

# JQuery の命令記述の基本

```
$function(){  
    $("セクター").jQuery の命令  
};
```

セクターは html・css で使用しているものになります。  
セクターとして使える主なものは以下の通りです。

## 1、css で使用するセクター

css の指定と同様の指定を行います。

- html の要素（タグ）
- id セクター
- class セクター
- 子孫セクター
- ユニバーサルセクター（全ての要素）  
 \$("\*")
- グループセクター（複数のセクターに同時指定）  
 \$("セクター 1, セクター 2, . . .")

- 子セレクター

`$(" 親要素名 > 子要素名 ")`

- 隣接セレクター

`$(" 要素 1 + 要素 2")`

- child 系の疑似クラス

特定の要素内の指定した番号の要素

`$(" 要素 :nth-child( 番号 )")`

特定の要素内の最初の要素

`$(" 要素 :first-child")`

特定の要素内の最後の要素

`$(" 要素 :last-child")`

などがあります。

- 間接セレクター（特定の要素の後に出現する要素）

`$(" 要素 1 ~ 要素 2")`

- 否定疑似クラス（特定の要素内の指定した要素以外の要素）

`$(" 要素 1:not( 要素 2)")`

- empty 疑似クラス（子要素やテキストを含まない要素）

`$(" 要素 :empty")`

## 2、jQuery 独自のフィルター

- first フィルター（指定した要素の最初の要素）

`$("要素 :first")`

（例）最初の li の文字色を赤くする。

```
$(function(){  
    $("li:first").css("color","red");  
});
```

- last フィルター（指定した要素の最後の要素）

`$("要素 :last")`

（例）最後の li の文字色を赤くする。

```
$(function(){  
    $("li:last").css("color","red");  
});
```

- even フィルター（指定した要素の偶数番目の要素）

`$("要素 :even")`

（例）li の偶数番目の文字色を赤くする。

```
$(function(){  
    $("li:even").css("color","red");  
});
```

- odd フィルター（指定した要素の奇数番目の要素）

`$("要素:odd")`

（例）li の奇数番目の文字色を赤くする。

```
$(function(){  
    $("li:odd").css("color","red");  
});
```

- eq フィルター（指定した番号の要素）

`$("要素:eq(番号)")` ※番号は 0 から

（例）li の 1 番目の文字色を赤くする。

```
$(function(){  
    $("li:eq(0)").css("color","red");  
});
```

（例）li の 4 番目の文字色を赤くする。

```
$(function(){  
    $("li:eq(3)").css("color","red");  
});
```

※この後出てくるものも含め、jQuery では数をカウントするとき「0」から開始するため、1 番目の要素は 0 となり、以下 2 番目は 1、3 番目は 2・・・となります。

- gt フィルター（指定した番号より後の要素）

`$("要素 :gt( 番号 )")` ※番号は 0 から

（例）li の 5 番目以降の文字色を赤くする。

```
$(function(){  
    $("li:gt(3)").css("color","red");  
});
```

※li の 5 番目は jQuery の 4 となり、このフィルターは「より後」のため 3 を指定する。

- lt フィルター（指定した番号より前の要素）

`$("要素 :lt( 番号 )")` ※番号は 0 から

（例）li の 1 ～ 3 番目の文字色を赤くする。

```
$(function(){  
    $("li:lt(3)").css("color","red");  
});
```

※「より前」なので、gt フィルターと同様に考える。

- header フィルター（h1 ～ h6 までの header 要素）

`$("要素 :header")`

（例）全ての見出し要素の文字色を赤くする。

```
$(function(){  
    $(":header").css("color","red");  
});
```

- contains フィルター（特定の文字列が含まれている要素）

`$("要素 :contains( 文字列 )")`

（例）「コーディング」という文字列を含む要素の文字色を赤くする。

```
$(function(){  
    $("li:contains(' コーディング ')").css("color","red");  
});
```

- has フィルター（特定の要素が含まれている要素）

`$("要素 1:has( 要素 2)")`

（例）p 要素のうち、中に span 要素を持つものの文字色を赤くする。

```
$(function(){  
    $("p:has(span)").css("color","red");  
});
```

- parent フィルター（子要素やテキストを含む要素）

`$("要素 :parent")`

（例）p 要素の中で子要素かテキストが含まれているものの文字色を赤くする。

```
$(function(){  
    $("p:parent").css("color","red");  
});
```

※empty 疑似クラスの反対の意味となります。

### 3、属性 (attribute) セレクター

- [attribute] (特定の属性を持つ要素)

`$("[ 属性名 ]")`

(例) id 属性を使用している要素の文字を赤くする。

```
$(function(){  
    $("#id").css("color","red");  
});
```

- [attribute='value'] (特定の属性が指定した値を持つ要素)

`$("[ 属性名 =' 値 ' ]")`

(例) name 属性が namae である要素の文字を赤くする。

```
$(function(){  
    $("[name= 'namae']").css("color","red");  
});
```

- [attribute != 'value'] (特定の属性が指定した値を持たない要素)

`$(" 要素名 [ 属性名 != ' 値 ' ]")`

(例) name 属性が namae 以外である要素の文字を赤くする。

```
$(function(){  
    $("[name != 'namae']").css("color","red");  
});
```

- [attribute^='value'] (特定の属性が特定した値で始まっている要素)

`$("[属性名 ^=' 値 '"])`

(例) name 属性が n で始まる要素の文字を赤くする。

```
$(function(){  
    $("[name ^= 'n']").css("color","red");  
});
```

- [attribute\$='value'] (特定の属性が特定した値で終わっている要素)

`$("[属性名 $=' 値 '"])`

(例) name 属性が e で終わる要素の文字を赤くする。

```
$(function(){  
    $("[name $= 'e']").css("color","red");  
});
```

- [attribute\*='value'] (特定の属性が特定した値を含んでいる要素)

`$("[属性名 *=' 値 '"])`

(例) name 属性に ma が含まれている要素の文字を赤くする。

```
$(function(){  
    $("[name *= 'ma']").css("color","red");  
});
```



- [attributeFilter1][attributeFilter2]（複数の属性セレクターの同時指定）

`$("[属性セレクター 1][属性セレクター 2]")`

（例）name 属性が n で始まり、更に e で終わる要素の文字を赤くする。

```
$(function(){  
    $("[name ^= 'n'][name $= 'e']").css("color","red");  
});
```