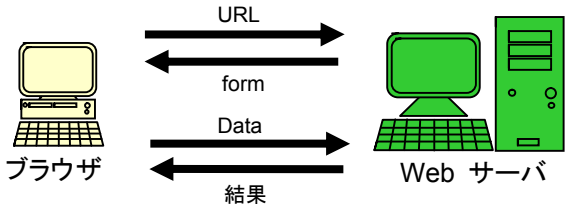
Web システム論 2

1. Form

Web サーバにデータを送る.



1.1. GET メソッド

- データを URL の一部として送る.
- 長さ制限あり.
- 直接ユーザがデータを確認, 操作できる.

http://el.mml.tuis.ac.jp/moodle/blocks/autoattend/index.php?course=14&class=0&from=home

1.2. POST メソッド

- データを HTTP のボディデータとして送る.
- サイズ制限なし.
- 直接ユーザがデータを確認, 操作することはできない. ツールを使えば可能.

GET メソッド

```
<html>

<head>
<meta http-equiv="content-type" content="text/html; charset=EUC-JP" />
<title>GET Test of CGI</title>
</head>


<body bgcolor = #COCOCO>
<br />
<h1> GET メソッドによるデータの転送</h1>
<br />

<form action="get_post.cgi" method="GET">
    <p />ラジオボタン<br />
    <input type="radio" name="Q1" value="A" /> ラジオボタン１<br />
    <input type="radio" name="Q1" value="B" /> ラジオボタン２<br />
    <input type="radio" name="Q1" value="C" /> ラジオボタン３<br />
    <input type="radio" name="Q1" value="D" /> ラジオボタン４<br />
    <input type="radio" name="Q1" value="E" /> ラジオボタン５<br />
    <br /><br />
    <p />チェックボックス<br />
    <input type="checkbox" name="Q2" value="1" /> チェックボックス１<br />
    <input type="checkbox" name="Q2" value="2" /> チェックボックス２<br />
    <input type="checkbox" name="Q2" value="3" /> チェックボックス３<br />
    <input type="checkbox" name="Q2" value="4" /> チェックボックス４<br />
    <input type="checkbox" name="Q2" value="5" /> チェックボックス５<br />
    <br /><br />

    <p />テキスト<br />
    <input type="text" name="TXT" value=""><br />
    <br /><br />

    <p />テキストエリア<br />
    <Textarea name="TXTA" cols=50 rows=3 align=bottom></textarea><br />
    <br /><br />

    <input type ="submit" value="送信！">
    <input type ="reset"  value="リセット"><br />
</form>

</body>
</html>
```

POST メソッド

```
<html>

<head>
<meta http-equiv="content-TYPE" content="text/html; charset=EUC-JP" />
<title>PUT Test of CGI</title>
</head>


<body bgcolor = #COCOCO>
<br />
<h1> POST メソッドによるデータの転送</h1>
<br />

<form action="get_post.cgi" method="POST">
    <p />ラジオボタン<br />
    <input type="radio" name="Q1" value="A" /> ラジオボタン１<br />
    <input type="radio" name="Q1" value="B" /> ラジオボタン２<br />
    <input type="radio" name="Q1" value="C" /> ラジオボタン３<br />
    <input type="radio" name="Q1" value="D" /> ラジオボタン４<br />
    <input type="radio" name="Q1" value="E" /> ラジオボタン５<br />
    <br /><br />

    <p />チェックボックス<br />
    <input type="checkbox" name="Q2" value="1" /> チェックボックス１<br />
    <input type="checkbox" name="Q2" value="2" /> チェックボックス２<br />
    <input type="checkbox" name="Q2" value="3" /> チェックボックス３<br />
    <input type="checkbox" name="Q2" value="4" /> チェックボックス４<br />
    <input type="checkbox" name="Q2" value="5" /> チェックボックス５<br />
    <br /><br />

    <p />テキスト<br />
    <input type="text" name="TXT" value=""><br />
    <br /><br />

    <p />
    テキストエリア<br />
    <textarea name="TXTA" cols=50 rows=3 align=BOTTOM></textarea><br />
    <br /><br />

    <input type ="submit" value="送信！">
    <input type ="reset"  value="リセット"><br />
</form>

</body>

</html>
```

1.3 セキュリティ上の注意（サーバ側）

・ セッションを維持する場合はセッション管理を行う．安易にクッキーなどで誤魔化さない．

・ 受け取ったデータは必ずサニタイジング（無害化）する．etc.

XSS，SQL インジェクション

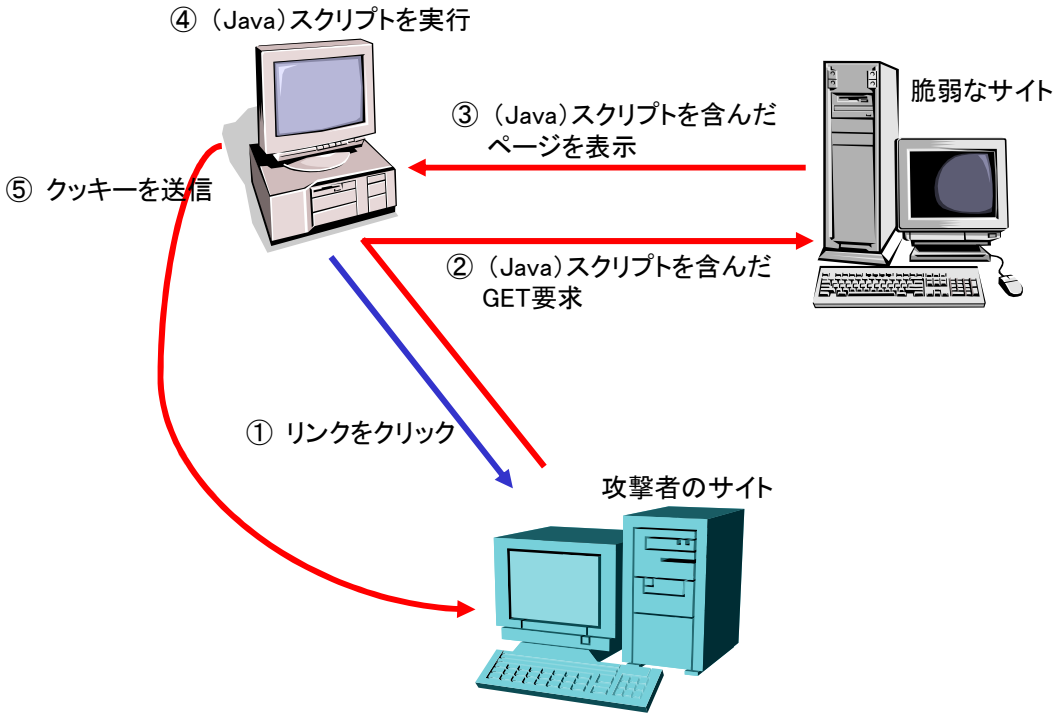
1.3.1 XSS（クロスサイトスクリプティング）

もしネット上に脆弱なサイト（入力に任意のスクリプトを流し込み，そのサイトのページの一部のように見せることが可能なサイト）があったとする．攻撃者はこのサイトにスクリプトを流し込むようなリンクを作成し，一般ユーザにこのリンクを踏ませる（クリックさせる）．その結果，一般ユーザがその脆弱なサイトに対してそのスクリプトを流し込んだことになる．

通常この流し込むスクリプトには，攻撃者へcookieを転送するものや，偽のページを表示させるものなどが多い．

WEBサーバの対策：入力データに対するサニタイジング（無害化） ただし条件により完全ではない．

ユーザの対策：脆弱なサイトを利用しない．怪しいサイトにアクセスしない．



1.3.2 SQL Injection（SQL の挿入）

例えば，サーバ側でSQL を実行する場合，SQL 文として

$SQL="SELECT * FROM user WHERE user='$userid' AND pass='$passwd'";

が用意されていたとする．ここで，$user として適当な文字

列(hoge)，$passwd として unknown’ or ’A’=’A を入力したとすれば，SQL 文は

SELECT * FROM user WHERE user='hoge' AND pass='unknown’ or ‘A’=’A’

で，WHERE節の条件はいつでも真となり，データベースへのアクセスが可能となる．

対策：入力データのサニタイジングを行う．(PHP の場合は addslashes() 関数など．この場合‘ は ¥’に変換される)

http://www.netp.tuis.ac.jp/lecture/tomcat/sql_name.html