

## 1.1 jQueryの大まかな概念

ここからは、具体的にjQueryの基本的な機能ひとつひとつを紹介していきます。とは言いまでも、jQueryの機能は膨大なので、本書では利用頻度の高いものに絞って解説しています。

「利用頻度の高いもの」と言いながらも、それだけでかなりの数があるのですが、初めて触る人が、jQueryで何ができるのかわかりやすいよう、体系立てて組み立てたつもりです。ここで解説していないjQueryの機能、各機能の組み合わせ方の例は、サンプル解説の章に少しずつ混ぜています。徐々に難易度の高いものを習得していけるよう、本を組み立てたつもりですので、ひとまず、この章では、ざっとどんなことができるのかをつかんでいただければと思います。

まずは、jQueryの大まかな概念、開発環境の作り方を解説します。

### 「何かを取ってくる」→「それに何かする」

さっそく、jQueryの基本的な概念に触れていきましょう。いきなりですが、jQueryを使って書かれたコードを見えます。

```
$('#sample1').hide();
```

このコードでやっていることは、「id属性がsample1である要素を隠す」ということです。ページのなかに、次のような要素があったとしましょう。

```
<div id="sample1">サンプルです</div>
```

こんな要素があったら、これが非表示になります。

ものすごく単純ですね。

ここでまず覚えておいていただきたいのは、jQueryを使う上での、基本的な流れです。jQueryは

#### 「何かを取ってくる」→「それに何かする」

という2段階のステップを踏みます。

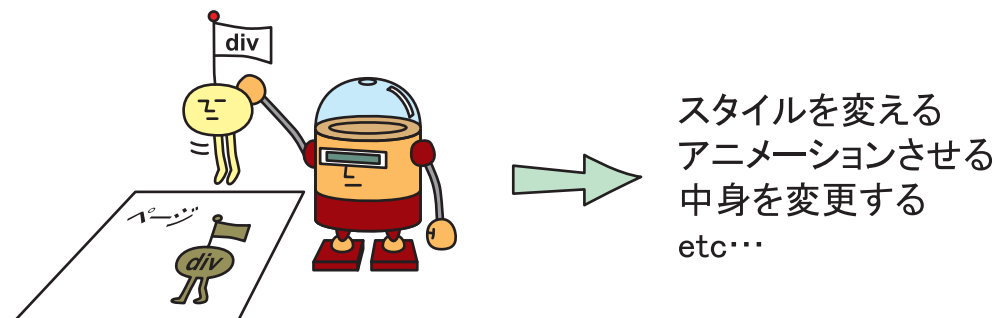


図1 div要素に対して作用するjQueryの概念図

具体的に見てみると、今例に出したサンプルは、次のような組み合わせでひとつの命令となっています。

表 命令とコードの対応表

	具体的な対象・命令	コード
何かを取ってくる	idがsample1の要素	<code>\$('#sample1')</code>
それに何かをする	隠す	<code>.hide()</code>

まず、要素を取ってきて、それを隠すという流れになっています。

普段スタイルシートを書いている人であれば、`#sample1`は「idがsample1な要素」を示すことがわかるでしょう。jQueryでは、CSSのセレクタを利用することができます。hideってというのは「隠す」の意味そのままです。

こんな感じで、「何かを取ってくる」→「それに何かする」をするのがjQueryの基本的な使い方です。セレクタについては、CSSを普段使っている人なら何の抵抗もないでしょうし、hideなんていう命令もそのままの意味です。

この章で、まずこのようなjQueryの大まかな概念を解説したのち、

- 「何かを取ってくる」
- 「それに何かする」

の2つに大きく分けてひとつひとつ解説していきます。

#### 「何かを取ってくる」→「それに何かする」

ここではこれだけを覚えて次に進みましょう。

ヒント◎ 文の終わりを示す ; (セミコロン)

行の最後にある ; は、命令の終わりを意味します。日本語の。と同じようなものと考えてください。付け忘れるとエラーになってしまいますので、注意してください。

## \$ と jQuery オブジェクト

jQuery を使うとき、\$ という記号が必ず登場します。

\$ は jQuery を理解するうえで必要不可欠なものですので、これが何かを初めに解説しておきましょう。先ほどのコードは次のようなものでした。

```
$('#sample1').hide();
```

頭に \$ がついていますがね。実は、この \$こそが、「jQuery」そのものなのです。

\$( ) で HTML の要素を jQuery オブジェクト化するところから、すべてがはじまります。

### \$( ) は何かを取ってくる

「jQuery オブジェクト化」などと言われてもなんのことやらさっぱりわからないと思いますので、この1行でどのようなことが行われるのか、詳しく解説しましょう。

まず、先ほど jQuery は、「何かを取ってくる」→「それに何かする」という流れで動くと書きました。つまり、jQuery が初めに行う仕事は、「何かを取ってくる<sup>※1</sup>」ことです。それをやっているのが次のハイライトした部分です。

```
$('#sample1').hide();
```

ここでは、jQuery の基本となる、**セレクト**という機能を使っています。

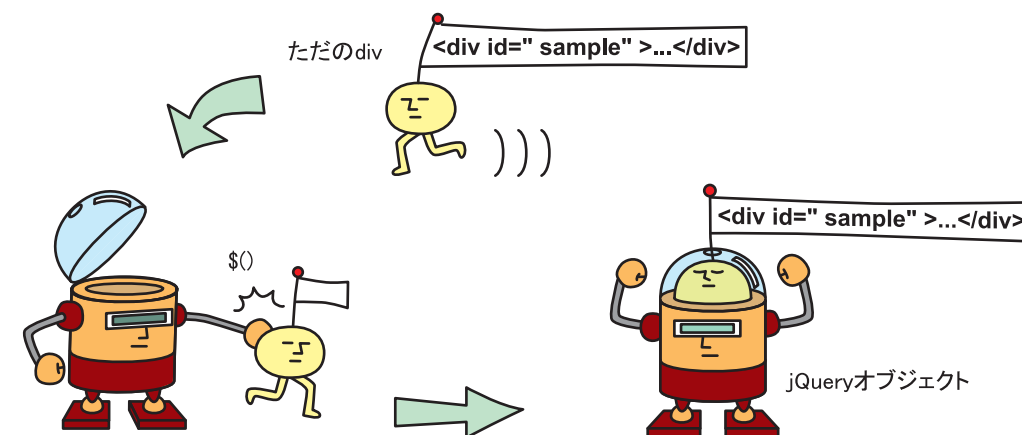
\$( ) でくるんだ部分に、取ってきたい要素を示すセレクトを指定することで、ほしい要素を取得してきてることができます。id が sample1 である要素がほしかったので、#sample1 というセレクトを書いています。

これが「何かを取ってくる」機能なのですが、実は \$( ) が行っているのは、「何かを取ってくる」ということだけではありません。要素を取ってくるのに加え、**jQuery オブジェクト**を作成するという作業を行っています。なので、「何かを取ってくる」は、正確には、「何かを取ってきて jQuery オブジェクトを作る」ということをしています。

## jQuery オブジェクト化する

さて、突然でてきた jQuery オブジェクトというものは何なのか。これは、たとえば、パワードスーツを着た人間のようなものです<sup>※1</sup>。ただの人間にパワードスーツを着せると超人的な力を発揮できるように、HTML の要素を jQuery オブジェクト化すると、JavaScript だけで行うにはとても大変な操作を、いとも簡単に行うことができるようになります。

※1 ほかにととえて言うならば、正露丸に対する正露丸糖衣、粉薬に対するカプセル錠剤、ファミコンに対するディスクシステム、よろいを着た戦士、変身した仮面ライダー、ガンダムにのったパイロット……のようなものと言ったらいいでしょうか。



■ 図2 jQuery オブジェクト化の概念

パワードスーツを着ることで、jQuery オブジェクトになった HTML の要素は、自由自在に jQuery の「それに何かする」機能を使うことができるというわけです。その「それに何かする」機能を使うには、後ろに . をつけて書きます。今回の例だと hide という機能<sup>※2</sup>を呼んでいます。

```
$('#sample1').hide();
```

これが jQuery の使い方の基本の基本です。基本の基本ではありますが、このように、要素を取ってきて jQuery オブジェクトを作り、それに何かするという流れはいつも同じです。

※2 JavaScript にはもともと、hide という名前の機能はありません。これは、簡単に要素を非表示にできるよう、jQuery が用意した機能です。hide は、その名の通り、要素を隠します。

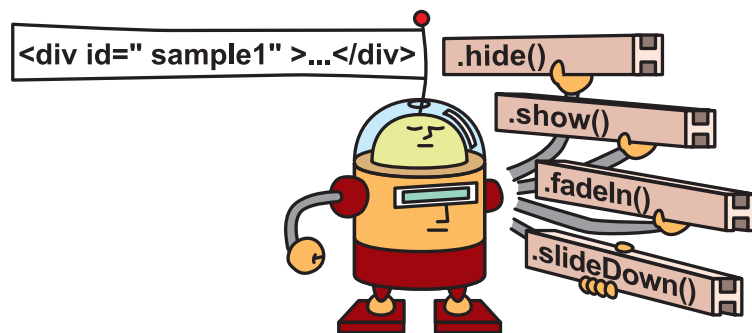
## jQueryオブジェクトって何?

このように、`$()`でjQueryオブジェクトを作ること初めて、「何かを取ってくる」の次に行う、「それに何かする」ことが可能になります\*。

ちょっと別の「それに何かする」機能を見てみましょう。

```
$('#sample1').show(); // 表示させる
$('#sample1').fadeIn(); // フェードインさせる
$('#sample1').slideDown(); // スライドアニメーション付で表示させる
$('#sample1').css('border','1px solid red'); // 1pxの赤線を付ける
$('#sample1').remove(); // 削除する
```

上記は、「それに何かする」機能のごく一部です。各機能は追って解説していきますが、コードのコメントに書いたような動作になります。スライドアニメーションやフェードアニメーションも非常に簡単に行うことができます。



■図3 jQueryの「それに何かする」機能のいろいろ

ところで、初めに、ブラウザごとにさまざまな特徴があり、それをjQueryはなんとかしてくれるといったようなことを書きました。このパワードスーツ……もといjQueryオブジェクトの裏で行われていることが、まさしくこれにあたります。JavaScriptで何かしようとするれば、犬も転べば棒に当たると言わんばかりに、ブラウザごとの差異に悩まされることになります。例えば、「ボタンをクリックされたらメッセージを出す」というような単純な動作でも、2通りの方法で書かなければならなかったりするので。

jQueryは、そういった小難しいことを、裏側でなんとかしてくれる、便利なパワードスーツです。パワードスーツを着たdiv要素がjQueryオブジェクトであるとイメージしていただけたらいいかと思います。

### ヒント◎コメントとは

JavaScriptで`//`を書くと、その行で`//`以降に書いた文字列はコメントとして扱われます。何かメモを残しておきたい時などに利用します。本書では、ちょっとした解説を入れるため、このコメント書式を使っています。

## 何がそんなに大変なの

ここまでの内容をまとめるとこんな感じになります。

- \$がjQuery
- \$()で何かを取ってくる
- そして取ってきた要素をjQueryオブジェクト化する
- jQueryオブジェクトにすると便利機能が使える
- その便利機能なしだと何んなことが大変

最後に挙げた、いろんなことが大変であるという点について少し解説します。

jQueryがなくなっても、いろんなことはできます。できますし、どう書けばどう動くのかという、れっきとした仕様があります。DOM (Document Object Model) という仕様です。DOMに則ってJavaScriptを書けば、ほとんどの場合、新しいブラウザでは意図した通りに動作します。ですが、DOMへの準拠がいまいちなブラウザもたくさんあります。そういったブラウザでは、「クリックしてダイアログを出す」のような単純なことをするJavaScriptのコードを素直に書いても、エラーになり動作しません。

では、どうすればいいのでしょうか。DOMへの準拠がいまいちなブラウザは、クリックしてダイアログを出すことすら不可能なのでしょうか。もちろん、そんなことはありません。そういったしくみは昔からほとんどのブラウザで実現可能でした。ですが、DOMの仕様として定義されているしくみとは違った方法です。仕様が固まっていなかったため、各ブラウザが独自にしくみを用意していたのです。要するに、何かをしたかったら、そういった問題をすべてクリアしなければなりません。例えば、「クリックしたら何かを起こさせる方法を実現するためには、次のステップを踏む必要があります。

- クリックイベントについて、DOMで定義されている仕様を調べる
- 対象とする各ブラウザについて、対応状況を調べる
- 対応していない各ブラウザについて、そのブラウザ固有の仕様を調べる
- 対応状況を判別して、別の処理を書く

実際に、「クリックしたらダイアログを出す」という動作を、現在主に使われている各ブラウザで問題なく動作させるには、次のように書く必要があります。

\* \$というのは、JavaScript的にはなんの意味もないただの文字で、aとかbとかcとかと同じです。よく使う機能を何度もタイプするのは面倒であるという理由で、このような意味ありげな一文字に機能を持たせてあります。