アクセス URL | : http://localhost/intro_test_名字のローマ字/echoCountDown.php |

概要

配列の要素を繰り返し処理の中で表示するプログラム。

作成仕様

- ・ '浜松町', '新橋', '有楽町', '東京', '神田' の 5 つの要素値を持つ配列 \$station を作成する。
- ・ 実行結果と同じ表示になるように、作成した配列の要素を 1 つずつ表示する。

実行結果

次は浜松町に止まります。神田まであと 5 駅です。 次は新橋に止まります。神田まであと 4 駅です。 次は有楽町に止まります。神田まであと 3 駅です。 次は東京に止まります。神田まであと 2 駅です。 次は神田に止まります。神田まであと 1 駅です。
--

問4 以下の仕様と実行結果から **echoGenderCount.php** を作成しなさい。

ソースコード

①ソースフォルダ	: intro_test_名字のローマ字
②ファイル名	: echoGenderCount.php
③アクセス URL	: http://localhost/intro_test_名字のローマ字/echoGenderCount.php

概要

繰り返し処理の中で配列要素の数を数えるプログラム。

作成仕様

- ・ '男性', '女性', '男性', '男性', '女性', '男性' の 6 つの要素値を持つ配列 \$gender_list を作成する。
- ・ 作成した配列の要素を 1 つずつ調べ、男性の場合は変数 \$male に、女性の場合は変数 \$female に加算する。
- ・ 実行結果と同じ表示になるように、合計値を表示する。

実行結果

男性の人数は 4 名、女性の人数は 2 名です。

問5 以下の仕様と実行結果から **echoMaxAge.php** を作成しなさい。

ソースコード

- | | |
|-----------|--|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : echoMaxAge.php |
| ③アクセス URL | : http://localhost/intro_test_名字のローマ字/echoMaxAge.php |

概要

配列の要素の大小を繰り返し処理の中で比較し、一番大きい値を表示するプログラム。

作成仕様

- ・ 'A' => 23, 'B' => 31, 'C' => 18 の 3 つの要素値を持つ配列\$age_list を作成する。
- ・ 作成した配列の要素を 1 つずつ調べ、最も年上の人の名前を変数\$max_name に、年齢を変数\$max_age に格納する。
- ・ 実行結果と同じ表示になるように、集計結果を表示する。

実行結果

3 名の中で最も年上なのは B さんで 31 歳です。

問6 以下の仕様と実行結果から **formGet.php** と **formGetResult.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : formGet.php |
| ③アクセス URL | : http://localhost/intro_test_名字のローマ字/formGet.php |

概要

フォームから入力され、GET 送信されたデータを結果画面で出力するプログラム。

作成仕様

以下の仕様を基に画面を作成します。

- ・「氏名」「年令」を入力する 2 つのテキストボックスと送信ボタンを作成する。
- ・送信ボタンをクリックした場合は「**GET 送信**」で、送信先は「formGetResult.php」に送信されるように設定する。

実行結果

The screenshot shows a web browser window with the title 'GET送信フォーム'. The address bar displays 'localhost/intro_test_kanda/formGet.php'. The main content area of the browser shows a form with the heading '氏名と年令を入力してください'. Below the heading, there are two empty text input fields side-by-side, followed by a button labeled '送信'.

ソースコード

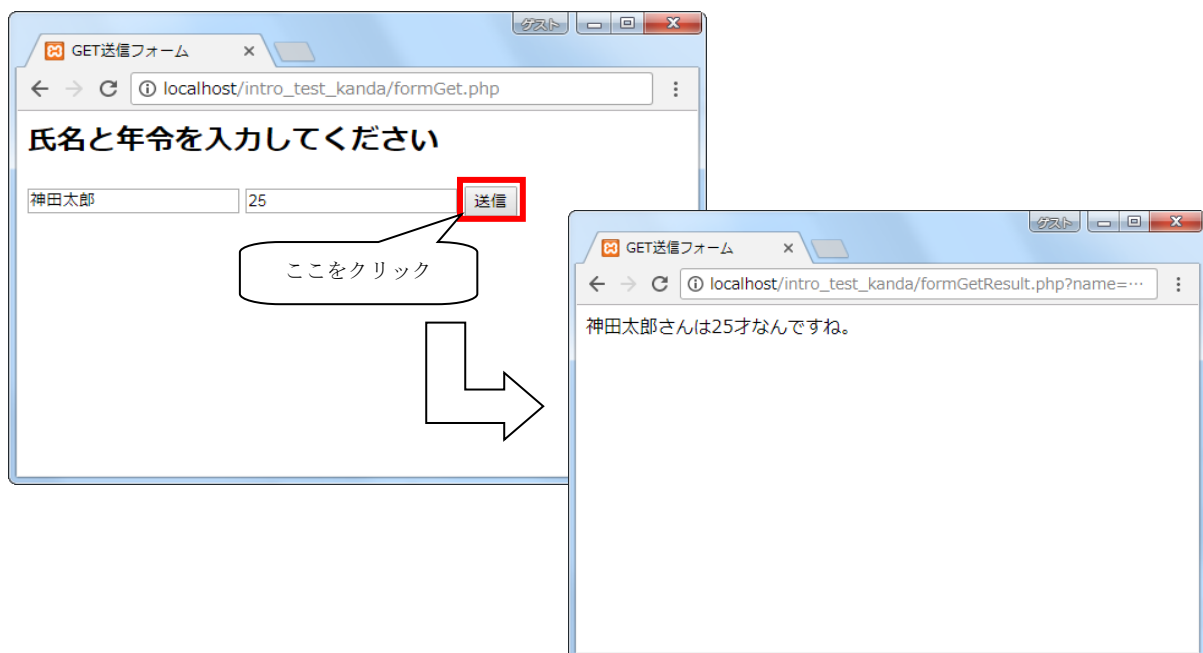
- | | |
|-----------|-------------------------------|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : formGetResult.php |
| ③アクセス URL | : formGet.php からの画面遷移でアクセスされる |

作成仕様

以下の仕様を基に画面を作成します。

- ・実行結果と同じ表示になるように、フォームから送信された「氏名」「年令」を表示する。

実行結果



問7 以下の仕様と実行結果から **formPost.php** と **formPostResult.php** を作成しなさい。

ソースコード

- | | |
|-----------|--|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : formPost.php |
| ③アクセス URL | : http://localhost/intro_test_名字のローマ字/formPost.php |

概要

フォームから入力され、POST 送信されたデータを結果画面で出力するプログラム。

作成仕様

以下の仕様を基に画面を作成します。

- ・「好きな食べ物」を選択する 3 つのラジオボタンと送信ボタンを作成する。
- ・送信ボタンをクリックした場合は「**POST 送信**」で、送信先は「formPostResult.php」に送信されるように設定する。

実行結果



ソースコード

- | | |
|-----------|--------------------------------|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : formPostResult.php |
| ③アクセス URL | : formPost.php からの画面遷移でアクセスされる |

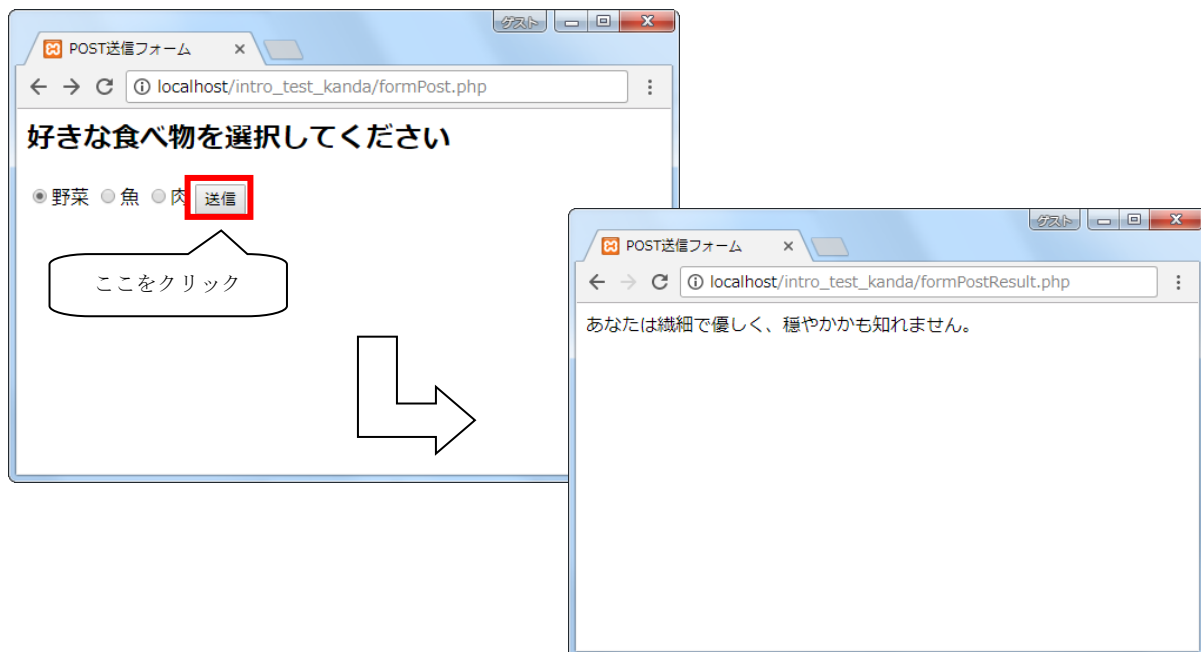
作成仕様

以下の仕様を基に画面を作成します。

- ・実行結果と同じ表示になるように、フォームから送信された「好きな食べ物」に応じたメッセージを表示する。

野菜が選択された場合	→	あなたは繊細で優しく、穏やかかも知れません。
魚が選択された場合	→	あなたは辛抱強く努力家で、冷静かも知れません。
肉が選択された場合	→	あなたはパワフルで強く、積極的かも知れません。

実行結果



問8 以下の仕様と実行結果から **formLoopCount.php** と **formLoopCountResult.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : formLoopCount.php |
| ③アクセス URL | : http://localhost/intro_test_名字のローマ字/formLoopCount.php |

概要

フォームから入力されたデータを用いて、1 からその入力値までの合計値を繰り返し処理の中で計算するプログラム。

作成仕様

以下の仕様を基に画面を作成します。

- ・「繰り返す回数」を入力するテキストボックスと送信ボタンを作成する。
- ・送信ボタンをクリックした場合は「**POST 送信**」で、送信先は「formLoopCountResult.php」に送信されるように設定する。

実行結果



POST送信フォーム

localhost/intro_test_kanda/formLoopCount.php

繰り返す回数 (1~100) を入力してください

送信

ソースコード

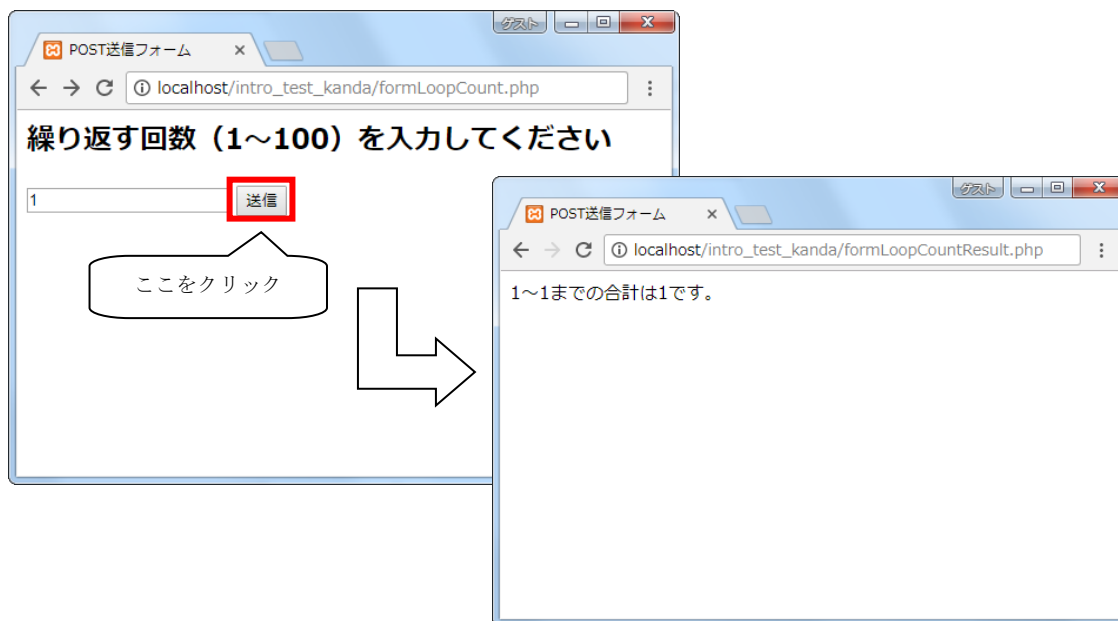
- | | |
|-----------|-------------------------------------|
| ①ソースフォルダ | : intro_test_名字のローマ字 |
| ②ファイル名 | : formLoopCountResult.php |
| ③アクセス URL | : formLoopCount.php からの画面遷移でアクセスされる |

作成仕様

以下の仕様を基に画面を作成します。

- ・フォームから送信された「繰り返す回数」分、1 から順に加算を繰り返すし、最後に合計値を表示する。
(例) 5 が入力された場合、 $1+2+3+4+5$ の計算結果である 15 が表示される。
- ・1~100 以外の値が入力された場合は、「不正な値が入力されました。」と表示する。

実行結果



以上、お疲れ様でした。

③関数理解度テスト

<<テストの流れ>>

条件 ■eclipse を使用し、プログラムを作成する。
テキストは使ってよい。インターネットは使ってはいけません。

Step ■プロジェクトを「function_test_名字のローマ字」として作成します。
問題を解き、プログラムを完成させます。

※ファイルの作成方法は以前の関数理解度テストの方法に沿って作成を行ってください。

問1 以下の仕様と実行結果から **rollDice.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : function_test_名字のローマ字 |
| ②ファイル名 | : rollDice.php |
| ③アクセス URL | : http://localhost/function_test_名字のローマ字/rollDice.php |

概要

ランダムに生成された数字に応じて結果を表示するプログラム。

作成仕様

@呼び出し元

- ・rollDice 関数を呼び出す。

@ rollDice 関数

- ・引数 なし
- ・戻り値 なし
- ・サイコロの結果（1～6 までのランダムな数字）を表示する。

ヒント：次のように記述すると、変数\$*n* には 0,1,2,3,4 のいずれかの値がランダムに代入される。

```
$n = mt_rand(0, 4);
```

実行結果

- ・実行する度にランダムに表示されるようにしてください。

サイコロを振った結果、「3」の目が出ました！

問2 以下の仕様と実行結果から **printSquareAsterisk.php** を作成しなさい。

ソースコード

①ソースフォルダ	: function_test_名字のローマ字
②ファイル名	: printSquareAsterisk.php
③アクセス URL	: http://localhost/function_test_名字のローマ字/printSquareAsterisk.php

概要

指定された数値を 2 乗し、その数分「*」を表示するプログラム。

作成仕様

@呼び出し元

- ・変数\$num に数字を格納する。
- ・変数\$num を引数として、printSquareAsterisk 関数を呼び出す。(実行結果参照)

@ printSquareAsterisk 関数

- ・引数 int 型 \$num
- ・戻り値 なし
- ・引数で渡された\$num の 2 乗分だけ 「*」を出力する。

実行結果

- ・ \$num が 3 の場合

```
*****
```

問3 以下の仕様と実行結果から **kuku.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : function_test_名字のローマ字 |
| ②ファイル名 | : kuku.php |
| ③アクセス URL | : http://localhost/function_test_名字のローマ字/kuku.php |

概要

1~9 が格納された配列を使い、繰り返し処理の中で九九の計算結果を表示するプログラム。

作成仕様

@呼び出し元

- ・配列\$kuku に九九の値を格納する。
- ・\$kuku を引数とし、kuku 関数を呼び出す。

@ kuku 関数

- ・引数 int 型二次元配列 \$kuku
- ・戻り値 なし
- ・foreach 文を使って\$kuku の値を表示する。(実行結果参照)

実行結果

- ・省略されている部分も表示させること。

1の段:	1	2	3	4	5	6	7	8	9
2の段:	2	4	6	8	10	12	14	16	18
...									
9の段:	9	18	27	36	45	54	63	72	81

問4 以下の仕様と実行結果から **getSeason.php** を作成しなさい。

ソースコード

- | | |
|-----------|--|
| ①ソースフォルダ | : function_test_名字のローマ字 |
| ②ファイル名 | : getSeason.php |
| ③アクセス URL | : http://localhost/function_test_名字のローマ字/getSeason.php |

概要

ランダムに生成された数字の月に応じて季節の文字列を表示するプログラム。

作成仕様

@呼び出し元

- ・ 1 から 12 までのランダムな数字を出力し、ランダムに生成した整数を引数に **getSeason** 関数を呼び出す。
- ・ **getSeason** 関数の戻り値を変数 **\$season** に格納し表示する。(実行結果参照)

@ **getSeason** 関数

- ・ 引数 **int** 型 **\$month**
- ・ 戻り値 **string** 型
- ・ 引数で与えられた **\$month** の月に応じた季節（春、夏、秋、冬）の文字を返す。

ただし、各月の季節は以下のようにする。

1 月…「冬」、 2 月…「冬」、 3 月…「春」、 4 月…「春」、 5 月…「春」、 6 月…「夏」
7 月…「夏」、 8 月…「夏」、 9 月…「秋」、10 月…「秋」、11 月…「秋」、12 月…「冬」

実行結果

3 月の季節は「春」です。

問5 以下の仕様と実行結果から **mergeArray.php** を作成しなさい。

ソースコード

- | | |
|------------|---|
| ① ソースフォルダ | : function_test_名字のローマ字 |
| ② ファイル名 | : mergeArray.php |
| ③ アクセス URL | : http://localhost/function_test_名字のローマ字/mergeArray.php |

概要

アルファベットが格納された 2 つの配列の要素を全て格納する配列の要素を表示するプログラム。

作成仕様

@呼び出し元

- ・「H」、「e」、「l」、「l」、「o」の値を持つ配列\$array1 と、「W」、「o」、「r」、「l」、「d」の値を持つ配列\$array2 を定義する。
- ・\$array1 と\$array2 を引数とし、mergeArray 関数を呼び出し、\$array1 と\$array2 の配列を連結させた配列を受け取る。
- ・受け取った配列の値を表示する。(実行結果参照)

@ mergeArray 関数

- ・引数 string 型配列 \$array1、string 型配列 \$array2
- ・戻り値 string 型配列
- ・引数で与えられた\$array1 と\$array2 を連結させた配列を返す。
(\$array1 の要素が先頭、\$array2 の要素が後ろになるようにする。)

実行結果

HelloWorld

問6 以下の仕様と実行結果から **squareElements.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : function_test_名字のローマ字 |
| ②ファイル名 | : squareElements.php |
| ③アクセス URL | : http://localhost/function_test_名字のローマ字/squareElements.php |

概要

配列の要素の中身を 2 乗して表示するプログラム。

作成仕様

@呼び出し元

- ・ 1 から 7 までの 7 つの要素値を持つ int 型配列 \$array を定義する。
- ・ squareElement 関数を呼び出し、配列の各要素を全て 2 乗する。
- ・ 配列の要素を出力する。(実行結果参照)

@ squareElements 関数

- ・ 引数 int 型配列 \$array
- ・ 戻り値 int 型配列 \$array
- ・ 引数で受け取る配列の要素を全て 2 乗する。

実行結果

1 4 9 16 25 36 49

問7 以下の仕様と実行結果から **isUniqueArrayElement.php** を作成しなさい。

ソースコード

- | | |
|-----------|---|
| ①ソースフォルダ | : function_test_名字のローマ字 |
| ②ファイル名 | : isUniqueArrayElement.php |
| ③アクセス URL | : http://localhost/function_test_名字のローマ字/isUniqueArrayElement.php |

概要

2つの配列に格納された数値が重複しているかを判定するプログラム。

作成仕様

@呼び出し元

- ・ 12,24,8,15 の要素を持つ配列\$array1 と、14,25,11 の要素を持つ配列\$array2 を定義する。
- ・ \$array1 と\$array2 を引数として、isUniqueArrayElement 関数を呼び出し、戻り値を受け取る。
- ・ 重複した値が1つもなければ「全てユニークな値になっています。」と表示し、1つでも重複した値があれば「重複した値が含まれています！」と表示する。(実行結果参照)

@ isUniqueArrayElement 関数

- ・ 引数 int 型配列 \$array1、int 型配列 \$array2
- ・ 戻り値 boolean 型
- ・ 2つの配列の要素を確認し、1つも重複した値がなければ true、1つでもあれば false を返す。

実行結果

--- 配列 array1 と配列 array2 を比較 --- 全てユニークな値になっています。
--

問8 以下の仕様と実行結果から **countThreeTimes.php** を作成しなさい。

ソースコード

①ソースフォルダ	: function_test_名字のローマ字
②ファイル名	: countThreeTimes.php
③アクセス URL	: http://localhost/function_test_名字のローマ字/countThreeTimes.php

概要

配列の要素のうち3の倍数を見つけ、その数と値を表示するプログラム。

作成仕様

@呼び出し元

※ここでは、関数を2つ作成します。

- ・1,3,5,7,9,11,13,15の要素を持つ配列\$arrayを定義する。
- ・countThreeTimes関数を呼び出し、配列内にある3の倍数の個数を変数\$countで受け取る。
- ・戻り値として受け取った\$countを表示する。(実行結果参照)

@ countThreeTimes 関数

- ・引数 int 型配列 \$array
- ・戻り値 int 型 \$count
- ・引数で受け取った配列の要素を1つずつ調べ、3の倍数の数字が何個あるかを数える。
- ・また3の倍数の数字が見つかるたびにその配列番号と値を引数に printThreeTimesElement 関数を呼び出し3の倍数が格納されている配列番号とその値を表示する。

@ printThreeTimesElement 関数

- ・引数 int 型 \$index、int 型 \$num
- ・戻り値 なし
- ・引数で与えられた\$index と\$num の値を表示する。(実行結果参照)

実行結果

配列番号 1 番目の値 3 は 3 の倍数です。
配列番号 4 番目の値 9 は 3 の倍数です。
配列番号 7 番目の値 15 は 3 の倍数です。

配列内にある 3 の倍数は合計で 3 個です。

③Web アプリ理解度テスト

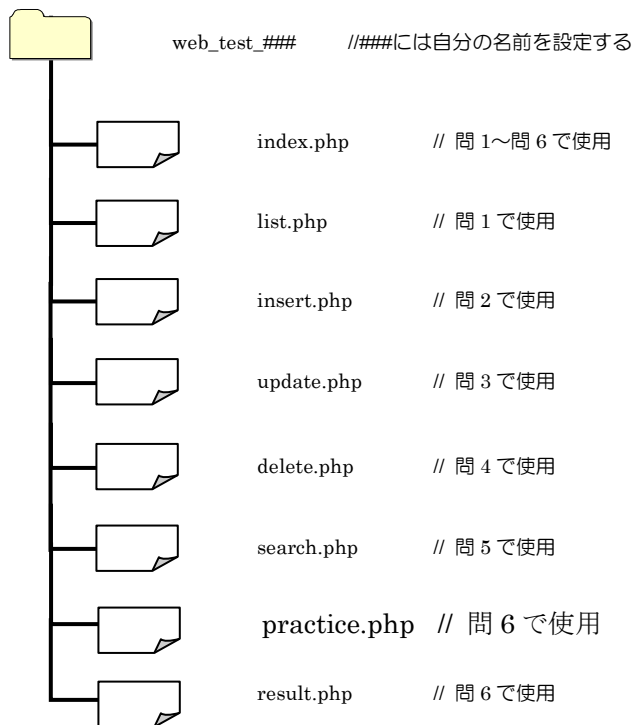
<<テストの流れ>>

条件 ■**eclipse** を使用し、プログラムを作成する。
テキストは使ってよい。インターネットは使ってはいけません。

Step ■プロジェクトを「web_test_名字のローマ字」として作成します。
問題を解き、プログラムを完成させます。

<<プログラムのファイル構成>>

作成するプログラムのファイル構成は以下のようになります。



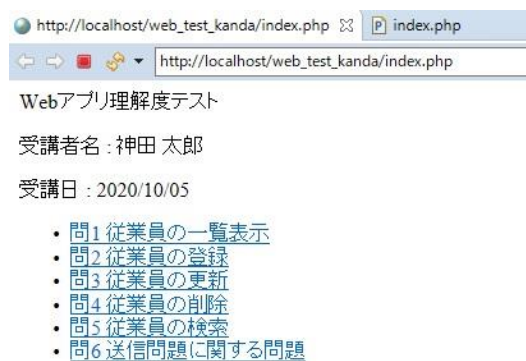
※上記が最低限必要なファイルです。その他追加したいファイルがある場合は追加してもかまいません。

<<前提>>

問題を解く前に、以下のような画面を表示する「`index.php`」を作成します。

以下の画面を表示する URL は「`http://localhost/web_test_####/index.php`」になります。

`index.php` ファイル内の各 URL は全て「`href="#"`」を設定します。(URL は各問題を解く時に設定します。)



問 1 以下の仕様と実行結果から list.php を作成しなさい。

概要

employeeinfo テーブルのデータを取得し、list.php で画面表示するプログラム。

前提

MySQL を使用する。

- ・ユーザー名：root
- ・パスワード：root123
- ・データベース名：testdb
- ・テーブル名：employeeinfo
- ・下記の SQL 文を実行し、データベースとテーブルを作成してください。

```
drop database testdb;
```

```
create database testdb;
```

```
use testdb;
```

```
create table employeeinfo (  
  id varchar(4) primary key not null,  
  name varchar(20),  
  age int(3),  
  store varchar(20)  
);
```

```
insert into employeeinfo (id, name, age, store) values ('0001', 'Tom', 20, '神田店');
```

```
insert into employeeinfo (id, name, age, store) values ('0002', 'Cathy', 17, '神田店');
```

```
insert into employeeinfo (id, name, age, store) values ('0003', 'Mike', 35, '東京店');
```

ソースコード

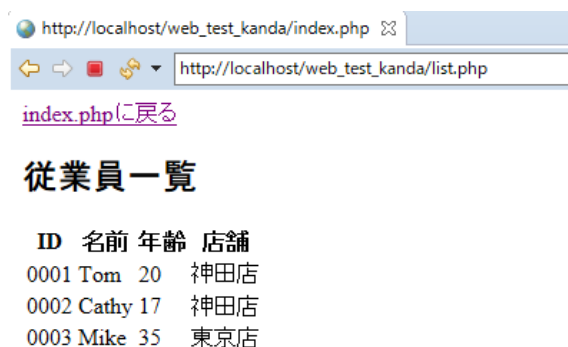
- | | |
|----------|--------------------------------------|
| ①ソースフォルダ | : web_test_名字のローマ字 |
| ③名前 | : list.php |
| ④アクセス方法 | : index.php の画面から「問 1 従業員の一覧表示」をクリック |

作成仕様

- ・DB の従業員テーブル内に格納されている情報を全件取得し、従業員の一覧を表示

実行結果

index.php から「問 1 従業員の一覧」押下時



問 2 以下の仕様と実行結果から **insert.php** を作成しなさい。

概要

employeeinfo テーブルに新規データを登録し、結果を表示するプログラム。

ソースコード

①ソースフォルダ	: web_test_名字のローマ字
②名前	: insert.php
③アクセス方法	: index.php の画面から「問 2 従業員の登録」をクリック

作成仕様

以下の仕様を基に画面を作成します。

- ・従業員を登録できるフォームを作成する。
- ・ID、名前、年齢の入力は「テキストボックス」、店舗の選択は「ラジオボタン」で設定できるようにする。
- ・form タグに URL 「**insert.php**」を設定し、「登録」ボタンを押すと登録処理が動作する。

実行結果

index.php から「問 2 従業員の登録」押下時

http://localhost/web_test_kanda/index.php

http://localhost/web_test_kanda/insert.php

[index.phpに戻る](#)

従業員の登録

ID:

名前:

年齢:

店舗: ☒ 神田店 ☐ 東京店

作成仕様

- ・登録フォームから入力したパラメータ（従業員情報）を全て取得する。
- ・取得した従業員情報を DB に登録する。
- ・登録完了メッセージをセッションに保存する。
- ・問 2 で作成した従業員一覧画面に遷移（リダイレクト）し、セッションに保存したメッセージを表示する。

【リダイレクト方法】

```
header('Location: 遷移先の URL');  
exit;
```

※次ページに「実行結果」の記載があります。

実行結果

IDに「0004」、名前に「Mary」、年齢に「23」を入力し、店舗「神田店」を選択した場合

The image shows a two-step process for registering an employee in a web application.

Step 1: Registration Form

The browser address bar shows `http://localhost/web_test_kanda/index.php`. The page title is "従業員登録" (Employee Registration). The form contains the following fields:

- ID: 0004
- 名前 (Name): Mary
- 年齢 (Age): 23
- 店舗 (Store): ☒ 神田店 ☐ 東京店

A red box highlights the "登録" (Register) button, with a callout bubble saying "クリック" (Click). Another callout bubble says "登録したい 従業員情報を入力" (I want to register, enter employee information).

Step 2: Confirmation and List

The browser address bar shows `http://localhost/web_test_kanda/list.php`. A red box highlights the message "従業員の登録が完了しました!" (Employee registration is complete!). A callout bubble says "メッセージが表示される" (Message is displayed).

Below the message is the "従業員一覧" (Employee List) table:

ID	名前	年齢	店舗
0001	Tom	20	神田店
0002	Cathy	17	神田店
0003	Mike	35	東京店
0004	Mary	23	神田店

A callout bubble says "「登録」をクリックすると、新しく従業員が登録される。" (When you click "Register", a new employee is registered). A red box highlights the new entry (0004 Mary 23 神田店).

問 3 以下の仕様と実行結果から **update.php** を作成しなさい。

概要

employeeinfo テーブルのデータを一部変更し、結果を表示するプログラム。

ソースコード

- | | |
|----------|------------------------------------|
| ①ソースフォルダ | : web_test_名字のローマ字 |
| ②名前 | : update.php |
| ③アクセス方法 | : index.php の画面から「問 3 従業員の更新」をクリック |

作成仕様

以下の仕様を基に画面を作成します。

- ・すでに登録されている従業員情報の更新ができるフォームを作成する。
- ・変更対象の ID、名前、年齢の入力は「テキストボックス」、店舗の選択は「ラジオボタン」で設定できるようにする。form タグに URL 「**update.php**」を設定し、「変更」ボタンを押すと更新処理が動作する。

実行結果

index.php から「問 4 従業員の更新」押下時

http://localhost/web_test_kanda/index.php

http://localhost/web_test_kanda/update.php

[index.phpに戻る](#)

従業員更新

更新対象のID:

名前:

年齢:

店舗: ☒ 神田店 ☐ 東京店

作成仕様

- ・更新画面から入力したパラメータ（従業員情報）を全て取得する。
- ・更新対象の ID を持つ従業員を取得した従業員情報で更新する。
- ・更新完了メッセージをセッションに保存する。
- ・一覧画面に遷移（リダイレクト）し、セッションに保存したメッセージを表示する。

※次ページに「実行結果」の記載があります。

実行結果

ID「0004」を持つ従業員の名前を「Brian」、年齢を「25」、店舗を「東京店」に更新する場合

http://localhost/web_test_kanda/index.php

index.phpに戻る

従業員の更新

更新対象のID: 0004

名前: Brian

年齢: 25

店舗: ○神田店 ●東京店

変更

変更したい従業員のIDを入力し、変更情報を入力する。

クリック

http://localhost/web_test_kanda/update.php

メッセージが表示される

従業員の更新が完了しました!

index.phpに戻る

従業員一覧

「変更」をクリックすると、対象のIDを持つ従業員情報が更新される。

ID	名前	年齢	店舗
0001	Tom	20	神田店
0002	Cathy	17	神田店
0003	Mike	35	東京店
0004	Brian	25	東京店

問 4 以下の仕様と実行結果から **delete.php** を作成しなさい。

概要

employeeinfo テーブルのデータを削除し、結果を表示するプログラム。

ソースコード

- | | |
|----------|------------------------------------|
| ①ソースフォルダ | : web_test_名字のローマ字 |
| ②ファイル名 | : delete.php |
| ③アクセス方法 | : index.php の画面から「問 4 従業員の削除」をクリック |

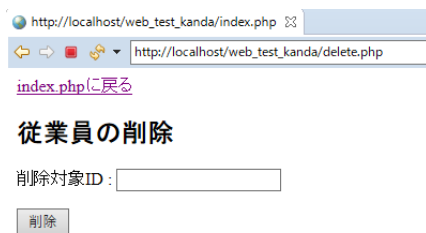
作成仕様

以下の仕様を基に画面を作成します。

- ・DBに登録されている従業員を削除できるフォームを作成する。
- ・削除対象の ID は「テキストボックス」で入力できるようにする。
- ・form タグに URL 「**delete.php**」を設定し、「削除」ボタンを押すと削除処理が動作する。

実行結果

index.php から「問 5 従業員の削除」押下時



作成仕様

- ・削除画面から入力したパラメータ（削除対象 ID）を取得する。
- ・削除対象の ID を持つ従業員を削除する。
- ・削除完了メッセージをセッションに保存する。
- ・一覧画面に遷移（リダイレクト）し、セッションに保存したメッセージを表示する。

※次ページに「実行結果」の記載があります。

実行結果

ID「0004」を持つ従業員を削除する場合

The screenshot shows a web browser at `http://localhost/web_test_kanda/index.php`. The page title is **従業員の削除** (Employee Deletion). It contains a link [index.phpに戻る](#), a text input field with the value `0004`, and a button labeled **削除** (Delete). A callout bubble points to the input field with the text: **削除したい従業員の ID を入力する。** (Enter the ID of the employee to be deleted). Another callout bubble points to the **削除** button with the text: **クリック** (Click). A large arrow points from this screen to the next screen.

The next screen shows the browser at `http://localhost/web_test_kanda/list.php`. The page title is **従業員一覧** (Employee List). It contains a link [index.phpに戻る](#) and a table of employees:

ID	名前	年齢	店舗
0001	Tom	20	神田店
0002	Cathy	17	神田店
0003	Mike	35	東京店

A red box highlights the bottom of the table. A callout bubble points to the table with the text: **「削除」をクリックすると、対象の ID を持つ従業員が削除される。** (When you click 'Delete', the employee with the target ID will be deleted). Another callout bubble points to the top of the page with the text: **メッセージが表示される** (A message is displayed).

問 5 以下の仕様と実行結果から **search.php** を作成しなさい。

概要

id の値を用いて employeeinfo テーブルのデータを検索し、結果を表示するプログラム。

ソースコード

- | | |
|----------|------------------------------------|
| ①ソースフォルダ | : web_test_名字のローマ字 |
| ②名前 | : search.php |
| ③アクセス方法 | : index.php の画面から「問 5 従業員の検索」をクリック |

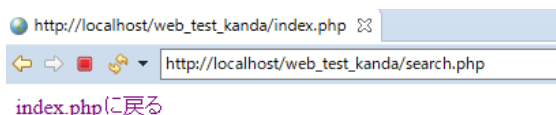
作成仕様

以下の仕様を基に画面を作成します。

- ・DB に登録されている従業員を削除できるフォームを作成する。
- ・検索対象の ID は「テキストボックス」で入力できるようにする。
- ・form タグに URL 「**search.php**」を設定し、「検索」ボタンを押すと検索処理が動作する。

実行結果

index.php から「問 6 従業員の検索」押下時



従業員の検索

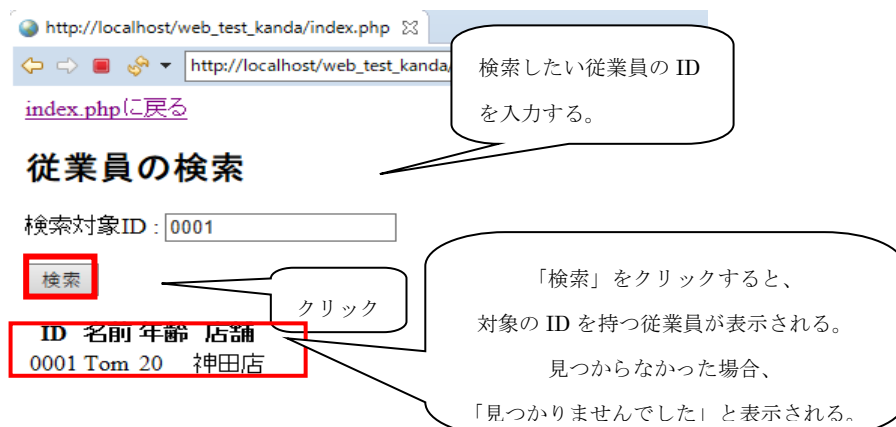
検索対象ID :

作成仕様

- ・検索画面から入力したパラメータ（ID）を取得する。
- ・検索対象の ID を持つ従業員を表示する。

実行結果

ID 「0001」 を持つ従業員を検索する場合



問 6 以下の仕様と実行結果から **practice.php** を作成しなさい。

概要

データの送信方法に応じて結果メッセージを表示するプログラム。

ソースコード

- | | |
|-----------|--|
| ①ソースフォルダ | : web_test_名字のローマ字 |
| ②ファイル名 | : practice.php |
| ③アクセス URL | : index.php の画面から「問 6 送信問題に関する問題」をクリック |

作成仕様

以下の仕様を基に画面を作成します。

- ・画面内の各ボタンまたはリンクの URL を「**./result.php**」とする。
- ・2 つのボタンについては、一番上のボタンをクリックした場合は「**GET 送信**」、中央のボタンをクリックした場合は「**POST 送信**」で送信されるように設定する。
- ・画面遷移をおこなうとき、画面内のどのボタン（またはリンク）から送信されたのかを識別する為のパラメータ「**cmd**」を送信する。パラメータ **cmd** の内容は以下のようにする。
 - フォームからの GET 送信の場合 : **cmd** の値に「**get**」を設定する。
 - フォームからの POST 送信の場合 : **cmd** の値に「**post**」を設定する。
 - リンクからの送信の場合 : **cmd** の値に「**link**」を設定する。

実行結果



ソースコード

- | | |
|----------|--------------------------------|
| ①ソースフォルダ | : web_test_名字のローマ字 |
| ②ファイル名 | : result.php |
| ③アクセス方法 | : practice.php からの画面遷移でアクセスされる |

作成仕様

以下の仕様を基に画面を作成します。

- ・画面表示用メッセージを格納する変数を用意し、フォームから送信された「**cmd**」に対応したメッセージを

格納する。

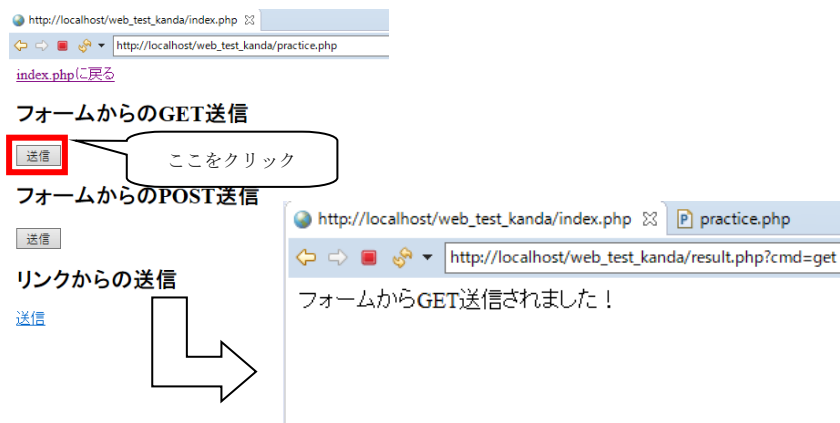
設定するメッセージは以下のようにする。

- ①フォームから GET 送信された場合：「フォームから GET 送信されました！」
 - ②フォームから POST 送信された場合：「フォームから POST 送信されました！」
 - ③リンクから送信された場合：「リンクから送信されました！」
- ・設定したメッセージを表示する。

実行結果

以下のように、クリックした箇所によってメッセージを変化させる。

1. フォームから GET 送信された場合



2. フォームから POST 送信された場合



3. リンクから送信された場合



以上、お疲れ様でした。