

# ネットワークアプリケーション

## 第8回 ウェブプログラミングについて

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)**

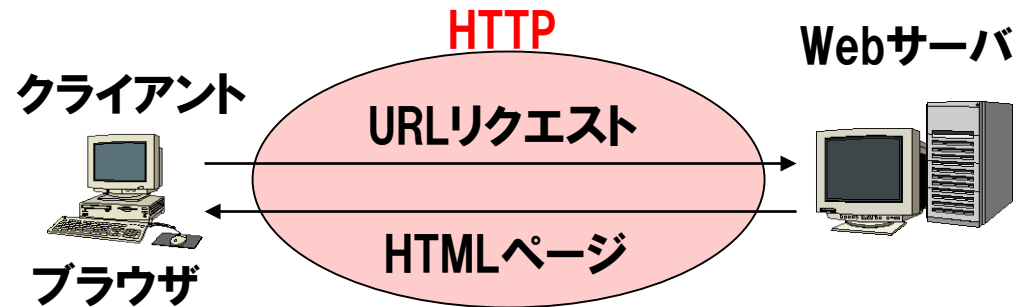
# スケジュール

- 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」

- **演習課題を提出してください**

# サーバサイド・クライアントサイド

- 静的なウェブページ

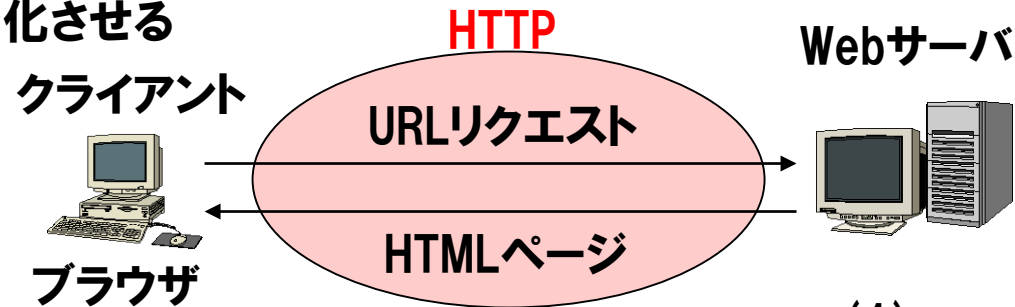


# サーバサイド・クライアントサイド

- 動的なウェブページを実現するには大きく2つの方法がある

(2)

クライアント側 (ブラウザ) で  
ページを動的に変化させる



(1)

サーバ側で  
HTTPのデータ (HTML) を  
動的に生成する

# サーバサイドウェブプログラミング

- CGI (Common Gateway Interface)
  - ウェブサーバ上でプログラムを動作させる仕組み
  - URLでプログラムを指定 → プログラムが実行されてHTTPデータを生成
- CGIはリクエストごとにプログラムを起動 → 動作に時間がかかる
- mod\_perl, FastCGI, PHP
  - あらかじめウェブサーバの一部としてプログラムを起動しておく  
→ 高速化を実現

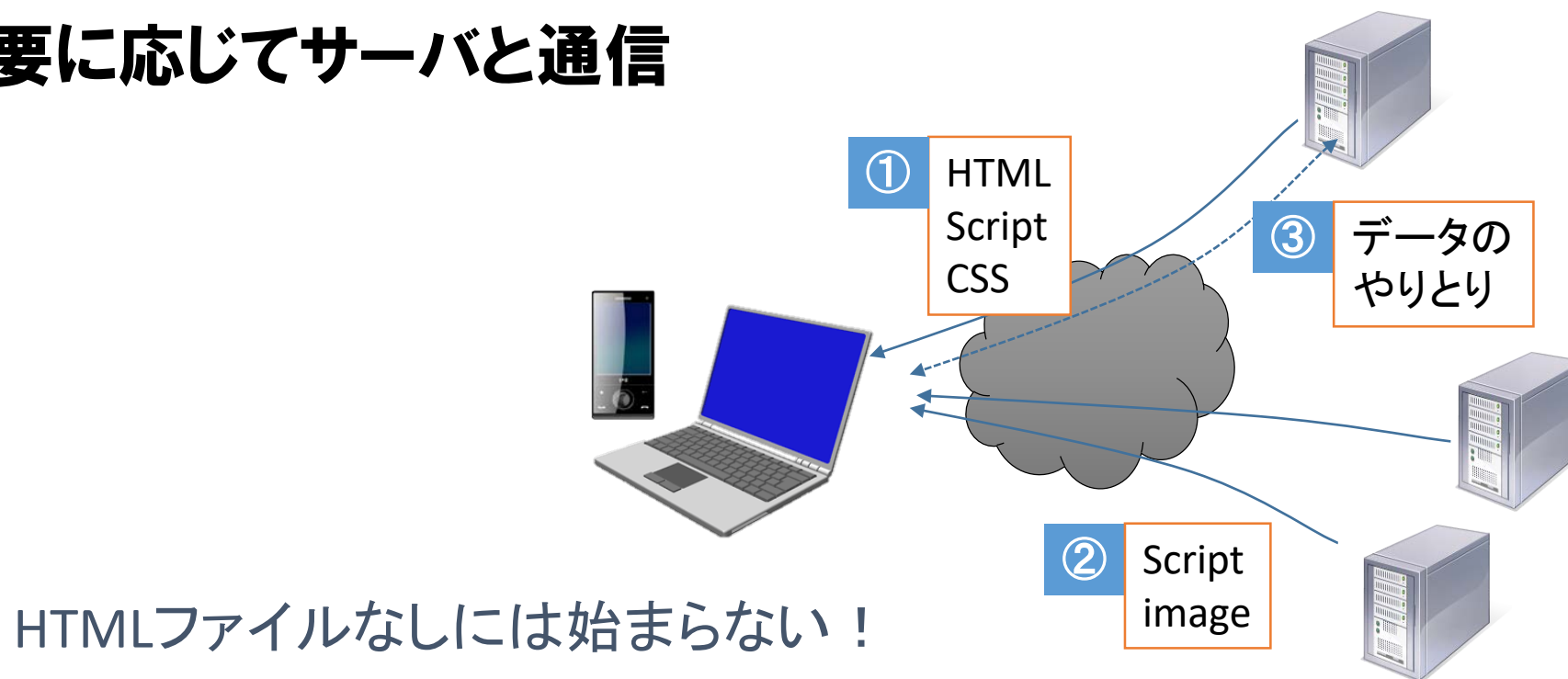
# JavaScript, ActionScript, Java Web Start

- **ブラウザで実行されるプログラム**
  - **ブラウザさえ対応していれば**
    - 様々な端末(PC、スマホ)に対応できる
    - 端末にインストールしなくても実行できる
- **現在は JavaScript の1人勝ち状態**
  - **JavaScript の欠点を補うために新しい言語も開発されている**
    - TypeScript
    - CoffeeScript
- **この講義では JavaScript をあつかう**



# JavaScriptの実行モデル

1. サーバからHTMLが配布されてブラウザ上で実行
2. 追加のスクリプトや画像を必要に応じロードされ実行
3. 必要に応じてサーバと通信



# JavaScriptの特徴

- **ウェブブラウザ上で実行するデファクトスタンダードの言語**
- **様々な端末(PCもスマートフォンも)に対応できる**
- **ブラウザの更新にともない処理系も更新される**
- **HTML5により端末のグラフィック機能が自在に使えるようになった**
- **ライブラリが急速に充実**
  - **ソースが見えるので、良いコードを開発者がマネしやすい(技術の伝搬が早い)**

# マニュアル

- Mozilla Developer Networkの資料がおすすめ



MDN > Web technology for developers > JavaScript

## JavaScript

JavaScript® (しばしばJSと略される)は軽量なオブジェクト指向言語です。Web ページで使用されるスクリプト言語として最もよく知られていますが、[ブラウザ以外の多くの環境においても使用されています。](#) (JavaScript についてさらに詳しく読む。)

[JavaScript「再」入門](#)  
世界で最も誤解されているであろうプログラミング言語「JavaScript」の本当の姿

- <https://developer.mozilla.org/ja/docs/Web/JavaScript>

# JavaScript開発環境

- Dreamweaver
- Microsoft Visual Web Studio
- Atom Editor
- jsdo.it

- 
- Eclipse JSDT
  - Aptana Studio
  - エディタ( Emacs・EmEditor・TeraPad・メモ帳等 )

# Dreamweaver

- **CSSのデザインのサポートが強力**
- **PhotoshopやIllustrator等の連携がしやすい**
- **JavaScriptの入力支援機能**
- **Webサーバへのコンテンツのアップロードが容易**
  - **1クリックでサーバと同期できる**
- **デメリット**
  - **機能が多すぎて少し難しい**
  - **高価(自宅では使えない)**

# Visual Web Studio

- Microsoftの開発環境

<http://www.microsoft.com/japan/msdn/vstudio/express/>



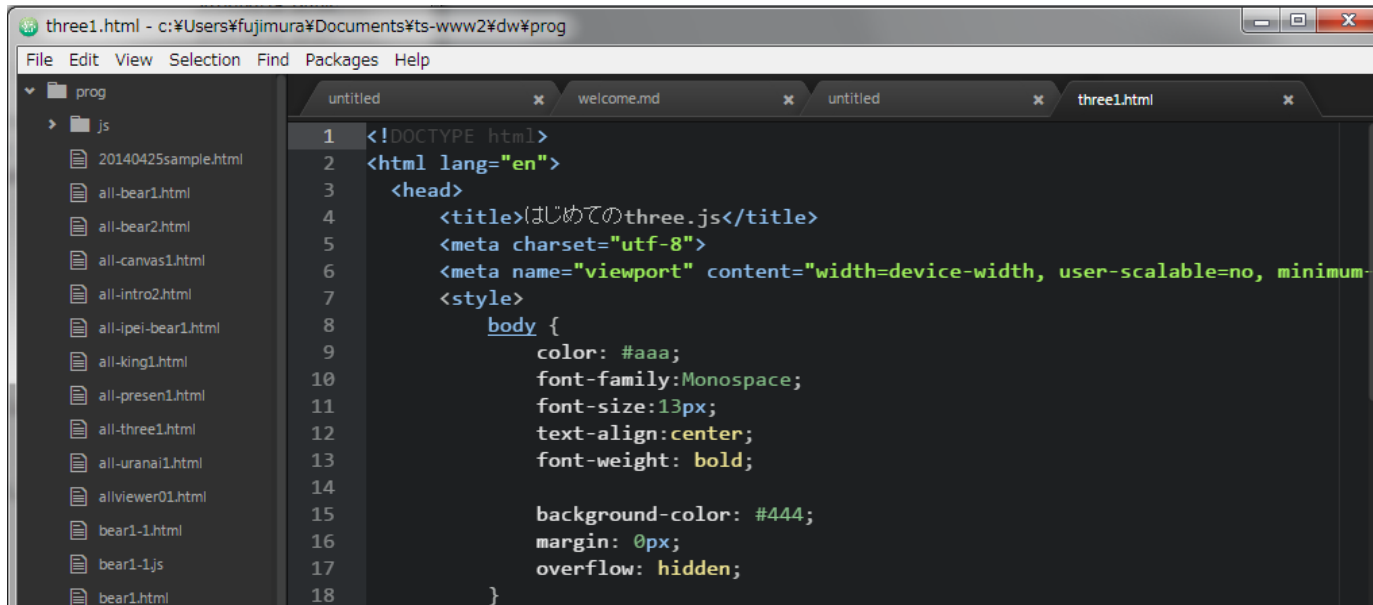
Microsoft  
Visual Web Developer 2010  
Express

Microsoft Web Platform Installer (Web PI) は、Visual Web Developer や IIS などの Microsoft Web Platform の最新コンポーネントのダウンロード、インストール、最新版への更新が簡単に可能になる無償ツールです。さらに、人気のあるオープンソースの ASP.NET と PHP Web アプリケーションをインストールします。

» [Web PI のダウンロード ページへ](#)

# Atom Editor

- 高機能で無料の注目のエディタ
- <https://atom.io/>
- 使いやすくカスタマイズできる

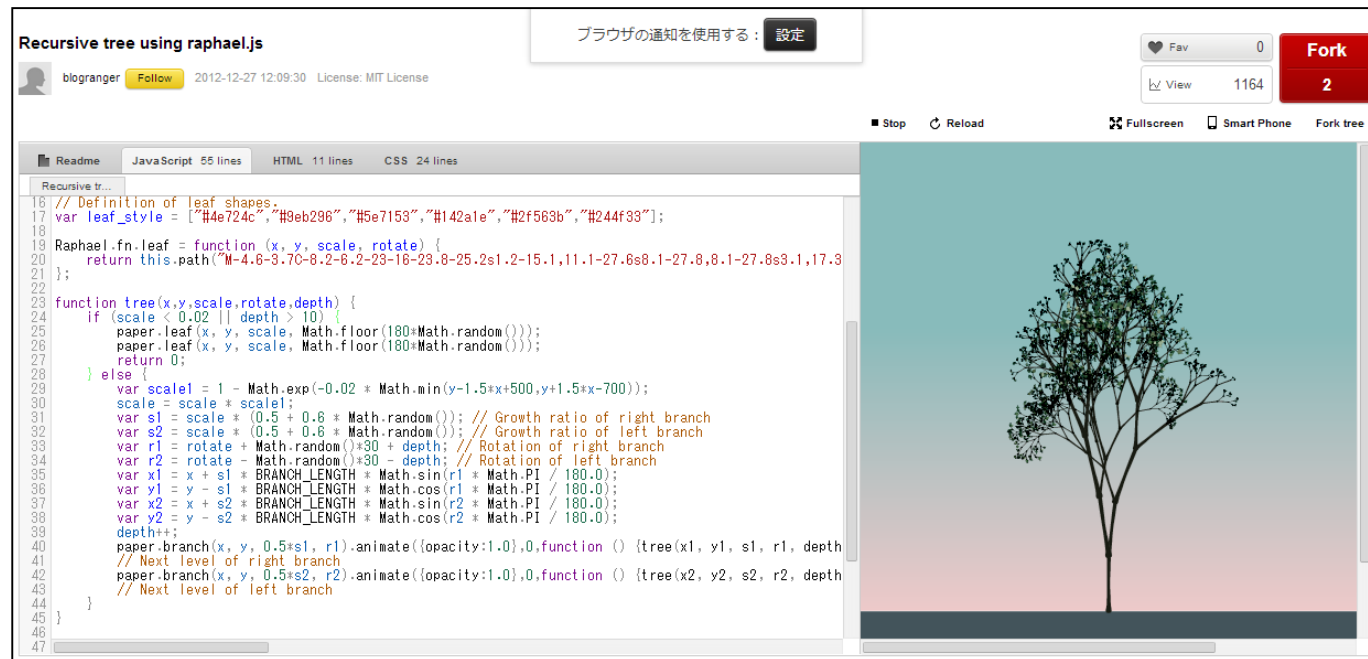


```
three1.html - c:\Users%fujimura\Documents\ts-www2\dw\prog
File Edit View Selection Find Packages Help
└─ prog
  └─ js
    └─ 20140425sample.html
    └─ all-bear1.html
    └─ all-bear2.html
    └─ all-canvas1.html
    └─ all-intro2.html
    └─ all-ipei-bear1.html
    └─ all-king1.html
    └─ all-presen1.html
    └─ all-three1.html
    └─ all-uranai1.html
    └─ allviewer01.html
    └─ bear1-1.html
    └─ bear1-1.js
    └─ bear1.html
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>はじめてのthree.js</title>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, user-scalable=no, minimum-
7     <style>
8       body {
9         color: #aaa;
10        font-family:Monospace;
11        font-size:13px;
12        text-align:center;
13        font-weight: bold;
14
15        background-color: #444;
16        margin: 0px;
17        overflow: hidden;
18      }
```

# jsdo.it

- JavaScriptで書かれたJavaScriptのための開発環境
- ブラウザでプログラムを書く

<http://jsdo.it>



The screenshot displays the jsdo.it online code editor interface. At the top, the title is "Recursive tree using raphael.js". The user profile shows "blogranger" with a "Follow" button, the date "2012-12-27 12:09:30", and the license "License: MIT License". There are buttons for "ブラウザの通知を使用する: 設定", "Fav 0", "Fork", "View 1164", and "2". Below the header, there are tabs for "Readme", "JavaScript 55 lines", "HTML 11 lines", and "CSS 24 lines". The main editor area shows JavaScript code for a recursive tree function. The code defines a function `tree(x,y,scale,rotate,depth)` that uses the `Raphael` library to draw a tree. The tree is rendered on the right side of the editor, showing a single tree with a brown trunk and green foliage against a light blue background.

```
16 // Definition of leaf shapes.
17 var leaf_style = ["#4e724c", "#9eb296", "#5e7153", "#142a1e", "#2f563b", "#244f33"];
18
19 Raphael.fn.leaf = function (x, y, scale, rotate) {
20   return this.path("M-4.6-3.7C-8.2-6.2-23-16-23.8-25.2s1.2-15.1,11.1-27.6s8.1-27.8,8.1-27.8s3.1,17.3
21 ");
22 };
23
24 function tree(x,y,scale,rotate,depth) {
25   if (scale < 0.02 || depth > 10) {
26     paper.leaf(x, y, scale, Math.floor(180*Math.random()));
27     paper.leaf(x, y, scale, Math.floor(180*Math.random()));
28     return 0;
29   } else {
30     var scale1 = 1 - Math.exp(-0.02 * Math.min(y-1.5*x+500,y+1.5*x-700));
31     scale = scale * scale1;
32     var s1 = scale * (0.5 + 0.6 * Math.random()); // Growth ratio of right branch
33     var s2 = scale * (0.5 + 0.6 * Math.random()); // Growth ratio of left branch
34     var r1 = rotate + Math.random()*30 + depth; // Rotation of right branch
35     var r2 = rotate - Math.random()*30 - depth; // Rotation of left branch
36     var x1 = x + s1 * BRANCH_LENGTH * Math.sin(r1 * Math.PI / 180.0);
37     var y1 = y - s1 * BRANCH_LENGTH * Math.cos(r1 * Math.PI / 180.0);
38     var x2 = x + s2 * BRANCH_LENGTH * Math.sin(r2 * Math.PI / 180.0);
39     var y2 = y - s2 * BRANCH_LENGTH * Math.cos(r2 * Math.PI / 180.0);
40     depth++;
41     paper.branch(x, y, 0.5*s1, r1).animate({opacity:1.0},0,function () {tree(x1, y1, s1, r1, depth
42 // Next level of right branch
43     paper.branch(x, y, 0.5*s2, r2).animate({opacity:1.0},0,function () {tree(x2, y2, s2, r2, depth
44 // Next level of left branch
45   }
46 }
47
```

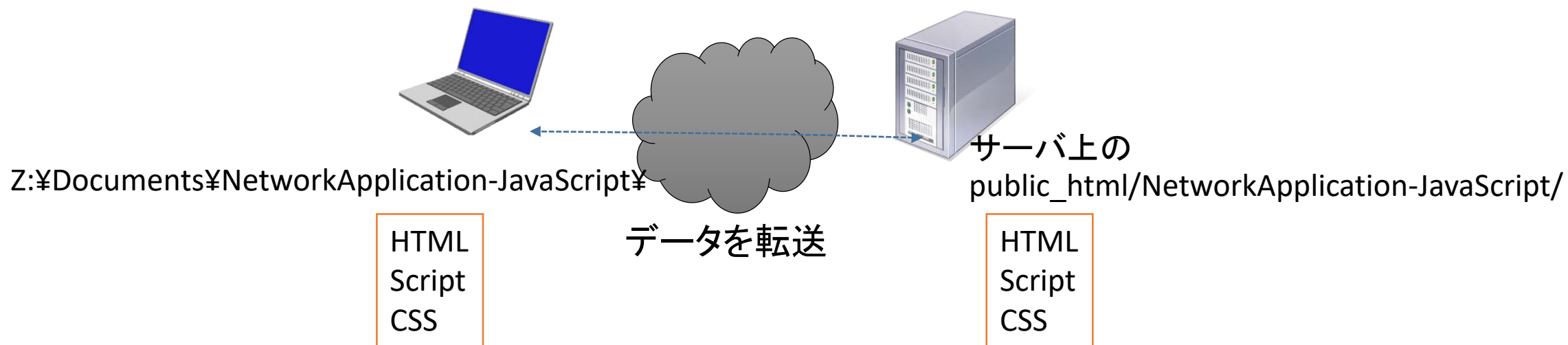


# 講義で用いるエディタ

- 授業ではDreamweaverを採用
  - 演習室・自習室で作業するときもDreamweaverを使うとよい
- 自宅でプログラミングを行う場合には,  
Atom Editor または jsdo.it を利用するとよい

# ローカルフォルダとサーバ上のディレクトリ

- Zドライブに本講義用のフォルダを作り、サーバ上の公開ディレクトリに転送する
  - ログインサーバ: o-vnc.center.oit.ac.jp

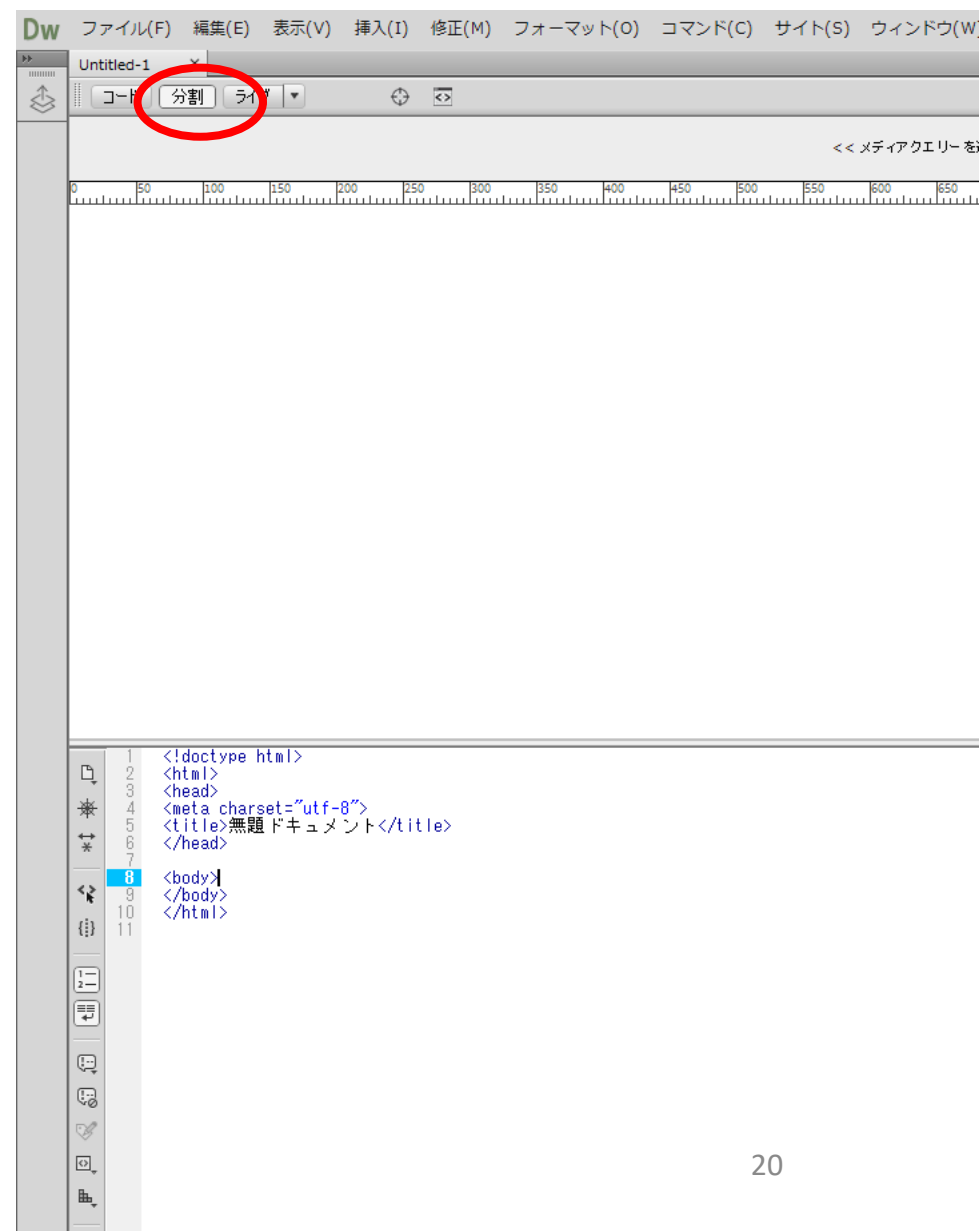


# Dreamweaverの起動

- **スタート> 下向き矢印( ↓ )> Adobe Dreamweaver CC 2015**
- **新規ドキュメントでHTML5を選ぶ**

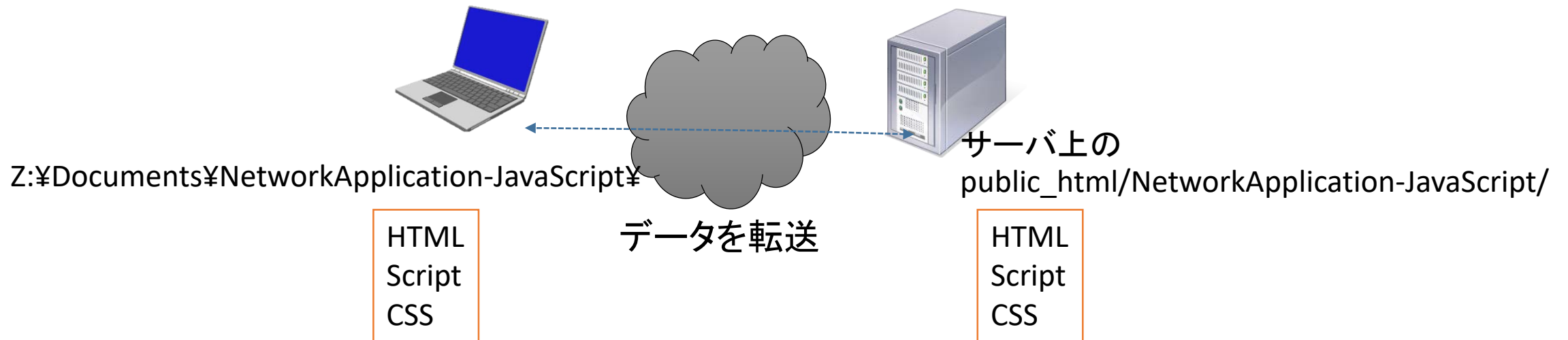
# Dreamweaverの起動

- コードとデザインを分割して表示した場合



# 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: o-vnc.center.oit.ac.jp
- 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)
- Toolbar: Standard FileZilla toolbar icons.
- Host (H): nc.center.oit.ac.jp | User (U): t2015039 | Password (W): [ ] | Port (P): [ ] | Quick Connect (Q): [v]
- 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

- .htaccess に認証に関する設定を書く
- .htpasswd に認証情報を書く  
(実際は htdigest コマンドを使って生成する)



```
150.89.250.195:22 - kenta@lecture: ~/public_html VT
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
AuthType Digest
AuthName "e1n14XXX"
AuthUserFile /home/e1n14XXX/public_html/NetworkApplication-JavaScript/.htpasswd
require valid-user
~
~
~
~
~
~
~
~
~
~
~
```

# 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、ページを公開することによって誰でも他人の演習結果を見ることができてしまうので、認証を実装することを実践する
  - 基本課題
    - ページのどこかに、学生番号と氏名を記載する
    - 自分の興味のあることについて、5行以内の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(e1n14XX)
      - パスワードはシステムのログインパスワードから推測されないものを自由に決めてよい
  - 発展課題
    - 画像の挿入をはじめ、HTMLでできる範囲で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:¥IN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