

Webプログラミング実習1

2017/5/10

Kazuma Sekiguchi

class@cieds.jp

告知

- Adobe XD Meeting #05
 - 5月19日（金）19:00～（予定）開催予定
 - 参加費は無料。懇親会は実費
 - Adobe本社会議室（大崎駅）
 - ワイヤーフレーム会
- 参加希望の方は、@adobexd_studyをフォローするか、<https://xd-study.connpass.com/>でメンバー登録しておくと、申し込み受付が始まれば連絡があります

DOM

- DOM=Document Object Model
- JSからHTMLやCSSを操作するための仕組み
 - HTMLによるタグ構造もDOMツリーやDOMと呼ぶことも
 - HTMLは<html>を最上位にした構造になっている
- JSからCSSなどを操作してさまざまな表現が可能となる

DOM

- JSからCSSを操作する
 - CSSはHTMLタグに紐付いて動作するため、JSからHTMLへアクセスして、CSSを操作する
 - 要素をオブジェクトとして扱い、CSSを変更する
- HTMLタグを指定する必要
 - HTMLタグは複数の種類、複数記述することでHTMLを構成
 - 表現を変えるためには、表現を変えたいところピンポイントでHTMLタグを指示する必要がある

DOM

- 指し示す方法
 - タグを利用
 - ID属性を利用
 - CLASS属性を利用
- 一番簡単なのはID属性を利用してピンポイントに指し示すことが可能
 - ID属性は单一（ほかに同一のIDは存在してはいけないため）
- CLASS属性及びタグは同一のものが複数存在する可能性がある
 - 複数の該当からピンポイントに指し示す必要があるため、順番を指定することでピンポイントに指し示す

ID属性を利用して要素取得

```
document.getElementById("id名");
```

- `document.getElementById`を利用し、ID名を指定する
と要素を指定可能
 - 要素を取得するとCSSやHTMLの属性を変更可能

タグやCLASS属性を利用して要素取得

```
document.getElementsByTagName("タグ名");  
document.getElementsByClassName("クラス名");
```

- どちらも複数の要素にマッチする可能性があるため、単一の要素は取得できない
 - 要素にアクセスする場合は、インデックス（順番）を付与してアクセスする必要がある

取得した要素でHTML書き換え

```
var main = document.getElementById("main");
    main.innerHTML = "テキストを書き換え";
var sub = document.getElementsByTagName("p");
sub[0].innerHTML = "1番目のp要素のテキストを書き換え";
```

- 取得した要素（変数に格納したもの）のinnerHTMLプロパティへアクセスすると、要素内のテキストが取得可能
 - =で繋げると、新しいテキスト（HTML）を設定可能
- TagNameなどの複数取得される場合は、[順番]を付与して利用する
 - 順番は0から始まる点、注意

取得した要素でCSS書き換え

```
var body = document.getElementsByTagName("body");
body[0].style.backgroundColor="#FF9900";
```

- 取得した要素（変数に格納したもの）のstyleプロパティへアクセスし、CSSプロパティ名を指定すると現在の値を取得可能
 - =で繋げると、新しい値を設定可能
- CSSプロパティ名は「-」を抜いて、抜いたハイフンの次の文字を大文字にする

取得した要素で属性書き換え

```
var pElem = document.getElementsByTagName("p");
    pElem[0].setAttribute("class","move");
```

- 取得した要素（変数に格納したもの）のsetAttribute
メソッドを利用すると、要素に対して属性を付与可能
 - 既に属性が付与されていた場合は、上書きになる

取得した要素にイベント付与

```
var pElem = document.getElementsByTagName("p");
pElem[0].onclick=function(){
    alert("click");
}
```

- 取得した要素（変数に格納したもの）にonClickなどのイベントを付与すると、イベントを紐付けることが可能
- =の後ろ側には、function()を付けて、イベント実行時にたいたい操作を記述
 - 無名関数
 - 名前を付けずに利用することが可能な独自関数
 - 即時に実行されるためJSでは非常に良く利用される（イベントでは関数を呼び出すことになっているため、無名関数はコード数を減らすことができる）

windowオブジェクト

- ・ブラウザー自身を指示するオブジェクト
 - ・ブラウザーウィンドウサイズなどをプロパティとして保持している
 - ・ブラウザーウィンドウに合わせて表示を変える場合などで利用

window.innerWidth → ブラウザーウィンドウの幅サイズ(実際には描画領域幅サイズ)

window.innerHeight → ブラウザーウィンドウの高さサイズ(実際には描画領域幅サイズ)

```
var w = window.innerWidth;  
var h = window.innerHeight;
```

windowオブジェクト

- DOMにアクセスする場合、windowオブジェクトがロードし終わった段階でアクセスする必要がある
 - JSはDOMツリーが構築される前に実行される
 - DOMツリーが構築されていない状態でタグにアクセスしてしまうため、エラーになる

```
window.onload=function(){  
    //DOMツリーが構築されたら実行したい部分  
}
```