ネットワークアプリケーション

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

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

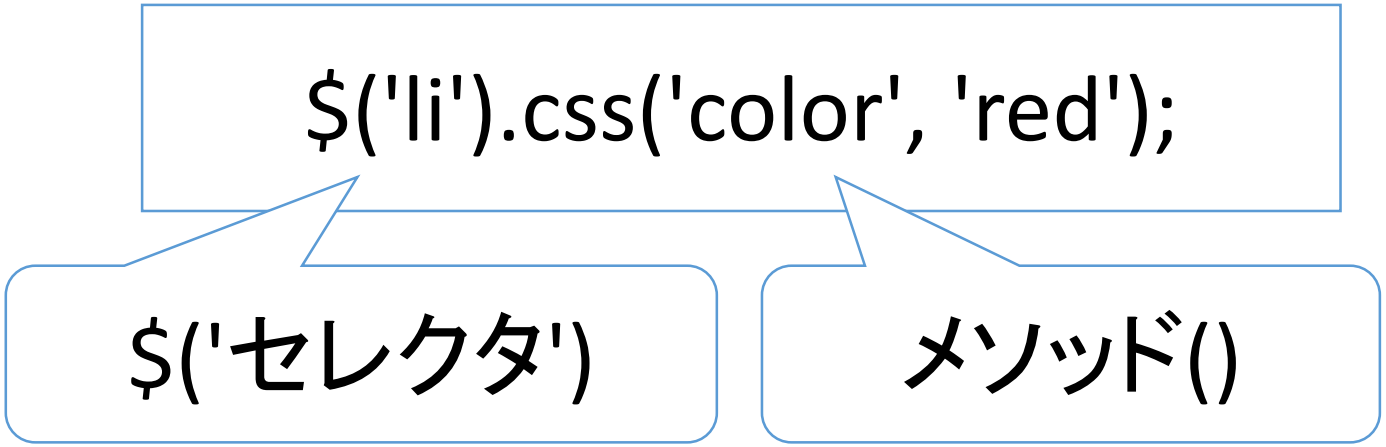
9月27日　第1回「TCP/IPプロトコルスイート」
10月4日　第2回「Javaによるウィンドウプログラミング」
10月11日　第3回「ネットワークアプリケーションのプログラミングモデル」
10月18日　第4回「Javaによるネットワークプログラミング」
10月25日　第5回「Javaによるネットワークプログラミング」
11月8日　第6回「Javaによるネットワークプログラミング」
11月15日　第7回「Javaによるネットワークプログラミング」　最終課題（1）
11月17日　第8回「ウェブプログラミングについて」

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...」	最終課題 (2)
1月10日	第14回「まとめと演習」	課題の演習
1月17日	第15回「まとめと演習」	課題の演習

jQueryの使いかた

- 以下の2ステップ
  - STEP1: 要素を選択して
  - STEP2: 操作を行う

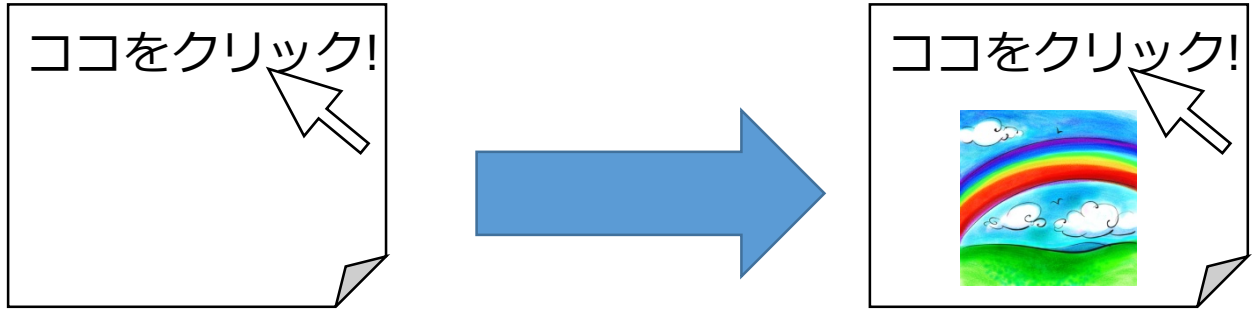


メソッド

- 多様な種類が存在
  - スタイル操作: CSS
  - 属性操作: Attributes
  - 要素間の移動・検索: Traversing
  - 要素の追加・削除: Manipulation
  - イベント: Events
  - エフェクト: Effects
  - Ajax(非同期通信): Ajax
- 全ての使い方を覚える必要はなく，リファレンスを見ながら使えればOK
  - http://api.jquery.com（本家）
  - http://semooh.jp/jquery/（日本語）

イベントを利用するプログラム

- ある文字列がクリックされた時に，
  - 画像を表示する
- ある画像の上にマウスカーソルをのせた時に，
  - 画像の説明を表示する
- ある文字列がダブルクリックされた時に，
  - 文字列を赤色にする

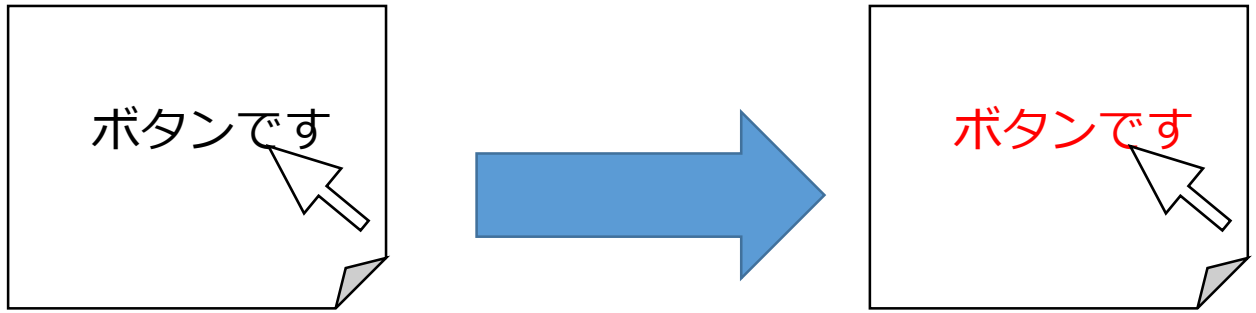


イベント: Events

```
<div id="button">ボタンです</div>


$('#button').click(function() {
        // ここにクリックされた時の動作を記述
        $(this).css('color', 'red');
});
```

イベントが設定された
要素自身を選択する場合
$(this)を用いる



イベント: Events

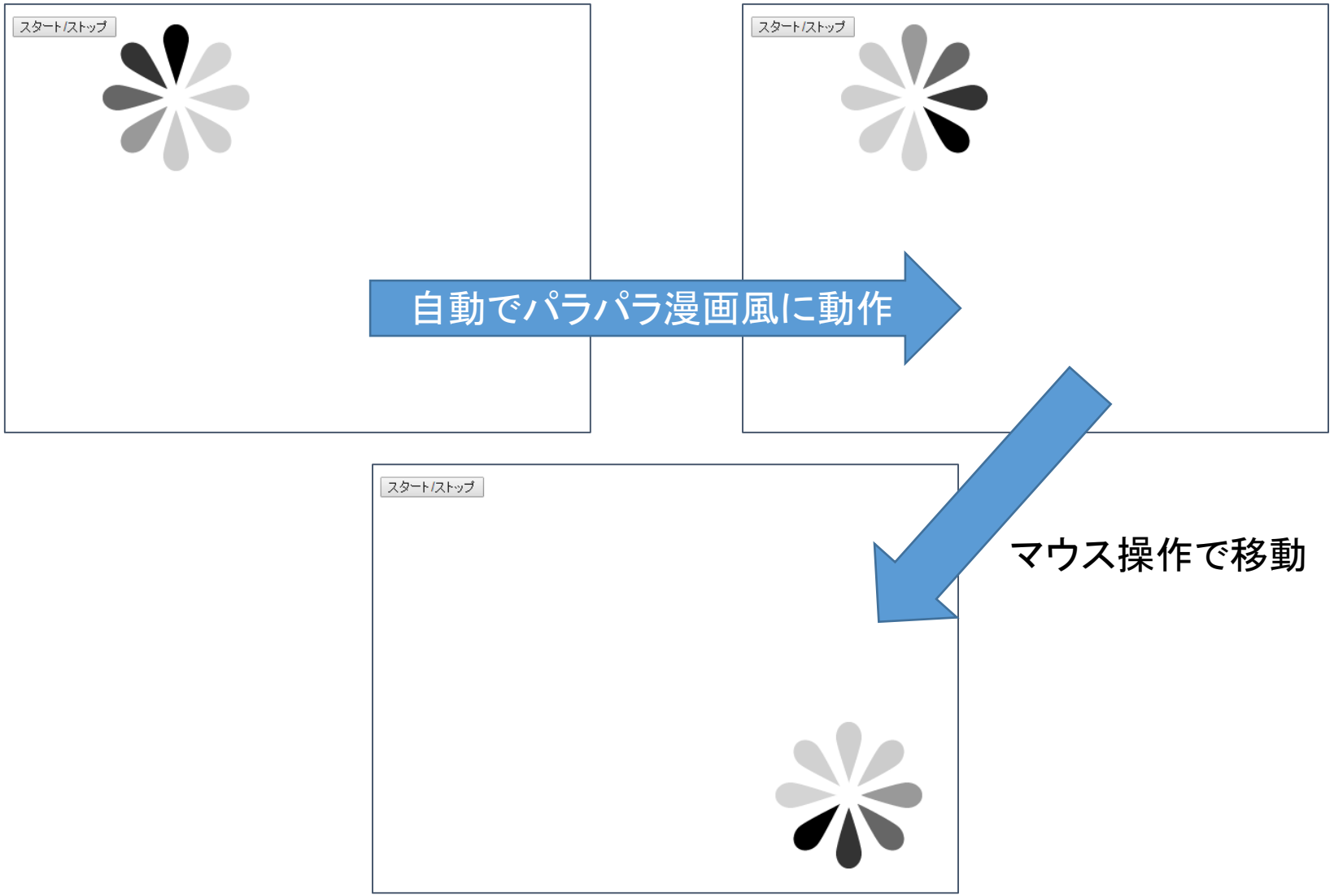
- ```
  $('div').hover(
        function()  { $('#image').fadeOut()  /* カーソルが乗った時 */ },
        function()  { $('#image').fadeIn()  /* カーソルが外れた時 */ }
  );
  ```

- ```
  $(document).ready(function()  {
        //要素の準備が出来たら実行
  });
  ```

ブラウザの処理の流れ
1. HTMLファイルの読み込み完了
2. $(document).ready()の実行
3. 画像読み込み・ブラウザ表示

- ```
  $(function()  {
        //jQueryを使うときにいつも書くこれは上記の省略記法
  });
  ```
  ```

アニメーション



一定時間後に繰り返す

- timerID = setTimeout ()
  - 指定した時間の経過後に関数を呼び出す
  - 関数の最後で自分自身を呼び出すように使うと
    繰り返し動作する関数を実現できる

```
function tokei() {
  $('#target').append("ほ");
  setTimeout(tokei, 200);
}
```

- clearTimeout (timerID)
  - タイマーを解除する
    - 繰り返し動作する関数もストップする

画像のプリロード

- **画像の読み込みには時間がかかる場合があるので，タイマー関数で画像を表示する前にブラウザに読み込んでおくとよい**

```
$("<img>").attr("src", "image0.png");
$("<img>").attr("src", "image1.png");
$("<img>").attr("src", "image2.png");
$("<img>").attr("src", "image3.png");
$("<img>").attr("src", "image4.png");
```

```
for (var i = 0; i < 5; i++) {
  $("<img>").attr("src", "image" + i + ".png");
}
```

練習: ぱらぱらマンガ

- 配布フォルダの画像を用いて，
  ぱらぱらマンガでアニメーションを作ってみよう
  - ヒント: setTimeout () を関数の最後に使う

- アニメーションのスピードを変えるとどうなるか試してみましょう
- 画像のプリロードにもトライしてみましょう

animate () メソッド

- **CSSのさまざまなプロパティ(色・サイズ・場所など)を徐々に変化させるメソッド**

```
$('#image').animate(CSSプロパティ,
                    変化時間,
                    変化スピード,
                    終了後の処理);
```

animate () メソッド

- **CSSのさまざまなプロパティ(色・サイズ・場所など)を徐々に変化させるメソッド**

```
$('#image').animate(CSSプロパティ,
                    変化時間,
                    変化スピード,
                    終了後の処理);
```

連想配列で複数指定が可能:

```
{
        プロパティ1: "変化後の値"
        …
        プロパティn: "変化後の値"
}
```

```
#image {
        position:absolute;
}
```

- **注意!**
  - **animate () メソッドを用いて「移動」を行う場合には,動かす対象のCSSの"position"プロパティをデフォルトから"relative"(相対位置で指定)もしくは"absolute"(絶対位置で指定)に変更する必要がある**

animate () メソッド

- **CSSのさまざまなプロパティ(色・サイズ・場所など)を徐々に変化させるメソッド**

```
$('#image').animate(CSSプロパティ,
                    変化時間,
                    変化スピード,
                    終了後の処理);
```

アニメーション全体にかかる時間:
"slow", "normal", "fast" か
ミリ秒単位の時間

アニメーションの変化スピード:
"linear":一定のスピードで変化
"swing":最初速くのちにゆっくり
など

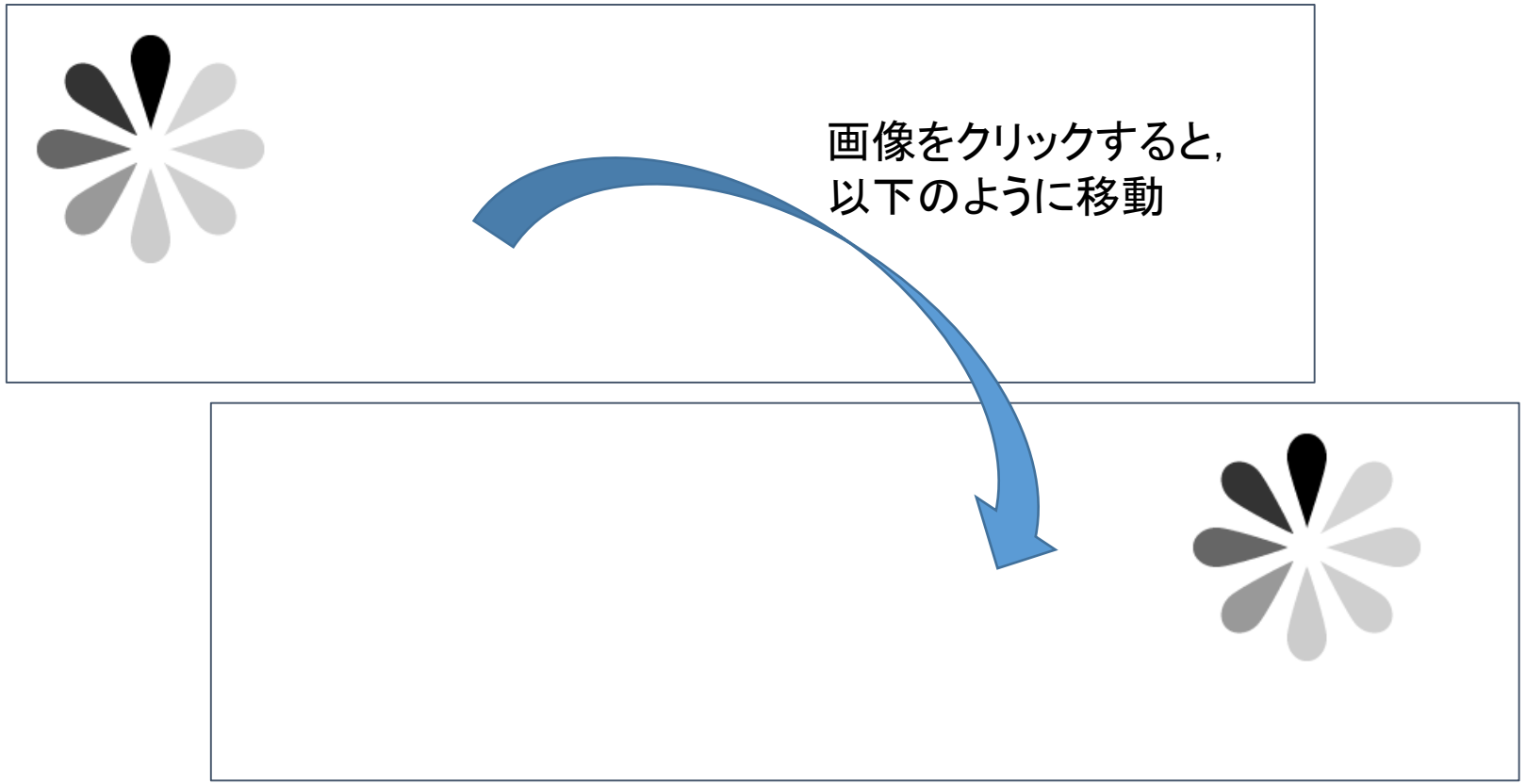
stop () メソッド

- animate () メソッドによるアニメーションは
  stop () メソッドで中止できる

```
$('#image').animate({'left','500px'}, 'slow', 'swing');
$('#image').stop();
```

練習: animate () メソッド

- **画像をクリックすると**
  決まった位置に動くプログラムを作ってみましょう



Eventオブジェクト

- click () メソッド や dblclick () メソッド に指定する関数に引数をとると Eventオブジェクト を取得できる

```
$(document).dblclick(function(event) {
  …
  var x = event.pageX;  // 文章内のx座標
  var y = event.pageY;  // 文章内のy座標
  …
}
```

- より詳細はリファレンスを参照のこと
  - http://api.jquery.com/category/events/event-object/
  - http://semooh.jp/jquery/cont/doc/event/

演習

- アニメーションのあるページを作成して公開する
  - 基本課題
    - ボタンをクリックでぱらぱらマンガのスタートストップ
    - 画面上をダブルクリックで画像の移動
  - 発展課題
    - 画像の移動中にダブルクリックが行われたら即座に新しい場所に向かう
    - 画像の移動中はぱらぱらマンガを一時停止する
    - ボタンをクリックしたら「スタート」「ストップ」と文字を変える
    - クリックした位置に画像の左上がきている場合は，クリックした位置が画像の中央になるように変更する
    - 画像やデザインやアニメーションスピードの変更などさまざまにアレンジしてください
- どうしてもわからない人へのヒント
  http://lss.oit.ac.jp/~t2015039/NetworkApplication-JavaScript/animation1.html

- ウェブページを作成・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に，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12回
  - ファイル名は「<学生番号>.txt」とする（ハイフンなし）　例: N14999.txt

メソッド

- 多様な種類が存在
  - スタイル操作: CSS
  - 属性操作: Attributes
  - 要素間の移動・検索: Traversing
  - 要素の追加・削除: Manipulation
  - イベント: Events
  - エフェクト: Effects
  - Ajax(非同期通信): Ajax
- 全ての使い方を覚える必要はなく，リファレンスを見ながら使えればOK
  - http://api.jquery.com（本家）
  - http://semooh.jp/jquery/（日本語）

JSON

- JavaScript構文を基にしたデータの表記法
  - 角括弧の中は配列
  - 波括弧の中はオブジェクト（連想配列）
  - 配列の配列（連想配列の配列）・連想配列の値が配列もできる

```
{"member":
        [
                {"firstName":"さとし", "lastName":"おおの"},
                {"firstName":"しょう", "lastName":"さくらい"},
                {"firstName":"まさき", "lastName":"あいば"},
                {"firstName":"かずなり", "lastName":"にのみや"},
                {"firstName":"じゅん", "lastName":"まつもと"}
        ]
}
```

JSON

- JavaScript構文を基にしたデータの表記法
  - 角括弧の中は配列
  - 波括弧の中はオブジェクト（連想配列）
  - 配列の配列（連想配列の配列）・連想配列の値が配列もできる

```
["おおの", "さくらい", "あいば", "にのみや", "まつもと"]
```

JSONのデータとして使えるもの
・数値
・文字列（ダブルクォートで囲む）
・ブール値（true/false）
・配列（角括弧でくくる）
・オブジェクト（波括弧でくくる）
・null

JSON

- JavaScript構文を基にしたデータの表記法
  - 角括弧の中は配列
  - 波括弧の中はオブジェクト（連想配列）
  - 配列の配列（連想配列の配列）・連想配列の値が配列もできる

{"firstName" : "さとし", "lastName" : "おおの"}

必ずダブルクォートで囲まなければならない

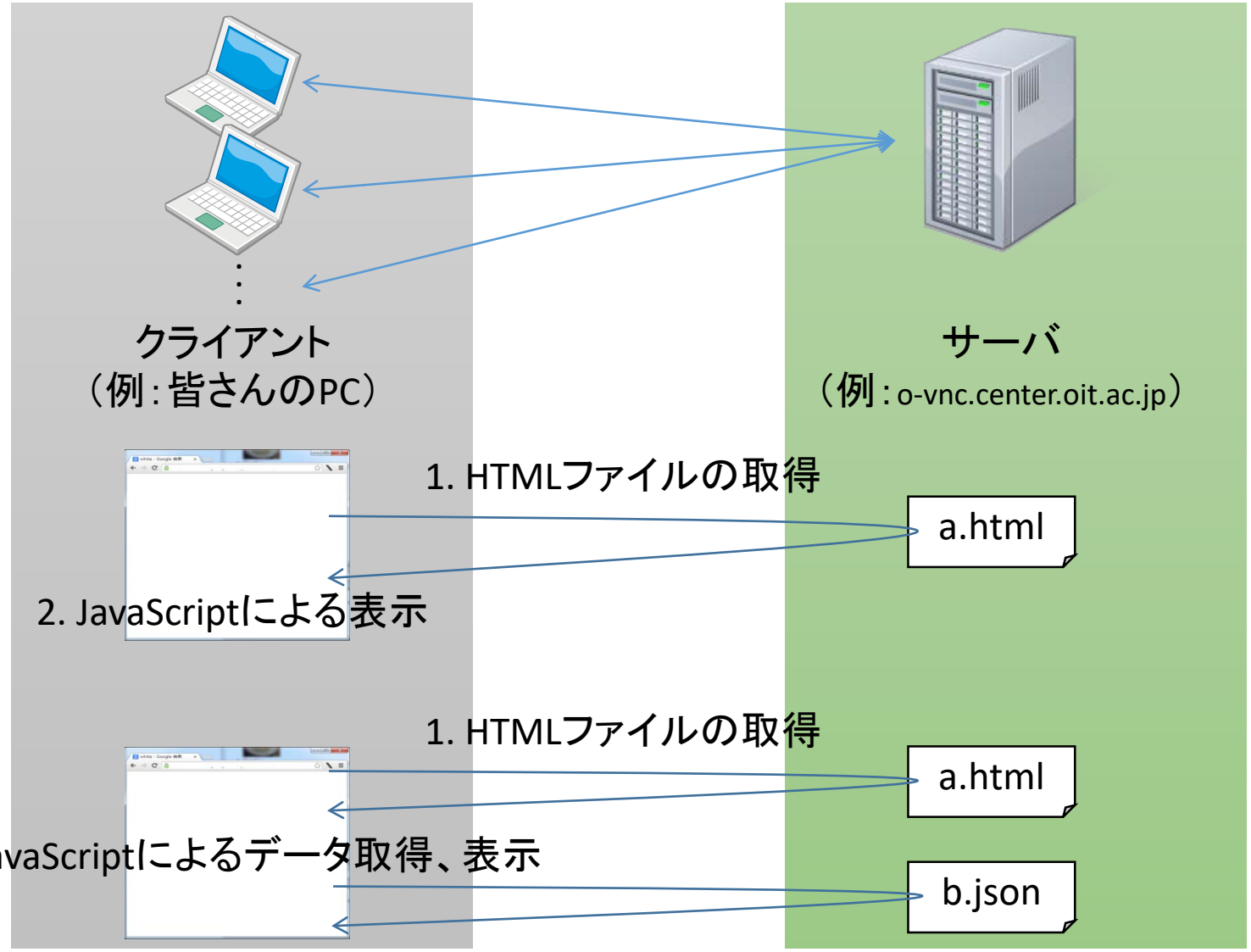
JSONのデータとして使えるもの
・数値
・文字列（ダブルクォートで囲む）
・ブール値（true/false）
・配列（角括弧でくくる）
・オブジェクト（波括弧でくくる）
・null

JSON

- JavaScript構文を基にしたデータの表記法
  - 角括弧の中は配列
  - 波括弧の中はオブジェクト（連想配列）
  - 配列の配列（連想配列の配列）・連想配列の値が配列もできる

```
{"member":
        [
                {"firstName":"さとし", "lastName":"おおの"},
                {"firstName":"しょう", "lastName":"さくらい"},
                {"firstName":"まさき", "lastName":"あいば"},
                {"firstName":"かずなり", "lastName":"にのみや"},
                {"firstName":"じゅん", "lastName":"まつもと"}
        ]
}
```

Ajax（非同期通信）


クライアント
（例：皆さんのPC）
サーバ
（例：o-vnc.center.oit.ac.jp）
これまで
1. HTMLファイルの取得
a.html
2. JavaScriptによる表示
今日
1. HTMLファイルの取得
a.html
2. JavaScriptによるデータ取得、表示
b.json

ajaxメソッド

```
$.ajax({
        type: "GET",
        url: "http://example.com/test.json",
        dataType: "json",
        success: function(data) {
          // データ取得後の処理を記述
        }
});
```

HTTP通信の種類を規定
GET or POST

リクエスト先のURLを記述

取得するデータタイプ
"html", "script", "json" 等

取得したデータは、
関数の引数（この例ではdata）に
格納されている

- **アクセス先のURLは、取得したHTMLと同じドメインのみ**
  - http://example.com/a.html
    ドメイン
- **異なるドメインからデータを取得する**
  JSONPという方法もある

getメソッド: ajaxをより簡単に

```
$.get(
        "http://example.com/test.json",
        "",
        function(data) {
          // データ取得後の処理を記述
        },
        "json"
);
```

リクエスト先のURLを記述

キーと値の組合せ
（静的ページ取得時は空文字）

取得したデータは、
関数の引数（この例ではdata）に
格納されている

取得するデータタイプ
"html", "script", "json" 等