

ソフトウェア開発 第 8 回目授業

平野 照比古

2015/11/20

Google Maps の利用 (1)

ソフトウェア開発
第 8 回目授業

平野 照比古

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

レポート問題

次のリストは Google Maps を利用して地図を表示するものである。

```
1  <!DOCTYPE html>
   <html>
   <head>
   <meta http-equiv="Content-Type"
5   content="text/html; charset=utf-8"/>
   <title>初めての GoogleMaps</title>
   <script type="text/javascript"
     src="http://maps.google.com/maps/api/js?sensor=false"></script>
```

Google Maps の利用 (2)

ソフトウェア開発
第 8 回目授業

平野 照比古

```
<script type="text/javascript">
10 window.onload = function() {
    var latlng = new google.maps.LatLng(35.486210,139.341443);
    var myOptions = {
        zoom: 10,
        center: latlng,
15    mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var mapCanvas = document.getElementById("map_canvas")
    var map = new google.maps.Map(mapCanvas, myOptions);
    }
20 </script>
<link rel="stylesheet" type="text/css" href="map.css" />
</head>
<body>
    <div id="map_canvas" ></div>
25 </body>
</html>
```

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

レポート問題

リストの解説 (1)

ソフトウェア開発
第 8 回授業

平野 照比古

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

レポート問題

- ▶ 1 行目は HTML 文書の DOCTYPE 宣言である。この形は HTML5 におけるもの
- ▶ 2 行目はこの HTML 文書のルート要素と呼ばれるものである。最後の 26 行目の `</html>` まだが有効となる。
- ▶ すべての要素はこの範囲になければならない。

リストの解説 (2)-<head>

3 行目から始まる<head>はブラウザに表示されない、いろいろな HTML 文書の情報を表す。

- ▶ 4,5 行目はこの文書の形式や文字集合を記述している。ここでは内容は text/html の形式、つまり、テキストで書かれた html の形式で書かれていることを表す。¹
- ▶ 6 行目の<title>はブラウザのタブに表示される文字列を指定している。
- ▶ 7,8 行目は Google Maps のライブラリーを読み込むためのものである。このように JavaScript のプログラムは外部ファイルとすることができる。
- ▶ 9 行目から 20 行目は HTML 文書内に書かれた JavaScript である。詳しい解説は後の授業で行う。
- ▶ 20 行目は HTML 文書の見栄えなどを規定する CSS ファイルを外部から読み込むことをしている。

¹このような方法でファイルのデータ形式を表すことを MIME(Multipurpose Internet Mail Extension) タイプと呼ぶ。元来、テキストデータしか扱えない電子メールに様々なフォーマットのデータを扱えるようにする規格である。

HTML 文書で実際にブラウザ内で表示される情報は
<body>要素内に現れる。

- ▶ このリストでは Google Maps を表示するための
<div>要素が一つあるだけである。このとき、<div>は
<body>の子要素であるといい、<body>は<div>の親要素という。
- ▶ 各要素名または要素の終了を示すタグ (<...>) の間に
文字列がある場合、その部分はテキストノードと呼ば
れるノードが作成されている。

各要素は<との中に現れる。初めに現れる文字列が要素名であり、そのあとに属性と属性値がいくつか並ぶ。

- ▶ 属性とその属性値は=で結ばれる。
- ▶ 属性値は”ではさまれた文字列として記述
- ▶ <script>要素では属性 type と src が設定
- ▶ 24 行目の<div>要素では属性 id に属性値 map_canvas を設定している。なお、この要素は CSS によっても属性が定義されている。

21 行目で参照している CSS ファイル

```
#map_canvas{
    width:500px;
    height:500px;
    float:left;
    margin:5px 10px 5px 10px;
}
```

- ▶ CSS の各構成要素は HTML 文書の要素を選択するセレクタ (ここでは #map_canvas) とそれに対する属性値の並び ([属性]:[属性値];) からなる。
- ▶ #で始まるセレクタはそのあとの文字列を <id> の属性値に持つ要素に適用される。
- ▶ したがって、ここの規則は 24 行目の <div> 要素に適用
- ▶ その内容は Google Maps が表示される画面の大きさ (width と height)、配置の位置 (float) と要素の外に配置される空白 (margin) を指定

- ▶ カスケーディングスタイルシート (CSS) は HTML 文書の要素の表示方法を指定するもの
- ▶ CSS は JavaScript から制御可能
- ▶ 文書のある要素に適用されるスタイルルールは、複数の異なるルールを結合 (カスケード) したもの
- ▶ スタイルを適用するためには要素を選択するセレクタで選ぶ。

セレクトの種類は配布資料を参照のこといくつか注意する点を挙げる。

- ▶ 属性 `id` の属性値の前に `#` をつけることでその要素が選ばれる。
- ▶ 属性 `class` の属性値の前に `.` をつけることでその要素が選ばれる。
- ▶ `nth-child(n)` には単純な式を書くことができる。このセレクトは複数書いてもよい。
- ▶ `E F` と `E > F` の違いを理解しておくこと。たとえば `div div` というセレクトは途中で別の要素が挟まれていてもよい。また、`<div>` 要素が 3 つある場合にはどのような 2 つの組み合わせも対象となる。

- ▶ Document Object Model(DOM) は HTML 文書などの要素をノードとしたツリー構造で管理する方法
- ▶ DOM のメソッドやプロパティを使うことで各要素にアクセスしたり、属性値やツリーの構造を変化させることが可能
- ▶ DOM の構造は開発者ツールなどで見ることができる。

代表的なブラウザの開発者ツール

- ▶ Opera の開発者ツールから Web インスペクタを開き、Elements タブで確認する。ここで要素上で右クリックして Edit as HTML を選択するとテキストとして編集できる。
- ▶ Google Chrome でも同様な操作が可能
- ▶ Firefox では開発ツールから「開発ツールを表示」を選択し、インスペクタタブで DOM ツリーが確認できる。要素上で右クリックから「HTML として編集」とするとテキストとして編集できる。
- ▶ IE では開発者ツールを開き左にあるタブの一番上にある DOM Explorer で同様のことができる。

DOM では DOM ツリーを操作するためにメソッドやプロパティが規定されている。メソッドとはそのオブジェクトに対する操作である。次のような手段を提供している。

- ▶ 条件に合う要素または要素のリストを得る。
- ▶ 要素の属性を参照、変更ができる。
- ▶ 要素を新規に作成する。
- ▶ ある要素に子要素を追加したり、取り除いたりする。

条件に合う要素または要素のリストを得る (2)

- ▶ `querySelector(selectors)`
`selectors` で指定された CSS のセレクトタに該当する一番初めの要素を得る
- ▶ `querySelectorAll(selectors)`
`selectors` で指定された CSS のセレクトタに該当する要素のリストを得る。配列と同様に [] で参照できる

要素の属性を参照、変更ができる

- ▶ `getAttribute(Attrib)`
対象要素の属性 `Attrib` の値を得る。戻り値の型は文字列である。
- ▶ `setAttribute(Attrib,Val)`
対象要素の属性 `Attrib` の値を `Val` にする。数を渡しても文字列に変換される。
- ▶ `hasAttribute(Attrib)`
対象要素に属性 `Attrib` がある場合は `true` を、ない場合は `false` を返す。
- ▶ `removeAttribute(Attrib)`
対象要素の属性 `Attrib` を削除

要素を新規に作成する

ソフトウェア開発
第 8 回目授業

平野 照比古

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

レポート問題

- ▶ `createElement(Name)`
Name で指定した要素を作成
- ▶ `createElementNS(NS,Name)`
名前空間 NS で定義されている要素 Name を作成
- ▶ `createTextNode(text)`
text を持つテキストノードを作成
- ▶ `cloneNode(bool)`
bool が true のときは対象要素の子要素すべてを、
false のときは対象要素だけの複製を作る。

- ▶ 指定した要素が定義されている規格を指定するもの
- ▶ 一つの文書内で複数の規格を使用する場合、作成する要素がどこで定義されているのかを指定
- ▶ 異なる規格で同じ要素名が定義されていてもそれらを区別することが可能
- ▶ 通常の HTML 文書では
`http://www.w3.org/1999/xhtml` を指定

- ▶ `appendChild(Elm)`
Elm を対象要素の最後の子要素として付け加える。Elm がすでに対称要素の子要素のときは元の位置から最後の位置に移動
- ▶ `insertBefore(newElm, PElm)`
対象要素の子要素 PElm の前に newElm を子要素として付け加える。Elm がすでに対称要素の子要素のときは元の位置から指定された位置に移動
- ▶ `removeChild(Elm)`
対象要素の子要素 Elm を取り除く。
- ▶ `replaceChild(NewElm, OldElm)` 対象要素に含まれる子要素 OldElm を NewElm で置き換える。
- ▶ `setValue(value)`
対象のテキストノードの値を value にする。

プルダウンメニューの例

1 月から 12 月までを選択できるプルダウンメニューを作成

```
<body>
  <form id="menu">
    <select>
      <option value="1">1 月</option>
      <option value="2">2 月</option>
      <option value="3">3 月</option>
      <option value="4">4 月</option>
      <option value="5">5 月</option>
      <option value="6">6 月</option>
      <option value="7">7 月</option>
      <option value="8">8 月</option>
      <option value="9">9 月</option>
      <option value="10">10 月</option>
      <option value="11">11 月</option>
      <option value="12">12 月</option>
    </select>
  </form>
</body>
```

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

レポート問題

- ▶ ユーザからの入力を受け付ける要素は通常、`<form>`要素内に記述
- ▶ プルダウンメニュー の要素名は`<select>`
- ▶ 選択する内容は`<option>`要素
- ▶ `<option>`要素の属性 `value` の値が選択した値として利用できる。
- ▶ `<option>`要素内の文字列 (テキストノード) がプルダウンメニューに表示される
- ▶ `<select>`は`<form>`の子要素であり、各`<option>`は`<select>`の子要素

プルダウンメニュー——プログラムで作成 (1)

ソフトウェア開発
第 8 回目授業

平野 照比古

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

レポート問題

```
1  <!DOCTYPE html>
   <html>
   <head>
   <meta http-equiv="Content-Type" content="text/html; charset=
5  <title>プルダウンメニューの作成</title>
   <script type="text/ecmascript">
   //<![CDATA[
       window.onload = function(){
           var i;
10      var Form = document.getElementById("menu");
           var Select = document.createElement("select");
           var Option, Text;
           Form.appendChild(Select);
```

プルダウンメニュー——プログラムで作成 (2)

ソフトウェア開発
第 8 回目授業

平野 照比古

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

レポート問題

```
15     for(i=1;i<=12;i++) {  
        Option = document.createElement("option");  
        Option.setAttribute("value",i);  
        Select.appendChild(Option);  
        Text = document.createTextNode(i+"月");  
        Option.appendChild(Text);  
20     }  
    }  
//]]>  
</script>  
</head>  
25 <body>  
    <form id="menu">  
    </form>  
</body>  
</html>
```


◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ≡ ↺ 🔍 ↻

Table: DOM 要素に対するプロパティ(2)—DOM4

プロパティ名	説明
children	子要素のうち通常の要素だけ からなる要素の配列
firstElementChild	指定された要素の先頭にある 通常の要素である子要素
lastElementChild	指定された要素の最後にある 通常の要素である子要素
nextElementSibling	指定された子要素の次の通常 の要素
previousElementSibling	現在の子要素の前にある通常 の要素

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは
DOM のメソッド
DOM のプロパティ

レポート問題

レポート問題

ソフトウェア開発
第 8 回授業

平野 照比古

HTML 文書の構成

CSS の利用

DOM の利用

DOM とは

DOM のメソッド

DOM のプロパティ

レポート問題

課題 7.1 から 7.4 までレポートにして提出のこと。