

jQuery 日本語リファレンス

<http://semooh.jp/jquery/>

jQuery の参考サイト

IBM : jQuery を使って Ajax 開発を単純化する

<http://www.ibm.com/developerworks/jp/xml/library/x-ajaxjquery.html>

セレクトタ

Selectors

<http://semooh.jp/jquery/api/selectors/>

```
$('#div').css('color', '#f00');
$('#id').css('color', '#f00');
$('.class').css('color', '#f00');
$('div[name=foo]').css('color', '#f00');
$('dl > dt:first').css('background', '#eee');
```

画面がロードされたときに実行させる

```
$(function(){
// 画面がロードされたときに実行させる処理
});
```

画面がロードされたときに input にカーソルを移す

```
<input type="text" id="email" value="" />
```

```
$(function(){
  $('#email').focus();
});
```

入力欄を目立たせる

.html

```
<input id="user" name="user" type="text" value="" />
<input id="password" name="password" type="password" value="" />
```

.css

```
.focus {
  color: #333;
  border: 2px solid #0cf;
  background-color: #fff;
}
```

.js

```
$(function(){
```

```

    $('input[type="text"], input[type="password"], textarea')
      .focus(function(){
        $(this).addClass("focus");
      })

      .blur(function(){
        $(this).removeClass("focus");
      });
    $("#user").focus();
  });

```

複数 DOM の部分は下のようによく書くこともできるみたい。

```

$('input:text, input:password, textarea')

```

return false と event.stopPropagation と event.preventDefault()

1分でわかる return false; preventDefault(); stopPropagation() の違い

<http://iwb.jp/return-false-preventdefault-stoppropagation/>

HTML タグが入れ子の関係にあるとき、子のイベントを親に伝播するかとかの違い。

スコープと this と event.target

```

$(".foo").click(function(e){
  $(this)...
  $(e.target)...

```

class="foo" を持つ要素を click したとき、クリックされた要素 this と e.target は同じ対象を指す。

```

$(".foo").click(function(e){
  $.get("bar.php?baz=1", function(){
    $(this)...
    $(e.target)...

```

スコープが変わると this が変わる。この例での this はスコープで引き継がれた e.target (クリックされた要素) と違い、\$.get の this みたいな感じになる。

javascript の「this」は「4種類」??

<http://qiita.com/takeharu/items/9935ce476a17d6258e27>

Ajax を使った送受信の際のエラー取得

```

$.ajax( {
  dataType : 'json',
  type : "POST",
  url : url,
  data : postData,
  :
  error : function(obj, status, errorThrown) {
    alert(obj.responseText.replace(/<\/?[^>]+>/gi, ''));

```

のような感じに obj.responseText で PHP などが吐くエラーを取得できる。

replace() しているのは、HTML 形式のエラーを alert() で表示する際に、HTML タグを除去するため。

[professional resumes](#)

コールバックが動作しない

Ajax ([jQuery](#)) とかで非同期通信を行う時、コールバックが動かなくてハマった (汗)

ローカルのテスト環境では動くのに、サーバーにアップすると動かない。
ググってみると、戻りのフォーマットがよろしくないとかコールバックが動かないんだとか。

で、[PHP](#) から JSON の戻りがローカルでは OK、サーバーでは何を返してんの？ とをチェックしてみると ...

```
Call to undefined function json_encode() ...
```

[PHP](#) のエラー出力やん (汗)

ローカルは [PHP-5.3.8](#)、サーバーは [PHP-5.2.17](#) で [PHP-5.2](#) に `json_encode()` がない？ と [PHP](#) マニュアルを見てみても

[PHP](#) : `json_encode`

<http://php.net/manual/en/function.json-encode.php>

pecl 入れないといけないってこと？

とりあえず `Zend\Json::encode()` で済ますことにした。

読み込みと記述する場所

いままでは普通 `<head>` の中で `jquery` を読み込んでおいて、その後 (その下のほうの行) で `jquery` を使用したスクリプトを書く、みたいなのが主流だったと思う。

それが最近では

```
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
</body>
</html>
```

みたいに、`body` の最後で読み込めば読み込み処理が最適化されんじゃね？ っていう流れに変わった。

普通に考えて

```
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
```

```
<script>
// ここにスクリプトを書く
</script>
</body>
</html>
```

となるわけだけど、フレームワークとか使ってる場合

```
<body>
{{ content }}
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
</body>
```

なのね。

```
<body>
{{ content }}
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
{{ javascript }}
</body>
```

みたいにして任意のコードがあったら挿入すりゃいいんだけど、なるべくファイルの読み込みは増やしたくない。

そこで「ライブラリ読み込み前に jquery を使ったスクリプトを記述できないか？」と調べた。

```
<script>
window.onload = function(){
$.foo().click(function(){
alert("OK");
});
};
</script>
:
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
</body>
</html>
```

呼び出しの部分を javascript にしておいて、ロードが完了したら jquery を使うって感じなら行けるっぽい。