

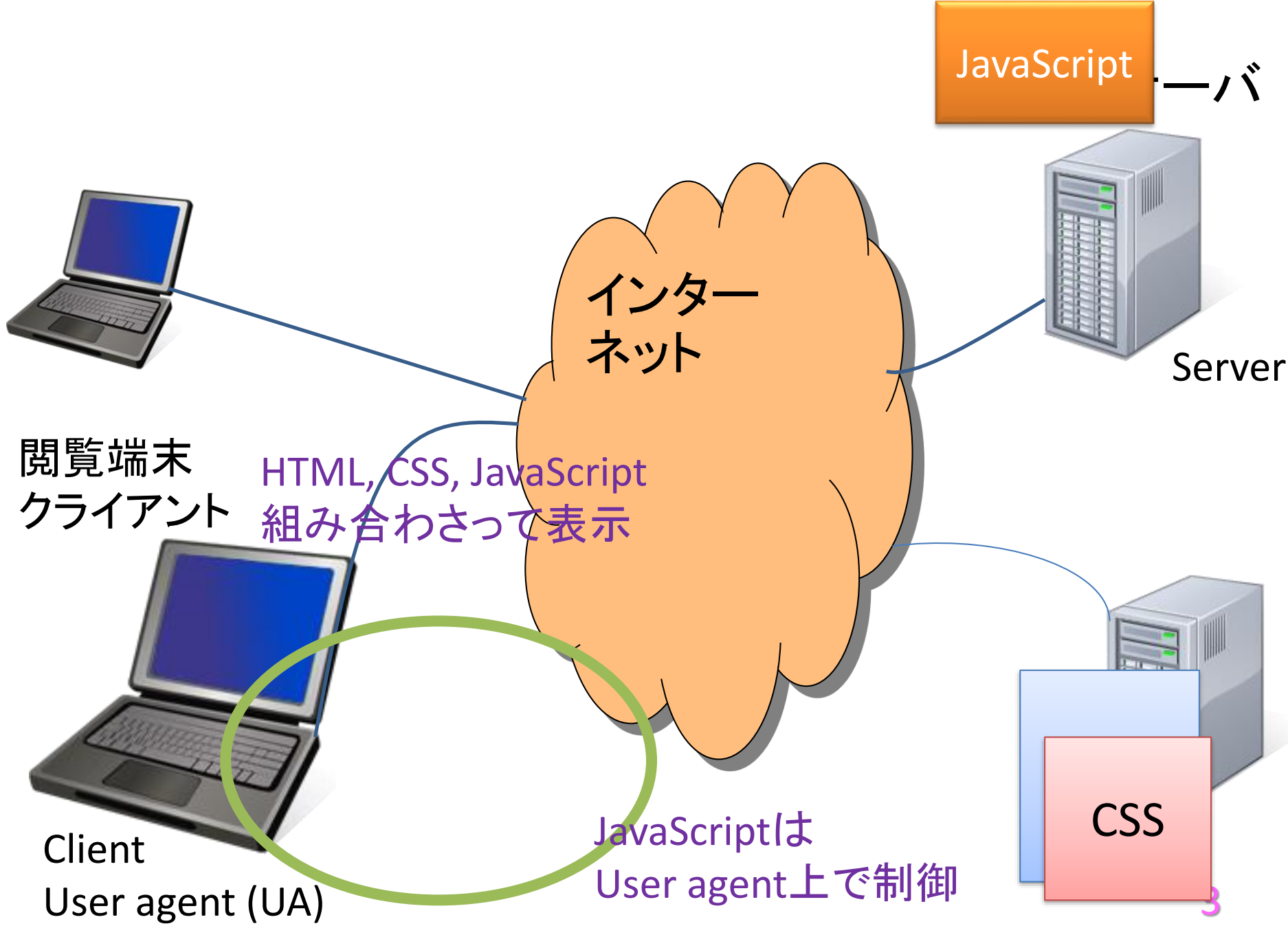
Web情報システム

—マルチメディア情報通信ソフトウェア

JavaScript

Webクライアントの制御

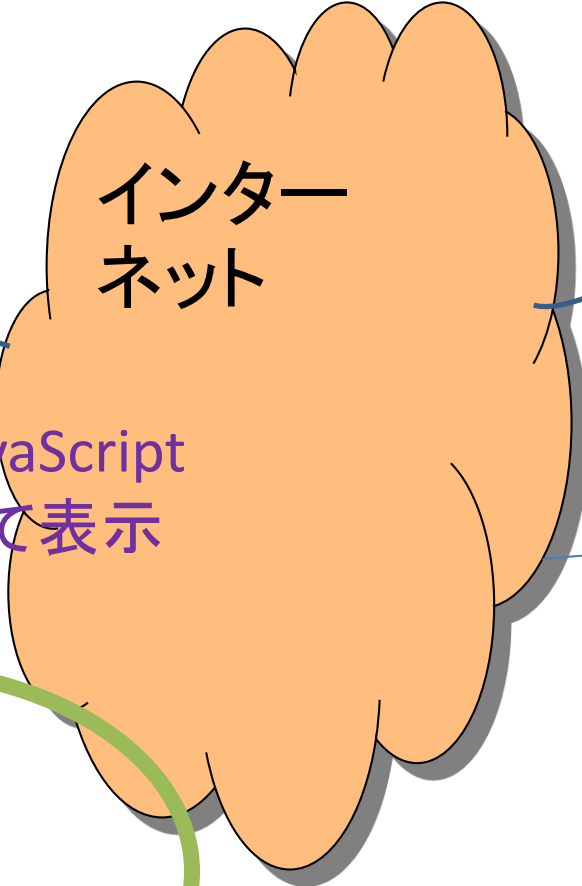
JAVASCRIPT



JavaScriptサーバ



Server



インターネット

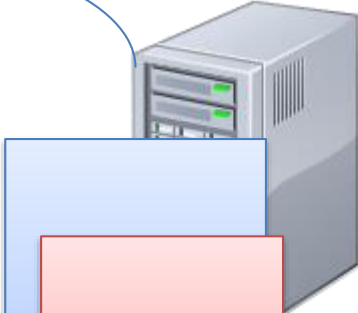
HTML, CSS, JavaScript
組み合わさって表示



閲覧端末
クライアント



Client
User agent (UA)



CSS

JavaScriptは
User agent上で制御

JavaScript

- WWWのUser agent上で使われることが多い
- この場合
- Webブラウザ上での制御
 - HTMLの読み書き
 - CSSの読み書き
 - 利用者からの入力
 - 時間経過による処理

JavaScript

□ JavaScript

➤ JAVA言語とは全く異なる

- ・マーケティング上の理由で似た名前を付けただけ

□ オブジェクト指向の手続き型構造化言語

➤ オブジェクト指向: オブジェクトを操作する

➤ 手続き型: 処理手順を順々に並べて書く

➤ 構造化: 処理のまとまりをわかりやすく書ける

□ イベントドリブン型 event driven

➤ 何か(イベント)が起こったら、これをする

JavaScriptの読み込み

□ プログラムファイル

- ▶ JavaScriptのプログラムファイル
- ▶ URLでアクセスできるように配置
- ▶ 拡張子 `js`

□ HTMLへの読み込み

- ▶ Script要素を使って指示

```
<script src="<プログラムのURL>"></script>
```

- ▶ 歴史的経緯もあり空要素ではない

```
cf. <link rel="stylesheet" href="<スタイルシートのURL>" />
```

HTML側の準備

- Script要素でJavaScript読み込みの指示

- HTMLで操作されるための仕込み

- 操作対象にid属性による名前をつける

例) `<p id="sample1">~~~~</p>`

- 他の方法

- class 属性による名前付け

- 何もしなくても、JavaScriptで順に探すこともできる

```
<html>  
<head>  
<script src="ch07.js"></script>  
~~~~~  
</head>  
<body>  
~~~~~  
<h2 id="sample1"> ~~~~~ </h2>  
~~~~~  
</body>  
</html>
```



```
window.onload = function()
{
  /* sample1 イベントハンドラ設定 */
  var node_sample1 = document.getElementById('sample1');
  node_sample1.onclick = sample1;
};

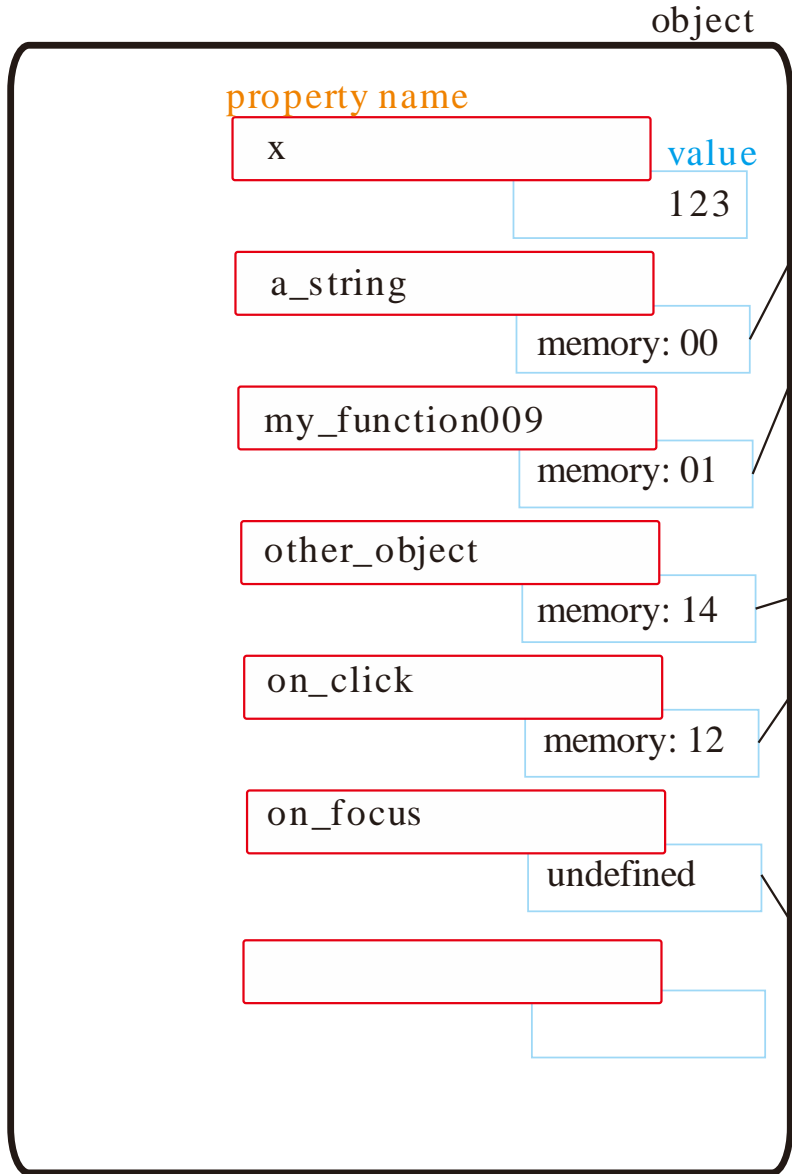
// Sample 1
function sample1(event)
{
  window.alert('Hello, world!');
}
```

オブジェクト Object

□ 操作盤のようなもの

- 現在の状態を示す(気温、風向)
- 操作指示(ライトをつける、プロペラを回す)

□ プログラム上で操作できるような記法



memory	
00	こんにちは。
01	function {
02	window.focus();
03	window.scroll();
04	var r = window.confirm();
05	if (r == true) {
06	var r2 = 0;
07	return r2;
08	} else {
09	return 3;
10	}
11	}
12	<i>function some_method</i>
13	memory: 01
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	object: undefined

object

property name	value

オブジェクトの構造

□ 操作盤の項目

- Property name とその値 value
- 読み取り専用も書き込める(操作)もある
- 自分で項目を新しく作れる

□ window オブジェクト

- Webブラウザ全体
- この中に document (HTML)オブジェクトなど

すべてがオブジェクト

□ window オブジェクト

▶ window.alert() メソッド

- 操作手順を書いたものが値
- 「呼び出す」と手順書が実行される

▶ window.onload プロパティ

- ここに手順書をセットしておく、
- HTMLの読み込みが完了した時に実行される

いくつか文法を

□ var キーワード

- ▶ 新しくpropertyを作成する
- ▶ var node_sample1

□ = 代入

- ▶ 左辺のpropertyで右辺の値を参照できるようにする。

□ ; セミコロン

- ▶ 文末を示す

```
var node_sample1 =  
document.getElementById('sample1');
```

オブジェクトのプロパティ

□.ドット

- ▶ オブジェクトのプロパティを示す
- ▶ `window.onload`

□値(中身)がメソッド(関数)の場合

- ▶ `window.alert`
 - 呼び出すときは、`window.alert()`
- ▶ `document.getElementById`
 - 呼び出すときは、`document.getElementById()`

メソッド

□メソッド method / 関数 function

□手順書

▶ バイトくんにわかるように指示したマニュアル

□例)

手順：商品を棚にしまう(商品は x とする)

扉を開ける

棚に x を置く

扉を閉める

棚の残りを報告

function キーワード

手順: 商品を棚にしまう
(商品は x とする)

扉を開ける

棚に x を置く

扉を閉める

棚の残りを報告

```
function メソッド名(引数リスト)
{
    手順1;
    手順2;
    手順3;
    return 報告する値;
}
```

```
window.onload = function()
{
    /* sample1 イベントハンドラ設定 */
    var node_sample1 = document.getElementById('sample1');
    node_sample1.onclick = sample1;
};

// Sample 1
function sample1(event)
{
    window.alert('Hello, world!');
}
```