

●foreach(フォーイーチ)の利用

- ・配列から順番に要素を取り出して処理する繰り返し構文

```
<?php
    $data = [98, 79, 56, 83, 69];
    $total = 0;
    foreach($data as $item){
        $total += $item;
    }
    echo "データの合計は、{$total}です。";
?>
```

と
り
あ
え
ず
\$item
に
格
納

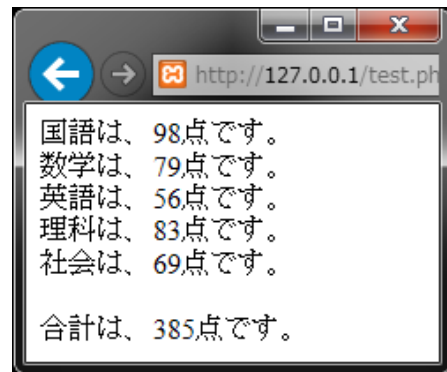
\$data[0]	98
\$data[1]	79
\$data[2]	56
\$data[3]	83
\$data[4]	69

●連想配列で foreach

```
<?php
    $data = [
        '国語' => 98,
        '数学' => 79,
        '英語' => 56,
        '理科' => 83,
        '社会' => 69];
    $total = 0;
    foreach($data as $key => $item){
        echo "{$key}は、{$item}点です。<br>";
        $total += $item;
    }
    echo "<br>合計は、{$total}点です。";
?>
```

連想配列の作成

← 配列から順番に取り出してと
りあえずの名前をつける



●関数の定義と利用

```
<?php
    echo '<table>';
    $data = getData('hanako');
    printData($data);
    $data = getData('taro');
    printData($data);
    echo '</table>';

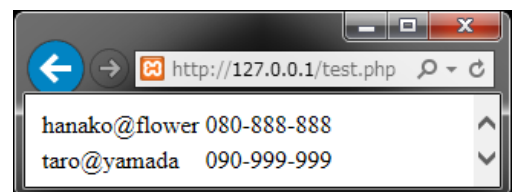
    function getData($name){
        $data = [
            'taro' => ['taro@yamada', '090-999-999'],
            'hanako' => ['hanako@flower', '080-888-888'],
            'sachiko' => ['sachiko@happy', '070-777-777'],
            'jiro' => ['jiro@ramen', '060-666-666']
        ];
        return $data[$name];
    }

    function printData($arr){
        echo "<tr><td>{$arr[0]}</td><td>{$arr[1]}</td></tr>";
    }
?>
```

引数付き関数呼び出し→戻り値：'taro@yamada', '090-999-999'

その戻り値を引数として printData 関数に渡す

関数内部で2つに分けて表示



●クエリーパラメータ（問い合わせ(る)・媒介変数)の利用

Web ページに情報(検索ワード など)を入力して、サーバに渡すと、それに応じた結果が返ってくる。



http://〇〇〇〇〇.com/〇〇?XX=XX&XX=XX&XX=XX... という形になっている。

http://ドメイン/パス?キー=値&キー=値&キー=値... キー=値 が繋がって渡される。

上の例だと、`r = 宇多田ヒカル` & `t = DISTANCE` となっている。これがクエリパラメータ。

●サーバ側でクエリパラメータを受け取るには

`$変数=$_GET[キー]`; キーを指定すると、その値が`$変数`に格納される。

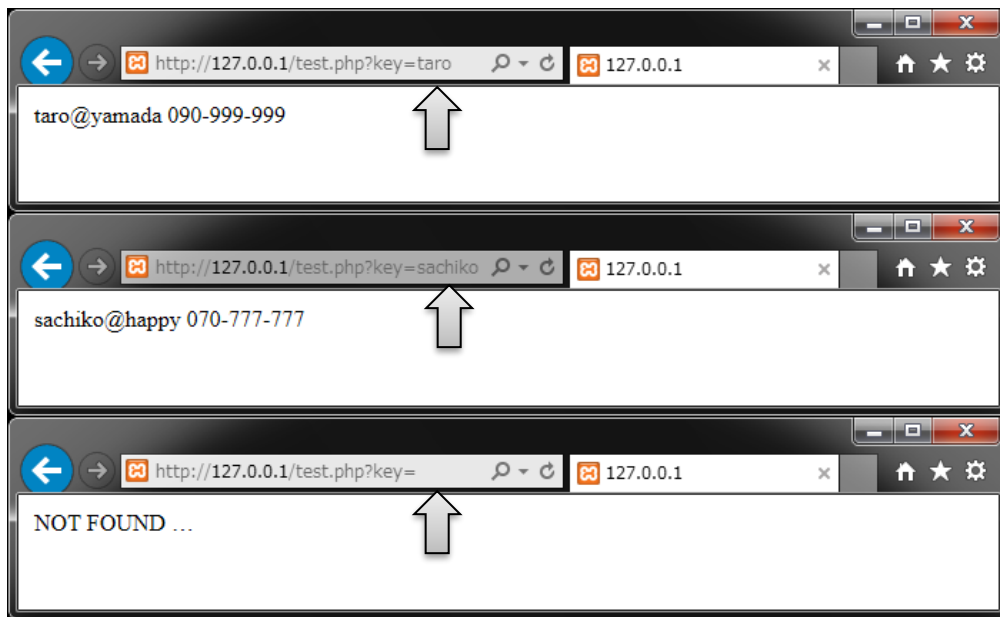
`<?php`

```
$key=$_GET['key'];
echo '<table>';
$data = getData($key);
printData($data);
echo '</table>';

function getData($key){
    $data = [
        'taro' => ['taro@yamada','090-999-999'],
        'sachiko' => ['sachiko@happy','070-777-777'],
    ];
    if(isset($data[$key])){          ←$data[$key]に値がセットされていたら
        return $data[$key];
    }else{
        return ['NOT FOUND','...']; ←$keyが無い場合
    }
}
```

```
function printData($arr){
    echo "<tr><td>{$arr[0]}</td><td>{$arr[1]}</td></tr>";
}
```

`?>`



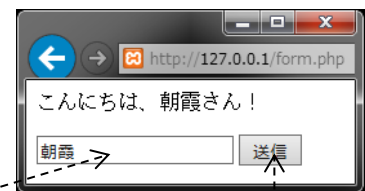
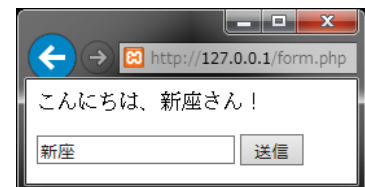
ブラウザで
key=●●●●
を指定すると、
それに応じた
表示になる。

←該当が無い
場合も対応する
必要がある。

入力された値(文字列)によって、サーバが表示するものを変える。これが「サーバサイドプログラミング」

●フォームの利用(ファイル名: form.php)

```
<?php
    $msg = '';
    $result = 'お名前は?';
    if(isset($_POST['msg'])){ ←もし送信データがあったら
        $msg = $_POST['msg']; ←送信データを取得する
        $result = "こんにちは、{$msg}さん!";
    }
?>
<body>
    <p><?php echo $result; ?></p>
    <form method="post" action="form.php">
        <input type="text" name="msg" value="<?php echo $msg; ?>">
        <input type="submit" value="送信">
    </form>
</body>
```

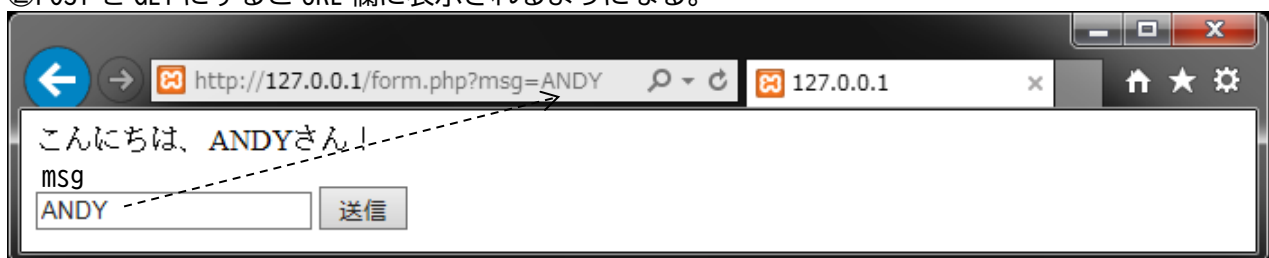


テキスト枠

ボタン

《特徴》

- ①POST で送信すると、送った内容がブラウザ上(URL 欄)に表示されない。
- ②POST を GET にすると URL 欄に表示されるようになる。



- ③<form>から</form>までに挟まれた内容が form.php に送られる。
- ④form タグで、method=送信方法、action=送信された内容进行处理するファイル名を指示する。
- ⑤送信ボタンでブラウザ上に入力したデータを読み取ってサーバに送信する。
- ⑥送信側と取得側を別々のファイルにしても良い。

●いろいろなフォーム（視聴者投稿フォームを例に）

```
<?php
    echo $_POST['nae']. "<br>";
    echo $_POST['ndai']. "<br>";
    echo $_POST['seibetu']. "<br>";
    echo $_POST['kanso']. "<br>";
    echo "-----<br>";
?>
<body>
    <form method="post" action="form.php">
        お名前:<input type="text" name="nae"><br> ※1行テキスト入力
        年 代:<select name="ndai"> ※プルダウンメニューから選択
            <option value="10代">10代</option>
            <option value="20代">20代</option>
            <option value="30代">30代</option>
            <option value="40代">40代</option>
            <option value="50代">50代</option>
        </select><br>
        性 別:<input type="radio" name="seibetu" value="男" checked>男 ※ラジオボタンで選択
            <input type="radio" name="seibetu" value="女">女<br>
        ご感想:<br> ※複数行のテキスト入力
            <textarea name="kanso" rows="4" cols="40"></textarea><br><br>
        <input type="submit" value="送信"> ※送信ボタン
    </form>
</body>
```

※ピリオドで文字列連結可
 入力結果表示（ファイルに保存することも可）
 ↓
 CSV形式なら EXCEL でも表示/編集できる！

※太字は画面表示用であり、サーバには送信されない。

このようにいろいろな部品を組み合わせ、利用者からデータを収集する仕組みを作る。検索ワードやお問合せ、要望・ご意見・クレーム・プチ情報などを取得することにより、【双方向の意見交換】が可能となる。HTML しか無かった時代は発信者の一方通行だけだったが、現在は WEB 上で閲覧者からの声を収集し、HP の改善・商品の品質向上・新商品の開発などに活かすことが出来る。しかし、閲覧者の中には前向きな事を言わない者もいたり、HP 自体を改ざんする者もいる。また、送信ボタンで一度送信したデータは簡単には取り消せないの送信には注意が必要である。