



G's ACADEMY
TOKYO

Fileupload & Ajax



アジェンダ

- Fileupload
- 授業作成SYSTEMに組み込む
- Ajax
- Ajax&サーバーサイド連携

管理サイト側

Form ・ Camera

◇ Camera/写真選択

```
<form method="post" action="送信先" enctype="multipart/form-data">  
  <input type="file" accept="image/*" capture="camera" name="upfile">  
  <input type="submit" value="Fileアップロード">  
</form>
```

① FILE選択できるようにする

```
<input type="file" .....>
```

※他の例<input type="text">

② File送信時はenctype属性を指定

```
enctype="multipart/form-data"
```

③POST送信 (Action="送信先")

◇ Camera/写真選択

```
<form method="post" action="送信先" enctype="multipart/form-data">  
  <input type="file" accept="image/*" capture="camera" name="upfile">  
  <input type="submit" value="Fileアップロード">  
</form>
```

② カメラ起動＆画像選択可能

input accept="image/*" capture="camera"

※ 他の記述方法例) accept="image/jpeg, image/gif, image/png"

※ 他の記述方法例) accept="audio/*"

※ 他の記述方法例) accept="video/*"

※ 他の記述方法例) accept="text/comma-separated-values" //CSV

FileUpload

FileUpload : ①アップロードチェックの処理

1. ファイルアップロード パラメータ取得

- isset(パラメータ名) (パラメータがセットされているか? ※未入力チェックとは違う)
- \$_FILES["upfile"]["error"] == 0 (0は正常にアップロードしてることを意味する)

※参考URL : http://www.flatflag.nir87.com/move_uploaded_file-970#tablepress-21

```
if( isset( $_FILES["upfile"] ) && $_FILES["upfile"]["error"] == 0 ) {  
    // echo 'アップロードしてきている';  
} else {  
    // echo 'アップロードしてきてない OR なにかしらのErrorが発生';  
}
```

2. \$_FILES : ファイル名、アップロード先のPath取得

```
if( isset($_FILES["upfile"]) && $_FILES["upfile"]["error"]==0 ) {  
    $file_name = $_FILES["upfile"]["name"]; // "1.jpg"ファイル名取得  
    $tmp_path = $_FILES["upfile"]["tmp_name"]; // www/tmp/1.jpg : TempPath取得  
} else {  
    // echo 'アップロードしてきてない OR なにかしらのErrorが発生';  
}
```


FileUpload:②アップロード処理

3. アップロードに使用する関数

1. `is_uploaded_file` アップロードされたファイルが存在してるか？
2. `move_uploaded_file` Tempフォルダからimgフォルダへ移動
3. `chmod` ファイルに権限を付与する (0644を付与)

```
// FileUpload [--Start--]
if ( is_uploaded_file( $tmp_path ) ) {
    if ( move_uploaded_file( $tmp_path, $file_dir_path . $file_name ) ) {
        chmod( $file_dir_path . $file_name, 0644 );
        //echo $file_name . "をアップロードしました。";
        $img = '';
    } else {
        echo "Error:アップロードできませんでした。";
    }
}
// FileUpload [--End--]
```

◇ アップロードしたファイルを表示する

`$img = '';` //imgタグを作成
上記をHTMLのbody要素内に `<?=$img?>` と埋め込むことで画像を読み込める。

FileUpload : ③アップロード完成例

◇ シンプルバージョン

```
<?php
if (isset($_FILES["upfile"]) && $_FILES["upfile"]["error"] ==0 ) {
    $file_name = $_FILES["upfile"]["name"];          //"1.jpg"ファイル名取得
    $tmp_path  = $_FILES["upfile"]["tmp_name"];      //"/usr/www/tmp/1.jpg" Tempフォルダ
    $file_dir_path = "upload/"; //画像ファイル保管先
    $img=""; //画像表示orError文字を保持する変数
    // FileUpload [--Start--]
    if ( is_uploaded_file( $tmp_path ) ) {
        if ( move_uploaded_file( $tmp_path, $file_dir_path . $file_name ) ) {
            chmod( $file_dir_path . $file_name, 0644 );
            //echo $file_name . "をアップロードしました。";
            $img = ''; //画像表示用HTML
        } else {
            $img = "Error:アップロードできませんでした。"; //Error文字
        }
    }
    // FileUpload [--End--]
} else {
    $img = "画像が送信されていません"; //Error文字
}
?>
```

FileUpload : ④アップロード時のファイル名 重複問題

◇ ファイル名重複問題の解決策の1例

```
2 //Fileアップロードチェック
3 ▼ if (isset($_FILES["upfile"]) && $_FILES["upfile"]["error"] == 0) {
4     //情報取得
5     $file_name = $_FILES["upfile"]["name"]; // "1.jpg"ファイル名取得
6     $tmp_path = $_FILES["upfile"]["tmp_name"]; // "/usr/www/tmp/1.jpg"アップロード先のTempフォルダ
7     $file_dir_path = "upload/"; //画像ファイル保管先
8
9
10    //***File名の変更***
11    $extension = pathinfo($file_name, PATHINFO_EXTENSION); //拡張子取得(jpg, png, gif)
12    $uniq_name = date("YmdHis").md5(session_id()).".".$extension; //ユニークファイル名作成
13    $file_name = $uniq_name; //ユニークファイル名とパス|
14
15
16    $img=""; //画像表示orError文字を保存
17    // FileUpload [--Start--]
18 ▼ if (is_uploaded_file($tmp_path))
```

```
//***File名の変更***
$extension = pathinfo($file_name, PATHINFO_EXTENSION);
$uniq_name = date("YmdHis").md5(session_id()).".".$extension;
$file_name = $uniq_name;
```

前回授業のコードに追加！！！！

index.phpとinsert.phpの変更

- index.phpのフォームに画像選択とenctype属性追加
- gs_an_tableにカラムを1つ追加
image var_char(128) nullはOK
- bind変数も追加
`$stmt->bindValue(':image', $upload_file, PDO::PARAM_STR);`
- SQLも変更
`INSERT INTO gs_an_table(id, name, email, naiyou, indate, image)
VALUES(NULL, :a1, :a2, :a3, sysdate(), :image)`
- select.php 画面に表示して確認する

Ajax

Ajaxとは

- AjaxのAである「Asynchronous（非同期）」は、非同期でのクライアント・サーバ間の通信を指します
- JavaScriptによりクライアント上での動作が主で必要なデータのみ受信することで通信量の負担を軽減
- 特殊なサーバの設定等は必要としない

Ajaxとは (Chrome:Yahooサイトで確認)


◇ChromeブラウザでYahooサイトを表示 → 右クリック(検証)

The screenshot shows the Yahoo Japan homepage with the Chrome DevTools Network tab open. The following elements are highlighted with red boxes and numbered annotations:

- 1**: The 'エンタメ' (Entertainment) category in the main navigation bar.
- 2**: The 'Network' tab in the DevTools toolbar.
- 3**: The 'XHR' filter in the DevTools Network panel.
- 4**: The request URL: `?_m=Topics&_a=getTopics&s...`.
- 5**: The 'Preview' tab in the DevTools Network panel.
- 6**: The response data, which includes a timestamp and a list of topics.

A red arrow points from the 'エンタメ' category to the request URL, indicating the source of the data.

Ajaxのメリット

- Webページのリンクをクリックした時のレスポンス待ち時間の体感時間が少ない。
- 必要な部分の情報のみを取得変更し、必要なときに更新可能のため高速に動作する。
- 例)
5画面 -> 5HTMLファイル (通常の方法)

5画面 -> 1HTMLファイル (Ajaxだとこうできる！)

Ajaxのデメリット

- SEO対策には不向き
- スクリプトの知識が必要
- 作りが複雑になりがち
- 更新履歴が残りません、ブラウザの[戻る]ボタンでは1つ前の状態には戻りません。
- 更新後でも再表示すると初期状態に戻ってしまいます。

Ajaxの構文 (全体像)

```
$.ajax({  
    type: "get",    //default=get  
    url: "*****",  
    cache: false,  
    datatype: "jsonp",  
    success: function(data){  
        alert("通信成功");  
    },  
    error: function() {  
        alert("エラー");  
    },  
    complete: function() {  
        alert("完了");  
    }  
});
```

cacheプロパティ

```
$.ajax({  
  type: "GET",  
  url: "*****",  
  cache: false //cacheの有無をtrue,falseで指定。  
});
```

《引数》

cache: true or false

JSONの場合はデフォルトでcache:true

JSONをキャッシュを持たせない場合はcache:falseを指定する。

datatypeプロパティ

```
$.ajax({  
  type: "GET",  
  url: "*****",  
  cache: false,  
  datatype: "json"  
});
```

《引数》

datatype: text, html, xml, json, jsonp, script
(データ形式を指定)

dataプロパティ

```
$.ajax({  
  type: "GET",  
  url: "*****",  
  cache: false,  
  datatype: "text",  
  data:{  
    id: 1,  
    name: "yamazaki"  
  }  
});
```

Ajaxの構文 2

```
$.ajax({  
  type: "GET",  
  url: "select.php",  
  cache: false,  
  datatype: "html",  
  success: function(data){  
    alert("通信成功");  
  },  
  error: function() {  
    alert("エラー");  
  },  
  complete: function() {  
    alert("完了");  
  }  
});
```

通信が失敗したときの処理を記述




successメソッド

```
$.ajax({  
  type: "GET",  
  url: "search.php",  
  data: { search:"山崎" },  
  datatype: "html",  
  success: function(data){  
    document.querySelector("#list").innerHTML = data;  
  }  
});
```

通信が成功したときの処理を記述。第一引数を指定してデータを取得。

Elements Network Sources Timeline Profiles Resources Audits Console

Preserve log

Name	Path	Headers	Preview	Response	Timing
 method_jsonp.html	/jquery				
 jquery-1.11.0.min.js	code.jquery.com				
 method_jsonp.php?callb...	/jquery				

3 requests | 1.1 KB transferred

Console Search Emulation Rendering

Response

Remote Address: ::1:80
Request URL: http://localhost/jquery/method_jsonp.php?callback=json_data
Request Method: POST
Status Code: 200 OK

▼ Request Headers view source

Accept: text/javascript, application/javascript, application/ecmascript, application/x-ecmascript, */*; q=0.01
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ja,en-US;q=0.8,en;q=0.6
Cache-Control: max-age=0
Connection: keep-alive
Content-Length: 33
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Host: localhost
Origin: http://localhost
Referer: http://localhost/jquery/method_jsonp.html
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.114 Safari/537.36
X-Requested-With: XMLHttpRequest

▼ Query String Parameters view source view decoded

callback: ison_data

▼ Form Data view source view decoded

id: 1
mode: hoge
type: entry
sleep: 0

▼ Response Headers view source

PHP側の受け方

//PHP側の処理の書き方：一例

```
if(  
    isset($_SERVER['HTTP_REFERER']) &&  
    $_SERVER['HTTP_REFERER']== http://www.msn.com/index.php  
) {  
    //上記アドレスのリンクからAjaxでページを参照した場合はここを処理
```

```
    /* ここに正常動作する処理を記述：  
    戻したい値、text文字列、html文字列などをecho 関数を使用して戻す！！  
    */  
    echo "<div>テスト戻し文字列</div>";  
    exit();
```

```
}else{  
    //上記アドレスから参照されなかった場合Errorを返す。  
    echo "false"; //文字列でfalseをAjaxに戻す  
    exit();       //処理終了  
}
```

PHP最終課題

- 今作りたいものを
PHP&MySQLを使っていればなんでもOK！！

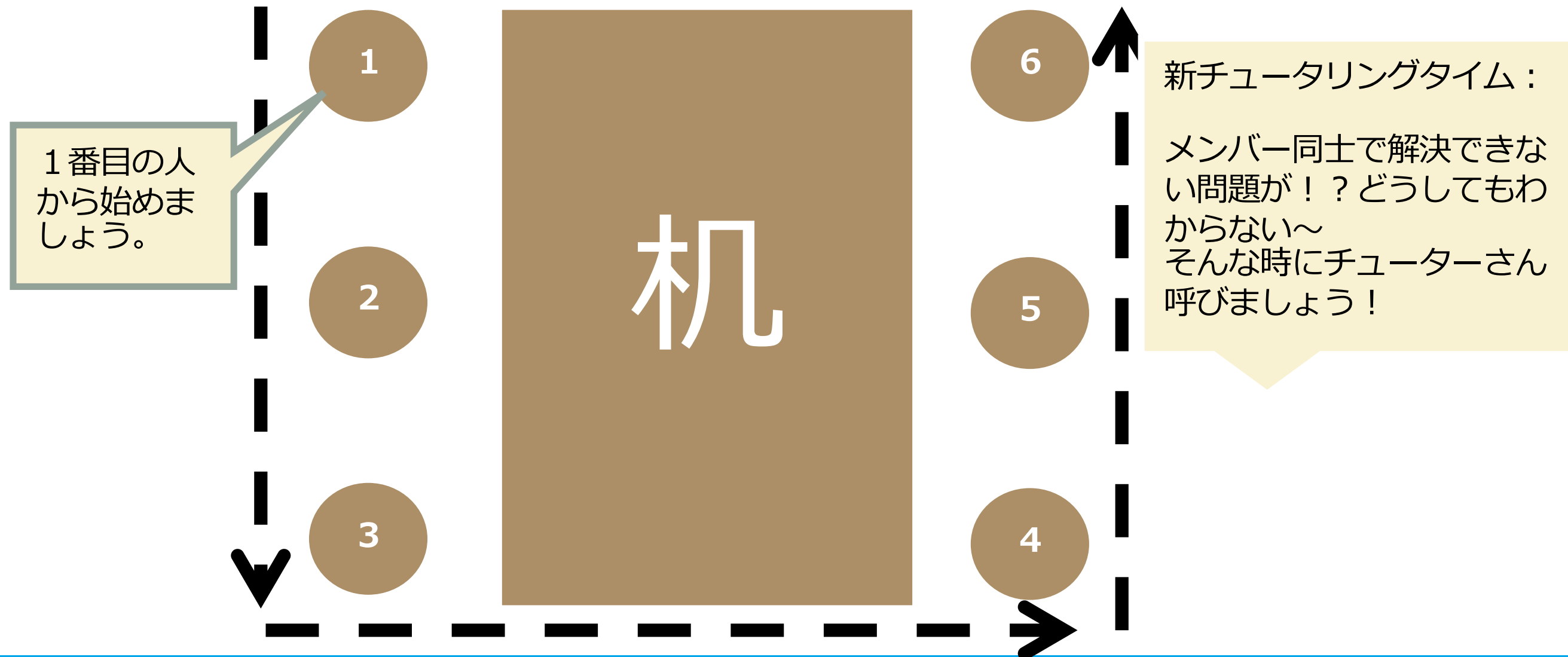
以上

チュータリングタイム

盛り上がってますね～！

メンバー同士の 進捗報告 & 教え合い！

プロジェクター側



こんなのやると面白い

- TextAreaをWordみたいにするライブラリを使ってみては？

<http://ckeditor.com>

CKEditor™

[Home](#)

Features

Add-ons

Blog

Pricing

Download

Documentation

Support

Contact us

CKEditor 4.6

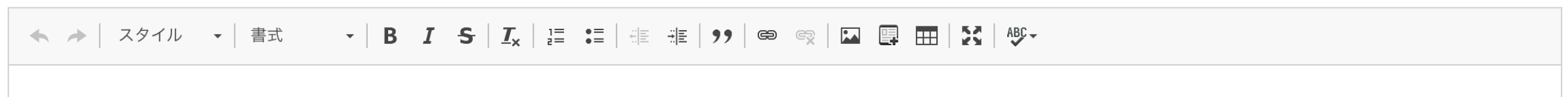
Copy Formatting, new Advanced Paste
from Word, refreshed skin.

Find out more

Article Editor

Document Editor

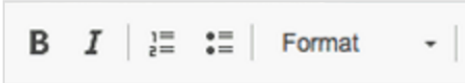
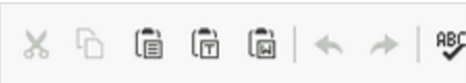

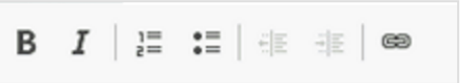
Inline Editor



Download

Download a ready-to-use CKEditor package that best suits your needs.

Version 4.6.2 • 12 Jan 2017

Basic Package	Standard Package	Full Package	Customize
 Minimal toolbar. For quick input fields. 17 Plugins	 Standards-compliant toolbar. Covers most editing needs for semantic-driven websites. 48 Plugins	 A heavier preset with a multiline toolbar. 73 Plugins	 Build your own version in 3 easy steps.
Download	Download	Download	Online Builder

[Compare packages](#) | [Release notes](#) | [Quick Start Guide](#) | [License](#)

Qiita風にするためEditorを機能追加

1. ダウンロードした「ckeditor.zip」解凍
2. 解凍してできた「ckeditor」フォルダを
本日の配布サンプルフォルダに入れます。
3. input.phpとdetail.phpの2ファイルを修正
※ckeditorライブラリ使用 (次ページ参照)
4. input.php にセッション機能を追加
5. menu.php に「データ登録」リンク追加

index.php

headに追加：

```
<script src="./ckeditor/ckeditor.js"></script>
```

コメント

```
30  <!--  
31      <label><textArea name="naiyou" rows="4" cols="40"></textArea>  
32      </label><br>  
33  -->
```

```
34  
35  <textarea name="naiyou" id="editor1" rows="10" cols="80">  
36      This is my textarea to be replaced with CKEditor.  
37  </textarea>  
38  <script>  
39      // Replace the <textarea id="editor1"> with a CKEditor  
40      // instance, using default configuration.  
41      CKEDITOR.replace( 'editor1' );  
42  </script>
```

追加

detail.php

headに追加：

```
<script src="./ckeditor/ckeditor.js"></script>
```

コメント

```
48 <!--      <label><textArea name="naiyou" rows="4" cols="40"><?
    =$row["naiyou"]?></textArea></label><br>-->
```

49

50

追加

```
51 <textarea name="naiyou" id="editor1" rows="10" cols="80">
52     <?=$row["naiyou"]?>
53 </textarea>
54 <script>
55     // Replace the <textarea id="editor1"> with a CKEditor
56     // instance, using default configuration.
57     CKEDITOR.replace( 'editor1' );
58 </script>
```

【 Ckeditor リファレンス 】

http://docs.ckeditor.com/#!/guide/dev_installation

Qiita風にするためEditorを機能追加

1. ダウンロードした「ckeditor.zip」解凍
2. 解凍してできた「ckeditor」フォルダを
本日の配布サンプルフォルダに入れます。
3. input.phpとdetail.phpの2ファイルを修正
※ckeditorライブラリ使用 (次ページ参照)
4. input.php にセッション機能を追加
5. menu.php に「データ登録」リンク追加