

Webの基本詰め合わせセット

2015/1/17

Webアプリの基礎的な内容をパパ〜っとまとめ。

Webアプリのアーキテクチャ

Webアプリを作るには、はじめにWebアプリがどんな仕組みか大まかな構成(アーキテクチャ)を押さえとく必要がある。図1が基本的なWebアプリの構成になってるよ。WebアプリはHTTPプロトコル(80ポート)を介してデータをやりとりすることでブラウザ上でアプリが見たり操作できるようになるよ。データを要求する側(ブラウザなど)のことをクライアント、データを返す側のことをサーバー(コンピュータ)っていうよ(クライアント・サーバーモデル)。

HTTPプロトコルでは、データを要求することをHTTPリクエストっていうよ。要求する操作(メソッド)には、GET・POST・PUT・DELETEなどが代表的。やりとりされるデータは基本的になんでもアリ、代表的なデータとしてはHTML(PHP)・画像・動画・JSONなど。要求を受けたサーバーは内容に応じてデータを返す、このことをレスポンスというよ。

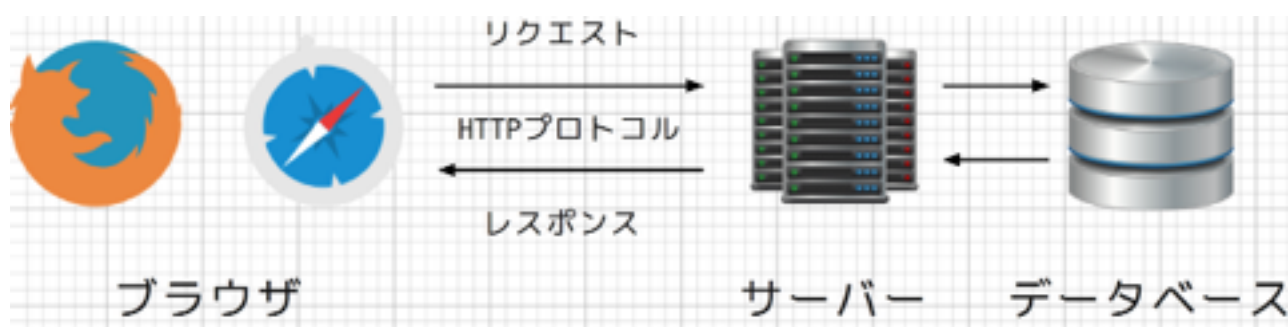


図1: Webアプリの基本アーキテクチャ

ブラウザ側、サーバー側では動かすことができるプログラミング言語が違うから下の表にまとめる。

言語	動作環境	用途
HTML	ブラウザ	ブラウザで表示される文書。タグによる階層構造を持つ。
Javascript	ブラウザ	HTMLを動的に操作するためのスクリプト言語。オブジェクト指向言語でもある。
CSS	ブラウザ	HTMLの見た目を操作するための言語。
PHP	サーバー	HTML文書にPHPタグを埋め込むことでサーバーサイドで実行展開されるスクリプト言語。

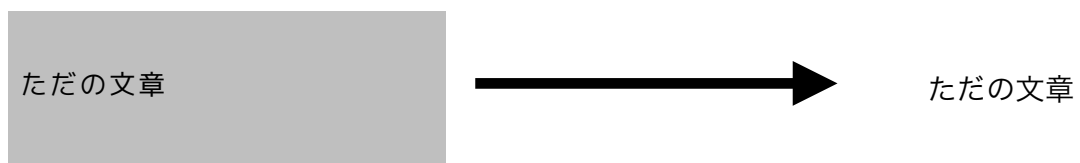
Java	サーバー	いろんな用途に使われるコンパイラ言語。オブジェクト指向言語。
SQL	サーバー	データベースを操作するためのDSL(特殊なケースに使われる言語のこと)。直接は実行せずに、PHPやJavaから実行される。

それぞれの言語は**実行される環境**が違うので注意！例えばPHPでHTMLを生成することはできるけどブラウザ上での動的な操作(ボタンをクリックしたらーとか)はできないし、Javascriptでデータベースを操作したりはできないよ！

HTMLのキホン

HTMLの基本をさっくりまとめるよ。HTMLはWeb上(今やいろんなところ)で文書を書くために作られたマークアップ言語だよ。マークアップっていうのは、ただ文字を羅列するんじゃなくて文章に意味を付加するって意味だよ。

例えばこれだと普通の文章だけど。



この場合だと、見出し(head line)の意味を持つ。

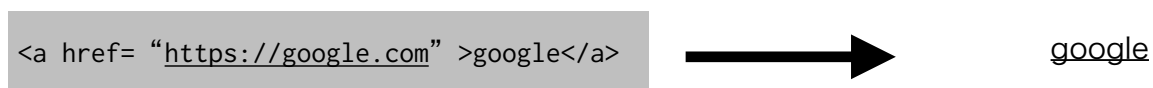


これだと、段落(paragraph)の意味を持つよ。



タグ名は大体意味のある名前の省略形だから意味も含めて覚えるとすぐ覚えれるよ。

HTMLタグは文字だけでなく、もっと色んな意味を持たせることができるよ。その意味のことを属性と言うよ。



ここでHTMLタグの文法を一般化すると以下のようなになるよ。タグによってどんな意味を持つようになるのか、属性はどんなものがあるのか、どんな意味を持つのかを考えるようにするとスムーズにHTMLが書けるようになるよ。

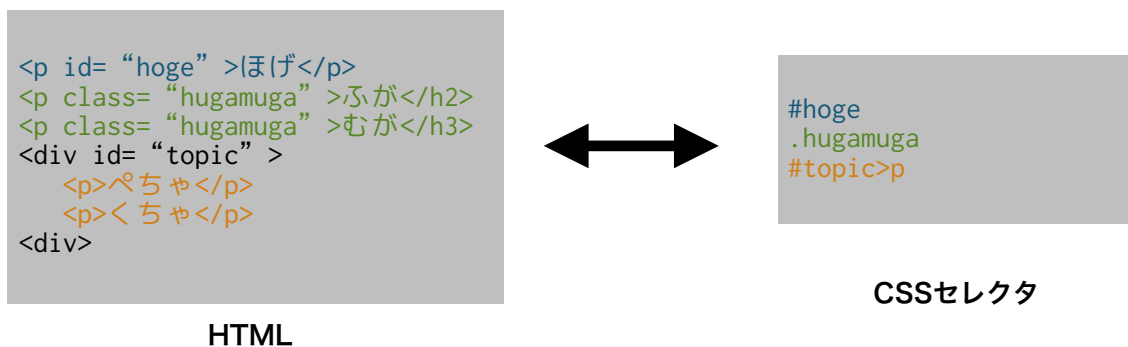
```
<タグ名 属性1=“値” 属性2=“値” …>innerHTML</タグ名>
```

HTMLとCSS

HTMLは文書に意味を付加するための言語で、意味を与えたところで大して何かが変わるわけではないよ。だけどブラウザの機能だったり別な言語の機能を併用することで機能を発揮してくるよ。例えば文書の見目を変えるためにはCSSという言語を使うよ。CSSのお仕事の基本は見た目を変えたい**タグを特定する**、そして**見た目を変える**。の2つだよ。



見た目を変えたいタグを特定するのはタグ名だけでは不十分。特定のタグだけや特定のタグの集まりだけを特定したい場合にはHTMLに**id**・**class**属性(idは複数のタグに同じ名前をつけてはならない。)を付加する。idを基準に特定する場合には「#」をclassを基準に特定する場合には「.」を文頭につける。innerHTMLを特定する場合には「>」記号を使う。このHTMLタグを特定する記述のことをCSSセレクタと呼ぶ。



HTMLとJavascript

JavascriptはHTML文書を操作するために作られたスクリプト言語だよ。けどそのまま使うとOS(Mac, Windows, Linux)やブラウザ(Safari, IE, Firefox, Chrome)ごとに書き方が違ったり動きが違ったりするし、コンパイルエラー(なので実行時の動きが見えるFirebugは絶対使おう!)がなかったりするからなかなか困った言語だよ。それらの問題を解消するためのすごいライブラリが開発されたよ！それが**jQuery**だよ。jQueryの使い方をちょっとだけ書くな。

Javascriptの基礎的な文法は他の言語と変わらないので、HTMLを操作する部分の説明だけ。HTMLのタグを扱うには**DOM(Document Object Model)オブジェクト**(ポインタ)を取得する必要があるよ。DOMを取得するにはjQueryオブジェクト「\$」という変数にCSSセレクタを渡してあげることによって取得できるよ。

```
$( "h1" );
$( "#hoge" );
$( ".hoge fuga" );
$( "div>p" );
```

DOMを取得したら色々な操作ができるよ。例えばh1タグのinnerHTMLを操作したい場合は以下のように書くよ。

```
$( "h1" ).html( "好きな文章" );
```

イベントを設置したい場合は、onという関数を呼び出してイベント種類の文字列とイベントの内容(関数)を引数に渡してあげると書けるよ。

```
$( "h1" ).on( "click", function(){
    $(this).html( "ほげ" );
});
```

PHP

ここまではクライアント側(ブラウザ)で動く言語を説明したから、これからはサーバー側で動くPHPをちょっとだけ説明。PHPはサーバー側に置かれてブラウザからアクセスされた瞬間に実行される(図1参照)。PHPのソースコードはHTMLにphpタグを埋め込むことでできる。

```
<!DOCTYPE html>
<head>
...
</head>
<body>
<p>
<?php
    print "Hello World!";
?>
</p>
</body>
</html>
```

サーバー側: 実行前



```
<!DOCTYPE html>
<head>
...
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

クライアント側: 実行後

どういうところに使うの？PHP!

HTMLはマークアップ言語なのでプログラミング言語ではない。PHPはプログラミング言語なので面倒なところ(条件分岐、ループ)やサーバー側でしかできないこと(データベースの操作)をしよう。例えば1~100これが100並んだHTML書きたいですか？こういう場合はPHPを使いましょうー。

```
<!DOCTYPE html>
<head>
...
</head>
<body>
<ul>
<?php
    for($i=1;$i<=100;$i++){
        print "<li>$i</li>" ;
    }
?>
</ul>
</body>
</html>
```

サーバー側: 実行前



```
<!DOCTYPE html>
<head>
...
</head>
<body>
<ul>
    <li>1</li>
    <li>2</li>
    <li>3</li>
    ...
    <li>100</li>
</ul>
</body>
</html>
```

クライアント側: 実行後

HTMLとPHP

実はここまでの話だと図1で説明したリクエストのうちGETメソッドしか使用していない。しかしHTMLからデータを送信しサーバー側で取得させたい場合がある。そのときはPOSTメソッドを利用する。HTML上からHTTPメソッドを利用するには**Form**という仕組みを利用する。

```
<!DOCTYPE html>
<head>
...
</head>
<body>
<form action= "hoge.php"
    method= "post" >
    <input type= "text"
name= "age" >
    <input type= "text"
name= "address" >
    <input type= "submit
value= "ボタン" >
</form>
</body>
</html>
```

クライアント側: FORM



HTTPリクエスト POST

```
<?php
if(isset($_POST[ "age" ] ) &&
    isset($_POST[ "address" ])){
    $age = $_POST[ "age" ];
    $address =
        $_POST[ "address" ];
    print $age;
    print $address;
}
?>
```

サーバー側: 値の受け取り

FormはHTTPリクエストを行いたい部分をformタグで囲み、送信したいデータをinputタグで用意する。データはinnerHTMLではなくvalueという属性でやりとりされる。inputタグはちょっと特殊なタグで閉じタグがない。またtypeがいくつか指定でき、テキストボックス・ラジオボタン・セレクトボックス・ボタンなどがある。特に**ボタンはtype="submit"と表記し、データを送信するときに使われる**。PHP側でPOSTされたデータを取得するには\$_POSTという変数を使う。この変数の型はarray(配列)型になっている。PHPの配列は**key**と**value**の組つまりハッシュである。データをPOSTするときはHTMLのinputタグの**name属性**が必須になる。これは、name=>valueでハッシュの形を作っているからである。上の例では以下のようなハッシュになる。

```
array(
    "age"      => "hoge" ,
    "address" => "fuga" ,
);
```

しかしPHP側で想定した通りにこのハッシュが送信されてくるとは限らない(inputの書き忘れ、valueが空など)。なので不測の事態に備えてハッシュにkeyがあるかどうかを判断する必要がある。判断する前にvalueを取得しようとするとエラーになってしまうからだ。そのための関数が**isset**関数である。

```
isset($_POST[ "age" ])
// ageというkeyがあればtrue, そうでなければfalseになる。
```

以上がちょっとしたWebアプリの作り方の基本! 中途半端な終わり方だけど勘弁ね!