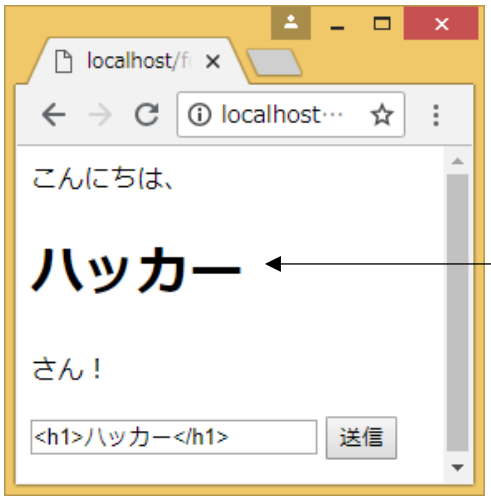
●クロスサイトスクリプティング(XSS)攻撃に備える

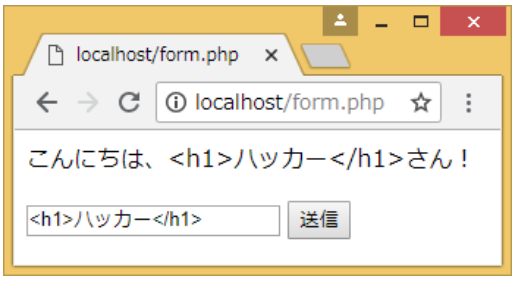


アクセスしてきた人が悪意を持って書き込む場合がある。
HTML や JavaScript に詳しい人間がテキスト枠にコードを入力し、画面を汚したり、作成者の意図しない動作をさせたりする。これに対応する。

←ﾃｷｽﾄ枠に HTML タグを一緒に埋め込むと反映されてしまう。
└→この場合：<h1>

そうならないように変更する。

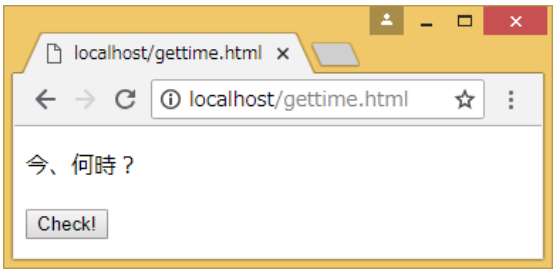
```
<p><?php echo htmlspecialchars($result); ?></p>
<form method="post" action="form.php">
        <input type="text" name="msg" value="<?php echo htmlspecialchars($msg); ?>">
        <input type="submit" value="送信">
</form>
```

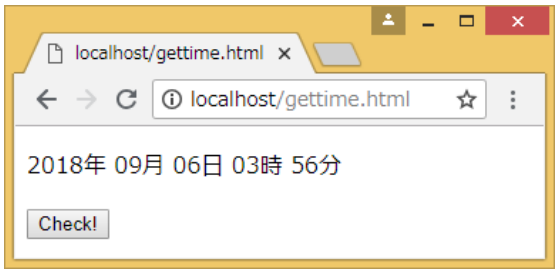


こうすることにより、タグとして認識しなくなる。
└→ただの半角文字として表示させている。

┌→通常はﾌﾞﾗｳｻﾞを再読込させる

●Javascript からサーバにアクセスする（Ａｊａｘ(エージャックス)通信）：ﾌｫｰﾑをﾘﾛｰﾄﾞしないで画面を更新する





Check!をクリックすると　　瞬時に現在時刻が表示される

手順１）gettime.html が開かれる→Check!がクリックされる。
手順２）gettime.js が起動する→gettime.php が起動する。
手順３）gettime.php が現在時刻を表示する→HTML に反映される。

html(入力用)、Javascript(通信用)、PHP(表示用)が連動して動く。

ﾌﾞﾗｳｻﾞの機能が向上したので、このような事が出来る。→古いﾌﾞﾗｳｻﾞは未対応。

①gettime.html

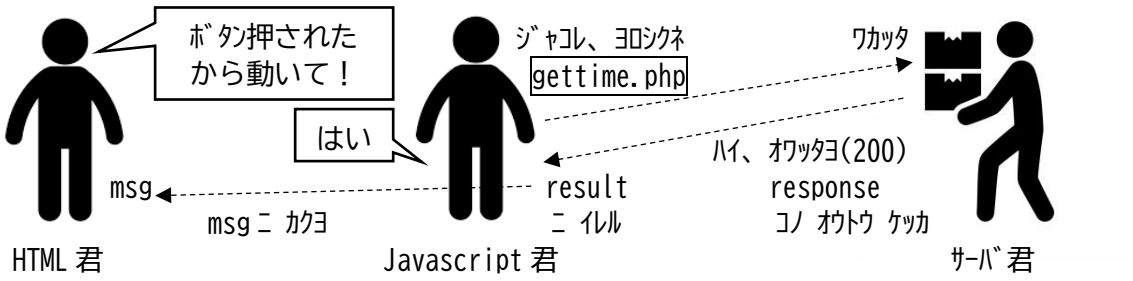
```
<!DOCTYPE html>
<html jang="ja">
<head>
        <script src="gettime.js"></script>
</head>
<body>
        <p id="msg">今、何時？</p>
        <button onclick="onClickBtn();">Check!</button> ←ﾎﾞﾀﾝが押されたら onClickBtn 関数
</body>
</html>
```

②gettime.js

```
function onClickBtn(){
        var xhr = new XMLHttpRequest();          ←http 通信要求を NEW する
        xhr.open('GET','/gettime.php',true);     ←接続する(ｱｸｾｽ方式,ｱｸｾｽ先,非同期)
        xhr.onload = function(e){
                if(this.status == 200){  ←XMLHttpRequest の status が 200 なら通信完了
                        var result = this.response;      ←通信結果は result へ
                        var msg = document.querySelector('#msg');  ←HTML の DOM 指定
                        msg.textContent = result;  ←結果を HTML 内へ書き込む
                }
        };
        xhr.send();      ←最後に「送信」する。（※忘れがちなので注意！）
}
```

③gettime.php

```
<?php
echo date("Y 年 m 月 d 日 h 時 i 分");
?>
```



●Ajax 通信を使ってフォームの内容を POST 送信する

①ajaxform.html

```
<!DOCTYPE html>
<html jang="ja">
<head>
        <script src="ajaxform.js"></script>
</head>
<body>
        <p id="msg">※番号を入力</p>
        <input type="text" id="number">                    ←ﾃｷｽﾄ枠の名前は number
        <button onclick="onClickBtn();">Check!</button>
</body>
</html>
```

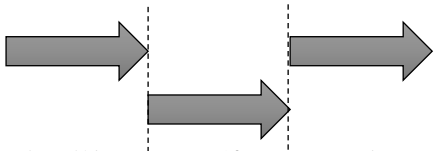
②ajaxform.js

```
function onClickBtn(){
        var val = document.querySelector('#number').value; ←number の部分の値
        var form = new FormData();                          ←Form 用ｵﾌﾞｼﾞｪｸﾄを NEW
        form.append('number',val);                          ←Form に number を追加
        var xhr = new XMLHttpRequest();                             (number=?)
        xhr.open('POST','/ajaxform.php',true);              ←POST で接続
        xhr.onload = function(e){
                if(this.status == 200){
                        var result = this.response;
                        var msg = document.querySelector('#msg');  ←HTML 内場所特定
                        msg.textContent = result;           ←HTML 書換え
                }
        };
        xhr.send(form);                                     ←form を送信
}
```

③ajaxform.php

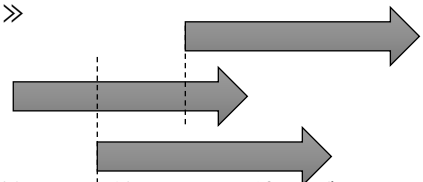
```
<?php
        $data = [
                "taro,taro@yamada,090-999-999",
                "hanako,hanako@flower,080-888-888",
                "sachiko,sachiko@happy,070-777-777"
        ];
        $msg = '123';
        if(isset($_POST['number'])){            ←通常の FORM の受け方と同じ方法で
                $n = $_POST['number'] * 1;                  ←1 をかけて'数字'にする
                if($n < 0){ $n = 0; }                       ←ﾏｲﾅｽだったら 0 にする
                if($n >= count($data)){ $n = count($data)-1; }  ←ﾃﾞｰﾀ数より大なら最大で
                $msg = $data[$n];                           ←該当の配列番号を msg へ
        }
        echo $msg;
?>
```

≪同期≫



通信が終わるまで待ってから次へ

≪非同期≫



通信を開始したら終わるまで待たずにどんどん先へ