JavaScript (1)

二宮 崇
東京大学 情報基盤センター 図書館電子化研究部門 講師

MBC 2008年10月1日

JavaScript

- JavaScript
  - プログラミング言語としてみて結構面白い
    - LISPっぽい? 無名関数、リテラル関数
    - プロパティやメソッドを文字列で得たり、連想配列でアクセスしたりと、かなりメタなプログラムが書ける
    - オブジェクト指向っぽい
    - Perlっぽい?型制約がほとんどない
    - ブラウザと連動したイベント処理
  - Ajaxで快適なリッチクライアント

JavaScript

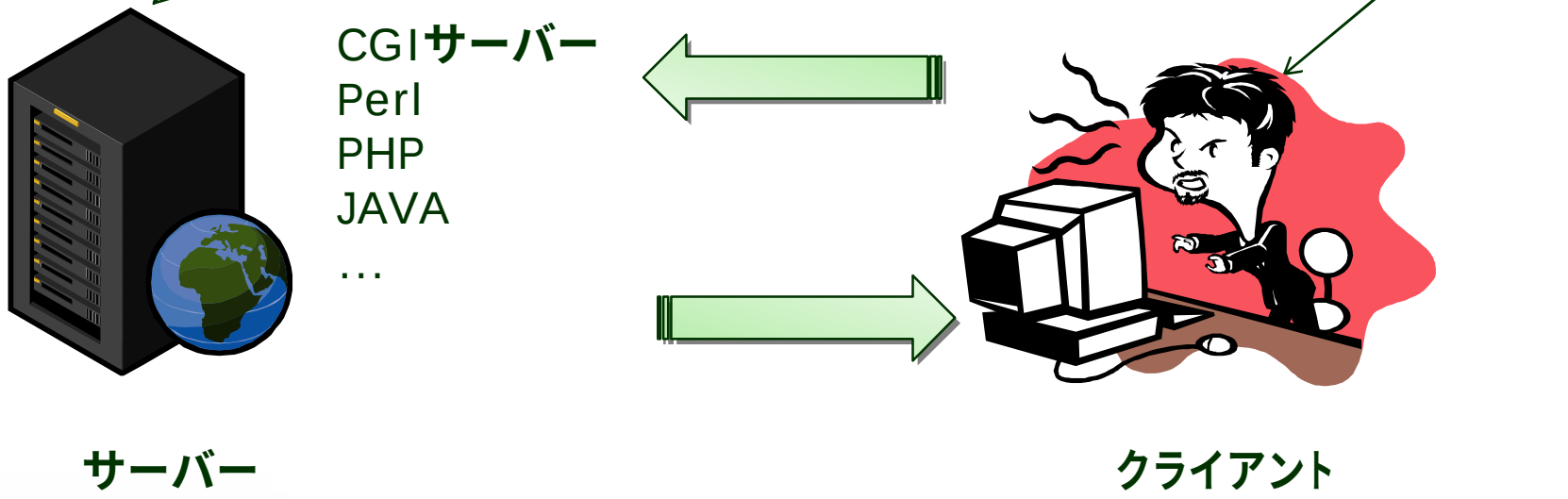
- (講師なのに) 最近この講習会のために勉強始めました。
  - まだわからないことがいろいろとあります
  - 一緒に勉強するぐらいのつもりでよろしくお願いします

JavaScript

- できること
  - GmailやGoogle Mapみたいなリッチクライアントなサービス
- できないこと?
  - ローカルファイルを開いて何か処理することができない?
- 難しいこと
  - Ajaxを使ってサーバーと非同期通信

サーバーとクライアント

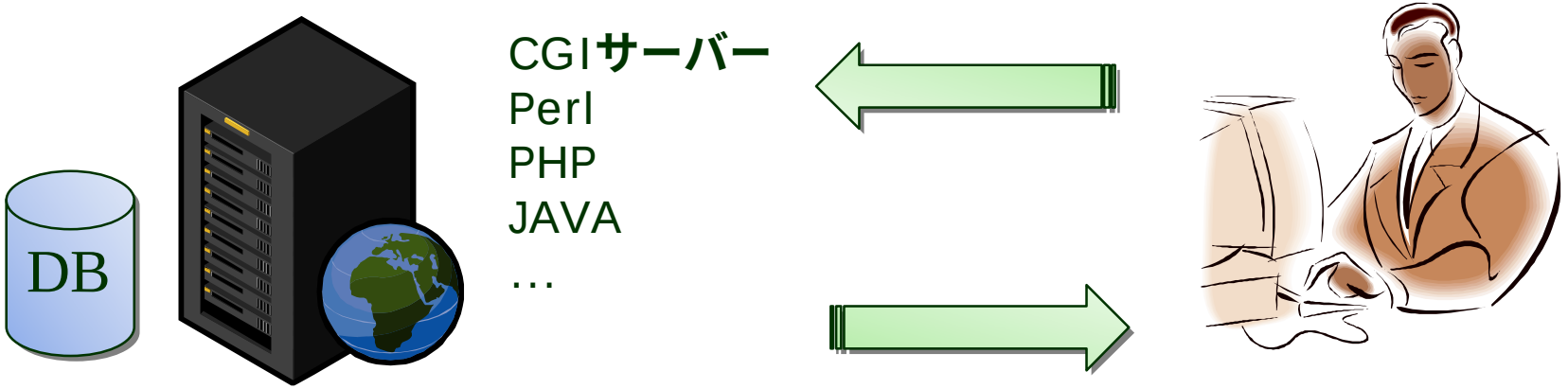
- 昔



- ほとんどの処理をサーバーで行っていた
- クライアントはデータを所持しない
- 些細なページの書き換えやデータの簡単なチェックのためには必ずサーバーへのアクセスが必要だった

サーバーとクライアント

- 今 (リッチクライアント)



- サーバーとクライアントで仕事をわけあう
  - サーバーはデータ管理
  - クライアントはユーザーインターフェース
  - クライアントにデータを一括ダウンロード
- クライアントはデータを保持 (クッキー)
- ページの書き換えやデータの簡単なチェックをクライアントで

僕もJavaScript使ってみました

- 良かったこと
  - JavaScriptはプログラミング言語としてもかなり面白いししっかりしている
  - クライアント側でデータの読み書き以外はほとんどのことができる
  - HTMLと連携してかなり面白いものが作れる
- よくわからなかったこと / 悪かったこと
  - データの読み書きができない…。
  - サーバー側も作らないといけない…。
  - 必然的にクライアント側だけだとおもちゃのようなものしか作れない
  - IEやFirefox等で互換性がない
    - 互換性をとるのはかなり大変とみた
  - オブジェクトへのアクセスの仕方 (nameとidの違いなど)

JavaScriptマスターへの道

1. まず、クライアント側のプログラムをいろいろと書けるようになろう!
2. サーバー側も用意してデータをやりとりしながら何か面白いものを作ろう
3. Ajax

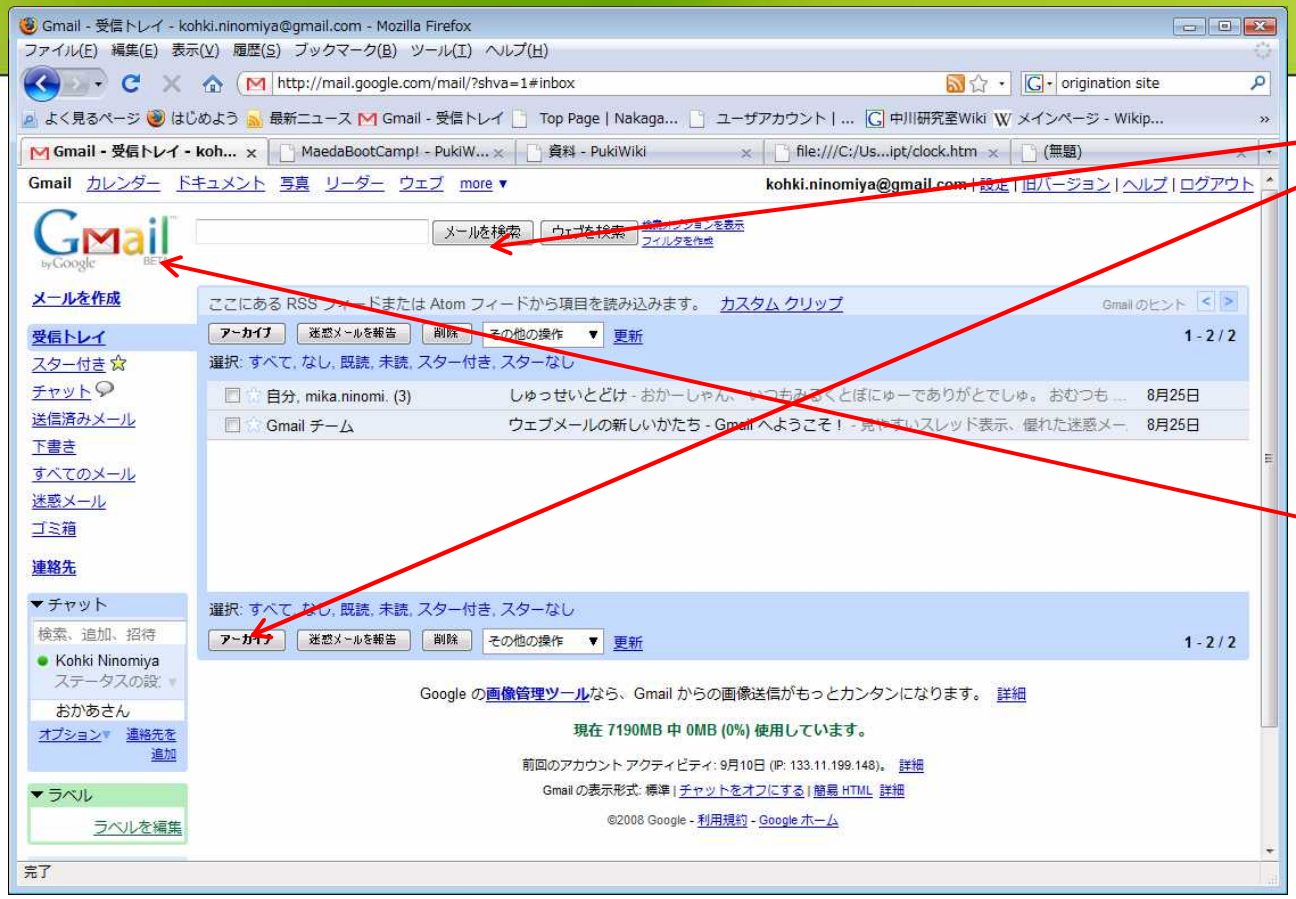
オブジェクト

- オブジェクト指向とは？
  - 巨大なプログラムを作るときのプログラムのモジュール化に関する考え方
  - 各オブジェクト（モジュール）は他のオブジェクトのことはあまりよく理解していないが、どういうメッセージを送ればどんな仕事をやってくれて、どういうメッセージが返ってくるかを知っている

ウィンドウとGUIとオブジェクト

- 昔は難しかった
  - ウィンドウ一つだすのも大変だった
  - マシンやOSごとにまったく異なるライブラリをつかっていた
  - ウィンドウの中にいろいろなGUIを用意するのはまた大変だった

ブラウザのオブジェクト

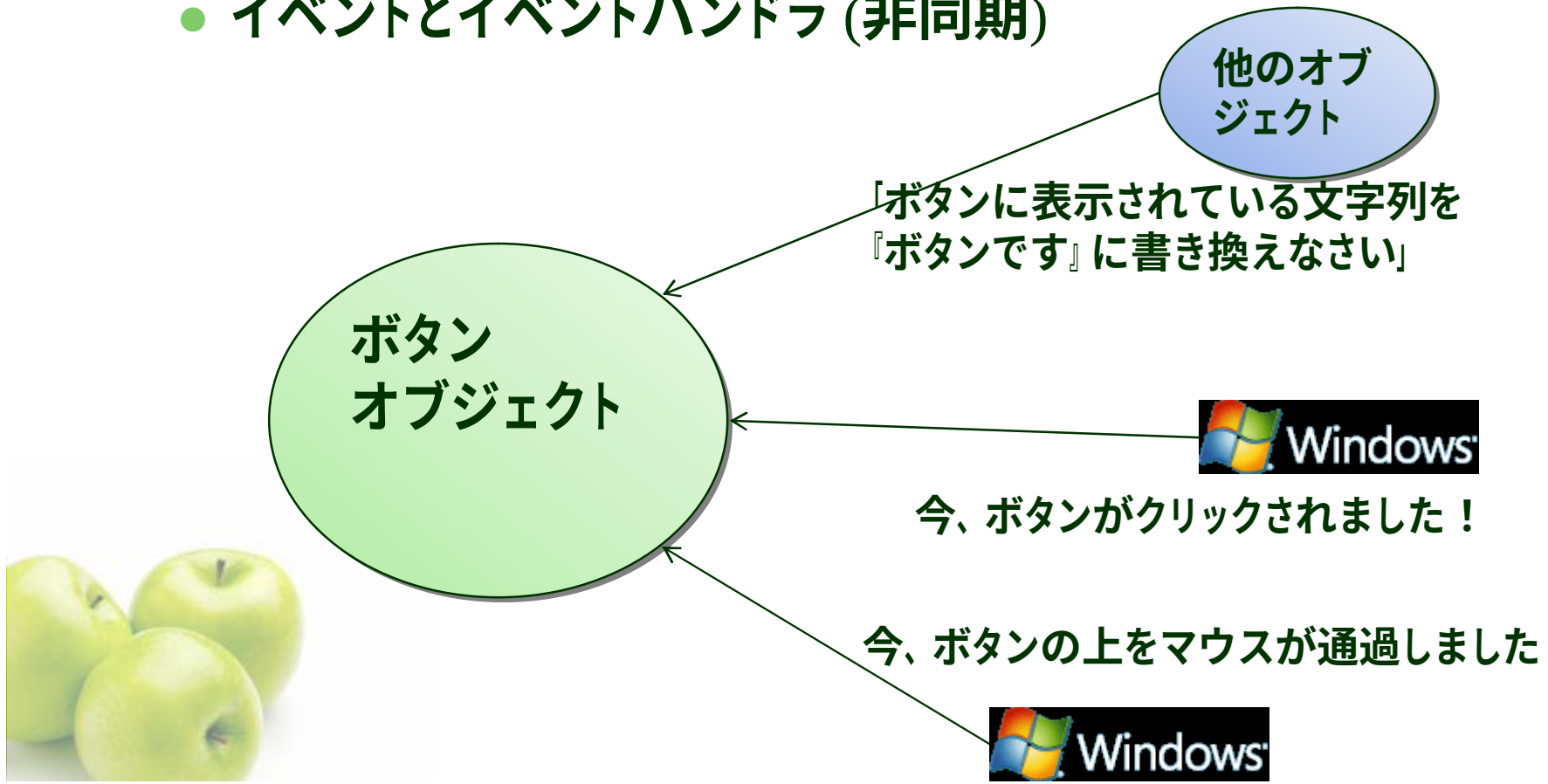


こういうフォームやボタンもオブジェクト

画像やリンクもオブジェクト

オブジェクトへのメッセージ

- 二種類のメッセージ
  - メソッドとプロパティ (同期)
  - イベントとイベントハンドラ (非同期)



JavaScript

- 課題
  - 九九の表を表示できるようになろう
  - 日付表示
  - カレンダを作ろう
  - 時計をつくろう
  - １５パズルをつくろう

はじめに：
次の雛形を作っておこう

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
</head>

<body>
<script type="text/javascript">
</script>
</body>

</html>
```

JavaScriptの変数

- JavaScriptでは変数は全てvarで宣言する
  - (varをつけなくても動くけど、とにかくつけた方が良い)
  - 例
    - var a = 1.0;
    - var b = “文字列です。”;
- 繰り返し文
  - for(初期化 ; 繰り返し条件 ; 繰り返しの時の処理) { … }

ドキュメントへの表示

- documentに対するメソッドwriteを使おう
  - document.write(..); で..の部分が表示される。
- <table>..</table>でテーブルをつくる
  - <tr>..</tr>で行をつくり、<td>..</td>で表の要素がつくれる

九九の表

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
</head>

<body>
<script type="text/javascript">
document.write("<center>");
document.write("<h1>九九の表</h1>");
document.write("<table border>");
for(var i = 1 ; i < 10 ; i++) {
        document.write("<tr>");
        for(var j = 1 ; j < 10 ; j++) {
                document.write("<td>", i*j, "</td>");
        }
        document.write("</tr>");
}
document.write("</table>");
document.write("</center>");
</script>
</body>
</html>
```

デバッグ

- Firefox
  - ツール→エラーコンソール
  - テキストボックスに表示
  - DOM インスペクタ
  - Firebug

カレンダー作成への道

- 関数の定義の仕方
- 文字列
- Dateオブジェクト
- 配列