

Lab 2: JavaScript

정보 비주얼라이제이션 2015 Fall

human-computer interaction + design lab.

Joonhwan Lee

DOM (Document Object Model)

DOM이란 무엇인가?

- ✦ DOM (문서객체모델, Document Object Model) 은 HTML 과 XML 문서에 대한 프로그래밍 인터페이스
- ✦ 문서에 대한 구조적 정보를 제공하고, 문서의 구조나 모습, 내용을 프로그래밍을 통해서 바꿀 수 있는 방법을 제공
- ✦ DOM은 문서 형태인 웹페이지를 Property 와 Method 를 가지는 객체와 노드의 트리형 구조로 표현 → 웹 문서를 객체지향적인 형태로

DOM이란 무엇인가?

- ✦ **D: 문서 (Document)**
 - ✦ 문서 객체 모델에서 문서는 가장 중요한 기본 단위
 - ✦ 웹 문서를 만들어 웹 브라우저에 띄우는 순간 웹 브라우저는 작성된 문서를 객체로 변환하여 DOM 형태로 구조화

DOM이란 무엇인가?

♦ O: 객체 (Object)

- ♦ 객체란 물리적으로 식별 가능한 것.
 - ♦ 프로그래밍 차원에서는 메모리 공간을 할당하고 있는 것.
- ♦ 객체는 주로 속성(property) 과 메소드(method) 를 가지고 있다.
 - ♦ 예: `car.brand = hyundai` `car.drive`
- ♦ DOM에서의 객체란 문서의 여러 요소들을 접근 가능한 물리적인 형태로 표현한 것
 - ♦ 예: `<h1>`, `<p>`, `<body>` 등의 태그로 둘러 싸인 요소들은 모두 객체로 인식

DOM이란 무엇인가?

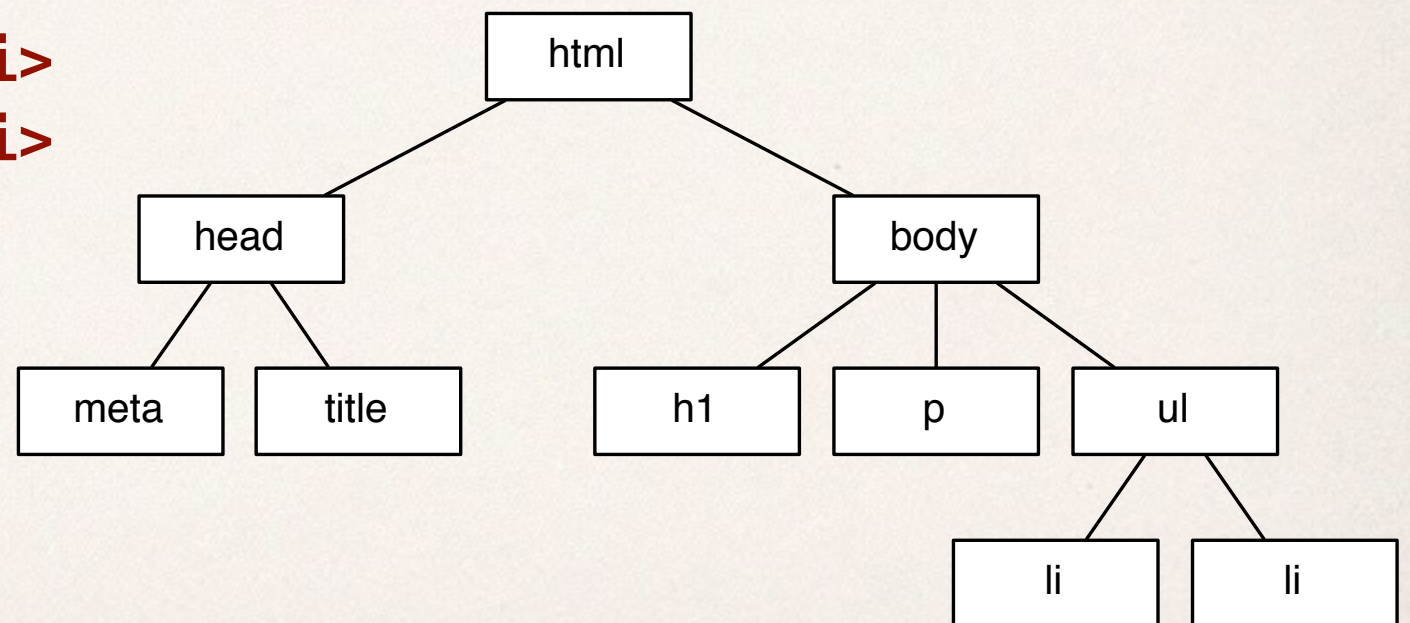
- ✦ **M: 모델 (Model)**
 - ✦ 개념을 구체적인 형태로 표현하는 것
 - ✦ DOM에서는 문서의 구조를 나뭇가지(tree)형 구조로 표시한다

DOM이란 무엇인가?

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/
    html; charset=utf-8"/>
    <title>테스트 페이지</title>
  </head>
  <body>
    <h1>큰 제목 1</h1>
    <p>본문내용.....</p>
    <ul id="list">
      <li>리스트1</li>
      <li>리스트2</li>
    </ul>
  </body>
</html>
```


DOM이란 무엇인가?

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/
    html; charset=utf-8"/>
    <title>테스트 페이지</title>
  </head>
  <body>
    <h1>큰 제목 1</h1>
    <p>본문내용.....</p>
    <ul id="list">
      <li>리스트1</li>
      <li>리스트2</li>
    </ul>
  </body>
</html>
```



DOM의 노드

- ◆ 노드 (node): 어떤 연결망에서 특정 지점과 지점을 연결하는 점.
- ◆ DOM에서 문서는 노드의 집합
 - ◆ 요소노드, 텍스트 노드, 속성 노드 등이 있다.

DOM의 노드

- ✦ 요소노드 (element node)

- ✦ DOM의 가장 기본적인 단위가 요소노드

- ✦ `<body>`, `<p>` 와 같이 웹 문서의 가장 기본적인 조각인 요소들을 지칭

- ✦ 요소는 다른 요소를 포함할 수 있다.

- `<ul>`

- `<li>.....</li>`

- `<li>.....</li>`

- `</ul>`

- ✦ `<ul>` 은 두개의 `<li>` 요소를 포함하고 있는 요소

DOM의 노드

- ✦ 텍스트노드 (text node)
 - ✦ 문서의 기본 단위는 요소 → 문서의 구조를 만든다
 - ✦ 요소만 존재한다면? → 구조는 만들어졌으나 내용이 X
 - ✦ 문서에 내용을 추가하기 위해 텍스트를 제공 → 텍스트노드
 - ✦ 예: `<p>Hello, HTML5 World!</p>`

DOM의 노드

- ✦ 속성노드 (attribute node)
 - ✦ 속성(attribute)은 요소에 대해 좀더 자세한 정보를 표현할 때 사용.
 - ✦ `<p title="my first html5 programming">Hello, HTML5 World!</p>`



Hello, HTML5 World!

my first html5 programming

DOM의 노드

- ✦ `<p title="my first html5 programming">Hello, HTML5 World!</p>`
 - ✦ 요소노드: p
 - ✦ 속성노드: title="my first html5 programming"
 - ✦ 텍스트노드: Hello, HTML5 World!

DOM과 CSS

- ✦ DOM은 웹 문서 구조와 상호 작용하기 위한 기술
 - ✦ 자바스크립등이 웹문서에 접근할 때 필요
- ✦ CSS 역시 웹 문서 구조와 상호 작용하기 위한 기술
 - ✦ CSS 에서 특정 요소의 스타일을 지정
 - ✦

```
p {  
    color : yellow;  
    font-family : "arial", sans-serif;  
    font-size : 1.2em;  
}
```

DOM과 JavaScript

- ✦ 자바스크립트: DOM 의 객체에 접근하기 위한 다양한 메소드를 제공
- ✦ **getElementById**
 - ✦ 도큐먼트 내에서 특정 id 를 사용하는 요소에 접근하기 위한 메소드
 - ✦ e.g., `document.getElementById("article")`
HTML문서 내에서 "article" 이라는 아이디를 가진 요소를 참조 → 객체로 인식
 - ✦ Example1.html
 - ✦ 버튼을 누르면 `changeColor` 메소드가 `para` 라는 아이디를 가진 요소를 찾아 그 요소 문자열의 칼라를 매개변수로 지정된 칼라로 바꾼다.

DOM과 JavaScript

✦ `getElementsByTagName`

- ✦ 아이디가 없는 요소를 찾을 때 사용 (예: `p`, `h1`, `li`, etc...)
- ✦ e.g., `document.getElementsByTagName("li")`
문서 내에서 `<li>` 를 가진 모든 요소를 찾는다 (array)
- ✦ 찾은 요소가 한개가 아닐 가능성이 높기 때문에 찾은 모든 요소
중 원하는 요소를 선택하는 방법을 찾아야.
- ✦ `Example2.html`
 - ✦ 문서 내에서 `<li>` 요소를 전부 찾은 후, 숫자를 출력 (alert 창 이용)

DOM과 JavaScript

✦ `getAttribute`

- ✦ 요소에 접근하기 위해서 `getElementById`나 `getElementsByTagName` 이라는 두가지 방법을 사용
- ✦ 원하는 요소에 접근한 후 속성값을 알기 위해 `getAttribute` 를 사용
- ✦ e.g.,

```
var paras = document.getElementsByTagName("p");  
paras[0].getAttribute("title");
```
- ✦ Example3.html
 - ✦ `<p>` 요소를 찾은 후, `title` 속성이 있는 것들을 alert 창에 출력

DOM과 JavaScript

✦ **setAttribute**

- ✦ 지금까지의 메소드는 정보를 추출하는 목적으로 사용
- ✦ 특정 속성 노드의 값을 바꾸기 위해서 `setAttribute` 를 사용

✦ e.g.,

```
var paras = document.getElementsByTagName("p");  
paras[0].setAttribute("title", "changed title");
```

✦ Example4.html

- ✦ `<p>` 요소를 찾은 후, `title` 속성을 `setAttribute` 로 교체

JavaScript Basics

자바스크립트

- ♦ 객체 기반의 스크립트 프로그래밍 언어
- ♦ 웹사이트에 주로 사용 (다른 응용프로그램에 내장되어 사용되기도 한다)
- ♦ 넷스케이프에서 처음 Mocha 라는 이름으로 개발 → LiveScript → JavaScript 로 개명
- ♦ Java 와는 전혀 상관없다 (구문은 유사한 점이 있음 → C 를 바탕으로 함)
- ♦ 현재 최근 버전은 1.8, 유럽의 ECMA 에서 표준을 제정, ECMAScript 라고도 부름
- ♦ 브라우저 전쟁 시대에 MS 는 Visual Basic 에 기반한 VBScript 로 경쟁

자바스크립트의 문법

- ◆ 기초 문법
- ◆ 변수와 배열
- ◆ 연산자
- ◆ 조건문과 반복문
- ◆ 함수와 객체

자바스크립트 프로그래밍의 기초

- ♦ 모든 코드는 (X)HTML로 된 문서 내에서 실행 됨
- ♦ `<script>` 태그에 자바스크립트 코드가 들어감
- ♦ 컴파일러가 필요없다
 - ♦ 웹브라우저가 코드를 real time 으로 실행하고 해석 (interpreter)

자바스크립트의 문법

- ✦ 모든 언어와 마찬가지로 자바스크립트도 나름대로의 문법이 존재
- ✦ C 로 부터 파생된 언어이기 때문에 자바나 C와 비슷한 문법 구조를 갖는다.

✦ 명령문

- ✦ 코드가 브라우저에서 로딩되면서 순차적으로 실행 됨.
- ✦ 단순히 줄바꿈만 하면 명령어를 구분할 수 있으나 일반적으로 세미콜론 (;) 으로 명령어를 구분하는 것이 올바른 습관

```
command line 1  
command line 2
```

```
command line 1;  
command line 2;
```

자바스크립트의 문법

♦ 주석 (comment)

- ♦ 자바스크립트 해석기가 모든 명령어 라인을 해석하지 않는다.
- ♦ 주석처리가 된 라인은 무시하고 진행 (// 혹은 /* */ 로 주석처리)

♦ e.g.

```
<script type="text/javascript">  
    // Write a heading  
    document.write("<h1>This is a heading</h1>");  
    // Write two paragraphs:  
    /* document.write("<p>This is a paragraph.</p>");  
    document.write("<p>This is another paragraph.</p>"); */  
</script>
```

자바스크립트의 문법

◆ 변수 (variables)

- ◆ 상황에 따라 변하는 값을 의미 (변하지 않는 값: 상수)
- ◆ 변수의 선언과 변수의 할당
- ◆ 변수의 선언
 - ◆ `var a;` 혹은 `var a, b;`
 - ◆ 값이 할당될 변수를 컴퓨터 메모리에 준비하는 과정
- ◆ 변수의 할당
 - ◆ `a = "apple";      b = "banana";      c = 2.54;`
 - ◆ `var a = "alphabet";` (선언과 동시에 할당)

자바스크립트의 문법

♦ 변수 (variables)

- ♦ 변수는 대소문자를 구별: myVar 와 myvar 는 서로 다른 변수
- ♦ 변수에 공백이나 마침표, \$ 등과 같은 특수 문자를 사용할 수 없음

자바스크립트의 문법

♦ 데이터 형식 (Data Type)

- ♦ 변수에 값을 할당할 때, 어떤 값이 할당 되는지 프로그래밍 언어에게 지시해 줄 필요가 있으나 자바스크립에서는 보통 생략

- ♦ `a = "hello";`
`a = 24.5;`
`a = true;`

- ♦ a에 할당된 세 값은 각각 문자열, 숫자, 불린(참/거짓) 값 → 사람은 값의 종류를 파악할 수 있으나 프로그래밍 언어는 어떤 값이 사용되었는지를 지정해 주어야 → 형지정 (typing)

- ♦ strongly typed language vs. weakly typed language (JScript → weakly...)

자바스크립트의 문법

◆ 데이터 형식 (Data Type)

◆ 문자열 (String)

- ◆ 한개이상의 문자로 이루어진 데이터 형
- ◆ 문자는 문자, 숫자, 마침표, 공백 등을 포함 함.
- ◆ 문자열을 할당할 때는 따옴표로 감싸 주어야 한다.

- ◆ `var mood = "happy";` (o)
- ◆ `var mood = 'happy';` (o)
- ◆ `var mood = "don't ask";` (o) (문자열 내에 따옴표가 있는 경우)
- ◆ `var mood = 'don\'t ask';` (o) (문자열 내에 따옴표가 있는 경우)
- ◆ `var mood = 'don't ask';` (x) (문자열 내에 따옴표가 있는 경우)

자바스크립트의 문법

♦ 데이터 형식 (Data Type)

♦ 숫자

- ♦ 변수의 값으로 숫자 사용 가능
- ♦ 소수를 포함해 어떤 숫자도 표시할 수 있다
- ♦ 다른 언어 (e.g., Java) 와 달리 특별히 형지정이 필요 없음

- ♦ `int num = 245;`

- ♦ `float num = 35.235;`

- ♦ `num = 245;        // JScript`

- ♦ `num = 35.235;    // JScript`

자바스크립트의 문법

♦ 데이터 형식 (Data Type)

♦ 불린 (Boolean)

- ♦ true or false

- ♦ 프로그래밍 내에서 스위치 용도로 많이 사용

- ♦ e.g.,

```
if checkbox_checked == true then
    do_something
else
    do_something_else
```

- ♦ 불린 값은 문자열과 달리 따옴표를 사용하지 않음

- ♦ `var isChecked = true;` // 불린값 지정

- ♦ `var isChecked = "true";` // 문자열 true 지정

자바스크립트의 문법

◆ 배열 (Array)

- ◆ 문자열, 숫자, 불린 → 단일 값을 가진다
- ◆ 변수에 여러 개의 값을 지정하고 싶을 때, 배열 (Array) 을 사용
- ◆ 배열: 하나의 변수에 여러 개의 값이 순서대로 지정되어 있는 것
- ◆ 배열 내에 지정된 각각의 값 → 배열 요소

Java	C++	Ruby	Obj C	JScript	HTML	CSS
[0]	[1]	[2]	[3]	[4]	[5]	[6]

총 7개의 요소를 가진 배열 (인덱스는 0부터 시작)

자바스크립트의 문법

◆ 배열 (Array)

- ◆ 배열의 선언: Array 라는 명령어로 배열을 선언할 수 있고, 배열에 포함될 요소의 갯수를 미리 지정 가능 (배열의 길이 지정)

// 배열의 길이를 7개로 지정 (7개 까지 저장)

```
var languages = Array(7);
```

// 배열 길이 미지정

```
var languages = Array();
```

- ◆ 배열의 할당

```
var languages = Array(7);
```

```
languages[0] = "Java";
```

```
languages[1] = "C++";
```

```
languages[2] = "Ruby";
```


자바스크립트의 문법

✦ 배열 (Array)

- ✦ 배열의 선언 및 할당

```
var languages = Array("Java", "C++",  
"Ruby");
```

```
var languages = ["Java", "C++", "Ruby"];
```

- ✦ 숫자나 불린 등을 지정할 수 있다. 섞어 써도 상관없음 (다른 언어와 차별점)

```
var myArray1 = Array(123, 456, 789);
```

```
var myArray2 = Array("Java", 1234, false);
```

자바스크립트의 문법

✦ 배열 (Array)

- ✦ 배열은 다른 배열을 포함할 수 있다

```
var person1 = Array("Jeff", 1970, "male");  
var person2 = Array("Jane", 1980, "female");  
var people = Array();  
people[0] = person1;  
people[1] = person2;  
alert(people[0][0]);
```


자바스크립트의 문법

✦ 연산자 (operator)

✦ 산술 계산을 하거나 데이터를 다루는 작업 → 연산(operation)

✦ 산술 연산자: = + - / *

(괄호를 사용해서 연산 순서 결정)

age = 10; //age 변수에 10을 지정

age = age + 1; //age 변수에 저장된 값에 1을 더함

age++; //age 변수에 저장된 값에 1을 더함

age = age - 1;

age--;

자바스크립트의 문법

✦ 연산자 (operator)

- ✦ + 연산자는 문자열의 접합에도 사용 (concatenation)

```
var myString = "I " + "program " +  
"JavaScript" + ".";  
var language = "Objective C";  
var myString = "I " + "program " +  
language + ".";  
var year = 2015;  
var myString = "올해는 " + year + "입니다.";
```

"123" + 123 → 123123

123 + 123 → 246

자바스크립트의 문법

♦ 조건문과 연산자 (operator)

♦ 조건문

- ♦ 명시한 조건이 참인지 거짓인지 판단하여 계산이나 결과를 수행하는 방법

```
if ( 조건1 ) {  
    조건1이 참일 경우 수행  
} else if ( 조건2 ) {  
    조건2가 참일 경우 수행  
} else {  
    조건1, 조건2가 모두 거짓일 경우 수행  
}
```

조건은 true/false 의 불린값을 가진다.

자바스크립트의 문법

✦ 조건문과 연산자 (operator)

```
✦ switch ( 조건 ) {  
    case 값1:  
        조건과 값1이 같으면 실행;  
        break;  
    case 값2:  
        조건과 값2가 같으면 실행;  
        break;  
    default:  
        조건과 맞는 case 가 없으면 실행;  
}
```


자바스크립트의 문법

♦ 조건문과 연산자 (operator)

- ♦ 비교 연산자: 조건이 참인지 거짓인지 판단하려면 비교 연산자가 필요
 - ♦ 예: $a = 5, b = 3 \rightarrow a > b$ 는 참, $a < b$ 는 거짓
 - ♦ 크기 비교: $>$, $<$, $>=$, $<=$, $==$, $!=$
- ♦ 논리 연산자: 조건이 여러개 있을 경우 논리연산자를 사용
 - ♦ 예: 조건1 과 조건2 를 모두 만족하는 경우 (AND 연산자)
조건1 과 조건2 둘 중 하나를 만족하는 경우 (OR 연산자)
 - ♦ $a \&\& b$ $a || b$
- ♦ 부정 연산자 (!) : $!a \rightarrow a$ 가 참이면 거짓, a 가 거짓이면 참

자바스크립트의 문법

◆ 반복문

- ◆ 특정 명령어를 반복적으로 수행하게 하는 프로그래밍 기법

- ◆ 반복문의 종류

- ◆ `for` (초기조건; 조건식; 조건 변경) : 초기조건을 바탕으로, 조건식에 따라 조건을 변경하여 반복조건을 만족할 때 까지 명령어를 반복 실행.
- ◆ `while` (조건) : 조건을 만족할 때 까지 (조건==true) 명령문을 반복한다. 내부에 조건을 반복하는 코드가 없으면 무한루프에 빠질 수 있음.
- ◆ `do...while` (조건) : `while` 과 비슷하나 `do...while` 은 특정 코드를 적어도 한번은 실행 가능

자바스크립트의 문법

♦ 반복문

♦ for 문

1 부터 10까지 출력하는 예제.

```
for (var i=1; i <= 10; i++) {  
    document.write(i + "<br>");  
}
```

Example9.html : for 문 예제

Example8.html : 어레이를 사용한 for 문 예제

자바스크립트의 문법

♦ 반복문

♦ while 문

1 부터 10까지 출력하는 예제.

```
var i = 1;
while (i < 11) {
    document.write(i + "<br>");
    i++;
}
```


자바스크립트의 문법

♦ 반복문

♦ do...while 문

1 부터 10까지 출력하는 예제.

```
var i = 1;
do {
    document.write(i + "<br>");
    i++;
} while (i < 11)
```

자바스크립트의 문법

✦ 함수 (function)

- ✦ 소스코드 안 어디서나 불러서 다시 사용할 수 있는 명령어의 묶음.

```
function countdown() {  
    var count = 10;  
    for (var count = 10; count > -1;  
count--) {  
        document.write(count + "<br>");  
    }  
}  
countdown();
```

자바스크립트의 문법

✦ 함수 (function)

✦ 인수(argument) 와 결과값 (return value)

- ✦ 함수는 목적은 같은 명령어를 반복하는데에만 있지 않음
- ✦ 주어진 값을 이용해 결과 값을 만들어 내는 것이 함수의 목적
- ✦ e.g., 섭씨/화씨 변환 프로그램

섭씨 = convertToCelsius(화씨);

화씨는 인수, 섭씨는 convertToCelsius 라는 함수의 결과 값.

convertToCelsius 함수는 인수만 달리하여 반복하여 사용할 수 있다.

자바스크립트의 문법

◆ 함수 (function)

- ◆ 화씨 → 섭씨 변환 코드

```
function convertToCelsius(temp) {  
    var result = temp - 32;  
    result = result / 1.8;  
    return result;  
}  
var temp_fahr = 95;  
cels = convertToCelsius(temp_fahr);
```

자바스크립트의 문법

✦ 함수 (function)

- ✦ 두개 이상의 인수를 사용할 때

```
function multiply(num1, num2) {  
    var total = num1 * num2;  
    return total;  
}  
  
multiply(5, 20);  
multiply(39, 250);
```


자바스크립트의 문법

✦ 자바스크립트의 실제 활용

✦ 섭씨/화씨 변환기

✦ 섭씨 `<input id="c" onkeyup="convert('C')">`도
= 화씨 `<input id="f" onkeyup="convert('F')">`도

```
✦ function convert(degree) {  
    if (degree == "C") {  
        F = document.getElementById("c").value * 9 / 5 + 32;  
        document.getElementById("f").value = F;  
    } else if (degree == "F") {  
        C = (document.getElementById("f").value - 32) * 5 / 9;  
        document.getElementById("c").value = C;  
    }  
}
```


Questions...?
