ソフトウェア開発
第 11 回目授業

平野 照比古

2018/12/14

問題 1

- PUT では URL にパラメータが記述される
- 直接記述してもサーバーのプログラムが起動される
- したがって、元のページでデータのチェックをしても、呼び出しの方でもデータのチェックをする必要がある。
- POST では直接見えないが、呼び出し側で直接記述が可能 (前回の Ajax 通信を POST で行うところを参照)
- サーバーを起動して実行していない場合もあった。

問題 2

- ブラウザのバージョンで内容が異なる。
- 最近のブラウザはそれほど多くの情報を与えないようである。
- セキュリティに関する考察が欲しい。

jQuery とは

jQuery の開発元[1] には次のように書かれている。

> *jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.*

[1] jQuery.com 参照日:2018/12/12

jQuery の利用法

1. jQuery はjquery.comのページの右上にある「Download jQuery」をクリックすることでダウンロードのページに行く
2. 次の 2 つの最新版がダウンロード可能
   Download the compressed, production jQuery 3.3.1
   Download the uncompressed, development jQuery 3.3.1
3. ファイルの違いはコメントなどがそのまま入っているもの (uncompressed) と、コメントを取り除き、変数名などを短いものに置き換えて、ファイルサイズを小さくしたもの (compressed)
4. ダウンロードしたファイルを取り込む<script>要素を記述すれば利用できる。
5. または、CDN(Contents Deliverly Network) を利用

```
<script
  src="https://code.jquery.com/jquery-3.3.1.min.js"
  integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8
  crossorigin="anonymous"></script>
```

jQuery オブジェクト

- jQuery ライブラリーは `jQuery()` というグローバル関数を一つだけ定義
- この関数の短縮形として、`$`というグローバル関数も定義
- jQuery() 関数はコンストラクタではない。
- jQuery 関数は引数の与え方により動作が異なる
- 戻り値として jQuery オブジェクトとと呼ばれる DOM の要素群を返す
- このオブジェクトには多数のメソッドが定義されていて、これらの要素群を操作可能

jQuery 関数の引数の型 (1)

jQuery() 関数の呼び出し方は引数の型により返される要素群が異なる。

- 引数が (拡張された)CSS セレクタ形式の場合(機能 **1**)
  - CSS セレクタにマッチした要素群を返す。
  - 省略可能な 2 番目の引数として要素や jQuery オブジェクトを指定した場合には、その要素の子要素からマッチしたものを返す。
  - この形はすでに解説した DOM のメソッド `querySelectorAll()` と似ている。
- 引数が要素や Document オブジェクトなどの場合(機能 **2**)
  与えられた要素を jQuery オブジェクトに変換する。

jQuery 関数の引数の型 (2)

- 引数として HTML テキストを渡す場合(機能 **3**)
  - テキストで表される要素を作成し、この要素を表す jQuery オブジェクトを返す。`createElement()` メソッドに相当する。
  - 省略可能な 2 番目の引数は属性を定義するものであり、オブジェクトリテラルの形式で与える。
- 引数として関数を渡す場合(機能 **4**)
  引数の関数はドキュメントがロードされ、DOM が操作で可能になった時に実行される。

jQuery オブジェクトのメソッドの基本

jQuery オブジェクトに対する多くの処理は HTML の属性や CSS スタイルの値を設定したり、読み出したりすること

- これらのメソッドに対してセッターとゲッターを同じメソッドを使う。メソッドに値を与えるとセッターになり、ないとゲッターになる。
- セッターとして使った場合は戻り値が jQuery オブジェクトとなるので、メソッドチェインが使える。
- ゲッターとして使った場合は要素群の最初の要素だけ問い合わせる。

HTML 属性の取得と設定

`attr()` メソッドは HTML 属性用の jQuery のゲッター/セッターである。

- 属性名だけを引数に与えるとゲッターとなる。
- 属性名と値の 2 つを与えるとセッターになる。
- 引数にオブジェクトリテラルを与えると複数の属性を一時に設定できる。
- 属性を取り除く `removeAttr()` もある。

CSS 属性の取得と設定

`css()` メソッドは CSS のスタイルを設定する。

- CSS スタイル名は元来の CSS スタイル名（ハイフン付）でも JavaScript のキャメル形式でも問い合わせ、設定が可能である。
- 戻り値は単位を含めて文字列で返される。

HTML フォーム要素の値の取得と設定

- `val()` は HTML フォーム要素の `value` 属性の値の設定や取得が可能
- これにより、`<select>`要素の選択された値を得ることなどが可能

要素のコンテンツの取得と設定

- `text()` と `html()` メソッドはそれぞれ要素のコンテンツを通常のテキストまたは HTML 形式で返す。
- 引数がある場合には、既存のコンテンツを置き換える。

DOM 構造を変える

Table: ドキュメントの構造の変更するメソッド

$(T).method(C)	$(C).method(T)	機能
append	appendTo	要素 T の最後の子要素として C を付け加える
prepend	prependTo	要素 T の初めの子要素として C を付け加える
before	insertBefore	要素 T の直前の要素として C を付け加える
after	insertAfter	要素 T の直後の要素として C を付け加える
replaceWith	replaceAll	要素 T と C を置き換える

DOM の操作 (補足)

- 要素をコピーする `clone()`
- 要素の子要素をすべて消す `empty()`
- 選択された要素 (とその子要素すべて) を削除する `remove()`

イベントハンドラーの登録

- イベントの種類ごとにメソッドが定義されていてる。
- 指定された jQuery オブジェクトが複数の場合にはそれぞれに対してイベントハンドラーが登録される。

イベントハンドラー登録のメソッド

```
blur()          focusin()     mouseenter()  scroll()
change()        keydown()     mouseleave()  select()
click()         keypress()    mousemove()   submit()
dbleclick()     keyup()       mouseover()   unload()
error()         load()        mouseup()
focus()         mousedown()   resize()
```

イベントハンドラーの登録

- `hover()` : `mouseenter` イベントと `mouseleave` イベントに対するハンドラー を同時に登録できる。
- `toggle()` はクリックイベントに複数のイベントハンドラー を登録し、イベントが発生するごとに順番に呼び出す。

イベントハンドラ– の登録解除

イベントハンドラ– の登録解除は unbind() メソッドで行う。

- removeEvenListener() と同様な形式として 1 番目の引数にイベントタイプ (文字列で与える)、2 番目の引数に登録した関数を与えるものがある
- 登録したイベントハンドラ– には名前が必要

jQuery.ajax() 関数

- jQuery では Ajax の処理に関するいろいろな方法を提供
- jQuery.ajax() 関数を紹介
- この関数は引数にオブジェクトリテラルをとる。

jQuery.ajax() 関数の引数

Table: jQuery.ajax() 関数で利用できる属性 (一部))

属性名	説明
type	通信の種類。通常は"POST"または"GET"を指定
url	サーバーのアドレス
data	"GET"のときは URL の後に続けるデータ。"POST"のときも同様
dataType	戻り値のデータの型を指定する。"text"、"html"、"script"、"json"、"xml"などがある
success	通信が正常終了したときに呼び出されるコールバック関数
error	通信が成功しなかったときに呼び出されるコールバック関数
timeout	タイムアウト時間をミリ秒単位で指定

今日は何の日—jQuery 版 (1)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>今日は何の日 (jQuery 版)</title>
6
7 <script type="text/ecmascript" src="jquery-3.3.1.min.js"></script>
```

6 行目はローカルに保存した jQuery ライブラリーを読み込んでいる。

今日は何の日—jQuery 版 (2)

```
 8 <script type="text/ecmascript">
 9 //<![CDATA[
10 $(window).ready(function(){
11     function makeSelectNumber(from, to, prefix, suffix, id, parent){
12       let Select = $("<select/>", {"id":id});
13       if(parent) parent.append(Select);//$(parent).append(Select);
14       for(let i=from; i<=to; i++) {
15         option = $("<option/>",{"value":i,"text":prefix+i+suffix});
16         Select.append(option);
17 //           $("<option/>",{"value":i,"text":prefix+i+suffix}).appendTo(Select);
18       }
19       return Select;
20     };
21     let today = new Date();
```

今日は何の日—jQuery 版 (2) 解説

- 10 行目の$(window).redy() はドキュメントが解釈されたときに引数の関数が呼び出されるイベントである。ここでの要素の参照は機能 2を用いている。
- 11 行目から 21 行目は連続した番号を値に持つプルダウンメニューを作成する関数。仕様は以前と同じ
  - 13 行目では<select>要素を作成し、同時に属性も設定 (機能 3)。
  - 15 行目から 19 行目で<option>要素を定義し、<select>要素の子要素にしている。なお、15 行目と 16 行目は appendTo() メソッドを用いると 18 行目のコメントアウトしてあるように記述することができる。

今日は何の日—jQuery 版 (3)

```
22    let y = today.getFullYear();
23    let m = today.getMonth();
24    let Form  = $("#menu");
25    let Year  = makeSelectNumber(2000,2020,"","年","year", Form);
26    let Month = makeSelectNumber(1,12,"","月","month", Form);
27    let Days = [];
28    for(let i=28; i<=31; i++) {
29      Days[i] = makeSelectNumber(1,i,"","日","day");
30    }
31    Year.val(y);      //$("#year").val(y);
32    Month.val(m+1);   //$("#month").val(m+1);
33    let d = new Date(y, m+1,0).getDate();
34    Form.append(Days[d]);
35    $("#day").val(today.getDate());
36    let changePulldown = function(){
```

今日は何の日—jQuery 版 (3) 解説

- 22 行目から 31 行目も依然とほとんど同じである。25 行目では機能 3を用いて要素を参照している。
- 32 行目、33 行目と 36 行目はプルダウンメニューの初期値を本日に設定している。

今日は何の日—jQuery 版 (4)

```
37        let d2 = $("#day").val();
38        d = new Date(Year.val(), Month.val(), 0).getDate();
39        if( d != $("#day option").length) {
40          $("#day").replaceWith(Days[d]);
41          $("#day").val(Math.min($("#day option").length, d2));
42        }
```

今日は何の日—jQuery 版 (4) 解説

- jQuery のメソッドを用いていることをのぞけは 38 行目から 43 行目までは以前と同じである。
- 40 行目のセレクタは id が day(#day) の下にある<option>要素を選択し、その数を length で調べている。

今日は何の日—jQuery 版 (5)

```
43        jQuery.ajax({
44          type:"GET",
45          url:       "./aniversary.php",
46          data:      "month=" + Month.val()+ "&day="+d2,
47          dataType: "text",
48          success : function(Data){
49            $("#details").text(Data);
50          },
51          error:function(){alert("error");}
52        });
53      };
54      Form.change(changePulldown);
55      changePulldown();
56  });
57  //]]>
58  </script>
59  </head>
```

今日は何の日—jQuery 版 (5) 解説

- 44 行目から 53 行目は Ajax の処理を定義
- ブラウザによる違いに対する対処や、XMLHTTPRequest の処理部分が大幅に減っていて見やすいコードになっている

今日は何の日—jQuery 版 (6)

```
60 <body>
61   <form id="menu"></form>
62   <p id="details"></p>
63 </body>
64 </html>
```