

jQuery はフリーのライブラリです。しかし、WEBアプリケーションにおける必要な JavaScript の処理を殆ど簡易化する事ができる優秀なライブラリです。JavaScript の HTML 要素参照はとても面倒ですが、jQuery ならばタイプ量が格段に少なくなります。ただ、注意すべきは画面上のコントロールが全て [] 後に、HTML要素を特定して処理記述をまとめて行うという記述方式になれる必要があるところです。

```
<script>
$(function(){
    // ここに処理を記述
});
</script>
```

例えば、ボタンの処理を記述する場合、HTML には オブジェクトを取得する為の [] 属性を指定し、値が btn だとします。この場合の jQuery での処理記述は以下のようになります。

```
$("#[ ]").on("[ ]", function() {
    alert("ボタンがクリックされました");
});
```

jQuery では HTML 要素の単独のオブジェクトを特定する為に、JavaScript と同じ [] 属性を使用しますが、その値の使用方法が [] における指定方法と同じになります。その選択方法は『セレクト』と呼ばれ、参照方法のバリエーションも多岐に渡る事になります。この場合、対象となるオブジェクトは一つだけですが、jQuery では HTML の構造や [] 属性や HTML要素の種類そのものも利用して対象を複数として考えて一括で処理する事を想定しています。

たとえば、FORM 要素の中にある INPUT 要素の背景色を全てピンクにするには以下のように記述します。

```
$("#[ ]").css("[ ]", "[ ]");
```

また、jQuery の殆どのメソッド(ここでは css) では、結果の値が再び参照していたオブジェクトそのものになるので、追加で全ての対象に値として 山田太郎 とセットする場合には以下のように記述できます。

```
$("#[ ]")
.css("[ ]", "[ ]")
. "[ ]" ("山田太郎");
```

この例では、css メソッドでスタイルを指定し、[] メソッドの引数に値をセットしています。値を取得したい場合は引数無しで使用すると戻り値として入力値を取得できます。(その場合は単独のオブジェクトを特定して使用します)

これ以外にも良く使用するメソッドして、HTML 要素の属性値を取得・設定する [] メソッドがあります。

そして、jQuery を使うと容易に新しい HTML 要素を新規に作成する事もできます。その際の仕様上、画面上に既に存在する特定の要素に追加する必要があるのでその要素の [] 属性の値を area として特定するとして、その中に INPUT 要素を追加するには以下ようになります。

```
$("#[ ]").appendTo("[ ]");
```