

# html の操作

html の文書の操作は、前記のセレクターを指定して行うことができます。

- 文章の変更

```
$("#セレクター").text("変更後の文章");
```

(例) クラス名 text の文章を「変更後」に変更

```
$(function(){  
    $(".text").text("変更後");  
});
```

- 文章を変更する際、<span> 等のタグを使用

```
$("#セレクター").html("変更後のタグ込みの文章");
```

(例) クラス名 text の文章を「<span> 変更後 </span>」に変更

```
$(function(){  
    $(".text").html("<span> 変更後 </span>");  
});
```

- 文章とタグの変更

```
$("#セレクター").replaceWith("変更後のタグ込み文章");
```

(例) p タグで囲まれた文章をタグを「<h1>」、文章を「変更後」に変更

```
$(function(){  
    $("p").replaceWith("<h1> 変更後 </h1>");  
});
```

- ・ 指定したセレクターを削除

```
$("#セレクター").remove();
```

(例) class 名「del」のついた p タグを削除

```
$(function(){  
    $("p.del").remove();  
});
```

- ・ 指定したセレクターの前に追加

```
$("#セレクター").before("追加するタグと文章");
```

(例) p タグの前に「<h1> 追加文 </h1>」を挿入

```
$(function(){  
    $("p").before("<h1> 追加文 </h1>");  
});
```

- ・ 指定したセレクターの後ろに追加

```
$("#セレクター").after("追加するタグと文章");
```

(例) p タグの後ろに「<h1> 追加文 </h1>」を挿入

```
$(function(){  
    $("p").after("<h1> 追加文 </h1>");  
});
```

# htmlの属性操作

## ・属性の追加・変更

指定した属性がすでに設定されている場合は変更し、設定されていない場合は追加します。複数も可能です。

### 1つの場合

```
$("#セレクター").attr("属性名","変更後の属性値");
```

### 複数の場合

```
$("#セレクター").attr({  
    属性名:"変更後の属性値",  
    (中略・・・上と同様のが続く)  
    属性名:"変更後の属性値"  
});
```

(例) img タグの src 属性を image/img1.jpg に変更

```
$(function(){  
    $("#img").attr("src","images/img1.jpg");  
});
```

(例) img タグの src 属性を「image/img1.jpg」に、alt 属性を「変更画像」に変更

```
$(function(){  
    $("#img").attr({  
        src:"images/img1.jpg",  
        alt:"変更画像の alt"  
    });  
});
```

- class 属性の追加

```
$("セレクター").addClass("クラス名");
```

(例) 最初の li タグに class 名「firstlist」を追加

```
$(function(){  
    $("li:first").addClass("firstlist");  
});
```

- 属性の削除

```
$("セレクター").removeAttr("属性名");
```

複数を削除する場合は、半角スペースで区切ります。

```
$("セレクター").removeAttr("属性名 1 属性名 2");
```

(例) img 要素の title 属性を削除

```
$(function(){  
    $("img").removeAttr("title");  
});
```

- class 属性の削除

```
$("セレクター").removeClass("クラス名");
```

(例) 最初の li タグから class 名「firstlist」を削除

```
$(function(){  
    $("li:first").removeClass("firstlist");  
});
```

## c s s の操作

css の操作は html 同様にのセレクターを指定して行います。

- css を 1 つ変更（追加）の場合

```
$("セレクター").css("プロパティ","値");
```

- css を複数変更（追加）の場合

```
$("セレクター").css({  
    "プロパティ ":" 値",  
    "プロパティ ":" 値",  
    (中略・・・上と同様のが続く)  
    "プロパティ ":" 値"  
});
```

※ここでは c s s と同様のプロパティ名を使用するために  
クォーテーションでプロパティを囲った記述にしています。

web 上で公開されているものの中には囲われていないものもありますが、  
囲わない場合、jQuery の書き方をする必要があります。

(例)

font-size → fontSize

background-color → backgroundColor

( 例)

<h1> のフォントサイズを 3 0 em へ変更

```
$("#h1").css("font-size","30em");
```

id 名 wrapper の文字サイズを 62.5%に

```
$("#wrapper").css("font-size","62.5%");
```

class 名 box の文字サイズを 1.6em、文字色を赤に

```
$(".box").css({  
    "font-size":"1.6em",  
    "color":"red"  
});
```

<li> の一つ目だけ文字色を赤に

```
$("#li:first-child").css("color","red");
```

## メソッドチェーン

html や css の操作について、同じプロパティに対して複数の命令を一度に行う方法があります。

メソッドチェーンといい、命令を「. (ドット)」でつなげていく方法になります。

この方法を使うと、同じプロパティを複数回探して命令を実行するのに対して 1 回ですむため、処理速度の向上につながります。

(例) p 要素の文章を「変更後」に、文字色を赤にする。

a、個別に書く場合

```
$(function){  
    $("p").text(" 変更後 ");  
    $("p").css("color", "red");  
});
```

b、メソッドチェーン

```
$(function){  
    $("p").text(" 変更後 ").css("color", "red");  
});
```