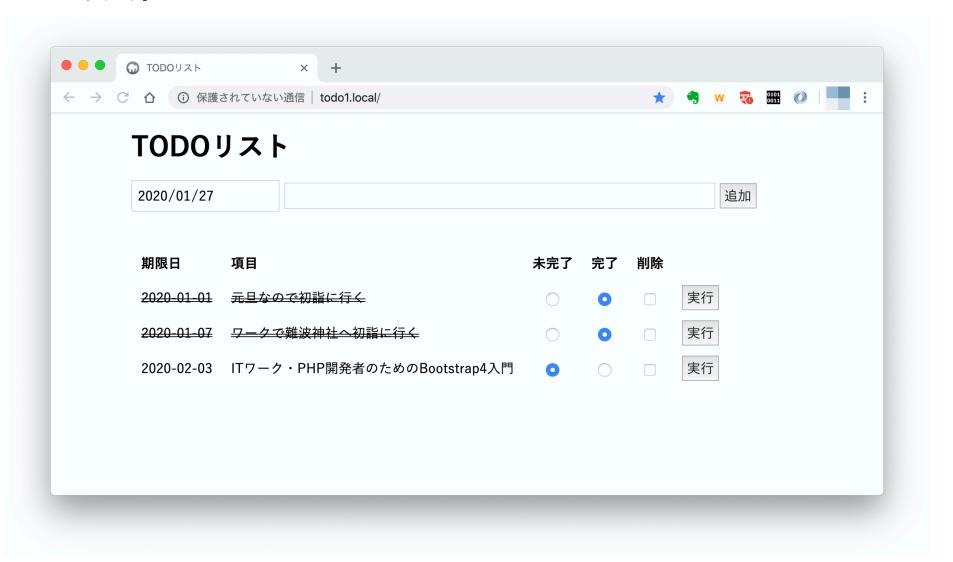
TODOリスト



TODOリスト・仕様(1)

- 個人で使う
- ログイン機能なし
- 入力項目
 - ✔ 期限日

デフォルト値:今日の日付

✔ TODO項目

デフォルト値:空

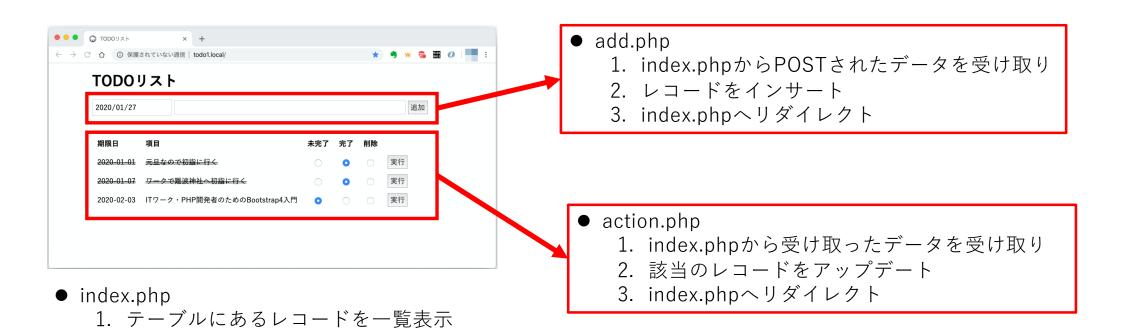
✓ 追加ボタン ボタンをクリックすると、TODO項目が追加される

- 表示項目
 - ✔ 期限日
 - ✓ TODO項目
 - ✔ 完了/未完了ラジオボタン
 - ✔ 削除チェックボックス

TODOリスト・仕様(2)

- 実行ボタンを押すとき
 - ✓ 項目を完了するときは「完了」にチェックを入れる
 - ✓ 項目が完了の場合は、「期限日」「TODO項目」に取り消し線を入れる
 - ✔ 項目を未完了にするときは「未完了」にチェックを入れる
- 一覧表示をするとき
 - ✔ 項目が「完了」のときは、「完了」のラジオボタンが選択状態になっている
 - ✔ 項目が「未完了」のときは、「未完了」のラジオボタンが選択状態になっている。
 - ✓ 削除されている項目は表示しない。
- 削除チェックボックス
 - ✔ 削除する場合はチェックを入れる
 - ✔ 削除したら一覧には表示されない
 - ✓ 実際に削除するのではなく、is_deleted=1にする
- 実行ボタン
 - ✓ 「完了」および「削除」を実行する

TODOリスト・画面遷移と処理内容



TODOリスト・フォームと値の受け取り(1)

HTML

TODOリスト・フォームと値の受け取り(2)フォームの部品

- テキストボックス
 value属性に値を指定することで、初期値を設定できます。
 <input type="text" name="item" value="やらなければならないこと">
- ラジオボタン<input type="radio">

name属性を同じ名前にすることで、同じグループになります。 value属性で、送信する値を設定できます。 「checked」を記載することで、選択状態にすることができます。

<input type="radio" name="complete" value="0" checked>
<input type="radio" name="complete" value="1">

チェックボックス
 value属性で、送信する値を設定できます。
 「checked」を記載することで、選択状態にすることができます。

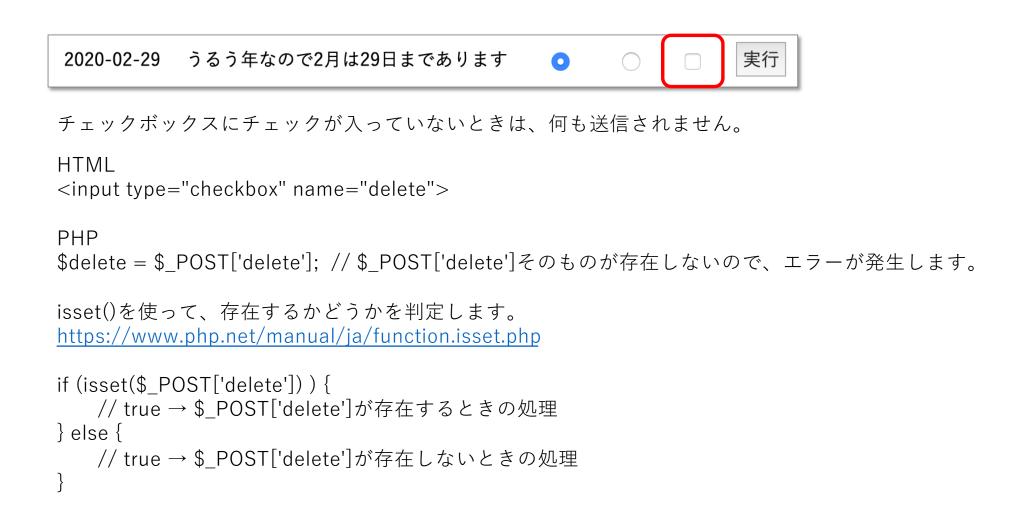
<input type="checkbox" name="delete" value="1">

TODOリスト・フォームと値の受け取り(2) 値の受け取り(1)

フォームから送信された値は、action先のPHPファイルで受け取ることができます。

```
HTML
   <input type="text" name="item" value="やらなければならないこと">
   <input type="radio" name="complete" value="0" checked>
   <input type="radio" name="complete" value="1">
   <input type="checkbox" name="delete" value="1">
PHP(method="post"で送信したとき)
   $name = $_POST['item'];
   $complete = $ POST['complete'];
   $delete = $ POST['delete']:
PHP(method="get"で送信したとき)
   ne = GET['item'];
   $complete = $_GET['complete'];
   $delete = $ GET['delete'];
```

TODOリスト・フォームと値の受け取り(3)値の受け取り(2)



TODOリスト・データベースへの接続とSQLの実行(1)

```
// データベースに接続
$dsn = 'mysql:dbname=todo list;host=localhost;charset=utf8';
$dbh = new PDO($dsn. 'root'. ''): // XAMPPの場合はパスワードなし
$dbh->setAttribute(PD0::ATTR ERRMODE, PD0::ERRMODE EXCEPTION);
// POSTデータをデータベースにインサートする
$sql = '';
$sql .= 'insert into todo items (';
$sql .= 'expiration_date,';
$sql = 'todo item';
$sql .= ') values (';
$sql .= ':expiration_date,';
$sql .= ':todo item';
$sal .= ')';
$stmt = $dbh->prepare($sql);
$stmt->bindValue(':expiration date', $post['expiration date'], PDO::PARAM STR);
$stmt->bindValue(':todo_item', $post['todo_item'], PDO::PARAM_STR);
$stmt->execute():
```

TODOリスト・SQLの実行(2) SELECT文

```
// 登録済みのT0D0リストを取得
$sql = '';
$sql = 'select ';
$sql .= 'where ';
$sql .= 'is_deleted=:is_deleted ';
$stmt = $dbh->prepare($sql);
$stmt->bindValue(':is_deleted', $post['is_deleted'], PDO::PARAM_INT);
$stmt->execute();
$list = $stmt->fetchAll(PD0::FETCH ASSOC);
foreach ($list as $v) {
      echo $v['todo_item']
```

TODOリスト・SQLの実行(3) bindValueメソッド

```
// バインドする値の型がintegerのとき、第3引数に「 PDO::PARAM_INT」を指定する $stmt->bindValue(':intValue', $intValue, PDO::PARAM_INT);

// バインドする値の型が文字列、日付型の文字列のとき、
// 第3引数に「 PDO::PARAM_STR」を指定する $stmt->bindValue(':strValue', $strValue, PDO::PARAM_STR); $stmt->bindValue(':dateValue', $dateValue, PDO::PARAM_STR);
```