



# 福井大学 リカレント 教育プログラム プログラミング応用

---

(3) PHPによる

バックエンドプログラミング (10/22)

# プログラミング応用 講義資料URL

<https://tsaitoh.net/~t-saitoh/2022-10-recp/>

login: guest

password: Guest

## 福井大学リカレント教育プログラム プログラミング応用

### リンク

- [Twitter @TohruSaitoh](#)
- [Facebook tsaitoh.net](#)
- [tsaitoh.net@google.com](mailto:tsaitoh.net@google.com)



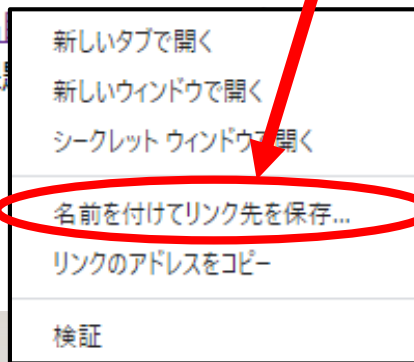
### 講義内容と講義資料

- [Webアプリケーションとプログラム言語\(11/07\)](#)
  - インターネットやWebの仕組みについて理解し、その中でJavaScriptやPHPなどのプログラム言語がどう使われるのか
  - 課題レポート
    1. [理解度確認\(11/07\)](#) (Google Formsに回答してください)
    2. nslookup コマンドで、www.fukui-nct.ac.jp のIPアドレスを調べてください。
    3. そのIPアドレスを使ってWebページを開いてください。  
最近のブラウザは http://x.x.x.x で開くと、「安全が確認できないけど開きますか?」といった警告がでますが、「危険性を理解したうえで開く」を実行してみてください。
    4. 2,3で確認した内容の画面をキャプチャしたものをレポートにまとめ、メールで[tsaitoh@fukui-nct.ac.jp](mailto:tsaitoh@fukui-nct.ac.jp)に提出してください。

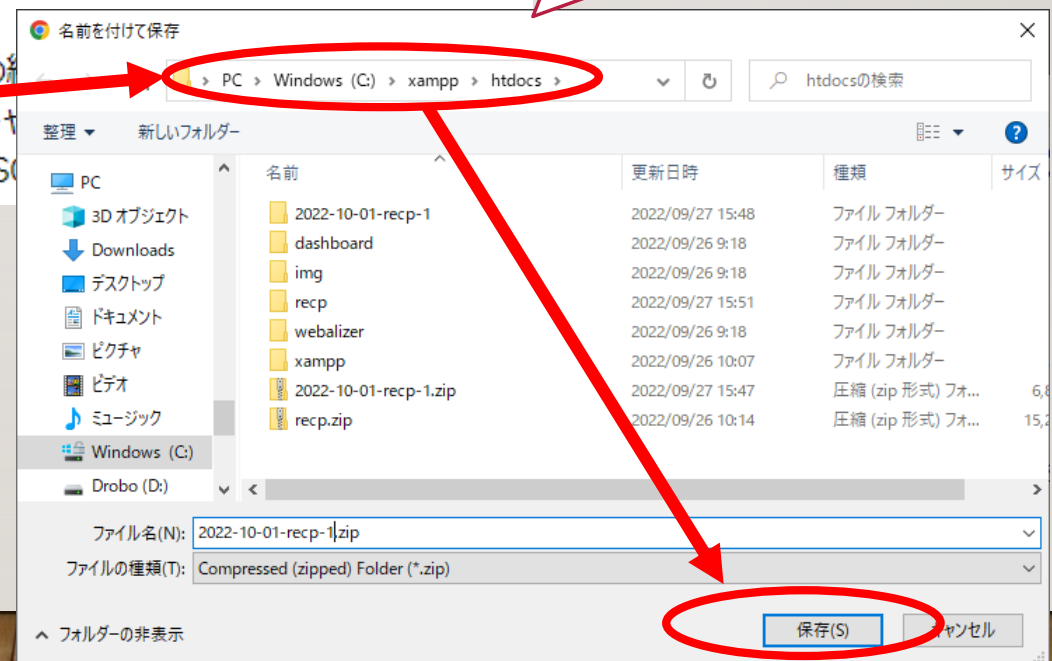
# 配布データのダウンロードと解凍-I

- [PHPによるバックエンドプログラミング\(10/22\)](#)

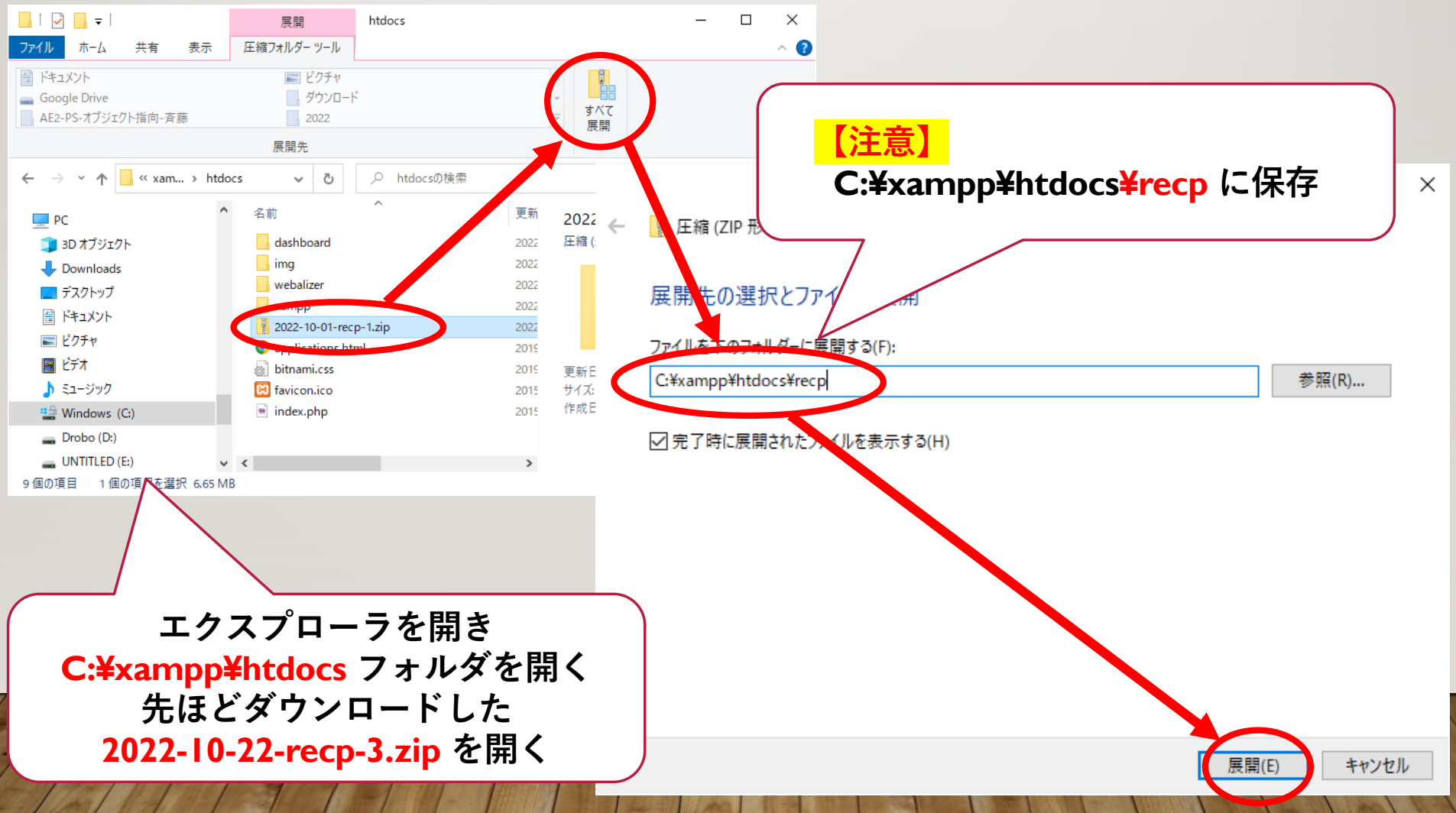
- 配布データ [2022-10-22-recp-3.zip](#)
- Webサーバで動くプログラム言語としての PHP について、 基本的な文法と
- [品](#) (Paiza.IO)
- 課



**C:\xampp\htdocs に保存**



## 配布データのダウンロードと解凍-2



# 本日の目標

## PHPによる

## バックエンドプログラミング(10/22)

---

- **PHP** とは

- PHPによるHello World
- 練習問題(スタイルシート)
- PHPでのデータの受け取り
- HTMLと処理の混在

注文ID	商品名	単価	個数
1010	みかん	50	1
1020	りんご	100	2
1022	パイナップル	1000	3
合計=3250円			発注

- データベース**SQL**とは

- 複数の表の組み合わせで表現
- SQLの使い方
- 表を組み合わせる処理
- 練習問題(複数の表を組み合わせる)
- 複数の表を組み合わせる

- PHPの中でSQLを使う



## 初回資料より抜粋

# サーバでPHPによってページを生成 (一般的なやり方)

- 一般的なやり方

### 商品一覧 HTMLファイル

注文ID	商品名	単価	購入
1010	みかん	50	購入
1020	りんご	100	購入
1022	パイナップル	1000	購入

ブラウザ

商品データベース

Webサーバ  
PHP

### \$itemlist

id	name	price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

```
<?php foreach( $itemlist as $item ) : ?>
<tr>
  <td class="ID">
    <?php echo $item.id ; ?></td>
  <td class="NAME">
    <?php echo $item.name ; ?></td>
  <td class="PRICE">
    <?php echo $item.price ;?></td>
  <td>... </td>
</tr>
<?php endforeach ; ?>
```

HTMLとPHPプログラムが混在

# JavaScriptによってページを生成 (最近のやり方)

## 商品一覧 HTMLファイル

注文ID	商品名	単価	個数
1010	みかん	50	5 個
1020	りんご	100	2 個
1022	パイナップル	1000	1 個
合計 1450 円			

1. JSON形式をもらってHTMLを生成
2. 買い物結果を送信

フロントエンド

HTMLとJavaScript  
プログラムを送信

商品データベース

データ送信  
JSON形式

データ受信  
JSON形式

id	name	price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

## 購入結果

id	count
1010	5
1020	2
1022	1

バックエンド

## 【JSON形式】

JavaScript で扱いやすい  
データ形式

# PHPとは

## PHPによるHELLO WORLD(I)

---

- PHPは、Webサーバと共に開発されてきた。
- HTMLの中に、サーバ側で動かす処理を埋め込む。
- PHPは、HTML部分はそのままブラウザに送られる。
- `<?php~?>`のPHPの部分は、サーバで処理され、`print` などの表示結果がブラウザに送られる。



- 変数の宣言は基本ナシ
  - JavaScript の `let` 命令みたいなのは不要

```
<html>
  <head><title>タイトル</title></head>
  <body>
    <?php print "Hello World" ;?>
  </body>
</html>
```



PHPの命令は <?php ...?> で囲む

## PHPによるHello World(2) sampleD.php

<http://localhost/recp/sampleD.php>  
をブラウザで開く

```
<?php
// 変数は $ マークで始まる。
$message = "Hello World" ;
?>
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8"/>
    <title>Sample Page 13(PHPの基本)</title>
  </head>
  <body>
    <h1>Sample Page 13(PHPの基本)</h1>
    <?php
      // "" の文字列の中の $ 変数は、変数を参照。
      print "<h2>$message</h2>" ;
      // 基本はC言語
      for( $i = 0 ; $i < 5 ; $i++ ) {
        printf( "%d<br />\n" , $i ) ;
      }
    ?>
    <!-- forの別の書き方 -->
    <?php for( $i = 0 ; $i < 3 ; $i++ ) : ?>
      こんにちは。<br/>
    <?php endfor ; ?>
  </body>
</html>
```

変数名は \$ を先頭に付ける

### Sample Page 13(PHPの基本)

Hello World

0  
1  
2  
3  
4

こんにちは。  
こんにちは。  
こんにちは。

# Paiza.IO で PHP プログラミング体験

- [PHPによるバックエンドプログラミング\(10/22\)](#)
  - 配布データ [2022-10-22-recp-3.zip](#)
  - Webサーバで動くプログラム言語としての PHP について
    - 関数と配列の繰り返し(Paiza.IO)
    - オブジェクト配列の参照(Paiza.IO)
    - オブジェクト配列の串刺し(Paiza.IO)

The screenshot displays the Paiza.IO web-based PHP editor. At the top, a green button labeled 'PHP' and a text input field containing '関数と配列の繰り返し' are visible. Below this, the editor window shows a file named 'Main.php' with the following PHP code:

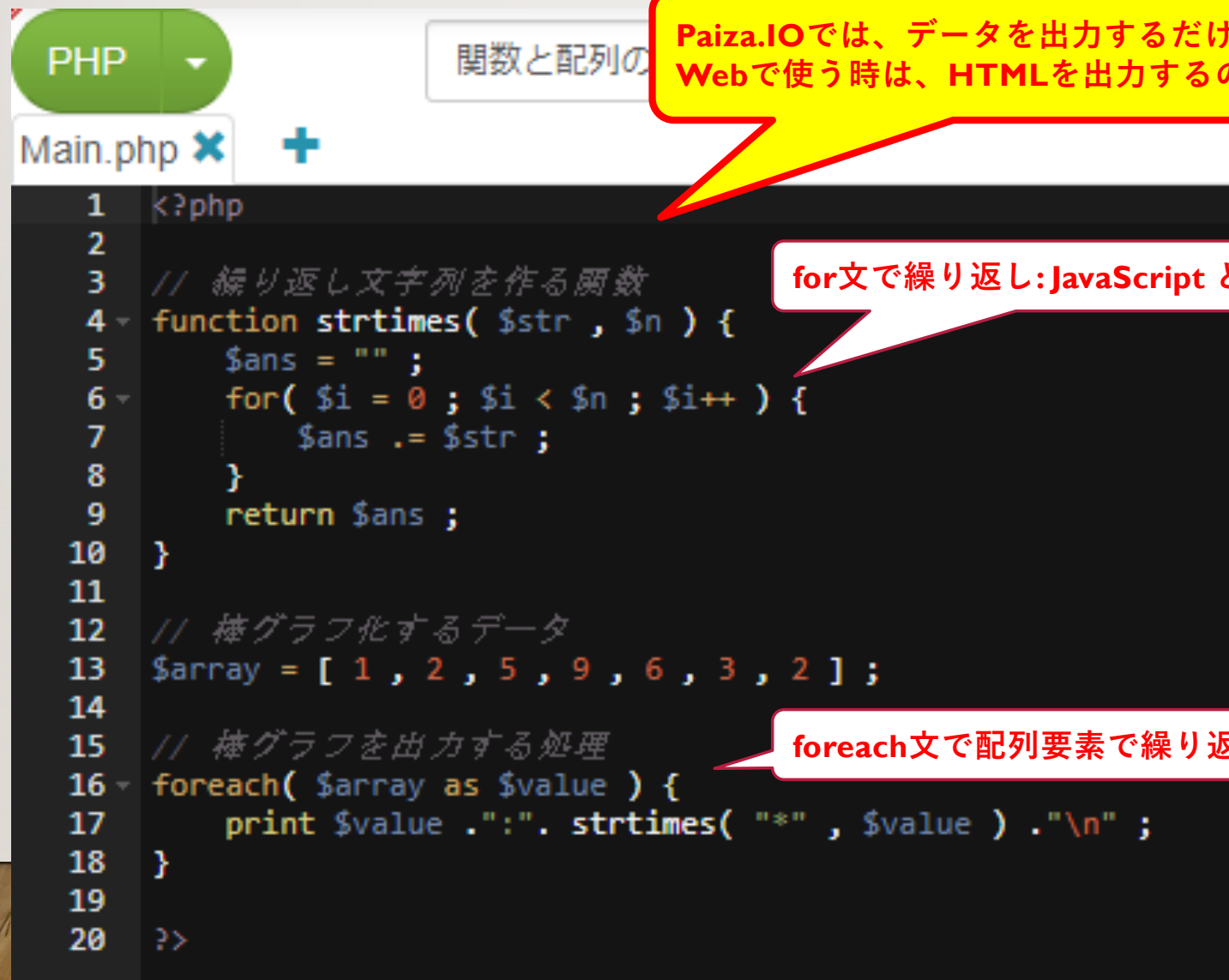
```
1 <?php
2
3 // 繰り返し文字列を作る関数
4 function strtimes( $str , $n ) {
5     $ans = "" ;
6     for( $i = 0 ; $i < $n ; $i++ ) {
7         $ans .= $str ;
8     }
9     return $ans ;
10 }
11
12 // 棒グラフ化するデータ
13 $array = [ 1 , 2 , 5 , 9 , 6 , 3 , 2 ] ;
14
15 // 棒グラフを出力する処理
16 foreach( $array as $value ) {
17     print $value . ":" . strtimes( "*" , $value ) . "\n" ;
18 }
19
20 ?>
```

Below the editor, a green button labeled '実行 (Ctrl-Enter)' is circled in red. To its right, a tab labeled 'PHPを学ぶ | プログラミン' is partially visible. Below the execution button, the output area shows the following text:

```
出力 入力 コメント 0
1:*
2:**
5:*****
9:*****
6:*****
3:***
2:**
```

# Paiza.IOでPHPプログラミング体験

## for文とforeach文で繰り返し



```
1 <?php
2
3 // 繰り返し文字列を作る関数
4 function strtimes( $str , $n ) {
5     $ans = "" ;
6     for( $i = 0 ; $i < $n ; $i++ ) {
7         $ans .= $str ;
8     }
9     return $ans ;
10 }
11
12 // 棒グラフ化するデータ
13 $array = [ 1 , 2 , 5 , 9 , 6 , 3 , 2 ] ;
14
15 // 棒グラフを出力する処理
16 foreach( $array as $value ) {
17     print $value . ":" . strtimes( "*" , $value ) . "\n" ;
18 }
19
20 ?>
```



Paiza.IOでは、データを出力するだけ。  
Webで使う時は、HTMLを出力するのが定番

for文で繰り返し: JavaScript と同じ

foreach文で配列要素で繰り返し

# Paiza.IOでPHPプログラミング体験

## オブジェクト配列の参照


Main.php  

```
1 <?php
2
3 // 品目オブジェクトの配列
4 $item_list = [
5     [ "id" => 1010 , "name" => "みかん" , "price" => 50 ],
6     [ "id" => 1020 , "name" => "りんご" , "price" => 500 ],
7     [ "id" => 1022 , "name" => "パイナップル" , "price" => 1000 ],
8 ];
9
10 // 全ての品目の繰り返し
11 foreach( $item_list as $item ) {
12     print "id=". $item["id"] ;
13     print ",name=". $item["name"] ;
14     print ",price=" . $item["price"] ;
15     print "\n" ;
16 }
17
18 ?>
```

オブジェクトは キー と 値 を書き並べる  
[ キー1 => 値1 , キー2 => 値2 ... ]

foreach 文で オブジェクトの配列を1つずつ参照

オブジェクトの要素は  
オブジェクト[ キー ]で参照



# Paiza.IOでPHPプログラミング体験

## オブジェクト配列の串刺し

```
3 // 品目のオブジェクト配列
4 $item_list = [
5     [ "id" => 1010 , "name" => "みかん" ,      "price" => 50 ] ,
6     [ "id" => 1020 , "name" => "りんご" ,      "price" => 100 ] ,
7     [ "id" => 1022 , "name" => "パイナップル" , "price" => 1000 ] ,
8 ];
9
10 // 購入のオブジェクト配列
11 $buy_list = [
12     [ "id" => 1010 , "count" => 5 ] ,
13     [ "id" => 1020 , "count" => 3 ] ,
14     [ "id" => 1022 , "count" => 1 ] ,
15 ];
16
17 // 合計を求める処理
18 $sum = 0 ;
19 foreach( $item_list as $item ) {
20     print "id=".    $item["id"] ;
21     print ",name=". $item["name"] ;
22     print ",price=".$item["price"] ;
23
24     // 串刺し処理
25     foreach( $buy_list as $buy ) {
26         if ( $item["id"] == $buy["id"] ) {
27             $sum += $item["price"] * $buy["count"] ;
28             print "\t累計=".$sum."\n" ;
29         }
30     }
31 }
```



## PHPによるHello World(2)

### 練習問題 sampleD-table.php

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8"/>
    <title>Sample Page 13-2(連想配列のデータベース)</title>
  </head>
  <body>
    <?php
      $item_list = [
        [ "id" => 1010 , "name"=> "みかん" ,      "price" => 50   ] ,
        [ "id" => 1020 , "name"=> "りんご" ,       "price" => 100  ] ,
        [ "id" => 1022 , "name"=> "パイナップル" , "price" => 1000 ] ,
      ] ;
    ?>
    <h1>Sample Page 13-2<br/>(連想配列のデータベース)</h1>
    <table border="1">
      <tr><td>id</td><td>name</td><td>price</td></tr>

      <?php foreach( $item_list as $item ) : ?>

        <tr>
          <td><?php print $item["id"] ; ?></td>
          <td><?php print $item["name"] ; ?></td>
          <td><?php print $item["price"] ; ?></td>
        </tr>

        <?php endforeach ; ?>

      </table>
    </body>
  </html>
```

連想配列は、  
JavaScript は : で区切る  
PHP は => で区切る

配列の繰り返しは  
foreach( 配列 as 要素 ) :  
処理...  
endforeach ;

\$item が繰り返し毎に、  
みかん,りんご,パイナップルを  
変えながら実行

**練習問題**  
前回のsampleA.htmlを参考に、  
スタイルシートを使ってみよう

# PHPによるデータの受け取り (I)

---

- HTMLの<form ...>は、サーバにデータを送るための  
入力フォームを作るタグ

<form method="GET" action="プログラム">

<input type="text" size="10" name="A" />

<input type="submit" value="実行" />

</form>

テキスト

type="text" name="A"

実行

type="submit"

実行を押す(submit)と、  
プログラムを呼び出す。  
\$\_REQUEST["A"]に  
入力内容を入れて実行。

## PHPによるデータの受け取り(2) sampleE.php

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8"/>
    <title>Sample Page 14(データの受け取り)</title>
  </head>
  <body>
    <h1>Sample Page 14<br/>(データの受け取り)</h1>
    <form method="GET" action="sampleE.php">
      <input type="text" size="5" name="A" />
      +
      <input type="text" size="5" name="B" />
      <input type="submit" value="計算" />
    </form>
    <?php
      // formで記入された値は $_REQUEST[] で参照できる
      if ( $_REQUEST["A"] != "" && $_REQUEST["B"] != "" )
        // 文字列の連結は . ピリオド
        print $_REQUEST["A"]
              . "+" . $_REQUEST["B"]
              . "=" . ( $_REQUEST["A"] + $_REQUEST["B"] ) ;
      } else {
        print "数値の足し算を行います" ;
      }
    ?>
  </body>
</html>
```

### Sample Page 14 (データの受け取り)

A 123 + B 345 計算

数値の足し算を行います

123 + 345 = 468

GETメソッドは、URL欄でA,Bの値が見える  
<https://...../sampleE.php?A=123&B=345>

# PHPによるデータの受け取り (3)

## sampleF.php

---

テキスト (1行)	<code>&lt;input type="text" name="..." /&gt;</code>
パスワード入力	<code>&lt;input type="password" name="..." /&gt;</code>
隠れた値	<code>&lt;input type="hidden"</code> <code>name="..." value="..." /&gt;</code>
テキスト (改行あり)	<code>&lt;textarea name="..."</code> <code>cols="..." rows="..."&gt;</code> <code>&lt;/textarea&gt;</code>

名前(text)	<input type="text" value="t-saitoh"/>
パスワード(password)	<input type="password" value="....."/>
隠れた値(hidden)	<input type="hidden"/>
ご意見(textarea)	<div></div>

演習環境で  
**sampleF.html**  
をみてください

# PHPによるデータの受け取り (4)

## sampleF.php

---

ラジオボタン

1つを選ぶ

```
<input type="radio"  
      name="..." value="" />
```

チェックボックス

複数を選ぶ

```
<input type="checkbox"  
      name="..." value="" />
```

プルダウンメニュー

```
<select name="...">  
  <option value="...">メニュー</option>  
</select>
```

性別(radio)	男性：○ 女性：○
ペット(checkbox)	犬：□ 猫：□ 鳥：□
職業(select+option)	職業を選んでください▼
(submit)	処理

演習環境で  
**sampleF.html**  
をみてください



# PHPによるデータの受け取り(5)

## 値はデータベースに要保存

---

- PHPでは、ブラウザで入力した値を、画面を表示
- 入力した値は、受け渡されただけ。
- 値を後で使いたい時は、**データベースに保存が必要**
- 保存しないと  
ブラウザの画面を閉じたら値は消滅。



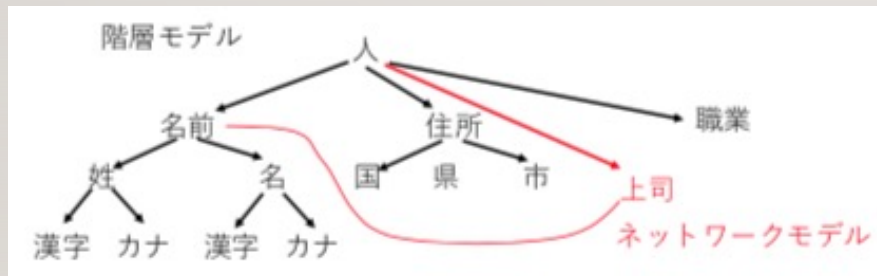
**データベースが必須**

# データベースとは

## (リレーショナルデータベース)



- データベースとは、検索や蓄積が容易にできるよう整理された情報の集まり。
- 階層モデル
  - LDAP



- 関係モデル すべてを表で表す
  - リレーショナルDB

### 関係モデル

学生			科目			
学生ID	学生名	学年	科目ID	科目名	担当	単位
31411	斉藤	4年	2111	情報構造論	斉藤	2
53213	田中	2年	2403	電磁気学	高久	2
:	:	:	:	:	:	:

### 受講

学生ID	科目ID	点数
31411	2111	96
53213	2403	83
31411	2403	65
:	:	:

不整合がおこらない  
複数の表で管理する

概念スキーマ

### 学生が科目を受講

外部からみるとわかりやすい表

学生ID	学生名	学年	科目ID	科目名	担当	単位	点数
31411	斉藤	4年	2111	情報構造論	斉藤	2	96
53213	田中	2年	2403	電磁気学	高久	2	83
31411	斉藤	4年	2403	電磁気学	高久	2	65
:	:	:	:	:	:	:	:

外部スキーマ

不整合の原因

# 複数の表の組み合わせで表現

品目リスト (ITEMLIST)

<u>i_id</u>	i_name	i_price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

顧客リスト (USERLIST)

<u>u_id</u>	u_name	u_age
10001	とおる	56
20002	ともこ	45
20003	あゆか	20

購入リスト (BUYLIST)

<u>i_id</u>	<u>u_id</u>	b_date	b_count
1010	10001	2021-11-01	5
1020	10001	2021-11-01	3
1020	20002	2021-10-25	2
1022	20003	2021-10-30	1

外部キー

外部キー

ダメなデータベース

同姓同名を避けるため  
顧客番号の方がいい

品目	単価	顧客	購入日	個数
みかん	50	とおる	2021-11-01	5
りんご	100	とおる	2021-11-01	3
りんご	100	ともこ	2021-10-25	2
パイナップル	1000	あゆか	2021-10-30	1

顧客名が変更になったら  
複数のレコードを修正

# SQLの使い方(I)

## 基本命令

- SQLは、データベースを扱うためのプログラム言語
- PHPなどの言語と連携して使う

品目リスト(ITEMLIST)

i_id	i_name	i_price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

表を作る命令

```
create table ITEMLIST (  
    i_id    integer    primary key ,  
    i_name  varchar(20) not null ,  
    i_price integer    not null  
);
```

データを登録する命令

```
insert into ITEMLIST ( i_id , i_name , i_price )  
values          ( 1010 , 'みかん' , 50 ) ;  
  
insert into ITEMLIST ( i_id , i_name , i_price )  
values          ( 1020 , 'りんご' , 100 ) ;
```

重複しないキー

# SQLの使い方(2)

## SELECT文

---

- `select` 命令で、表の中から目的のデータを探す

`select` 表示する内容 (射影)  
`from` 対象のテーブル (結合)  
`where` 条件 ; (選択)

((すべての要素を出力))

```
select * from ITEMLIST ;
```

((IDが1010のデータだけ出力))

```
select * from ITEMLIST where i_id=1010 ;
```

((単価が500円以上の品名を出力))

```
select i_name from ITEMLIST where i_price >= 500 ;
```

品目リスト(ITEMLIST)

i_id	i_name	i_price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000



# Paiza.IO でデータベースの練習

- [PHPによるバックエンドプログラミング\(10/22\)](#)
  - 配布データ [2022-10-22-recp-3.zip](#)
  - Webサーバで動くプログラム言語としての PHP について、基本的な文法
  - 品目データベース sampleG-itemlist.sql(Paiza.IO)
  - 品目・顧客・購入データベース sampleG.sql(Paiza.IO)
  - 課題

MySQL 品目・顧客・購入データベース

Main.sql

Success Tweet Share 0

```
1 -- 品目リスト(実体) --
2
3 create table ITEMLIST (
4     i_id integer primary key , -- 品目ID --
5     i_name varchar( 20 ) not null , -- 品目名 --
6     i_price integer not null -- 品目単価 --
7 );
8
9 insert into ITEMLIST ( i_id , i_name , i_price ) values( 1010 , 'みかん' , 50 );
10 insert into ITEMLIST ( i_id , i_name , i_price ) values( 1020 , 'りんご' , 100 );
11 insert into ITEMLIST ( i_id , i_name , i_price ) values( 1022 , 'パイナップル' , 1000 );
12
13 -- ここに SQL を記述 --
14 select * from ITEMLIST ;
15
```

Run (Ctrl-Enter)

Output Input Comments 0

i_id	i_name	i_price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

この部分を書き換える

# SQLの使い方(3)

## 有名なDBシステム

---

- ネットワーク対応の一般的なデータベースシステム
  - Oracle -- 有料 大量データを分散システム用 🏆
  - MySQL -- 無料→有料? 小規模システムでよく使われる
  - データベース利用者の制限なども充実

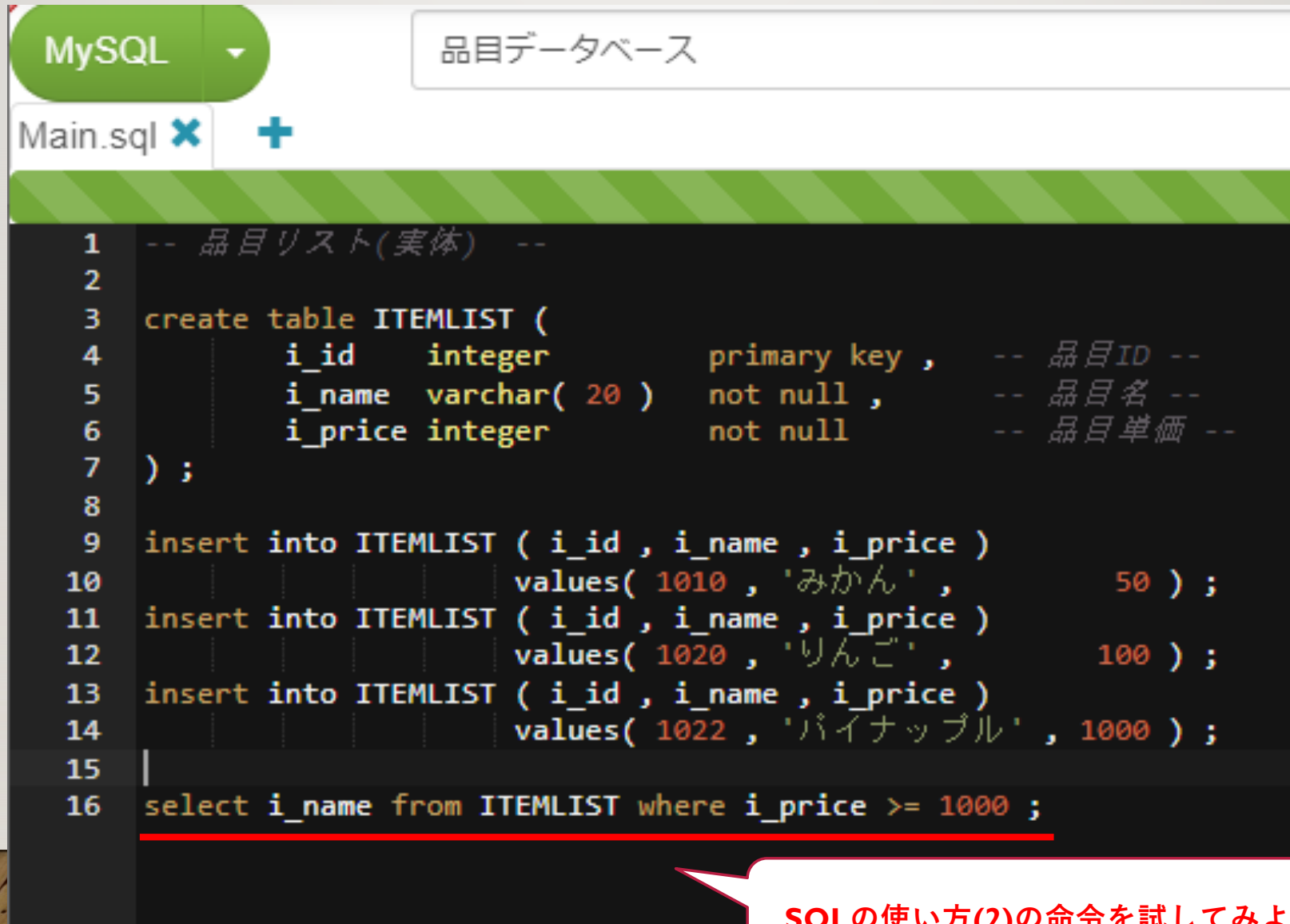


- 簡単なデータベース (ネットワーク非対応)
  - SQLite3 -- 無料



# SQLの使い方(4)

## 実際に使ってみよう



The screenshot shows a MySQL command-line interface. At the top, there is a green button labeled 'MySQL' and a text box containing '品目データベース'. Below this, there is a tab labeled 'Main.sql' with a close button (X) and a plus sign (+). The main area displays a series of SQL commands, numbered 1 through 16. The commands are as follows:

```
1  -- 品目リスト(実体)  --
2
3  create table ITEMLIST (
4      i_id      integer      primary key ,      -- 品目ID --
5      i_name    varchar( 20 ) not null ,      -- 品目名 --
6      i_price   integer      not null         -- 品目単価 --
7  ) ;
8
9  insert into ITEMLIST ( i_id , i_name , i_price )
10     values( 1010 , 'みかん' ,          50 ) ;
11  insert into ITEMLIST ( i_id , i_name , i_price )
12     values( 1020 , 'りんご' ,          100 ) ;
13  insert into ITEMLIST ( i_id , i_name , i_price )
14     values( 1022 , 'パイナップル' , 1000 ) ;
15
16  select i_name from ITEMLIST where i_price >= 1000 ;
```

A red underline is drawn under the final SELECT statement. A red speech bubble points to this line with the text: SQLの使い方(2)の命令を試してみよう

# 複数の表を組み合わせる(I)

品目リスト (ITEMLIST)

<u>i_id</u>	i_name	i_price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

顧客リスト (USERLIST)

<u>u_id</u>	u_name	u_age
10001	とおる	56
20002	とどこ	45
20003	あゆか	20

購入リスト (BUYLIST)

<u>i_id</u>	<u>u_id</u>	<u>b_date</u>	b_count
1010	10001	2021-11-01	5
1020	10001	2021-11-01	3
1020	20002	2021-10-25	2
1022	20003	2021-10-30	1

複数の表のデータベースの練習をしましょう。

- [PHPによるバックエンドプログラミング\(10/22\)](#)
  - 配布データ [2022-10-22-recp-3.zip](#)
  - Webサーバで動くプログラム言語としてのPHPについて、基本的な文法
  - 品目データベース [sampleG-itemlist.sql](#)(Paiza.IO)
  - 品目・顧客・購入データベース [sampleG.sql](#)(Paiza.IO)
  - 課題レポート

# 複数の表を組み合わせる(2) 練習問題

試してみよう

顧客リスト(USERLIST)

<u>u_id</u>	u_name	u_age
10001	とおる	56
20002	ともし	45
20003	あゆか	20

購入リスト(BUYLIST)

<u>i_id</u>	<u>u_id</u>	b_date	b_count
1010	10001	2021-11-01	5
1020	10001	2021-11-01	3
1020	20002	2021-10-25	2
1022	20003	2021-10-30	1

```
40  
41 insert into BUYLIST( i_id , u_id , b_date , b_count )  
42 insert into BUYLIST( i_id , u_id , b_date , b_count )  
43 insert into BUYLIST( i_id , u_id , b_date , b_count )  
44 insert into BUYLIST( i_id , u_id , b_date , b_count )  
45  
46 -- ここに実行したい SQL を記述 --  
47 select * from ITEMLIST ;  
48
```

顧客の情報を全て表示するSQL

select [ ] from [ ] ;

50歳以上の顧客の名前を表示するSQL

select u\_name from [ ]  
where [ ] ;

2022年11/1の購入情報を表示

select \* from [ ]  
where [ ] = '2022-11-01' ;



## 複数の表の組み合わせる(3)

テーブルの  
串刺し

品目リスト(ITEMLIST)

i_id	i_name	i_price
1010	みかん	50
1020	りんご	100
1022	パイナップル	1000

購入リスト(BUYLIST)

i_id	u_id	b_date	b_count
1010	10001	2021-11-01	5
1020	10001	2021-11-01	3
1020	20002	2021-10-25	2
1022	20003	2021-10-30	1

外部キーが一致している場所

```
select ITEMLIST.i_name , BUYLIST.b_count
```

```
from ITEMLIST, BUYLIST
```

```
where ITEMLIST.i_id = BUYLIST.i_id ;
```

ITEMLISTと  
BUYLISTの  
全ての組合せ  
を作る

```
46 -- ここに実行したい SQL を記述 --
47 select ITEMLIST.i_name , BUYLIST.b_count
48       from ITEMLIST, BUYLIST
49       where ITEMLIST.i_id = BUYLIST.i_id ;
50 |
51
```

Run (Ctrl-Enter)

Output Input Comments 0

```
i_name  b_count
みかん   5
```

ITEMLIST.i\_idと  
BUYLIST.i\_idが  
同じ所だけ抽出

# 複数の表を組み合わせる(4) 練習問題

複数の検索条件では  
where A and B

品目リスト(ITEMLIST)

i_id	i_name	i_price
I010	みかん	50
I020	りんご	100
I022	パイナップル	1000

顧客リスト(USERLIST)

u_id	u_name	u_age
U0001	とおる	56
U0002	ともこ	45
U0003	あゆか	20

購入リスト(BUYLIST)

i_id	u_id	b_date	b_count
I010	U0001	2021-11-01	5
I020	U0001	2021-11-01	3
I020	U0002	2021-10-25	2
I022	U0003	2021-10-30	1

穴埋めしてください

((購入があった品目と日付は?))

```
select IТЕMLIST.i_name, BUYLIST.b_date  
from _____, BUYLIST  
where _____.i_id = BUYLIST.i_id;
```

((とおるが買い物した日付は?))

```
select BUYLIST.b_date  
from USERLIST, _____  
where USERLIST.u_id = BUYLIST.u_id  
and USERLIST.u_name = "とおる";
```

((何を誰がいつ買いましたか?))

```
select IТЕMLIST.i_name, USERLIST.u_name, BUYLIST._____  
from IТЕMLIST, USERLIST, BUYLIST  
where IТЕMLIST.i_id = BUYLIST._____  
and USERLIST.u_id = BUYLIST.u_id ;
```

# PHPの中でSQLを使う (I)

## sampleH.php

演習環境で  
**sampleH.php**  
をみてください

- sampleD-table.php のプログラムをデータベースを参照するように修正

### 【 sampleD-table.php の品目の設定処理 】

```
<?php
$item_list = [
    [ "id" => 1010 , "name"=> "みかん" ,      "price" => 50  ] ,
    [ "id" => 1020 , "name"=> "りんご" ,      "price" => 100 ] ,
    [ "id" => 1022 , "name"=> "パイナップル" , "price" => 1000 ] ,
] ;
?>
```

### 【 sampleH.php はデータベースから読み込み 】

```
<?php
// データベースで読み込んだ内容で商品リストを表示
$dbh = new PDO( "sqlite:shopping.db" ) ;
$sql = "select i_id,i_name,i_price from ITEMLIST" ;
$item_list = $dbh->query( $sql , PDO::FETCH_ASSOC ) ;
?>
```

# PHPの中でSQLを使う (2)

## sampleH.php

演習環境で  
**sampleH.php**  
をみてください

```
// PHPからSQLのデータベースに接続
$dbh = new PDO( "sqlite:shopping.db" );

// 実行したいSQL命令の文字列
$sql = "select i_id, i_name, i_price from ITEMLIST ;" ;

// データベースに問い合わせ
$item_list = $dbh->query( $sql , PDO::FETCH_ASSOC ) ;
```

**PDO::FETCH\_ASSOC** 形式では、以下のような結果で参照できる

```
[ [ "i_id"=>1010 , "i_name"=>"みかん" , "i_price"=>50  ] ,  
  [ "i_id"=>1020 , "i_name"=>"りんご" , "i_price"=>100 ] ,  
  ... ]
```

問い合わせ以外の処理の場合は、**exec(...)**を使う

```
$dbh->exec( "insert into ... " ) ;
```

# PHPの中でSQLを使う (3)

## sample1.php

演習環境で  
sample1.php  
をみてください

```
<table border="1">
  <tr><td>i_id</td><td>i_name</td><td>i_price</td><td>b_count</td></tr>

  <?php foreach( $item_list as $item ) : ?>

  <tr>
    <td><?php print $item["i_id"] ; ?></td>
    <td><?php print $item["i_name"] ; ?></td>
    <td><?php print $item["i_price"] ; ?></td>
    <td>
      <?php
        // <input type=text name="b_count[1001]" />
        print "<input type='text' size='4' name='"
          . "b_count[" . $item["i_id"] . "]"
          . "' />" ;
      ?>
    </td>
  </tr>

  <?php endforeach ; ?>
  <tr><td colspan="4" align="right"><input type="submit" value="発注" /></td></tr>
</table>
```

**\$item は 品目の連想配列**

```
[ "i_id"    => 1010 ,
  "i_name"  => "みかん" ,
  "i_price" => 50  ]
```

**品目に対応した購入数の  
<input type="text">を仕込む**



# PHPの中でSQLを使う (4)

## sample1.php

演習環境で  
**sample1.php**  
をみてください

```
<?php
if ( isset( $_REQUEST["b_count"] ) ) {
    // ターミナルで、recp フォルダ, shopping.db への書き込み許可が必要
    // > chmod 777 ~/public_html/recp
    // > chmod 666 ~/public_html/recp/shopping.db
    // $dbh = new PDO( "sqlite:shopping.db" );
    foreach( $_REQUEST["b_count"] as $i_id => $b_count ) {

        // 作成するSQL命令: insert into BUYLIST(i_id,...) values(1001,...)
        $sql = "insert into BUYLIST"
            . " (i_id,u_id,b_date,b_count)"
            . " values($i_id, '$user[u_id]', '$user[b_date]', $b_count);" ;

        // 本当はSQLを exec( $sql ) で実行したいけど、
        // 書き込み権限の設定が面倒なので、SQLを確認するだけ
        print "$sql\n<br/>" ;
        // $ans = $dbh->exec( $sql ) ;
    }
}
?>
```


# JavaScript のフロントエンドと PHP のバックエンドの組み合わせ(I)

- PHPで生成したWebページだけでは、  
表示を細かく制御できない。
- JavaScript でフロントエンド(見栄えの制御)  
PHP でデータベースのやり取り→バックエンド
- JavaScript に、JSON形式でSQLのデータを渡す  
PHPプログラム

演習環境で  
**sampleJ.php**  
をみてください

```
// HTML形式でなく、JSON形式を返すことを指定
header( "Content-Type: application/json; charset=utf-8" );

// データベースで読み込んだ内容で商品リストを表示
$dbh = new PDO( "sqlite:shopping.db" );
$sql = "select i_id,i_name,i_price from ITEMLIST" ;
$item_list = $dbh->query( $sql , PDO::FETCH_ASSOC );
print json_encode( $item_list->fetchAll() );
```



```
[{"i_id":1010,"i_name":"¥u307f¥u304b¥u3093","i_price":50},
{"i_id":1022,"i_name":"¥u30d1¥u30a4¥u30ca¥u30c3¥u30d7¥u30e0"}]
```

# JavaScript のフロントエンドと PHP のバックエンドの組み合わせ(2)

演習環境で  
**sampleK.html**  
をみてください

```
<script type="text/javascript">
// このプログラムは、sampleC.html を
// sampleJ.php によって出力されたJSONファイルを読み込むように
// 書き換えたものです。
$(function() {
    $.getJSON( "sampleJ.php", function(item_list) {
        let text = "<table border='1'>" ;
        text += "<tr><th>id</th><th>name</th><th>price</th></tr>" ;
        for( let item of item_list ) {
            text += "<tr>"
                + "<td>"+item.i_id+"</td>"
                + "<td>"+item.i_name+"</td>"
                + "<td>"+item.i_price+"</td></tr>" ;
        }
        document.getElementById( "output" ).innerHTML
            = text ;
    })
}) ;
</script>
```

JSON形式で  
品目リストを返す  
PHP に変更

ITEMLISTの  
属性名に変更

# まとめ

---

- PHPは、HTMLの中にサーバで動くプログラムを記述

```
<?php print $item ; ?>
```

- 簡単なしくみで、ブラウザでの入力を扱える
- PHPは、値の保存にデータベースシステムが必要
- SQLは、データベースに命令するための言語

```
select * from ITEMLIST  
where i_price >= 100 ;
```

- 複数の表を組み合わせたデータを取り出せる
- PHPとSQLを組み合わせる使う