

マークアップエンジニア デザイナー向け JavaScript入門

2007年7月14日 UK

最近アイコン変えた



- 名前 : UK
- 職業 : Webプログラマ(といつつ雑用多し)
- UK STUDIO(<http://uk-studio.net/>)の管理人
- JS歴 : 半年ぐらい
- めがねっこ

JavaScriptの書き方 - HTMLに直接書く

```
<script type="text/javascript">  
<!--  
  (ry  
  //-->  
</script>
```

コメントアウトはしたほうがいい(モダンブラウザ対策)

コメントアウトしてないと？

ブラウザによってはソースコードが表示されちゃう(´・ω・｀)

JavaScriptの書き方 - HTMLに直接書く

<とか&は直接書きちゃだめ><

<とか &と書くべし

XHTMLならCDATAセクションもあり

```
<script type="text/javascript">  
<![CDATA[  
  (ry  
]]  
</script>
```

でもCDATAセクションに対応してないブラウザは？

```
<script type="text/javascript">  
/*<![CDATA[*/  
  (ry  
/*]]*/  
</script>
```


JavaScriptの書き方 - HTMLに直接書く

正直めどい

ぶっちゃけ外部スクリプトでかけばよくね？

```
<script type="text/javascript" src="./js/sample.js"></script>
```

変数

宣言

```
var hoge;  
var foo = "bar";  
var name = "UK", age = 20; //一行にまとめて
```

命名規則

アルファベット(a-z,A-Z)、数字、アンダースコア(_)の組み合わせ

最初の文字に数字は使えない

予約語は使えない

リテラル

プログラミングにおいて、ソースコード内に値を直接表記したものを言う(via Wikipedia)

数値

文字列

真偽値

Null

```
var integer = 100;  
var string = “文字列”;  
var death_flag = true;  
var nullpo = null;
```

配列

複数のデータに番号をつけて管理できる

番号のことをインデックス(添字)と言い0から始まる

```
var browser = ["InternetExplorer","Firefox","Safari","Netscape"];
```

```
var browser = Array("InternetExplorer","Firefox","Safari","Netscape");
```

```
var browser = new Array("InternetExplorer","Firefox","Safari","Netscape");
```

配列へのアクセスと代入

```
alert(browser[0]);      // InternetExplorer  
browser[0] = "Opera";  
alert(browser[0]);      // Opera
```


連想配列

インデックスに文字列を使用できる

```
var UK = new Array();  
UK["name"] = "赤松 祐希";  
UK["age"] = 20;  
alert(UK["age"]); //20
```

演算子

算術演算子

+ - * / % ++ --

代入演算子

= += -=

比較演算子

== != > < >= <=

論理演算子

&& ||

ビット演算子

& | ^ ~ << >> >>>

制御構造

if文

```
if(条件式){  
    条件式がtrueだったら実行される処理  
}else if(条件式2){  
    条件式がfalse、条件式2がtrueだったら実行される処理  
}else{  
    どの条件式もfalseだったら実行される処理  
}
```

while文

```
while( 条件式){  
    処理  
}
```

制御構造

for文

```
for(初期条件; 条件式; 条件の変更){  
    処理  
}
```

do while文

switch文

今回は省略します＞＜

関数

数学

```
f(x,y) = x + y
```

JavaScript

```
function add(x,y){  
  var answer = x + y;  
  return answer;  
}
```

数学もJavaScriptも引数を与えて結果をもらうという点では同じ
(引数も返り値もない関数もあるけど)

```
function hello(){  
  alert("Hello,JavaScript");  
}
```

変数のスコープ

グローバル変数とローカル変数の2種類

```
var param = "global";  
  
function func_a(){  
    var param = "local";  
    alert(param);  
}  
func_a(); //local  
alert(param); //global
```

```
var param = "global";  
  
function func_a(){  
    param = "local";  
    alert(param);  
}  
func_a(); //local  
alert(param); //local
```

var の有無でスコープの範囲が変わる

オブジェクト

オブジェクトって結局なによ! ('▽') ヨーワカン
とりあえずつくってみればわかるかも!?
じゃあまずは一番シンプルなオブジェクトから

```
var obj = {};
```

これでもちゃんとしたオブジェクトなんです><

```
alert(obj); //[object Object]
```

ほらね!

もっかいオブジェクト

プロパティもあるオブジェクト

```
var obj = {  
  name: “オブジェクト”  
};  
alert(obj.name); //オブジェクト
```

プロパティの変更、追加もできるよ

```
obj.name = “すべらないななし”;  
obj.age = 55;  
alert(obj.name);    //すべらないななし  
alert(obj.age);     //55
```

プロパティ -> 変数と大差ないよ

まだまだオブジェクト

メソッドもあるオブジェクト

```
var obj = {  
  name : “オブジェクト”  
  
  getName : function(){  
    return this.name;  
  }  
};  
alert(obj.getName()); //オブジェクト
```

メソッド -> 関数と大差ないよ

もうちょっと具体的に言うと

メソッド -> Functionオブジェクトが代入されたプロパティ

隣のあいつもオブジェクト

文字列(Stringオブジェクト)

```
alert("Hello,JavaScript".length); //14 プロパティがある
```

数値(Numberオブジェクト)

```
var num = 100;  
alert(num.toString()); //”100” メソッドがある
```

文字列も数値もJavaScript側が用意してるオブジェクト

-> **ビルトインオブジェクト**

配列(Arrayオブジェクト)

真偽値(Booleanオブジェクト)

関数(Functionオブジェクト)

日付(Dateオブジェクト)

数学(Mathオブジェクト)

正規表現(RegExpオブジェクト)

エラー(Errorオブジェクト)

しつこいけどオブジェクトの話

ビルトインオブジェクトとは別に

ナビゲータオブジェクトというものが存在する

ビルトインオブジェクト -> 言語仕様レベル

ナビゲータオブジェクト -> ブラウザレベル

windowオブジェクト

locationオブジェクト

historyオブジェクト

documentオブジェクト

後ほど説明するDOMが扱うのはdocumentオブジェクト

オブジェクトまとめ

JavaScriptでいうオブジェクトとはプロパティとメソッドの集合体

プロパティ -> 変数みたいなもの

メソッド -> 関数みたいなもの

(正確にはFunctionオブジェクトが代入されてるプロパティ)

数値、文字列などはビルトインオブジェクトとして用意されてる

ブラウザが用意するオブジェクトはナビゲータオブジェクト

DOMが扱うのはナビゲータオブジェクトのdocumentオブジェクト

今回は話していませんでしたが、JavaScriptのオブジェクトには
prototypeという概念が存在します。重要な概念なので一度調べて
みることをお勧めします。

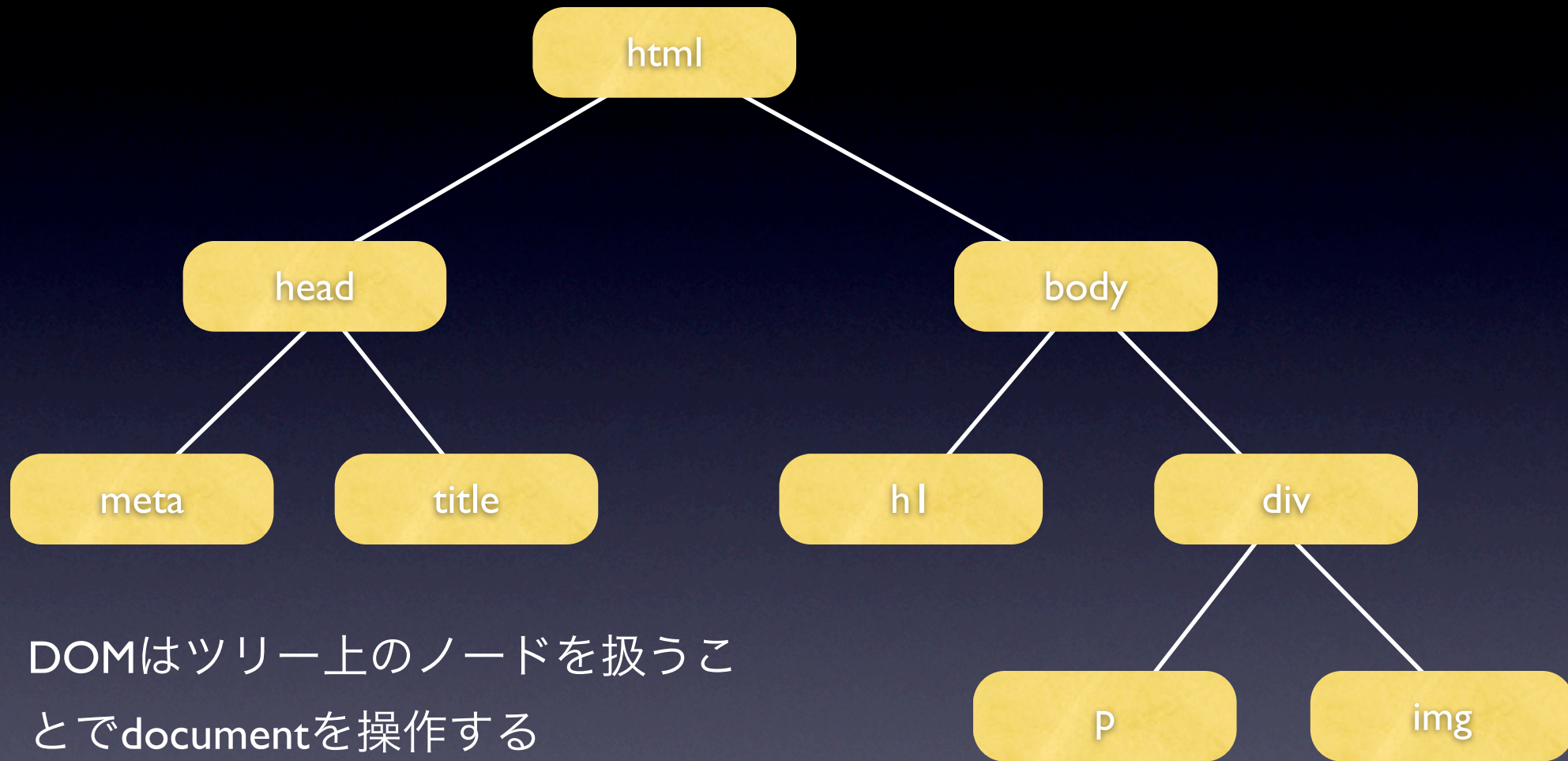
やっ とDOMの話

マークアップエンジニア、デザイナー向けの入門なんだから
DOMはやっておかないとだめでしょう！

とか言いつつ、かなりあっさりです。

目標としては動的にCSSを書き換える程度を目指してます。

document ツリー



DOMはツリー上のノードを扱うことでdocumentを操作する

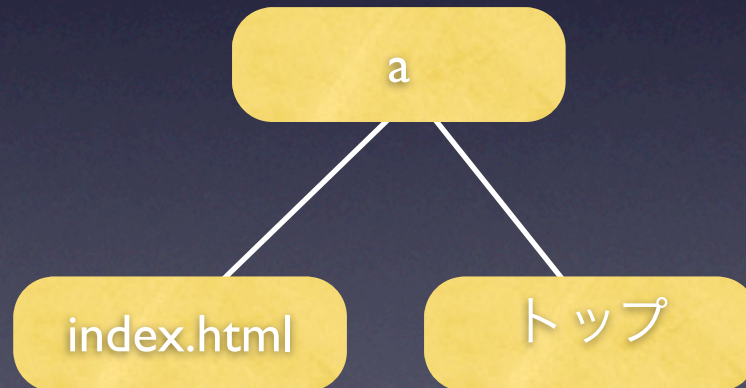
ノードとはdocumentをオブジェクトとして扱う際の区切りの単位

ノードの種類

ノードにはいくつか種類があり、それぞれ扱えるプロパティやメソッドが異なります

```
<a href="index.html">トップ</a>
```

要素ノード



属性ノード

テキストノード

他にも
コメントノードや
ドキュメントノードなどがある

getElementById

getElementByIdは要素ノードを取得するメソッド

```
var obj = document.getElementById("sample");
```

要素ノードを取得できたらそこからCSSの書き換えをする

```
obj.style.backgroundColor = "#000";
```

この2行だけでCSSの書き換えを行うことができる

簡単でしょ?(´・ω・`)

キャメルケース

background-colorじゃなくてbackgroundColorなのはなぜ？

JavaScriptではハイフンをプロパティや変数名に使えないから

だからハイフンの次の文字を大文字にすることで対応している

-> キャメルケース

font-family	->	fontFamily	
padding-left	->	paddingLeft	
letter-spacing	->	letterSpacing	etc...

イベントハンドラ

動的にというぐらいだからなんらかの動作時にJSを処理させたい

例：id:buttonがクリックされたらid:naviの背景を変えたい!

```
var btn = document.getElementById("button");  
var navi = document.getElementById("navi");  
  
btn.onclick = function(){  
    navi.style.backgroundColor = "#ccc";  
}
```

onclickがイベントハンドラになる。

指定の要素がクリックされたら対応する処理を呼び出すという意味

他にもonmouseoverやonloadなどのイベントハンドラがある。

最後に

今回の発表はいかがでしたでしょうか。

説明が不十分だったり、足りない部分があったと思いますが
皆様に「JSをやってみよう！」とと思っていただけたら幸いです。

JavaScriptは最初はツンツンしてるかもしれませんが、
話していくうちにデレな部分も見せてくれます。
ゆっくり打ち解けていってください。

ご清聴ありがとうございました