

# ネットワークアプリケーション

## 第9回 JavaScriptによる クライアントサイドウェブプログラミング

石井 健太郎

(423研究室・オフィスアワー火3限)

# スケジュール

- 9月27日 第1回「TCP/IPプロトコルスイート」
  - 10月4日 第2回「Javaによるウィンドウプログラミング」
  - 10月11日 第3回「ネットワークアプリケーションのプログラミングモデル」
  - 10月18日 第4回「Javaによるネットワークプログラミング」
  - 10月25日 第5回「Javaによるネットワークプログラミング」
  - 11月8日 第6回「Javaによるネットワークプログラミング」
  - 11月15日 第7回「Javaによるネットワークプログラミング」
  - 11月17日** 第8回「ウェブプログラミングについて」
- 最終課題(1)**

# スケジュール

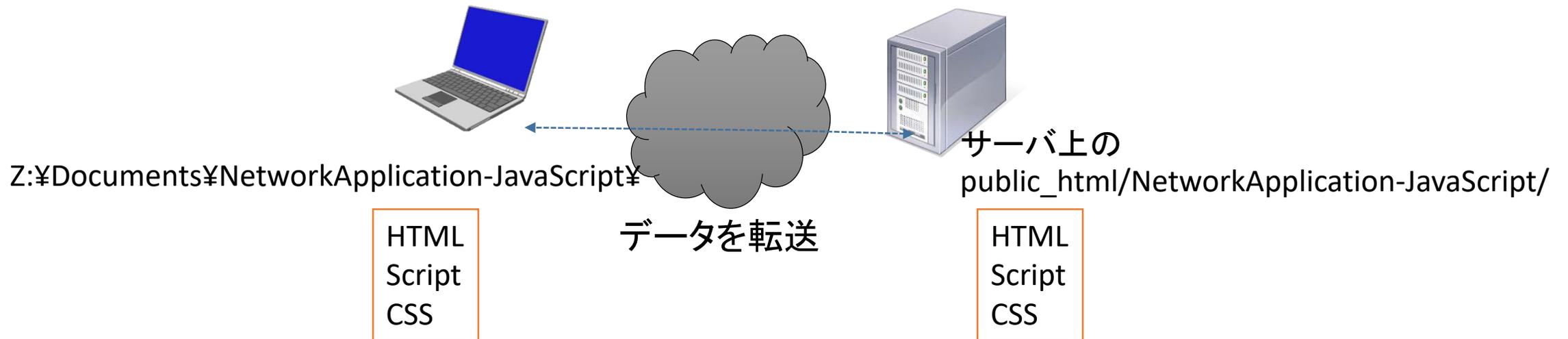
- 11月22日 第9回「JavaScriptによるクライアントサイドウェブプログラミング」
- 11月29日 第10回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月6日 第11回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月13日 第12回「JavaScriptによるクライアントサイドウェブプログラミング」
- 12月20日 第13回「JavaScriptによるクライアントサイドウェブプログラミング」
- 1月10日 第14回「JavaScriptによるクライアントサイド...」 **最終課題(2)**
- 1月17日 第15回「まとめと演習」

# ウェブサイトの変更

- 大学のサーバでは認証をかけられないことがわかったので、  
本講義専用サーバを用意しました
- 本講義専用サーバ
  - IPアドレス: **150.89.250.195**
- ユーザ名・ログインパスワードの初期設定を講義で伝えますので、  
ログインしてパスワードを変更してください
  - 方法は講義中に

# ウェブページの公開

- TeraTerm or FileZilla で public\_html 以下のディレクトリを作成する
  - ログインサーバ: **150.89.250.195**
- Zドライブのローカルフォルダの内容をサーバ上に作成したディレクトリに転送する



# ウェブページの公開

The screenshot shows the FileZilla interface with the following details:

- Window title: t2015039@o-vnc.center.oit.ac.jp - FileZilla
- Menu: ファイル(F) 編集(E) 表示(V) 転送(T) サーバ(S) ブックマーク(B) ヘルプ(H)
- Host (H): nc.center.oit.ac.jp | User (U): t2015039 | Password (W): | Port (P): | Quickconnect (Q)
- Status: 接続されました (Connected)
- Command: PWD
- Response: 257 "/home/teachers/t2015039" is your current location
- Local Site (ローカル サイト): ¥ (Local drive)
- Remote Site (リモート サイト): /home/teachers/t2015039

Two red boxes highlight the local and remote site panels. The local site panel is labeled **ローカル** (Local) and the remote site panel is labeled **サーバ** (Server).

ファイル名	ファイルサイズ	ファイルの種類	最終更新日時
C:		ローカル ディスク	
D:		CD-ROM/DVD	
X: (¥¥o-file01¥post)		ネットワーク ドラ...	
Y: (¥¥o-file01¥common)		ネットワーク ドラ...	
Z: (¥¥o-file01¥home¥teachers¥t2015039)		ネットワーク ドラ...	

ファイル名	ファイルサイズ	ファイルの種類	最終更新日時	パーミッション
..		ディレクトリ		
.adobe		ファイル フォ...	2015/09/11 ...	0700
.anthy		ファイル フォ...	2016/11/16 ...	0700
.cache		ファイル フォ...	2015/09/16 ...	0700
.compiz		ファイル フォ...	2016/11/16 ...	0700
.compi...		ファイル フォ...	2015/09/08 ...	0755
.config		ファイル フォ...	2016/11/16 ...	0700
.dbus		ファイル フォ...	2015/09/08 ...	0700

20 個のファイルと 35 個のディレクトリ。合計サイズ: 24,383,072 バイト

サーバ/ローカル ファイル	方向	リモート ファイル	サイズ	優先度	状態
---------------	----	-----------	-----	-----	----



# ウェブサイトの認証

- htdigest コマンドの使いかたは下記のとおり



```
150.89.250.195:22 - kenta@lecture: ~/public_html/NetworkApplication-J VT
ファイル(E) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
kenta@lecture:~/public_html/NetworkApplication-JavaScript$ htdigest -c .htpasswd e1n14XXX e1n14XXX
```

# JavaScript言語入門

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>こんにちは世界</title>
</head>
<body>
<script>
var x;
x ="Hello World";
alert(x);
</script>
</body>
</html>
```

# JavaScript

- **C言語と異なるところ**

- **インタープリタ言語**
- **コンパイルをあらかじめ行わない**  
(実行時にプログラムの意味を解釈し、実行される)
- **変数の宣言時に変数の型(char, string, int, float, double等)を指定しない**
  - **変数の型は存在する**
  - **すべての変数の宣言はvarによって行う**  
(varを省略することもできるがおすすめしない)
- **型が自動的に変換される(楽だがミスがわかりにくい)**
- **イベントドリブン**  
(同時に複数のプログラムコードの箇所が並行して動く)

# データ型

- 文字列
- 数値
- 真偽値(true / false)
- オブジェクト
  - 配列
  - 関数
  - 組み込みオブジェクト
- undefined 定義されていない
- null なにもない

# 数値と演算子

- 数値

10

2.5

-2.5

- 演算子

+   -   \*   /   %

+=   -=   \*=   /=

++   --

【注意】+ は「文字列の結合の演算子」でもある。左右のどちらかが文字のとき、文字に変換される

```
var x;  
x = "1" + 2;  
alert(x);  
x = +"1" + 2;  
alert(x);
```

のように単項演算子として使うと文字列を数値に変換する関数として使うこともできる

# 文字列

- **表現方法と特殊文字**

- 「"」か「'」で囲む

x = "hello world";

x = 'hello world';

x = "it's a pen";

x = 'it¥'s a pen';

- **エスケープ文字は「¥」(円マーク)**

- **特殊文字の改行やタブは「¥n」「¥t」**

# 制御構造

- if文 C言語と同じ
- for文 C言語と同じ(変数の宣言はvarに変わる)

```
for (var i=0; i<20; i++) {  
  console.log(i);  
}
```

- while文、do while文           C言語と同じ
- break文、continue文           C言語と同じ
- switch文
- 条件分岐の中にある比較演算子は二種類ある
  - === !==   型も含めて一致するか比較
  - == !=     異なる型の場合には変換してから比較

# 関数

- C言語と異なり、返り値の型の宣言は不要
- その代わりに、**function**というキーワードを書く
- 引数の型宣言も不要

```
var x = 100;
var y = 200;

function max(x, y) {
  if (x>=y) return x; else return y;
}

alert(max(x,y));
```

標準的な書き方

```
var x = 100;
var y = 200;

var max = function (x, y) {
  if (x>=y) return x; else return y;
}

alert(max(x,y));
```

このような書き方もできる

# 配列

## • 二つの表現方法

```
var student = new Array(3);  
student[0] = "櫻井";  
student[1] = "松本";  
student[2] = "大野";
```

```
var student = ["櫻井","松本","大野"];
```

## • 使い方

```
for (var i=0; i<3; i++) {  
  alert(student[i]);  
}
```

```
for (var i in student) {  
  alert(student[i]);  
}
```

**配列の各要素を処理するには"in"が便利**

# オブジェクト(連想配列)

## • キーと値の組

```
var hobby = new Object();  
hobby['櫻井'] = "ありません";  
hobby['松本'] = "買い物";  
hobby['大野'] = "釣り";  
//hobby.櫻井 = "ありません"; //でもOK
```

```
var hobby = {  
  櫻井:"ありません",  
  松本:"買い物",  
  大野:"釣り"  
}
```

## • 使い方

```
for (key in hobby) {  
  alert(key + "の趣味は" + hobby[key]);  
  //alert(key + "の趣味は" + hobby.key); //ではNG! keyというメンバー変数へのアクセス  
}
```

# オブジェクト(連想配列)

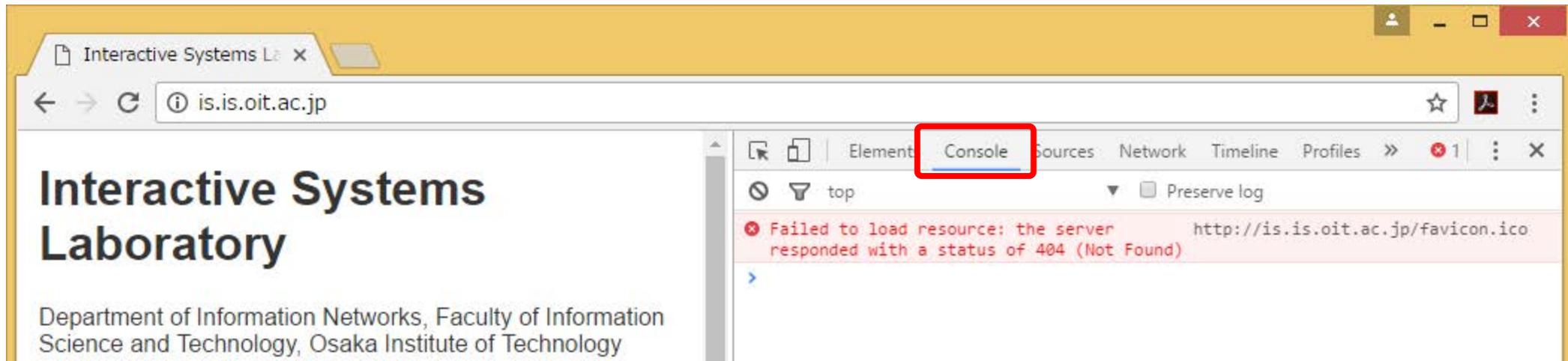
## • 「連想配列」の「配列」の例

```
var student = new Array(3);
student[0] = {
  名前:"櫻井",
  趣味:"なし",
  特技:"ラップ"
}
student[1] = {
  名前:"松本",
  趣味:"買い物",
  特技:"演技"
}
student[2] = {
  名前:"大野",
  趣味:"釣り",
  特技:"ダンス"
}
```

```
for (i in student) {
  alert(student[i]["名前"]
  +"の趣味は"
  + student[i]["趣味"]);
}
```

# JavaScriptでデータを出力する方法3種類

- `console.log ()`                      コンソールに書き出す
  - `alert ()`                                アラートウインドウ(ポップアップ)に書き出す
  - `document.write ()`                    HTML の body に書き出す
- 
- **それぞれ JavaScript のコードを書いて動作を確かめてみよう**



# 演習

- 来週から本格的にJavaScriptを開発するにあたり、ウェブページの公開が必須となる。そこで、まず、ウェブページを作成して公開する演習を行う。
- また、ページを公開するということは、だれでも他人の課題の演習結果を見ることができてしまうので、認証をかけることを実践する。
  - 基本課題
    - ページのどこかに、学生番号と氏名を明記する
    - 自分の興味のあることからについて、5行以上の内容を記述する。(自己紹介でもいいです)
    - 講義中に示したサイトの認証を実現する
      - ユーザ名は自分のアカウント名(e1n14XXXなど)とする
      - パスワードはシステムのログインパスワードとは異なるものかつ他人から推測されないものを自由に決めてよい
    - JavaScript の document.write () を使って、HTMLファイルに何か書き足してみよう
  - 発展課題
    - 画像の挿入をはじめ、HTMLのできることを思い出して(または調べて)、魅力的なウェブページを作成する
- ウェブページを公開するとともに、そのURLと認証情報(ユーザ名&パスワード)をテキストファイルに記載して、次回の講義開始時まで、提出フォルダ(Xドライブ)にファイルで提出
  - X:\N科専門¥石井講師¥ネットワークアプリケーション¥第8回
  - ファイル名は「<学生番号>.txt」とする(ハイフンなし) 例: N14999.txt

# 【練習問題1】第1版（提出課題ではない）

- サンプルプログラム:

<http://lss.oit.ac.jp/~t2015039/NetworkApplication-JavaScript/uranai1.html>

を自分のローカル環境で動くようにする

ブラウザで右クリックするとソースが見れるので、これをコピー & ペーストする(ダウンロードでも可)

- 占いの文面を5件以上作成し、配列に格納し、乱数でその内の一つの文を選んで出力するようにして修正する
- 占いの文面に合う画像を表示するようにする

# 【練習問題1】第2版(発展版1)

以下のように、占いの候補リスト(u)を以下のような連想配列の配列で表現し、占い文、画像、ラッキーアイテムを記述するように変更し、これらを表示するようにプログラムを修正する

```
var u = [  
  {text:"素晴らしいです。", img:"peach.jpg", lucky:"もも"},  
  {text:"すごいです。", img:"orange.jpg", lucky:"みかん"},  
  {text:"ひどすぎます。", img:"apple.jpg", lucky:"りんご"},  
];
```

- ヒント:例えば、配列の最初の要素のラッキーアイテムは以下のように取得できる

```
u[0]['lucky'] // これで"もも"が取得できる
```

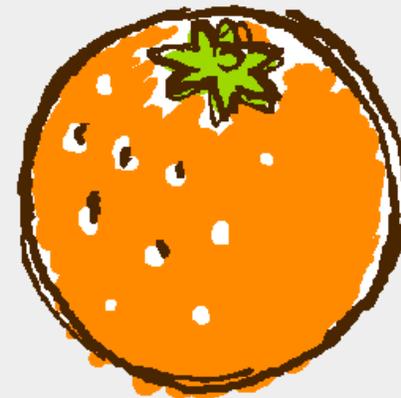
誕生日はいつですか？

10 月 3 日

運勢

あなたの今日の運勢は、

すごいです。本日のラッキーアイテムはみかんです。



# 【練習問題1】第3版(発展版2)

1. 毎回実行する度に、占い文が変化しないように、誕生日に応じて一意に決まる占い文が選択されるようにして下さい
  - ヒント: 誕生日を `parseInt()` で数値に変換し、月と日を加える等により、何らかの数値に変換し、そして、配列の長さで割った余りの値で配列の要素を選ぶ
2. 誕生日に応じて一意に決まる占い文が毎日変化するようにして下さい
  - ヒント: 本日の月(1~12)と日付(1~31)を取得する方法を調べ、その値と誕生日の日付を加えて適当な数値を作る(方法は様々)