

jQuery

[CSS][CSS3][レスポンス Web デザイン]

- ・ [逆引き](#)
- ・ [はじめに](#)

ready() メソッド

- ・ [jQuery](#) は ready() メソッドを実装している
- ・ ready() メソッドは、一般に [DOM](#) のドキュメントオブジェクトにバインドされるカスタムイベントハンドラ
- ・ 唯一の引数として関数を受け取る
- ・ この関数には [DOM](#) がトラバースや操作を実行できるようになったときに実行される [JavaScript](#) のコードが含まれる

ready() イベントハンドラは、[JavaScript](#) の window.onload イベントにかわる [jQuery](#) の機能であり、必要に応じて何度でも利用できる。

```
jQuery(document).ready(function(){
    alert('ready!');
});
```

ready() イベントを使うためのショートカット

```
jQuery(function(){
    alert('ready');
});
```

このイベントが必要になるのは、ページの先頭で <head> で囲まなければいけない場合のみ。 </body> の直前に配置することで、ready() イベントは使わずに済む。 </body> の位置では DOM 要素だけでなく、それよりも前にあるすべてのものを読み込んでから、[JavaScript](#) を実行することになる

\$ グローバルオブジェクト

- ・ \$ グローバルオブジェクトは、[jQuery](#) に置き換えられます。
- ・ \$() は、[jQuery\(\)](#) 関数のエイリアスとなります

\$ の競合を回避する

- ・ \$ は [jQuery](#) のエイリアスだが、'[jQuery](#)' という表現とはことなり、'\$' は、他のライブラリと競合する可能性がある。
- ・ しかしながら、[jQuery](#) ではなく、\$ を利用したい。
- ・ 匿名関数を作成し、その関数に [jQuery](#) オブジェクトを渡したのち、[jQuery](#) オブジェクトへのパラメータポイントとして、\$ を使う

```
(function($){ // $ パラメータを使ってプライベートスコープを作成する関数
// 競合の心配なく、$ を利用できる
})(jQuery) // 匿名関数を呼び出し、jQuery オブジェクトを渡す
```

セレクタとフィルタ

セレクタ

[CSS]

- ・ jQuery では、CSS1-3 仕様で規定されているセレクタのほとんどをサポートしているだけでなく、カスタムセレクタを採用している

名称	例
ユニバーサルセレクタ	<code>\$("*")</code>
ID セレクタ	<code>\$("#id 名 ")</code>
クラスセレクタ	<code>\$(". クラス名 ")</code>
要素セレクタ	<code>\$(" 要素名 ")</code>
グループセレクタ	<code>\$(" セレクタ , セレクタ ...")</code>
子孫セレクタ	<code>\$(" 要素 1 要素 2")</code>
子セレクタ	<code>\$(" 要素 1 > 要素 2")</code>
隣接セレクタ	<code>\$(" 要素 1 + 要素 2")</code>
間接セレクタ	<code>\$(" 要素 1 ~ 要素 2")</code>
first-child 疑似クラス	<code>\$(" 要素 :first-child")</code>
last-child 疑似クラス	<code>\$(" 要素 :last-child")</code>
nth-child 疑似クラス	<code>\$(" 要素 :nth-child(番号)")</code>
only-child 疑似クラス	<code>\$(" 要素 :only-child")</code>
[attribute]	<code>\$("[属性名]")</code>
[attribute='vvalue']	<code>\$("[属性名 =' 値 '")</code>
[attribute^='value']	<code>\$("[属性名 ^=' 値 '")</code>
[attribute \$='value']	<code>\$("[属性名 \$=' 値 '")</code>
[attribute*='value']	<code>\$("[属性名]*=' 値 '")</code>
[attribute ='value']	<code>\$("[属性名 =' 値 '")</code>
[attribute~='value']	<code>\$("[属性名 ~=' 値 '")</code>

フィルタ

first フィルタ	<code>\$(" 要素 :first")</code>
last フィルタ	<code>\$(" 要素 :last")</code>
even フィルタ	<code>\$(" 要素 :even")</code>
odd フィルタ	<code>\$(" 要素 :odd")</code>
eq フィルタ	<code>\$(" 要素 :eq(番号)")</code>
gt フィルタ	<code>\$(" 要素 :gt(番号)")</code>
lt フィルタ	<code>\$(" 要素 :lt(番号)")</code>
header フィルタ	<code>\$(":header")</code>
アニメーションフィルタ	<code>\$(" 要素 :animated")</code>

否定疑似クラス	\$("#要素 :not(セレクタ)")
contains フィルタ	\$("#要素 :contains(文字列 1)")
empty フィルタ	\$("#要素 :empty")
has フィルタ	\$("#要素 1:has(要素 2)")
parent フィルタ	\$("#要素 :parent")
button フィルタ	\$(":button")
checkbox フィルタ	\$(":checkbox")
checked フィルタ	\$(":checked")
disabled フィルタ	\$(":disabled")
enabled フィルタ	\$(":enabled")
file フィルタ	\$(":file")
image フィルタ	\$(":image")
input フィルタ	\$(":input")
password フィルタ	\$(":password")
radio フィルタ	\$(":radio")
reset フィルタ	\$(":reset")
selected フィルタ	\$(":selected")
submit フィルタ	\$(":submit")
text フィルタ	\$(":text")
hidden フィルタ	\$(":hidden")
visible フィルタ	\$(":visible")

API

- <http://api.jquery.com/>

.each()

- 合致した全てのエレメントに対して関数を実行

```
$("#img").each(function(){
    $(this).closest("a").attr("target","_blank");
});
```

.closest()

- 開始要素から最も近い親要素を選択

```
$("#img").each(function(){
    $(this).closest("a").attr("target","_blank");
});
```

length

- ・ [jQuery](#) オブジェクトのエレメント数を保持

```
$("#img").each(function(){  
  if ($("#this).closest("a[target]").length == 0) {  
    $("#this).closest("a").attr("target","_blank");  
  }  
});
```

プラグイン

作成

- ・ [jQuery プラグインの作成](#)

DataTables

- ・ [jQuery DataTables 動的に高さを変更](#)

Tips

- ・ [jQuery Tips](#)

jQuery Cheat Sheet

・