Dynamic Web Apps

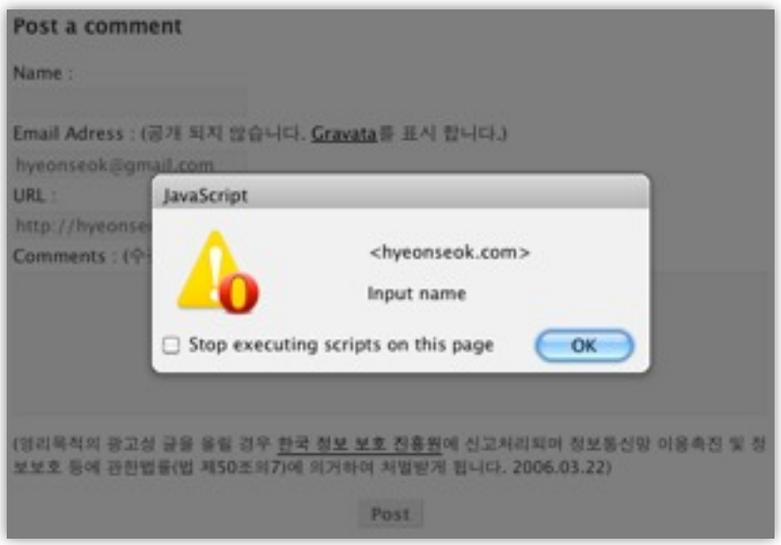
제주대학교 2011. 3. 30.

Introduction

Usage

◉ 메뉴

◉ 경고 메시지



◉ 탭, 이전/다음 UI

Introduction

Programming Language

- JavaScript != Java
- 인터프리트 언어
- 사용자 컴퓨터의 브라우저에서 작동
- HTML 문서에서 동적인 상호작용을 구현
- 브라우저 문제나 미지원 기능을 보완할 수 있음
- 최근 HTML5등이 부각되면서 중요 언어로 급부상 중

Introduction

<script> element

- <head>, <body> 안에만 위치할 수 있음
- HTML 파일 안에 스크립트를 직접 기술하는 방법

```
<script type="text/javascript">
alert('Hello world!');
</script>
```

- 외부 파일에 기술된 스크립트를 불러들이는 방법

```
<script type="text/javascript" src="hello.js"></script>
```

  - hello.js

```
alert('Hello world!');
```

Introduction

Variables

- Variable types
  - float, integer, string, boolean, arrays, objects

```
var i = 1;
var f = 1.0003
var s = 'Hello world';
var b = true;    // or false
var a = new Array('item1', 'item2', 'item3');    //a[0], a[1], a[2]

var o = new Object();
o.cpu = '1.6GHz Core2duo';
o.memory = '2GB';
o.size = '32.5cm x 22.7cm';
o.weight = '1.36Kg';
o.storage = '80GB HDD';
o.screenSize = '13.3 inch';
```

Introduction

Operators

- + - * / %
- ++ -- += -= *= /= %=
- == === != !== > < >= <= && || !
  - (5 == "5")는 true, (5 === "5")는 false
- String concatenation
  - "저는 " + something + "을 좋아합니다."

Introduction

Conditions

```
if (condition) {
    statement;
} else if (condition) {
    statement;
} else {
    statement;
}
```

```
if (a > 30) {
    result = 'a는 30보다 크다.';
} else if (a == 30) {
    result = 'a는 30이다.';
} else {
    result = 'a는 30보다 작다.';
}
```

Introduction

Loops

```
for (condition; end condition; change) {
    statement;
}
```

```
var names = new Array('Chris', 'Dion', 'Ben', 'Brendan');
var count = names.length;
for(var i = 0; i < count; i = i + 1){
    if (names[i] == 'Ben') {
        var message = 'It's him.';
        // do something with names[i]
    }
}
```

Introduction

Functions

- 반복되는 기능의 코드를 재사용하기 위한 기능 단위
- 하나의 함수에는 하나의 기능만 수행

```
function myFunction (arg1, arg2, arg3, ... ) {
    /* do something
       do something */
    return value;
}
```

```
function getCircleArea (rad) {
    var pi = 3.1415926535;
    var area = 2 * pi * rad * rad;
    return area;
}

var area1 = getCircleArea(15);
var area2 = getCircleArea(18);
```

H/W

1부터 n까지의 합

- 함수 선언
- for를 사용하여 1부터 n까지 루핑
- 루프 안에서 iteration 변수를 result 변수에 계속 더함
- 10에 대한 함수 실행
- 100에 대한 함수 실행
- 1000에 대한 함수 실행

DOM

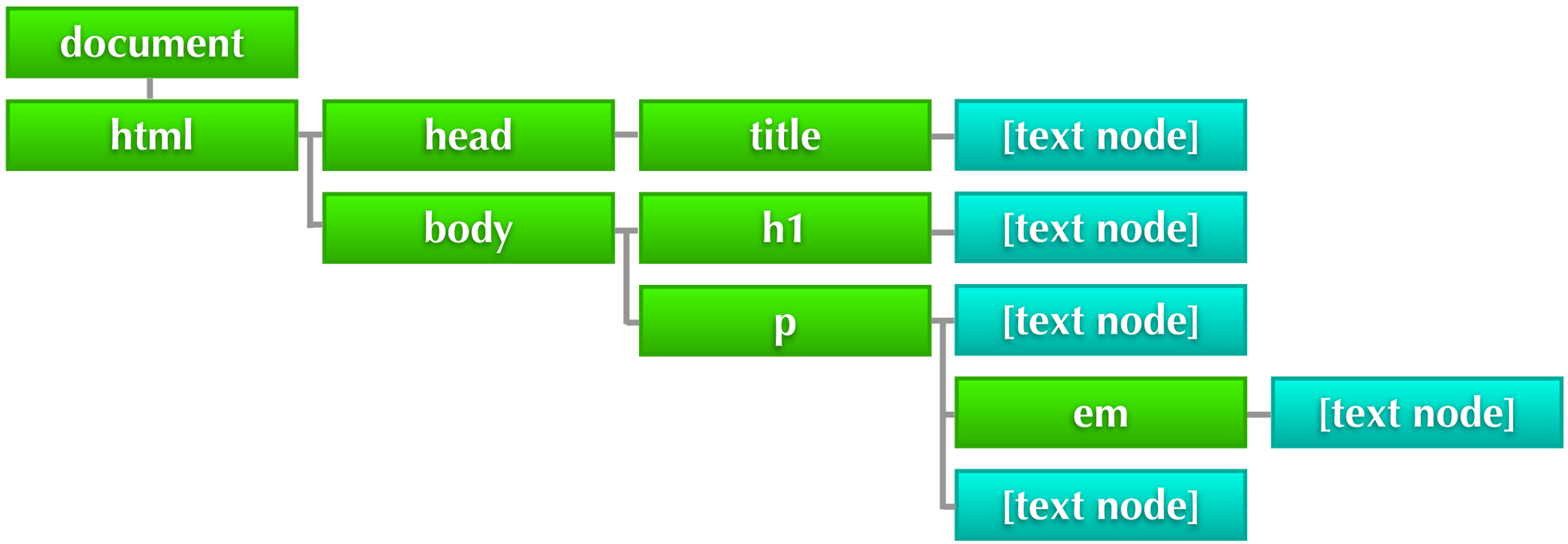
Document Object Model

- 문서의 객체 표현 모델
- 트리 구조
- HTML 문서의 태그, 속성, 텍스트, CSS의 속성, 값, 룰(rule) 정의 등 웹 문서의 구성 요소들은 DOM으로 표현된다.
- 웹에서 사용되는 자바스크립트는 많은 경우 DOM을 제어하는데 이용된다.
- http://www.w3.org/DOM/

DOM

Document Object Model

```
<html>
  <head>
    <title>This is a Document!</title>
  </head>
  <body>
    <h1>This is a header!</h1>
    <p id="exciting-text">This is a paragraph!<em>Excitement</em>!</p>
  </body>
</html>
```



DOM

Traversing the DOM

- Nodes
  - parentNode, childNodes

```
var theHtmlNode = document.childNodes[0];
if ( theHtmlNode.parentNode == document ) {
    alert( "The HTML node's parent is the document object!" );
}
```

```
var theHtmlNode = document.childNodes[0];
var theBodyNode = theHtmlNode.childNodes[1];
var theParagraphNode = theBodyNode.childNodes[1];
```

```
var theParagraphNode = document.childNodes[0].childNodes[1].childNodes
[1];
```

DOM

Traversing the DOM

- Nodes
  - firstChild, lastChild
    - 노드의 첫번째 자식

```
node.childNodes[0]  == node.firstChild
```

    - 노드의 마지막 자식

```
node.childNodes[node.childNodes.length - 1]  == node.lastChild
```

    - 따라서,

```
var theParagraphNode = document.firstChild.lastChild.childNodes[1];
```

DOM

Traversing the DOM

- Direct access
  - getElementById, getElementsByName

```
var theParagraphNode = document.getElementById('exciting-text');
```

  - getElementsByTagName

```
var theParagraphNode = document.getElementsByTagName('p').item(0);
```

  - querySelector, querySelectorAll

```
var theParagraphNode = document.querySelector('#exciting-text');
var theEmNode = document.querySelector('#exciting-text em');
```

DOM

Modifying DOM

- Modifying Child nodes
  - appendChild

```
node.appendChild(node);
```

  - removeChild

```
node.removeChild(node);
```

  - insertBefore

```
node.insertBefore(node, oldnode);
```

  - replaceChild

```
node.removeChild(node, oldnode);
```

DOM

Modifying DOM

- Adding Elements
  - createElement

```
var theParagraphNode = document.getElementById('exciting-text');
var newParagraphNode = document.createElement('p');
theParagraphNode.appendChild(newParagraphNode);
```

  - removeChild (again)

```
var theParagraphNode = document.getElementById('exciting-text');
theParagraphNode.parentNode.removeChild(theParagraphNode);
```

```
removeChild(newParagraphNode);     // (X)
```

Styling with DOM

DOM Styling properties

- element.style.cssProperties
  - 한단어로 된 CSS 속성: 그대로 표현

```
var el = document.getElementById('my-element');
el.style.color = '#000';
el.style.position = 'relative';
el.style.margin = '10px auto 15px';
el.style.background = 'url(mybg.png) top left #eee';
```

  - 대쉬를 포함한 CSS 속성: 카멜 케이징(CamelCasing)

```
var el = document.getElementById('my-element');
el.style.backgroundColor = '#000';
el.style.marginTop = '30px';
el.style.zIndex = '2';
el.style.fontSize = '3em';
el.style.letterSpacing = '1px';
```

Styling with DOM

DOM Attributes

- setAttribute
  - HTML 요소의 속성을 정의

```
var el = document.getElementById('my-element');
el.style.color = '#00f';
el.setAttribute('style', 'background-color: #ddd');
```

- getAttribute
  - HTML 요소의 속성을 반환

```
var el = document.getElementById('my-element');
el.getAttribute('id');     // my-element
```

H/W

스타일 변경

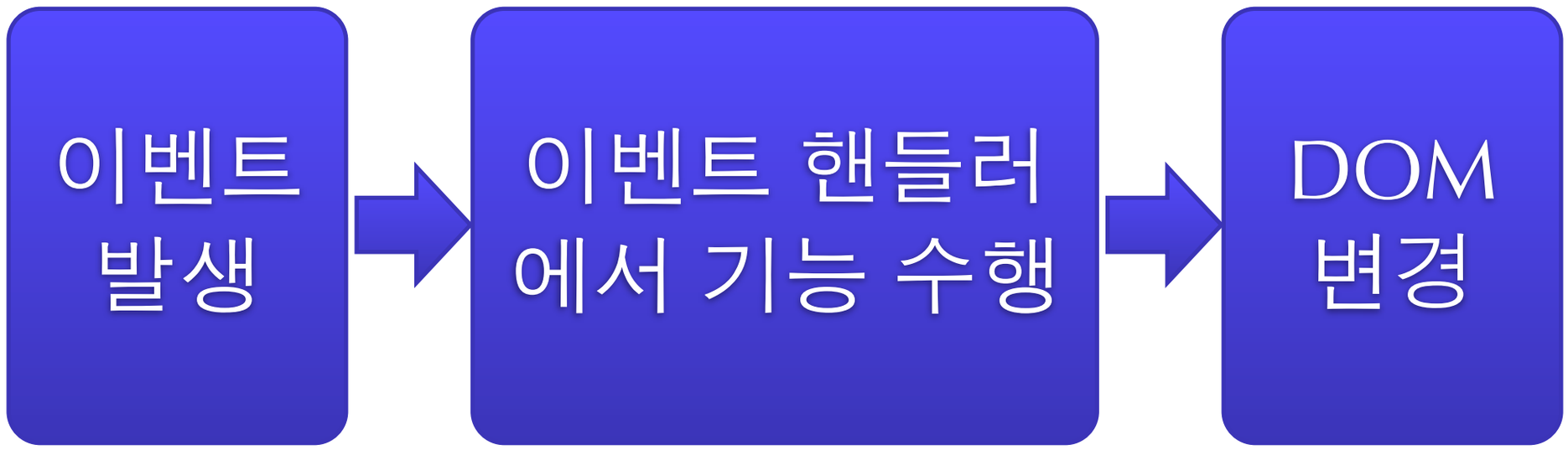
- 페이지가 열리면 붉은색 사각형을 녹색 사각형으로 변경
  - id로 HTML 요소를 참조
  - style 속성의 backgroundColor 속성에 색 대입
    - 붉은색: #f00, 녹색: #0f0

- changestyle.html

Event Handling

Introduction

- 사용자 동작에 따라서 웹 문서를 제어하기 위한 수단
  - 마우스 움직임, 클릭, 키보드 입력 등에 웹 문서가 반응하게 함



Event Handling

Image roll over/out

```
<script type="text/javascript">
function menuOn(imgEl) {
    imgEl.src = imgEl.src.replace(".gif", "_on.gif");
}

function menuOut(imgEl) {
    imgEl.src = imgEl.src.replace("_on.gif", ".gif");
}
</script>
<p><img src="hello.gif" onmouseover="menuOn(this);"
onmouseout="menuOut(this);" alt="안녕하세요."></p>
```

Event Handling

Type of events

- Mouse
  - Clicking: onmousedown, onmouseup, onclick, ondblclick
  - Movement: onmouseover, onmouseout, onmousemove
- Keyboard
  - onkeydown, onkeyup, onkeypress, onfocus, onblur
- Others
  - Form: onselect, onchange
  - Page: onload, onunload, onresize, onerror

Event Handling

Assign event

```
function menuOn() {
    this.src = this.src.replace(".gif", "_on.gif");
}

function menuOut() {
    this.src = this.src.replace("_on.gif", ".gif");
}
```

- element.event = function;

```
var el = document.getElementById('my-image');
el.onmouseover = menuOn;
el.onmouseout = menuOff;
```

- addEventListner

```
var el = document.getElementById('my-image');
el.addEventListner('mouseover', menuOn, false);
el.addEventListner('mouseout', menuOff, false);
```

Event Handling

addEventListner

- 여러개의 이벤트를 동시에 적용할 수 있다.
- W3C 표준과 IE 구현이 상이하다.
  - W3C 표준: addEventListener
  - MS IE: attachEvent
- 브라우저 호환을 위해서 중간 매핑 함수를 많이 쓴다.
  - http://dean.edwards.name/weblog/2005/10/add-event2/
  - http://dean.edwards.name/my/events.js

H/W

이벤트 발생시 스타일 변경

- 버튼 클릭 시 붉은색 사각형을 녹색 사각형으로 변경
  - 색을 바꾸는 함수 선언
    - id로 HTML 요소를 참조
    - style 속성의 backgroundColor 속성에 색 대입
      - 붉은색: #f00, 녹색: #0f0
  - 버튼의 onclick 이벤트에 함수 연결
- clicksetstyle.html

Debugger

Javascript debugging tools

- Firefox: Error console, Firebug
- Opera: Error console, DragonFly
- Safari: Error console, Debugging, Profiling
- MS IE: Internet Explorer Developer Toolbar

H/W

탭메뉴 만들기

- tabmenu.html
  - HTML element 탭을 누르면
    - HTML elements 내용만 보이고 나머지는 감춰짐
    - 탭의 배경색을 #aaa로 변경
  - HTML Attribute, CSS Properties 도 상동
- hyeonseok@gmail.com으로 제출
  - 메일 제목은 아래의 형식으로 제출
    - 제주대학교 학과명 이름 학번