

PHP基礎

フォームを使ってDBと連携しよう

フォームを理解しよう

formとは?

HTML側にformタグを設置し、利用することでPHP側でデータを受け取ることができるようになります。



A sample HTML form with a light blue background. It contains the following fields and elements:

- 説明会の希望日時**: A dropdown menu with the text "選択してください" and a downward arrow.
- 氏名**: A text input field.
- フリガナ**: A text input field.
- メールアドレス**: A text input field with a placeholder "(半角のみ: メールアドレス)".
- 希望コース**: Three radio button options:
 - ☐ DEVコース (週末集中型)
 - ☐ LABコース (フルタイム総合型)
 - ☐ どちらにも興味がある
- 電話番号**: Three text input fields separated by hyphens, with a placeholder "(電話番号は半角数字でご入力下さい)".
- 志望動機**: Five checkbox options:
 - ☐ 起業をしたい (直近に限らず将来的にも)
 - ☐ エンジニアとして就職したい
 - ☐ 現在エンジニアだがWeb系へ転職したい
 - ☐ エンジニアと関わる仕事で知識がほしい
 - ☐ 仕事の幅を広げたい
 - ☐ その他
- 紹介者名 (ご紹介者がいる場合のみ)**: A text input field.
- Submit Button**: An orange button with the text "確認画面へすすむ →".

フォームを理解しよう

formの準備を整えよう!

formを利用する際は、HTMLに<form>タグを利用します。
「action」属性にはデータを渡すURL(xxx.php)を指定します。

index.php

送信先

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

例 // ここにフォームタグを書いていく

```
<input type="text">
```

```
<input type="email">
```

```
</form>
```

PHP 実習！

演習 1

ではここで実際にパソコンで作業を進めていきましょう！

1. `<?php ?>`を最初を書いてphpを利用できるようにする

2. index.phpを作成しよう

3. index.phpにformタグを追加し、action属性をnext.phpに設定しよう

フォームを理解しよう

inputテキストを追加しよう

<form>タグの準備ができたところで次は<input>タグを追加してPHPでデータを 受け取れる準備を整えましょう!

index.php

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

```
//input タグを記述していこう
```

```
<input type="text" name="name">
```

```
</form>
```

入力データに名前をつける

PHP 実習！

演習2

ではここで実際にパソコンで作業を進めていきましょう！

1. 先程作成したformタグの中に<input>タグを追加しよう
2. type属性をtext name属性をname
3. 記述した内容を確認してみよう！

フォームを理解しよう

データを受け取れるようにしよう

inputタグで送る値を次のページ(actionで指定したページ)で受け取るためには `$_POST[''];` で受け取ります。

index.php

```
<?php
// ここは PHP の中で記載する
$name=$_POST['name'];
// 変数で受け取ることで「扱いやすく」する

echo $name;
    ※フォームから受け取るときは $_POST[''];
    で受け取る必要がある！
?>
```

PHP 実習！

演習3

ではここで実際にパソコンで作業を進めていきましょう！

1. next.phpでデータが受け取れる準備をしよう

2. **`$name = $_POST['name'];`** と記載しよう

3. **echo 文**を使って表示されるかチェックしてみよう！

PHP 実習！

演習4

ではここで実際にパソコンで作業を進めていきましょう！

1. グーグルでいろいろな<input>タグを調べてみよう
2. 調べたinputタグ実際に記述して利用できるようにしてみよう！
3. `$xxx = $_POST['xxx'];` ※xxxは自分で変更して受け取れるかチェック
4. **echo 文**を使って表示されるかチェックしてみよう！

※inputタイプの使い方を理解することでDBとの連携の基礎を理解することができます。
自分で調べて書くことで頭の中が整理されます！

データベースとは




DBの意味を理解しよう!

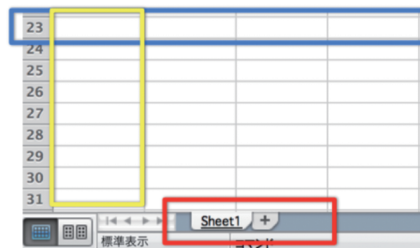
データベースを学ぼう

データベースとは？

データベース(DB)はエクセルのファイルをイメージしてください。
DBを利用することで登録されているデータを自由に扱うことができるようになります。

◇データベースはExcelファイルと似てます。

- ・データベース : Excelで言うとファイルです。
- ・テーブル (表) : ExcelではSheetです。 
- ・レコード (行・ロウ) : Excelでは横の行です。 
- ・フィールド (列・カラム) : Excelでは縦の列です。 



データベースに送る命令=SQL



SQL
→

←
データ



データベースを学ぼう

データベースとExcelの比較

◇ Excel の場合

ファイル名 a_db

Sheet名 a_table

項目名: id

項目名: name

項目名: email

項目名: naiyou

項目名: indate

データベースを学ぼう

データベースとExcelの比較

◇ データベース の場合

データベース名 a_db

テーブル名 a_table

項目名: id int(数値)

項目名: name varchar(文字列)

項目名: email varchar(文字列)

項目名: naiyou varchar(文字列)

項目名: indate datetime(日付)

データベース作成

phpmyadminを利用して作成してみよう

データベースを作成

演習5-1

Database作成 (XAMPP使用した場合)

1. 最初にデータベースを作成します。http://localhost/xampp/
2. 画面に左のメニュー "phpMyAdmin" を選択
3. 『データベース』タブをクリック
4. 『データベースを作成する』に半角英数で任意のデータベース名、
「a_db」を入力し作成。
5. 『照合順序』は utf8_unicode_ci を選択。
6. 『作成』ボタンで作成。



データベースを作成

演習5-2

■ テーブル作成

- DB名 **a_db**
- table名 **a_table**
- Field名

Id int(12) AUTO_INCREMENT PRIMARY_KEY

name var_char(64)

email var_char(128)

naiyou text

Indate datetime

データベース操作

phpmyadminを利用してSQLを体験しよう

SQLを学ぼう

SQLとは?

データベース(DB)はエクセルのファイルをイメージしてください。

DBを利用することで登録されているデータを自由に扱うことができるようになります。

■SQL(データを操作する命令)

INSERT: データを“登録”する事ができます。

SELECT: データを“表示”する事ができます。

UPDATE: データを“更新”する事ができます。

DELETE: データを“削除”する事ができます。

SQLで文字列を扱う際の注意! シングルクォートで囲む。ダブルクォートはNG

SQLを学ぼう

INSERT(データ登録)

書式:

```
INSERT INTO テーブル名(カラム1, カラム2,...) VALUES(値1, 値2,...);
```

```
INSERT INTO a_table(id, name, email, naiyou indate)  
VALUES(NULL, '就職キャンプ1', 'test1@test.test', 'テスト1','2015-06-15 00:00:00');
```

```
INSERT INTO a_table(id, name, email, naiyou, indate)  
VALUES(NULL, '就職キャンプ2', 'test2@test.test', 'テスト2', sysdate() );
```

※文字列を登録する場合**シングルクォート**で囲んで指定します。

SQLを学ぼう

SELECT(データ取得)

書式:

SELECT 表示するカラム FROM テーブル名;

SELECT * FROM a_table; --全指定

SELECT name FROM a_table; --単体指定

SELECT name, email FROM a_table; --複数指定

SELECT * FROM a_table WHERE name = '就職キャンプ1';

※**WHERE** を使用して特定のデータを抽出することが可能です。

SQLを学ぼう

条件付き検索(データ取得)

条件を指定することで更に細かな検索も可能です!

・演算子を使う

SELECT * FROM テーブル名 WHERE

SELECT * FROM テーブル名 WHERE

・AND, OR で検索条件を複数指定する

SELECT * FROM テーブル名 WHERE id = 1 OR id = 2;

SELECT * FROM テーブル名 WHERE id >= 1 AND id <= 3;

・あいまい検索をする

SELECT * FROM テーブル名 WHERE indate LIKE '2015-06%';

SELECT * FROM テーブル名 WHERE email LIKE '%@gmail.com';

SELECT * FROM テーブル名 WHERE email LIKE '%@%';

SQLを学ぼう

条件付き検索(ソートと制限)

書式:

SELECT * FROM テーブル名 ORDER BY ソートしたいカラム名 ***;

SELECT * FROM テーブル名 ORDER BY id DESC;

SELECT * FROM テーブル名 ORDER BY email, name DESC;

※ **DESC**は降順でSORT、**ASC**は昇順でSORT

書式:

SELECT 表示するカラム FROM テーブル名 LIMIT ***;

SELECT * FROM テーブル名 LIMIT 5;

SELECT * FROM テーブル名 LIMIT 3, 5;

SQLを学ぼう

UPDATE(データ更新)

書式:

UPDATE テーブル名 SET 変更データ WHERE 選択データ;

UPDATE a_table SET name= '就職キャンプ1' WHERE id=1;

UPDATE a_table SET name= '就職キャンプ2' WHERE id=2;

※WHERE を使用して特定のデータを更新します

※忘れると全てのデータが 更新されるので知っておきましょう。

SQLを学ぼう

DELETE(データ削除)

書式:

```
DELETE FROM テーブル名;
```

```
DELETE FROM a_table;
```

--WHEREを使用して特定のデータを削除します。

```
DELETE FROM a_table WHERE id=1;
```

※注意)DELETEするとデータは復旧できないので気をつけましょう。

課題 DBを作成しよう

今までのおさらいをしよう

授業で行ったことをおさらいして「データベース」を完成させよう!

必須条件!

1.新しくDBを作成し、その名前を「DB名 **b_db**」「table名 **b_table**」

「フィールド名 **id, name, score, naiyou indate**」

2.次にSQL1というファイルを開き、その中の文章をインサートで全て追加しよう

3.インサートができた人はセレクト表示を試みよう

4.条件付き検索を利用してスコアが高い人順に表示させよう