SVG 資料第 9 回目 (その 2)

クライアントとサーバーでのデータのやり取りの基本

メディア専門ユニット I(SVG)

2017/6/20

データを入力する

次のファイルを start.php とする。

```
 1<!DOCTYPE html>
 2<html>
 3<head>
 4<meta charset="utf-8"/>
 5<title>簡単なデータのやり取り</title>
 6</head>
 7<body>
 8  <form method="POST" action="hello.php">
 9    <input type="text" name="user"/>
10    <input type="submit" value="送信"/>
11  </form>
12</body>
13</html>
```

データを入力する–解説

- 4 行目でファイルが UTF-8 でエンコーディングされていることを示している
- 8 行目から 11 行目で入力フォームを定義
- `action` では `submit` ボタンが押されたときに呼び出される関数またはサーバー側のファイルを指定
- ここでは `hello.php` が呼び出される。

データを入力する–実行方法

- start.php を適当なサーバーにコピー
- ブラウザーで start.php を表示
- このファイルには PHP のプログラム部分がまったくないので拡張子を html にしてもよい
- テキストボックスにデータを入力した後、隣にある「送信」ボタンをクリック

データを受け取る


```
 1<!DOCTYPE html>
 2<html>
 3<head>
 4<meta charset="utf-8"/>
 5<title>簡単なデータのやり取りの結果</title>
 6</head>
 7  <body>
 8<?php
 9  if($_POST['user']!='') {
10    print("ようこそ{$_POST['user']}さん\n");
11  } else {
12?>
13  <form method="POST" action="start.php">
14    <input type="submit" value="戻る"/>
15  </form>
16<?php
17  }
18?>
19  </body>
20</html>
```

データを受け取る–解説

- hello.php を適当なサーバーにコピー
- start.php の<form>要素で属性属性 method を POST にしている (8 行目) のでこのフォームに含まれる input で指定されている値はスーパーグローバル変数 $_POST の連想配列のなかに入れられる。
- 値は$_POST['user'] で参照可能
- 9 行目で入力された文字があるかどうかの判定
- 空の文字列ではないときには 10 行目の文が実行されるのでメッセージを表示
- 空の文字列の場合には 13 行目から 15 行目までで「戻る」ボタンが表示
- このボタンを表示する部分は HTML の要素で記述
- その後に PHP で書かれている 9 行目の else 節のを閉じるために 16 行目から 18 行目で閉じるための } を挿入

やってみよう

- 上記のプログラムを実行した結果を報告
- 9 行目の method=P̈OST¨を method=P̈UT¨に変更し hello.php の 9 行目と 10 行目の$_POST を$_GET に変更したもので同じように実行させて動作することを確認
- method=P̈OST¨との method=P̈UT¨による違いがあるか調べる。

サーバーに伝えられる情報 (配布資料 214 ページ)


```
 1<!DOCTYPE html>
 2<html>
 3  <head>
 4    <meta charset="UTF-8"/>
 5    <title>サーバーに伝えられる情報</title>
 6    <link rel="stylesheet" type="text/css" href="table.css">
 7  </head>
 8  <body>
 9    <div class="table">
10<?php
11foreach($_SERVER as $key => $val) {
12  print <<<_EOL_
13      <div class="Row">
14        <div class="Cell">$key</div>
15        <div class="Cell">$val</div>
16      </div>
17_EOL_;
18}
19?>
20    </div>
21  </body>
22</html>
```

サーバーに伝えられる情報 (解説)

- 6 行目で外部の CSS ファイル (table.css) を読み込む。
  - <link>要素の属性 href でファイルを指定
  - 属性 rel で stylesheet を指定
- サーバーに伝えられる情報はスーパーグローバル変数 $_SERVER の連想配列内にある
- この配列内のキーをすべてわたるために foreach($Array as $key=>$val) 形式の構文を用いる。
  - $key にはキーの値、$val にはそれに対応する配列の値が入れられる (変数名は何でもよい)
  - $Array[$key]=$val の関係が成立

サーバーに伝えられる情報 (解説つづき)

これらを並べてテーブルの形で出力 (12 行目から 17 行目)

- 12 行目の<<<はヒアドキュメントと呼ばれる形式で文字列を定義することを示す。
- この後の文字列_EOL_はこの文字列の最後を示すためのマークを定義
- 同じ文字列が 17 行目にある。ここで終了となる。
- この文字列は行の先頭に置く。先頭に空白があってはいけない。
- 終了の文字列以外には分の終了を示す; 以外は書けない。
- 14 行目と 15 行目にある変数は展開される。

CSS ファイル

6 行目で読み込まれる CSS ファイル

```
 1.head {
 2                font-size:20px;
 3}
 4.table {
 5    display:table;
 6}
 7.Row{
 8    display:table-row;
 9}
10.Cell {
11    display:table-cell;
12    text-align:center;
13}
```

CSS ファイル (解説)

<table>要素を用いないで表組する CSS となっている。

- クラス table は<table>要素をまねるために属性 display に table を指定
- クラス Row は<tr>要素をまねるために属性 display に table-row を指定
- クラス Cell は<td>要素をまねるために属性 display に td を指定
- さらにテキストを中央揃えに指定

サーバーに伝えられる情報 (確認しよう)

これらの結果を見てどのようなデータが送られているか調査すること