

Web アプリ理解度テスト②プログラム作成問題(全 3 問:60 点)

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

テスト制限時間： 2 時間 10 分

タイピングチェック制限時間： 10 分

座席番号：

氏名：

会社名：

タイピングレベル：

<<テストの流れ>>

※eclipse を使用して、プログラムを作成してください。

※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。

※制限時間は厳守してください。

※取得した PDF ファイルを「Adobe Acrobat Reader DC」で開いてください。

★Adobe Acrobat Reader DC の使い方については下記でご確認ください。

URL：https://drive.google.com/drive/u/2/folders/1j28jbwdwSGZpxSlozVrvxgm5rS_BfRN5

Step1 ■配布されたテスト資料が 9 ページで構成されているか確認します。

Step2 ■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下さい。

☐ 「実施日」を記入しました。 ※通学時のみ

☐ 「開始時間」「終了時間」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。

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 ■プロジェクト「web_test_名字のローマ字」を作って問題を解き、プログラムを完成させます。

Step5 ■制限時間は厳守してください。

すべてのプログラムが完成、もしくは制限時間に達したら作業を止めます。

※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。

Step6 ■「タイピングレベル」の確認をします。

『腕試しレベルチェック』のページを開き、レベルチェックを行って下さい。(制限時間：10 分)

<https://www.e-typing.ne.jp/roma/check/>

Step7 ■レベルチェック実施後、結果が出たら、本紙上部の「タイピングレベル」に結果を記入して下さい。

※ 今回のレベルチェックで「A-」以上に達しなかった方は、改めてレベル確認を行いますので、練習を継続してください。

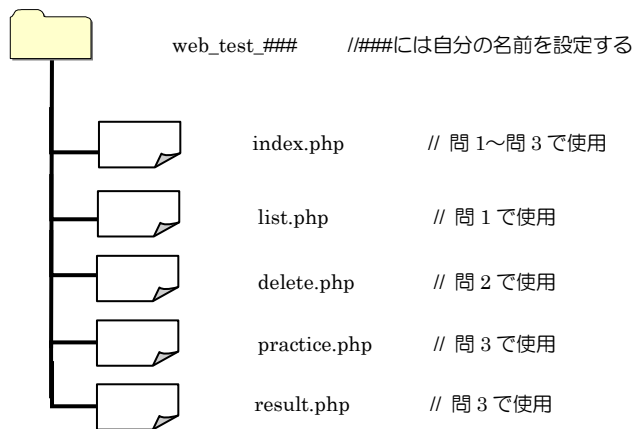
Step8 ■レベル確認が終わったら、本紙を講師に提出してください。

※ 作成したプロジェクトは zip で圧縮し、提出フォームから提出して下さい。

※ 自宅学習講の方は、記入内容に漏れがないか確認してから本ファイルを上書き保存し、プロジェクトと同じフォルダに格納して zip で圧縮し、提出フォームから提出して下さい。

<<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。余力があれば「12.7 データベースの操作を関数を使って共通化してみよう」を参考に、DB 接続情報をまとめたファイルを作成しても構いません。

<<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。

単元

PHP とデータベースの連携

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

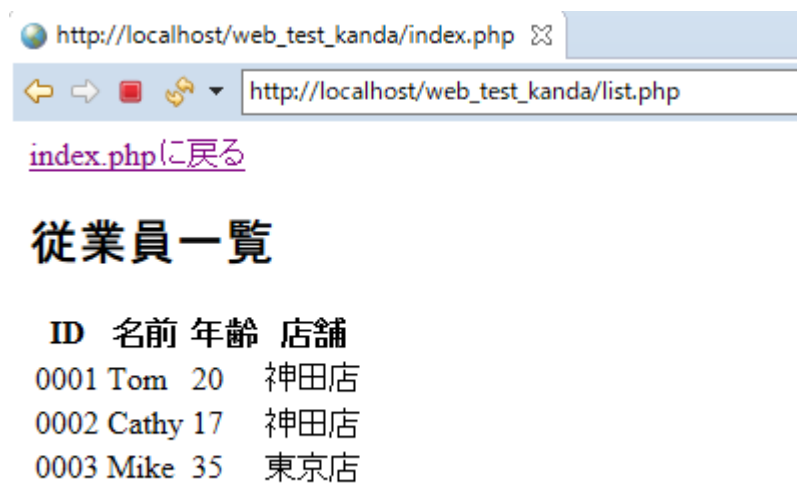
①ソースフォルダ	: web_test_名字のローマ字
③名前	: list.php
④アクセス方法	: index.php の画面から「問 2 従業員の一覧表示」をクリック

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
- ・セッションのメッセージをが登録されている場合、メッセージを受け取って、表示するロジックを追記する

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 **delete.php** を作成しなさい。

概要

employeeinfo テーブルのデータを 1 件削除し、改めて一覧を表示するプログラム。

単元

PHP とデータベースの連携

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 2 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 2 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。

【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「0003」を持つ従業員を削除する場合

http://localhost/web_test_kanda/index.php

index.phpに戻る

従業員の削除

削除対象ID

削除

クリック

削除したい従業員の ID を入力する。

メッセージが表示される

従業員の削除が完了しました！

index.phpに戻る

従業員一覧

ID	名前	年齢	店
0001	Tom	20	神田店
0002	Cathy	17	神田店
<input type="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。

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

概要

データを送信する 3 つの方法を使い、渡された `cmd` の値に応じて結果の文字列を表示するプログラム。

単元

フォームデータの扱い方

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 3 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。

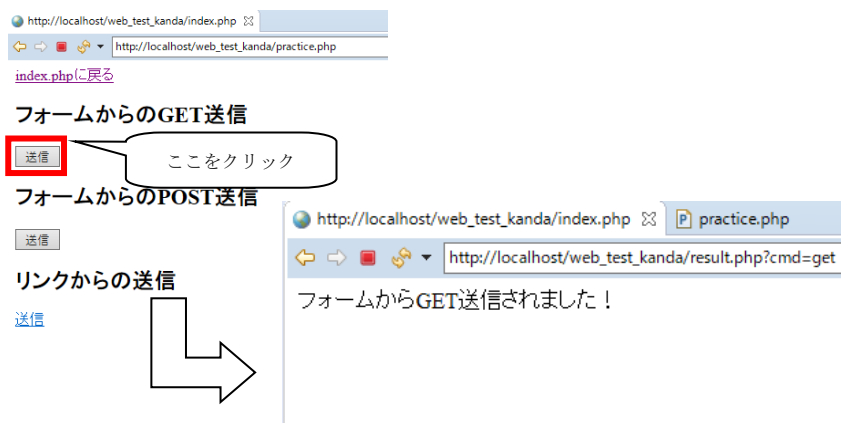
設定するメッセージは以下のようにする。

- 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。