

メディア情報応用 第5週

Advanced Topics in Media Informatics Week 05

金沢工業大学 情報フロンティア学部
メディア情報学科

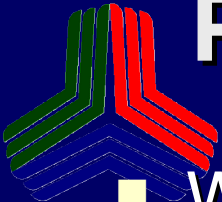
山岸 芳夫

Yoshio Yamagishi

Dept. of Media Informatics

Col. of Informatics and Human Communication

Kanazawa Inst. Tech.



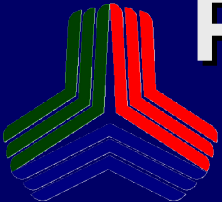
PHP(PHP Hypertext Preprocessor)

- Webアプリケーションの作成に用いられるスクリプト言語。データベースとの連携も容易。

A kind of script language, which is mainly used to develop web applications. The connection of the application to database can be easily done.

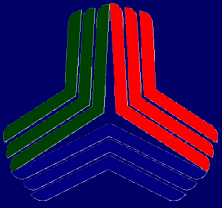
- CGIとして使うこともできるが、通常は**phpモジュール**をWebサーバに組み込み、CGIを介さず直接WebサーバがPHPスクリプトを実行させるように設定して用いる。

Although CGI program can be written in PHP, usually the Web server is configured to use built-in **php module** instead of CGI when it executes PHP scripts.



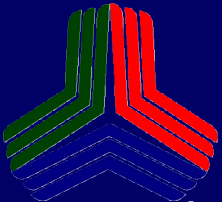
PHP(PHP Hypertext Preprocessor)

- PHPもそれ自身様々なモジュールを組み込むことで機能拡張が可能。データベースとの連携を行うためには **PDOモジュール**を用いるのが主流。
The function of PHP can be extended when some modules are included. Currently a **PDO module** has been mainly used to connect to database.
- PHPにどのようなモジュールが組み込まれているのかを知るには、phpinfo()関数を使う。
phpinfo() function shows the list of modules included in the PHP.



MariaDB

- MySQL互換の完全にオープンソースのデータベース。
基本的にMySQLの命令などがそのまま使える。
A MySQL-compatible open source relational database.
Basically the main commands of MySQL can be used.
- 今回はデータベースを使ったチャットプログラムを作るため、まず最初にデータベースを作成する。
At first we build database for “chat” web application.



データベースの作成

Database Creation

- 端末を開き、mysql コマンドでMariaDBサーバに接続する。
Open the terminal and connect MariaDB server with “mysql” command
- まずchat用データベースを作成する。
Firstly we create a database for chat application.

```
CREATE DATABASE chat DEFAULT CHARACTER SET utf8;
```

- 続いてユーザの作成、権限の設定を行う。
Then, we add a user with appropriate privileges.

```
GRANT ALL PRIVILEGES ON chat.* TO chat@localhost IDENTIFIED by 'chat';
```

データベースの作成(2)

Database Creation

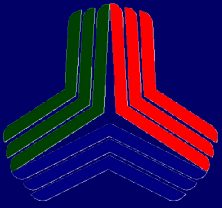
- いったんここでquitして、以降はchatユーザーでログインして作業する。パスワードを聞かれたら chat と入力しよう。

At this moment, quit the mysql and then re-login as “chat” user. The password is “chat”.

```
mysql -u chat -p chat
```

- log テーブルを作成する。
Create the table “log”.

```
CREATE TABLE log ( date int, name varchar(100),  
mesg text , PRIMARY KEY (date));
```



chat.php

- Moodle の第5週から chat.php.gz をダウンロードして展開し、/var/www/html/ にコピーすること。
Download “chat.php.gz” from topic 5 in the ATMI18 course of Moodle, then extract and copy it to /var/www/html/ .

```
gzip -dc chat.php.gz > /var/www/html/chat.php
```

- http://localhost/chat.php にアクセスして、動くことを確かめよう。
Try to access http://localhost/chat.php and check how it works.

Cookie の利用

The Use of Cookie

■ What's Cookie ?

- あるWebサーバの情報を一時的に保存しておくブラウザの機能（例：セッション情報、パスワードなど）

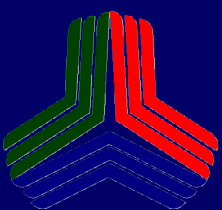
A browser function which saves some information (Ex. session, password, etc...) of the web server.

- Linux 版Firefox の場合は「設定」→「プライバシー」→「Cookieを個別に削除」で表示できる。

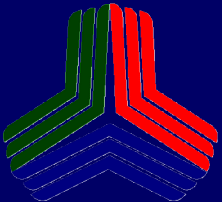
The contents of the Cookie can be shown by “Menu”->“Preferences”->“Privacy”->“remove individual cookies” for Linux version of Firefox. ⁸

Cookie の利用

The Use of Cookie

- 
- チャットの場合、Cookieを使わないと名前を毎回入力しなくてはならないため非常に不便

At the case of “chat” program, it is very inconvenient to enter the handle name at each time to submit the post when the cookie is disabled.



Cookie 文字列

Cookie String

■ 基本的なCookie文字列/Structure of Cookie String

(Cookie名) = (値) ; (Cookie名) = (値) ; ...expire= (日付) ; path= (URL)

セミコロン(;)の後に半角
スペースが入っている

(Cookie name)=(Value); (Cookie name)=(Value); ...expire=(date); path=(URL)

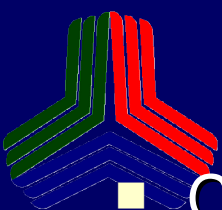
1-byte space is inserted after
each semi-colon(;)

- ただし、expire (有効期限) の日付はGMT (グリニッジ標準時) で書かなくてはならない。

Note that the format of the expire date must be GMT(Greenwich Mean Time).

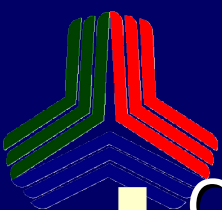
Cookie の使い方

Cookie Howto

- 
- Cookieを「焼く」（サーバがブラウザに情報を送る）
 - “Bake” the Cookie (send information to browser)
 - HTTPヘッダに以下のように表示させる
Add below string to HTTP header.
Set-Cookie: (Cookie String)
 - Cookieを「食べる」（ブラウザに保存されている情報をサーバが取得する）
 - “Eat” the Cookie (get information from browser)
 - ブラウザにそのURLに対応する有効期限内のCookieが保存されていれば、環境変数HTTP_COOKIEに、 expire やpathなどの情報を省略したCookie文字列が入る
The cookie string saved in the browser will be inserted into the environment variable HTTP_COOKIE if current URL corresponds to its path and it is not expired.

PHPによるCookie

Cookie Handling of PHP

- 
- Cookie を「焼く」 / "Bake" Cookie...`setcookie()` 関数
 - Cookie を「食べる」 / "Eat" Cookie...`$_COOKIE['(Cookie name)']`
 - Javascript やMETAタグでCookieを扱うこともできるが、この方法ならば、**そのページのHTMLソースを見てもCookieに書き込まれる内容が分からないので機密性が高いし、Cookie文字列を作らなくても良い**

Although Cookie can be handled by Javascript and META tag, we use above functions because it is highly secure that **the contents of the cookie do not appear in the HTML source**, and also we need not create the Cookie string with this method.

PHPによるCookie

Cookie Handling of PHP

- setcookie()の使い方/setcookie() syntax

```
setcookie(Cookie名,値,期限,パス);  
setcookie(Cookie name, value, expire date, path);
```

注意：Cookieはヘッダに含まれるため、**setcookie()** はHTMLを送信する前に実行すること

Notice: **setcookie()** must be executed before the transmission of HTML because cookies are included in HTTP header which is sent before HTML contents.

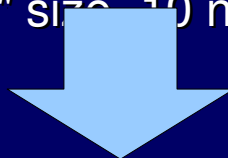
- 詳しくは/See also
<http://php.net/manual/ja/function.setcookie.php>

PHPによるCookieの利用

PHP Usage of Cookie

- /* COOKIE処理 */の下に以下の4行を追加
Add below 4 lines after /* COOKIE 処理 */

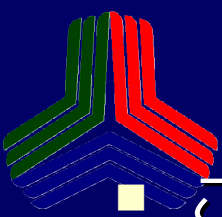
```
if($_POST['name'] != $_COOKIE['chat_name']){ // 名前が変更された場合  
    $_COOKIE['chat_name'] = $_POST['name']; // COOKIE 変数を更新  
    setcookie("chat_name", $_POST['name']); // COOKIE を焼く  
}
```
- 以下の行を次のように変更
Change below line as follows.
名前 : <input type="text" size=10 name="name">



名前 : <input type="text" size=10 name="name" value="<?php print
\$_COOKIE['chat_name']; ?>">

Javascriptの活用

The Use of Javascript

- 
- このチャットプログラムは書き込みをするのに一々メッセージのテキストフィールドをクリックしないといけないので非常に手間である。

Currently the posting to this chat program is really tedious because the user must click the message textfield at each time to post the message.

- ブラウザに読み込まれたときに自動的にメッセージのテキストフィールドにフォーカスするように、body タグに以下の Javascript コードを入れておく。

To focus on the message textfield automatically when the chat page is loaded in the browser, insert below Javascript code in the “body” tag.

```
onLoad="document.chat_form.mesg.focus()"
```

Javascriptのイベント

Event of Javascript

- イベント…コンピュータについてユーザーが何か操作をしたり、システムの状態が変化したりすることで発生する
“event” will be generated when the user operates with the interface or the system status is changed.
- イベントが発生した時に何らかの処理を行わせる命令を**イベントハンドラ**という
The command to execute some process at the time the corresponding event is generated is called “**event-handler**”.
- Ex:
 - マウスのボタンをクリック/click the mouse button.....**onclick**
 - ポインタを移動/move the pointer.....**onmousemove**
 - テキストを選択/select the text.....**onselect**
 - Webページの読み込みが完了/complete loading of the web page.....**onload**

Javascriptのイベント

Event of Javascript

- イベントハンドラは

イベントハンドラ名= “実行文”

の形で記述される。

The event-handler is described as

event handler name = “command”

- Ex :


`onclick = “window.alert('Handler Test')”`

A small window will pop up when click the object



“form” Object

- Javascriptでフォームを扱う場合、form オブジェクトのプロパティを用いる。
A property of the “form” object is the key to handle HTML form with Javascript.

例として以下のようなフォームを考える。

For example, consider the form below:

```
<form name="myForm"><p>
```

Textfield Example

```
<input type=text name="text1" size=40><br>
```

.....

この時、テキストフィールド"text1"に値を表示させたいときは、

If you want to show some values in the textfield named “text1”, describe as:

```
document.myForm.text1.value= (some values)
```

と書く。forms配列、およびelements配列で書く場合は、

You can rewrite above with the arrays of forms and elements like

```
documet.forms[0].elements[0].value = (some values)
```

となる。