

# 웹 프로그래밍(Part A)

Client-side web programming: JavaScript

강사: 박대철 교수

2016.07.25-28

# 학습 내용 목차[JavaScript]

## Part A

- 개요
- 1. 시작하기
- 2. 핵심 자바스크립트 구문
- 3. 데이터 유형과 객체
- 4. 문서 객체 모델(DOM) 이해하기
- 5. 문서 객체 모델 작업
- 6. 이벤트 와 이벤트 리스너 처리
- 7. 자바스크립트 프로그래밍 디버그

## Part B

- 8. 스마트 폼 작성
- 9. UI 향상 기법
- 10. 자바스크립트 실용 실습
- 11. 자바스크립트 라이브러리
- 12. 자바스크립트와 HTML5
- 13. 고급 자바스크립트 특징
- 14. 실무 프로젝트

## Part C

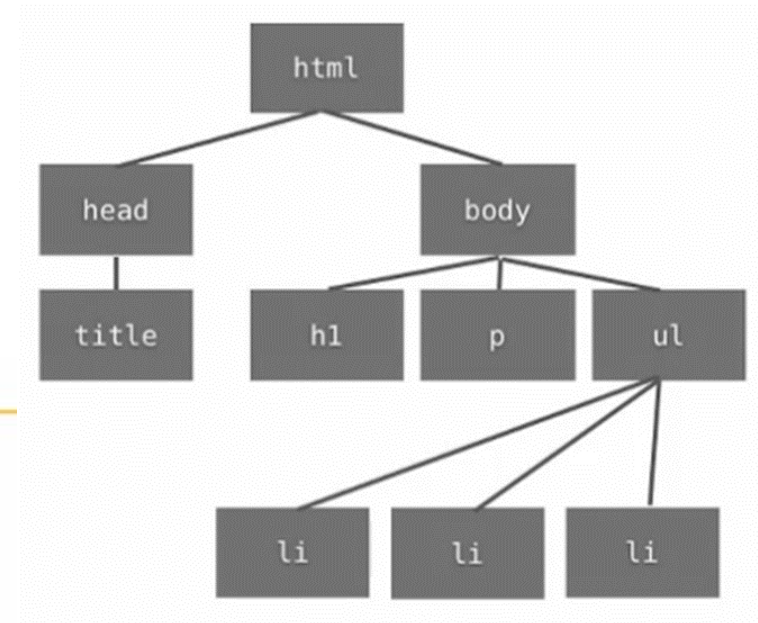
## Part D

## + 선수 학습 요구 사항(1/)

- 사전 요구 지식
  - HTML 구문

### HTML

```
<html>
  <head>
    <title>Simple Page</title>
  </head>
  <body>
    <div id="main">
      <h1>Today's Headline</h1>
      <p>Regular paragraph text goes here.</p>
    </div>
  </body>
</html>
```



## + 선수 학습 요구 사항 (2/)

- 사전 요구 지식
- CSS 구문

CSS

---

```
p {  
  color:#333;  
  font-size:1.2em;  
}  
  
#main {  
  width:90%;  
  background-color:#eee;  
}
```

## + 선수 학습 요구 사항(3/)

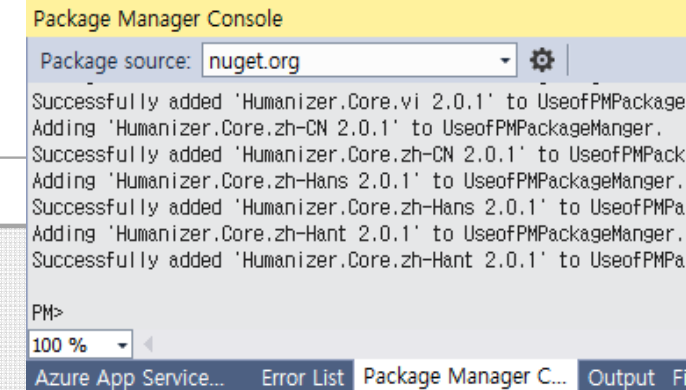
- 숙달된 프로그래머일 필요는 없다
- C언어에 기초한 자바스크립트 언어의 핵심 구문 학습
- C/C# 혹은 자바 등과 같은 컴파일 언어에 대한 사전 지식이 있다면 유사한 부분을 자바스크립트를 학습하며 만나게 될 것이다

# JavaScript 패키지 매니저

Npm 패키지 자바스크립트 서버사이드 패키지 매 니저	Bower 패키지 클라이언트사이드 웹 패키지 매니저	Jspm 패키지 브라우저 패키지 매니저
<a href="https://www.npmjs.com/">https://www.npmjs.com/</a>	<a href="https://bower.io/">https://bower.io/</a>	<a href="http://jspm.io/">http://jspm.io/</a>
download a zip file from <a href="https://nodejs.org/en/download/">https://nodejs.org/en/download/</a> and unpack it in the same folder where node.exe lives	Bower is a command line utility. Install it with npm. \$ npm install -g bower	jspm is a package manager for the <a href="#">SystemJS universal module loader</a> , built on top of the dynamic <a href="#">ES6 module loader</a>
Install packages with npm <a href="#">install</a> . npm installs packages to node_modules/.	Install packages with <a href="#">bower install</a> . Bower installs packages to bower_components/. \$ bower install <package>	<b>Install jspm CLI:</b> \$ npm install jspm -g
예제: \$ npm install colors \$ npm test.js  2016.07.25-28	예제: \$ bower install angular Index.html 수정 <script src="bower_components/jquery/dist /jquery.min.js"> </script>  1. Introduction	예제: \$ jspm install angular Index.html 수정 <script src="jspm_packages/system.js"> </s cript> <script src="config.js"> </script> <script> System.import("angular"); </script>  6

# Visual Studio.NET 패키지 매니저

- Command line tool → PM
- GUI tool → Nuget 패키지 매니저



## PM 패키지 사용 이전

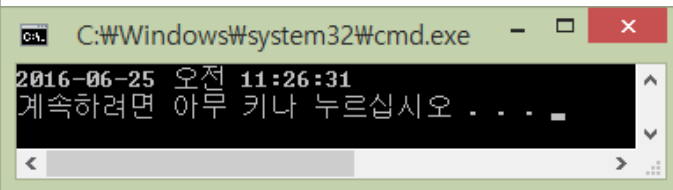
```
namespace UseofPMPackageManger
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine(DateTime.Now);
        }
    }
}
```

## PM 패키지 사용 이후

```
using Humanizer;
namespace UseofPMPackageManger
{
    class Program
    {
        static void Main(string[] args)
        {
            //Console.WriteLine(DateTime.Now.AddHours(-2));
            Console.WriteLine(DateTime.Now.AddHours(-2).Humanize());
        }
    }
}
```

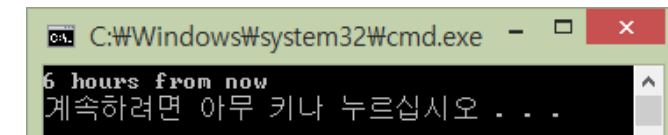
Tools>NuGet Package Manager>Package Manager Console 선택

PM>Install-Package Humanizer  
Using Humanizer 네임스페이스 추가  
C#코딩에서 Humanizer() 메소드 사용 가능



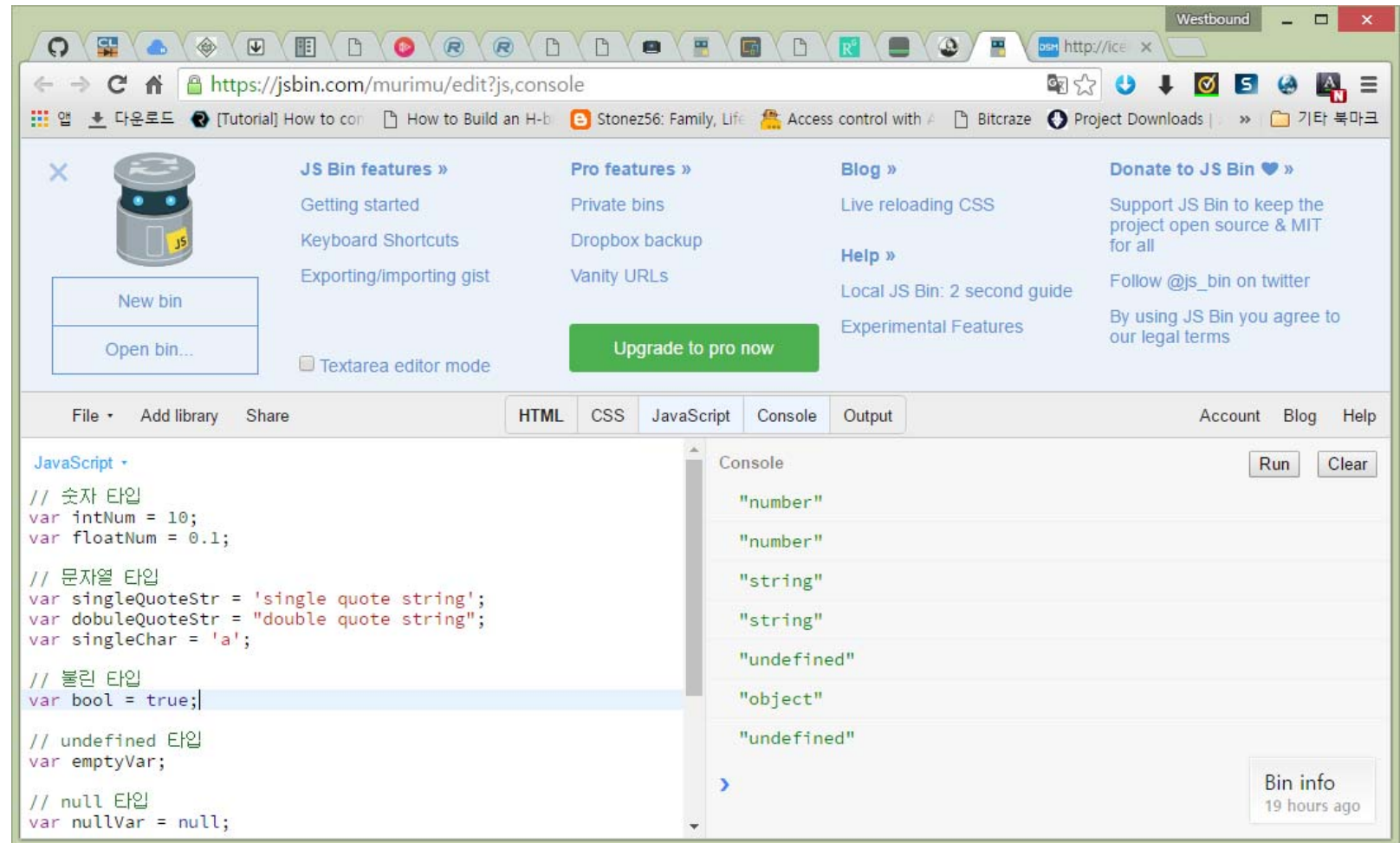
2016.07.25-28

1. Introduction



# 웹기반 HTML5/CSS3/JavaScript IDE-JS Bin

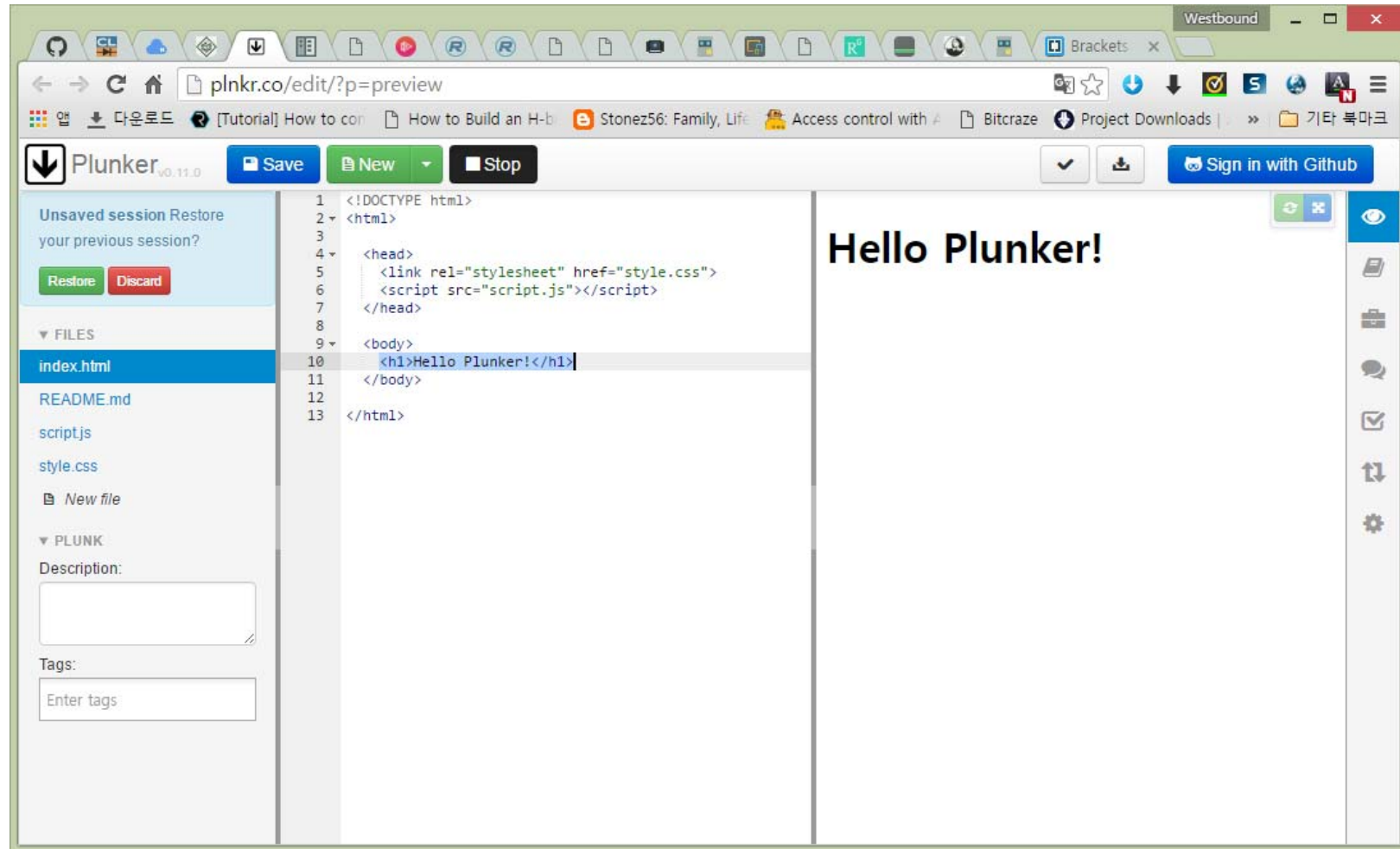
- JS Bin
- <https://jsbin.com/?js,console>





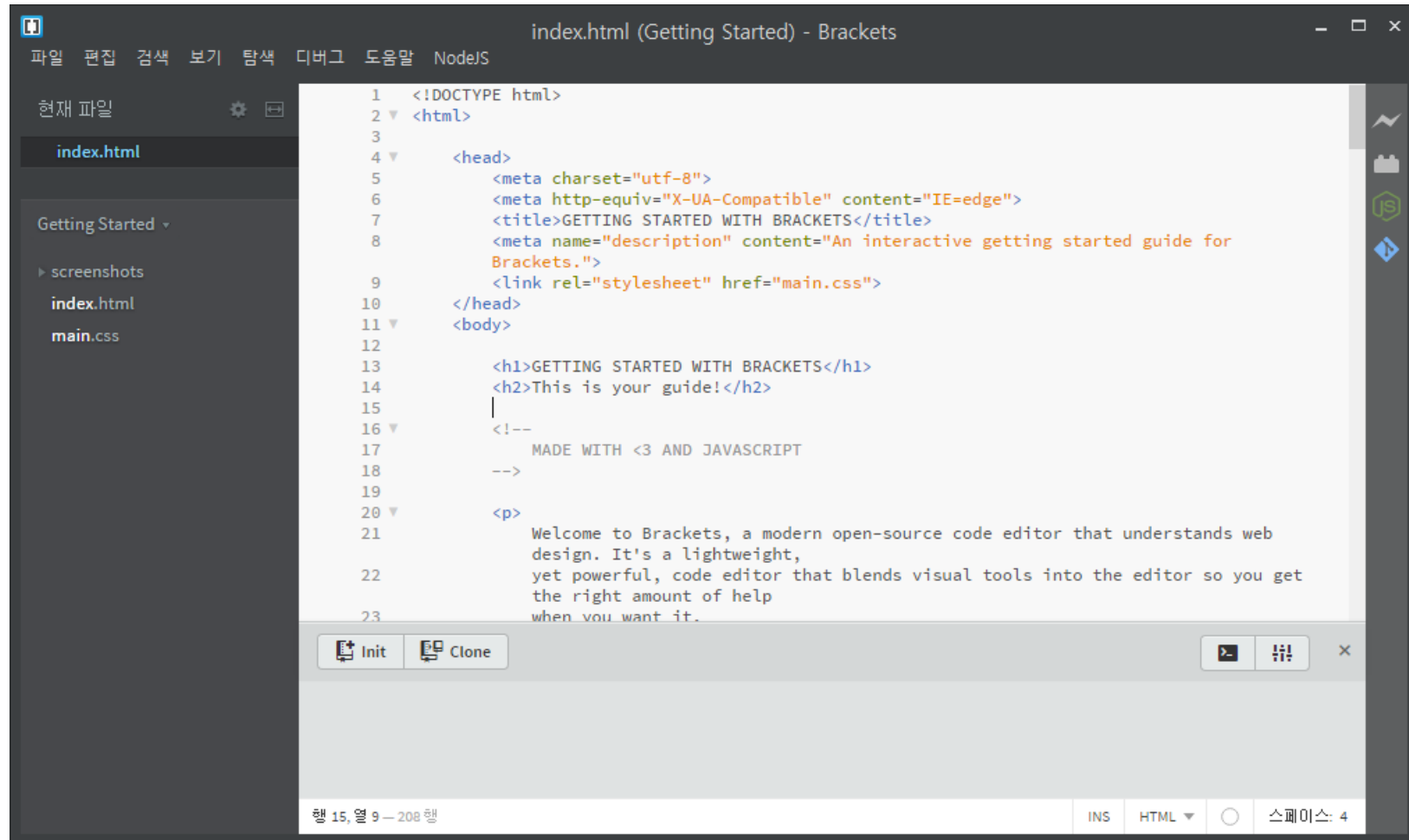
# 웹 기반 HTML5/CSS3/JavaScript IDE-plnkr

- Plnkr
- <http://plnkr.co/edit/?p=preview> 연  
결하여 사용



# HTML5/CSS3/JavaScript 데스크탑용 IDE

- Brackets
- <http://brackets.io/> 에서 다운  
로드 받아 설  
치



+

## 1. 시작하기(1/)

- 웹 페이지의 3개의 핵심 언어



markup language  
**content**

내용 과 페이지 구조 설계



style sheet language  
**presentation**

내용 과 페이지 표현 설계

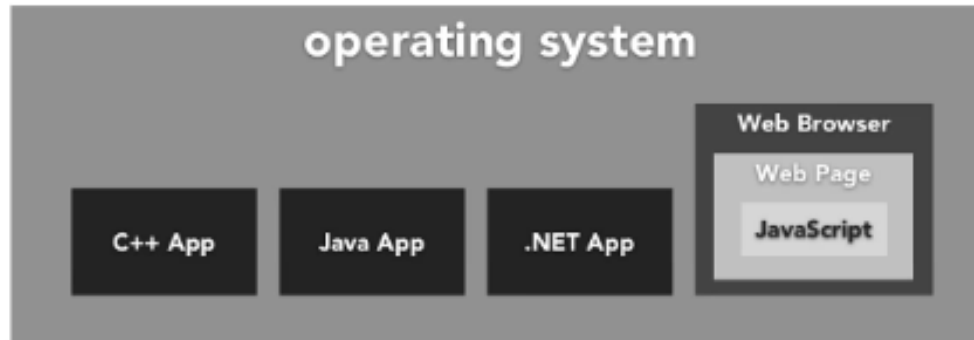


programming language  
**behavior**

내용 과 페이지 행위/이벤트 설계

## 1. 시작하기(2/)

자바스크립트 언어란?



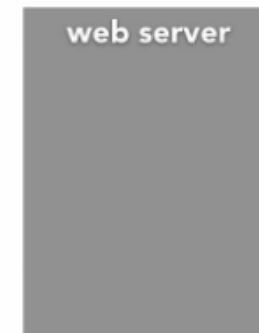
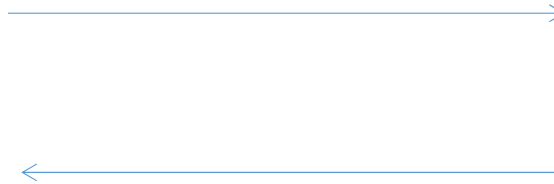
로컬 파일을 접근할 수 없다  
직접 데이터베이스를 접근할 수 없다  
하드웨어(USB등)을 접근할 수 없다

- JavaScript는 브라우저 앱 내에서만 동작한다. 즉, 웹 페이지를 조작하기 위해 설계된 언어이다.
- JavaScript는 클라이언트 쪽 언어이다.
- it was designed to manipulate web pages and it does that very well.

## 1. 시작하기(3/)

Javascript 는 클라이언트 사이드 언어이다(node.js는 서버사이드 언어)

클라이언트사이드 언어가 갖는 문제점



Javascript 는 클라이언트 사이드에서 차단될 수 있다.

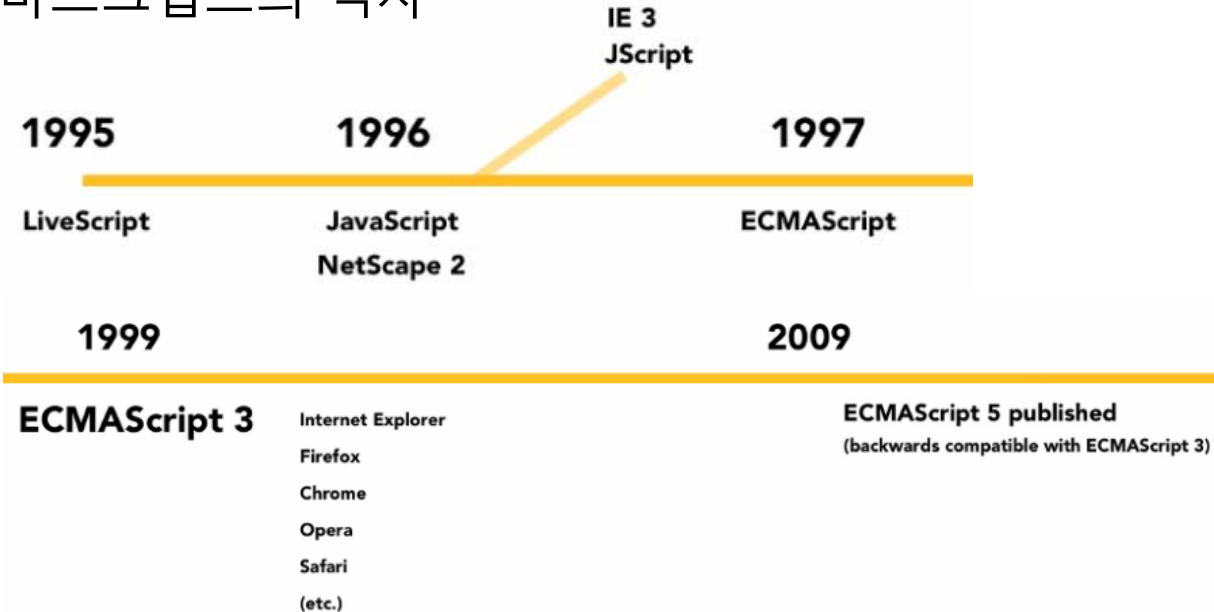
브라우저간 호환성 Cross-Browser compatibility(CSS핸들링 브라우저마다 다름)

**PHP, ASP.NET, Ruby on Rails node**

- Server-side languages
  - PHP, ASP.NET, Ruby on Rails, JSP
  - Node.js → acrobat :노드.js 사용, photoshop : 구글 앱스 스크립트 사용
  - Javascript is in server-side products like Node.js and Goole Apps Script.

## 1. 시작하기(4/)

### 자바스크립트의 역사



- JavaScript는 1995년 넷스케이프사의 Brendan Eich에 의해 개발
- It was originally going to be called LiveScript Netscape and Sun had an agreement to rename LiveScript to JavaScript
- 따라서 JavaScript는 Java가 아니다! It was not based on Java.
- The standardized edition of this language is officially called ECMAScript → 현재 2015년 ECMA6 발표됨

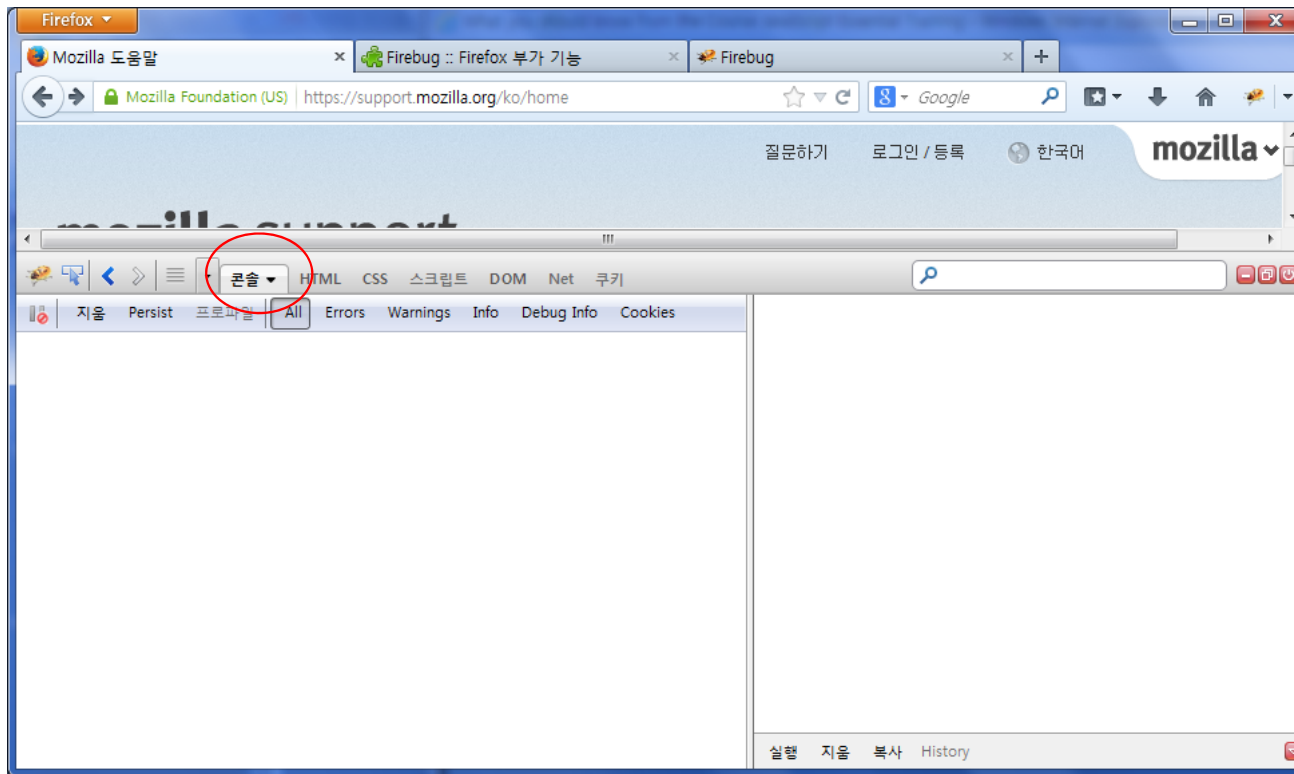
## 플랫폼 별 Javascript 도구들?



- 모든 현대 웹 개발 혹은 프로그래밍 혹은 프로그래밍 앱은 자바스크립트 편집을 지원함.
- -Dreamweaver, Xcode, Coda, TextMate, TextWrangler, BBEdit, Visual Studio, Webmatrix, Eclipse, Komodo, Emacs, Vi, Sublime, Brackets, WebStorm and more
- Recommend: Brackets, Aptana Studio 3 <http://www.aptna.com/products/studio3>
- Browser : Firefox (or Chrome)
- Online web version: <http://plnkr.co/edit/?p=preview> , <https://jsbin.com/?js,console>

## 웹 브라우저와 추가 기능들 Web Browser and Addons?

- Firefox : <https://addons.mozilla.org/ko/firefox/>
- Extension for firefox : <http://getfirebug.com/>
- <https://getfirebug.com/firstrun#Firebug%201.12.5>



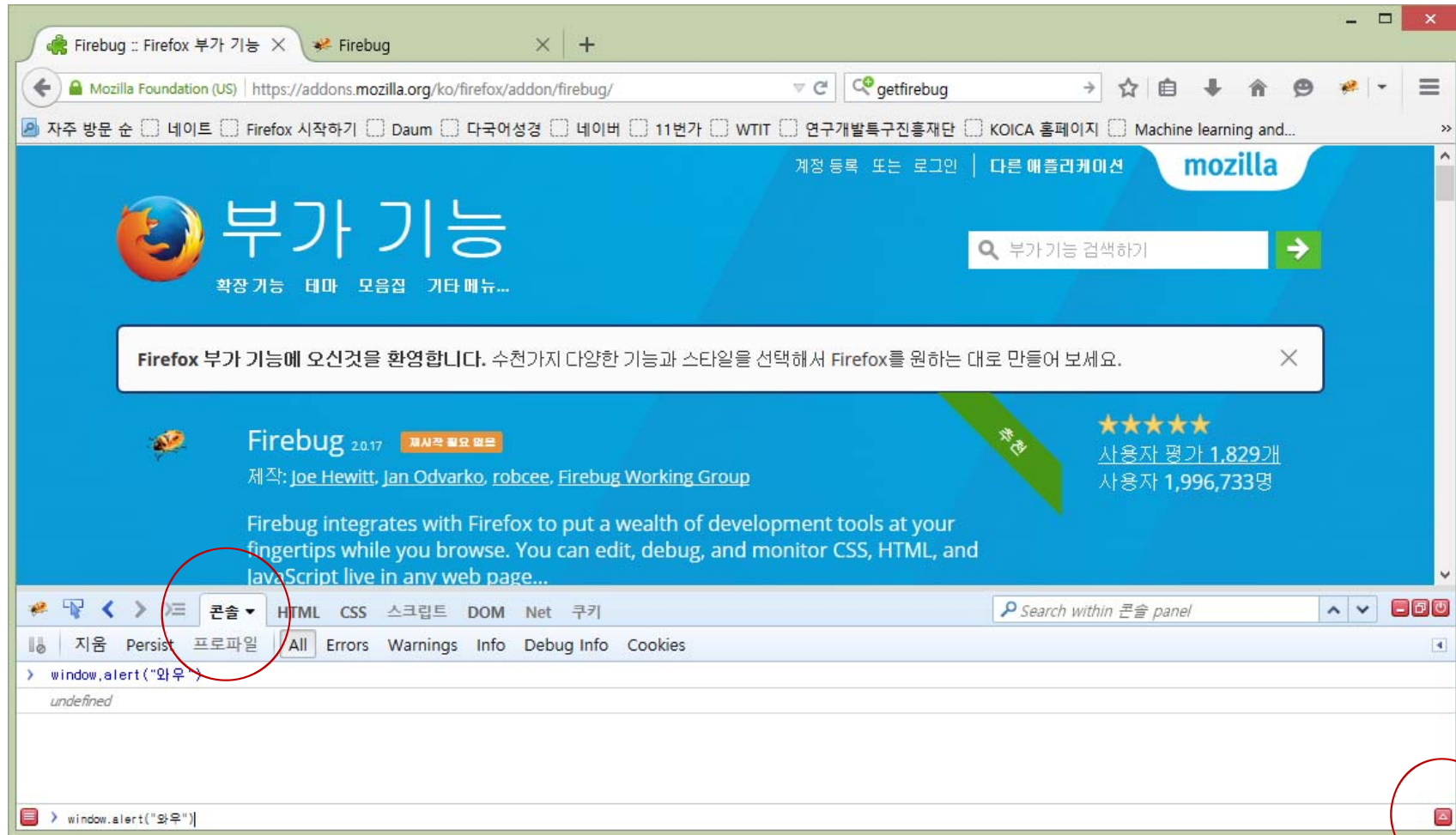
On/Off

Javascript  
Command  
Editor 창



2016년 클라우드 자  
바 개발자 과정 :  
청년취업아카데미

- 콘솔탭을 선택
- 콘솔탭을 enable시킨다
- 콘솔 입력 맨 오른쪽 끝 아래 show command editor 버튼을 클릭하여 입력창을 확대하여 연다



## 2. 자바스크립트 언어 핵심 구문

- JavaScript 컴파일 언어가 아니고 웹 브라우저가 해석해주는 인터프리터 언어이다
- JavaScript는 대소문자를 구별한다(case-sensitive 언어)!!
- JavaScript 문장은 한 줄에 보통 작성된다
- JavaScript는 여백이나 라인에 상관없다.

`alert("Hello world");` ✓  
`Alert("Hello world");` ✗

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello</title>
  </head>
  <body>
    <div>
      <p>regular body text</p>
    </div>
    <script>
      alert("hello");
    </script>
  </body>
</html>
```

web browser

# JavaScript를 웹페이지 어디에 둘 것인가?

simple.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Hello</title>
  </head>
  <body>
    <div>
      <p>regular body text</p>
    </div>
    <script src="myscript.js">
    </script>
  </body>
</html>
```

myscript.js

```
alert("hello");
```

← Location of Script Tag?

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

꼭 필요하지 않음

## 변수 Variables

- 변수는 값이나 표현(수식)을 저장하기 위해 사용된다
- 변수 생성: 대소문자 구분, 문자나 \_문자로 시작해야한다

```
var year;  
var customerEmail;  
var todaysDate;  
var foo;  
var x;  
var 99problems; ❌  
var problems99; ✔️
```

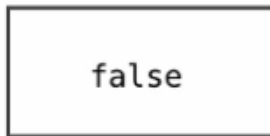
**letters**  
**numbers**  
—  
\$

```
var day, month, year;  
var day=20, month=10, year=2015;
```

## 변수 데이터 유형 Variable data types

- 일반 변수를 생성하기 위해 `var` 키워드를 사용하고 임의의 유형의 값을 할당하면 된다
- Built-in 데이터 유형들: Number, String, Boolean, and Array

```
var myVariable;
```



**myVariable**

```
myVariable = 200;  
myVariable = "Hello";  
myVariable = 'Hello';  
myVariable = true;  
myVariable = false;
```

```
var myVariable;
```



**myVariable**

**integer?**

**floating point number?**

**text string? boolean value? array? date? object?**

```
var name = new Array(2);  
name[0]="array1";  
name[1]="array2";
```

## 조건부 코드 작업 Working with conditional code

- 용어 : (parentheses), [brackets], {braces}
- 조건부 문장은 다른 조건에 기초한 다른 액션을 수행하기 위해 사용된다.

```
if ( condition )  
{  
    executed if condition is true... //code block goes here.  
}
```

```
if ( condition ) {  
    // code goes here  
    // ...  
} else {  
    // otherwise, different code  
    if ( ) {  
        // nested if  
    }  
}
```

```
Switch (n)  
{  
    case 1:  
        execute code block 1;  
        break;  
    case 2:  
        execute code block 2;  
        break;  
    default:  
        execute if n is not equal to 1 or 2  
}
```

## 연산자 사용

- 산술연산자 Arithmetic operators: +, -, /, \*,
- 할당 Assignment : =, +=, -=, \*=, /=
- Equality : ==, !=
- Strict equality : ===
- 모듈로 연산 Modulus : %
- Increment/decrement: ++, --
- 논리 연산자 Logical and/or: &&, ||
- Ternary: condition ? true : false

## 논리 비교 와 유니리 연산자 Logical comparisons/unary operators

- 비교와 증감(decrement/increment)

유형과값까지  
일치 ===

```
if (a == b) { ...  
if (a != b) { ...  
if (a === b) { ...  
if (a !== b) { ...  
if (a > b) { ...  
if (a < b) { ...  
if (a >= b) { ...  
if (a <= b) { ...
```

```
a = a + 1;
```

```
a += 1;
```

```
a++;
```

```
++a;
```

```
a = a - 1;
```

```
a -= 1;
```

```
a--;
```

```
--a;
```



## 터너리 연산자 Ternary

- Variablename = condtion ? True:false
- 터너리 연산 예제

```
var playerOne = 500;  
var playerTwo = 600;
```

```
// alternatively... condition ? true : false  
var highScore = (playerOne > playerTwo) ? playerOne :
```

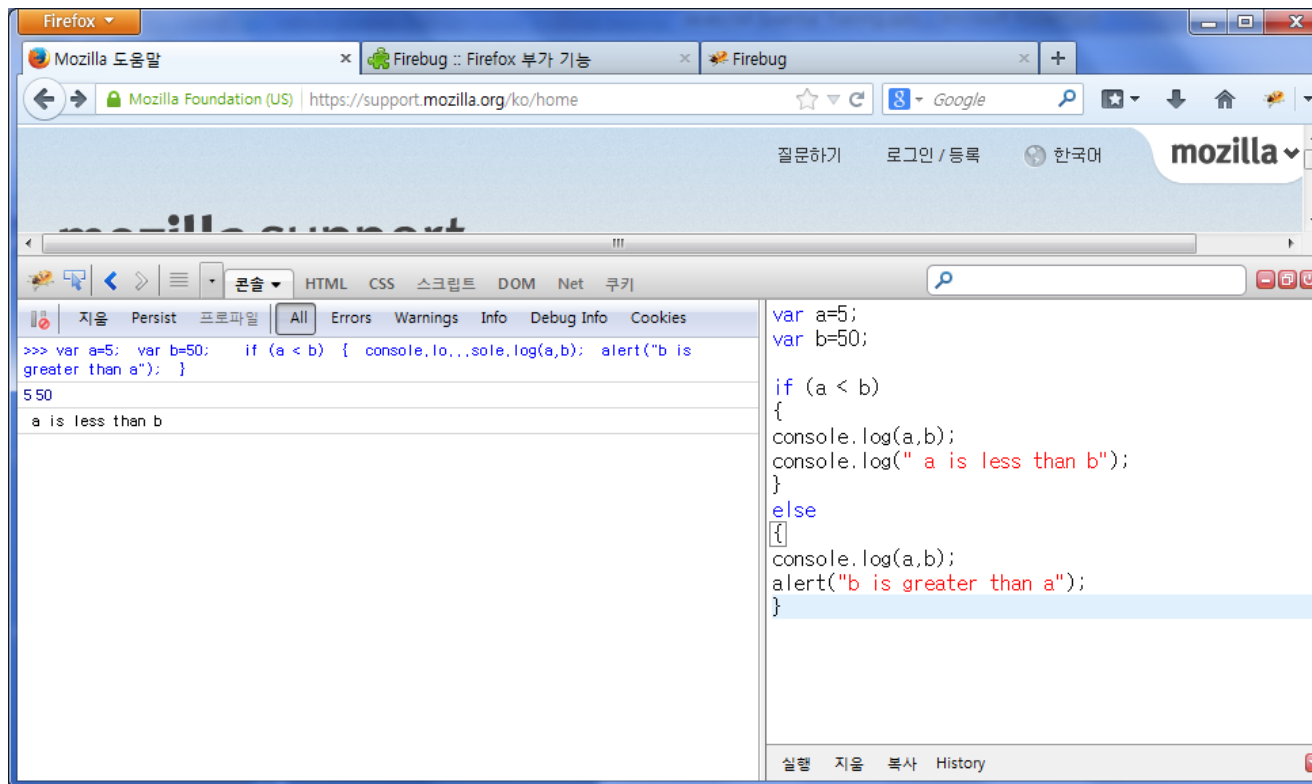
```
if (playerOne > playerTwo) {  
    highScore = playerOne;  
}  
else {  
    highScore = playerTwo;  
}
```

Ternary 연산자

```
→ // alternatively... condition ? true : false  
var highScore = (playerOne > playerTwo) ? playerOne : playerTwo ;
```

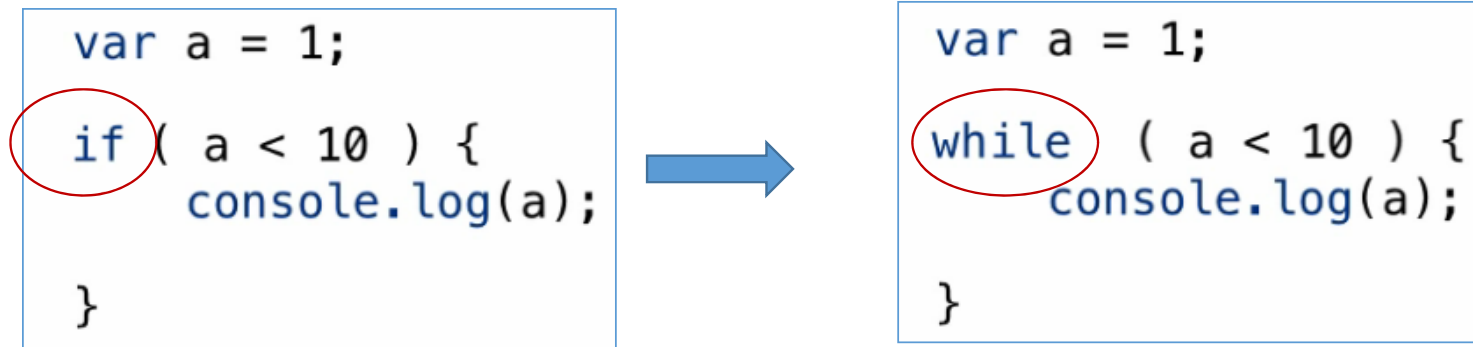
## 콘솔에 메시지 보내기

- firebug extension 사용하기 (in firefox browser)
- 기본 함수들 : console.log(), console.warn(), console.error(), console.info()



## 루핑 : 반복 Working with loops

- 루핑 문장은 한 블록의 코드를 지정된 횟수만큼 혹은 지정된 조건이 사실인 경우 반복을 수행하는 것이다. 여기에는 아래 4가지 방법이 있다.
  - For loop
  - Do ... while
  - For ... in
  - While loop



## Do... while loop

- The Do... while loop는 한블럭의 코드를 한번 수행하난 뒤 지정된 조건이 참인 경우만 루핑이 반복된다.

```
var a = 1;  
  
do {  
    console.log(a);  
    a++;  
} while ( a < 10);
```

- For loop

	setup index	check condition	increment index
for (	var i = 1	; i < 10	; i++ ) {
	// do stuff		
	// do stuff		
	// do stuff		
	// etc..		
}			

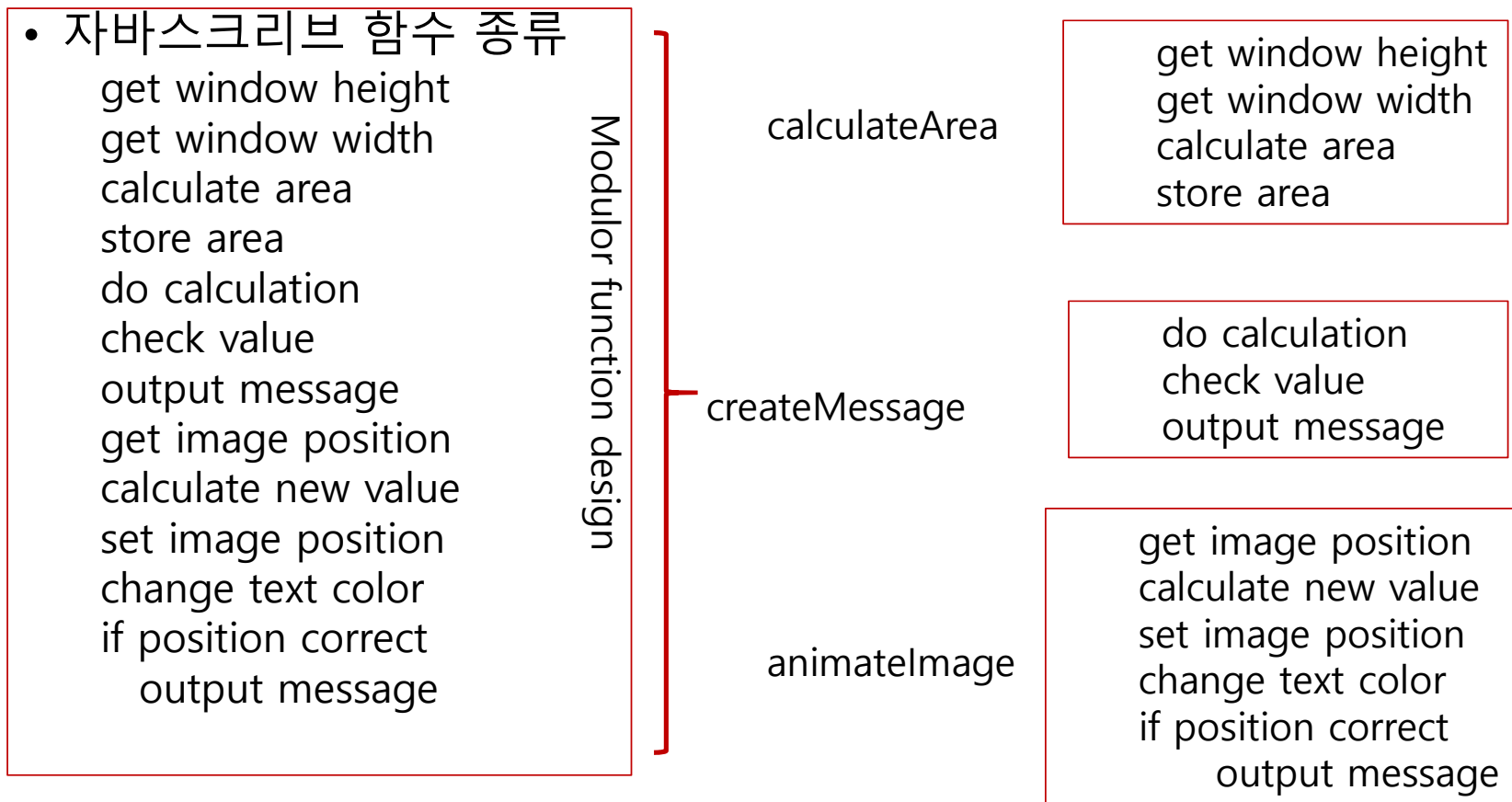
## Break and continue 문장

- The break and continue 문장은 루프내에서 흐름 제어를 위해 사용되어 질 수 있다.
- Break 문장
  - break 문장은 루프를 빠져 나와서 루프 이후의 코드 블록을 계속 수행한다
- Continue 문장
  - The continue 문장은 현재 루프를 빠져나와 다음 값을 가지고 계속 진행한다.

```
for ( var i = 1 ; i < 5000 ; i++ ) {  
    // do stuff  
    // do stuff  
    if (i % 5 == 0) {  
        continue; // done with this iteration  
    }  
    // do second set of stuff  
    // do second set of stuff  
}
```

# 함수 생성 Creating functions

함수란 이벤트 발생 혹은 함수 호출에 의해 어떤 임무를 수행하는 코드 블록 묶음이다



## 함수 정의 Defining a Function

- 구문 Syntax:
  - `function` functionName(var1,...varN)
  - {.....some code...;}
- 자바스크립트에서의 함수 예

```
var year = 2014, month = 01, day = 03;
function welcomeAlert()
{ alert("Welcome");
}
function printDate()
{ welcomeAlert();
  alert(Date());
  // loops, if statements, anything!
}
//sometime later
printDate();
```

- 파라미터를 갖는 함수

```
var year = 2014, month = 01, day = 03; ← global variable
function computeDays(yr, mo, dy)
{ var days = yr * 365 + mo * 30 + dy;
  return days; // optional
  alert(days);
}
computeDays(year, month, day);
var days = computeDays(year, month, day);
alert(days);
```



- 파라미터 수가 일치하지 않는 경우 처리

```
function calculateLoan(amount, months, interest, name)
{
  // lots of code
}
```

```
calculateLoan(10000, 60, 7, "홍길동"); // correct
calculateLoan(10000, 60, 7, "홍길동", "추가사항"); // extras
are ignored
calculateLoan(10000, 60); // missing are passed "undefined"
```

- 변수 인식 범위 : local vs global

```
function simpleFunction() {  
    // lots of code  
    var foo = 500; ← local variable  
    // lots of code  
    console.log(foo); 500  
}  
  
simpleFunction();  
console.log(foo); undefined
```

## Break

1. 배 자 나 너 표 나 버  
2. 드 르 모 표 사람 사람  
3. 오 등 사람 모 7 E 오  
4. 오 오 사람 사람 오 오 너  
5. 오 7 등 나 모 모 너

이게 무슨 암호일까요?  
5분안에 풀면 수제  
다 풀면 치매는 절대 안 걸린대요

### 3. 데이터 유형과 객체

- 기본 배열 Array : `var mutli = []; multi[0]=50; multi[1]=60; multi[2]="data"; multi[3]=true;`
  - 구문: `var varname = new Array();` //배열 객체
  - 구문 : `var varName = Array();`
- 결합 배열 Associated Array:
  - 구문 `array={key1:'value1', key2:'value2'};`
- 다차원 배열 Multidimensional Array
  - Syntax `var Name = new type[size1][size2];`
- 다차원 배열 작업 Working with multi-dimensional arrays

#### Output

Red, Yellow, Green  
Table, Chair, Fan, Bulb, TV  
JAVA, C#, CSS, HTML, .NET



```
<script language="javascript">
var Array = new Array(
["Red", "Yellow", "Green"],

["Table", "Chair", "Fan", "Bulb", "TV"],
["JAVA", "CSS", "HTML", ".NET"]);

document.write(Array[0]+"<br />");

document.write(Array[1]+"<br />");

document.write(Array[2]+"<br />");

</script>
```

```
var singleValue;  
singleValue = 99;
```

Creating array

singleValue

99

```
var multipleValues = [];
```

```
multipleValues[0] = 50;
```

```
multipleValues[1] = 60;
```

```
multipleValues[2] = "Mouse";
```

multipleValues

[0]

50

[1]

60

[2]

Mouse

**zero-based index**

```
console.log(multipleValues[2]);
```

```
var multipleValues = [ 50,60,"Mouse" ];
```

[0] [1] [2]

**zero-based**

## 배열 메소드

- 배열 메소드 Array methods

someFunction(params); //to call a function

### -메소드란 객체에 속한 함수

someObject.someMethod();// to call a mehod

-ex:

```
var multArray =[10, 20, 30, 40, 50];
```

```
var reverseArray =multArray.reverse();
```

```
.multArray.sort();
```

```
.join();
```

```
console.log(reverseArray.join() );// "50,40,30,20,10"
```

배열 객체의  
내장 배열 메소드들



## Arrays 는 모든 곳에!

- `document.getElementsByTagName("TagName")`

```
var myArrayOfLinks = document.getElementsByTagName("a");
```

<a /> 태그로 액세스된  
배열 링크

myArrayOfLinks	
[0]	<code>&lt;a href="somepage..."</code>
[1]	<code>&lt;a href="otherpage..."</code>
[2]	<code>&lt;a href="http://..."</code>
[3]	<code>&lt;a href="http://...."</code>

## Numbers로 작업

- Not a Number(NaN)

```
var foo = "55"; // could be "abc"

var myNumber = Number(foo); // make it a number

// double negative - if NOT NOT a number
if ( !isNaN(myNumber) ) {
    console.log("It IS a number");
}
```



- Addition vs Concatenation

```
var foo = 5;  
var bar = 5;  
console.log(foo + bar); // 10
```

← Number

```
var foo = "5";  
var bar = "5";  
console.log(foo + bar); // 55
```

← String

- Math 객체 사용하기

```
var x = 200.6;  
var y = Math.round(x); // 201  
  
var a = 200, b = 10000, c = 4;  
var biggest = Math.max(a,b,c);  
var smallest = Math.min(a,b,c);  
  
Math.PI    Math.random()    .sqrt()    .log()
```

## Strings로 작업

```
var myString = "Double quotes work."; ✓
```

```
var myString = 'Single quotes work too.'; ✓
```

```
var myString = "One or the other."; ✗
```

```
var phrase = 'Don't mix your quotes.'; ✗
```

```
var phrase = "Don't mix your quotes."; ✓
```

```
var phrase = "He said "that's fine," and left."; ✗
```

```
var phrase = "He said \"that's fine,\" and left."; ✓
```

## String 속성 과 메소드(1/)

- 스트링 속성

```
var phrase = "This is a simple phrase.";
console.log(phrase.length); // 24
```

- 스트링 메소드

```
var phrase = "This is a simple phrase";
console.log(phrase.toUpperCase());
var words = phrase.split(" ");
```

```
var position = phrase.indexOf("simple"); // 10
```

```
// it returns -1 if the term is not found
if ( phrase.indexOf("DDDD") == -1) {
    console.log("That word does not occur.");
}
```

```
// there is also .lastIndexOf()
```

0	This
1	is
2	a
3	simple
4	phrase.

## String 속성 과 메소드(2/)

```
var phrase = "Yet another phrase";  
var segment = phrase.slice(6,11); //other
```

`.substring(start,end)`     `.substr(start,length)`

```
var str1 = "Hello";  
var str2 = "hello";  
// str1 != str2  
if ( str1.toLowerCase() == str2.toLowerCase() ) {  
    console.log("Yes, equal");  
}
```

```
var str1 = "aardvark";  
var str2 = "beluga";  
if (str1 < str2 ) { ...    // true
```

```
var str1 = "aardvark";  
var str2 = "Beluga";  
if (str1 < str2 ) { ...    // false!  
ABCD... less than abcd...
```

- 자바스크립트 문서에 관련된 참조 사이트 → 간단한 소개
  - <https://developer.Mozilla.org/en/JavaScript/Reference>

## Dates로 작업하기

```
var today = new Date(); // 현재 날짜와 시간
//year, month, day
var y2k = new Date(2016, 0, 1);
//year, month, day, hours, minutes, seconds
var y2k = new Date(2016,0,1,0,0,0);
document.write(y2k + "<br />");// Fri Jan 01 2016 00:00:00 GMT+0900 (대한민국 표준시)
```

```
var today = new Date();
```

```
today.getMonth(); // returns 0-11
```

```
today.getFullYear(); // YYYY (not zero-based)
```

```
today.getFullYear(); // deprecated
```

```
today.getDate(); // 1-31 day of month
```

```
today.getDay(); // 0-6 day of week. 0 == sunday
```

```
today.getHours(); // 0-23
```

```
today.getTime(); // milliseconds since 1/1/1970
```

- Date 객체의 SET 메소드
- Date 객체 비교

```
var today = new Date();  
  
today.setMonth(5);  
today.setFullYear(2012);  
today.setDay(0);
```

```
var date1 = new Date(2000,0,1);  
var date2 = new Date(2000,0,1);  
  
if ( date1 == date2 ) { ... // false!  
  
if ( date1.getTime() == date2.getTime() ) { ...  
    949388400000 == 949388400000  
    // true!
```



## 객체(Objects)로 작업하기

- 자바스크립트의 객체들

```
var myArray = [10,20,30,"Forty",50];  
console.log(myArray.length);
```

0	10
1	20
2	30
3	"Forty"
4	50

```
var todaysDate = new Date();  
var ms = todaysDate.getTime();
```

- 자바스크립트의 객체 생성

```
var playerName = "Fred";  
var playerScore = 10000;  
var playerRank = 1;
```

Fred	playerName
10000	playerScore
1	playerRank
player	

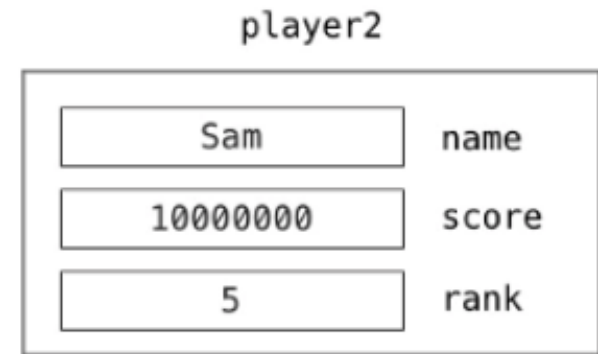
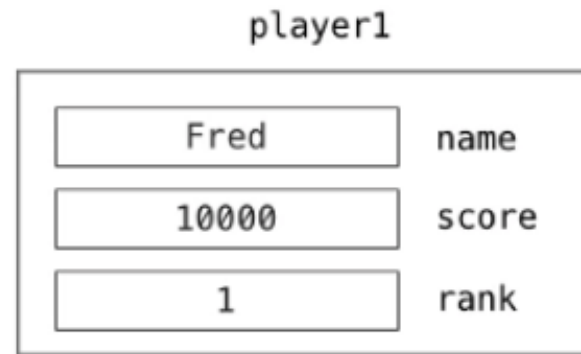
- 자바스크립트의 객체 생성  
(generic object container)

```
var player = new Object();  
player.name = "Fred";  
player.score = 10000;  
player.rank = 1;
```

properties	
Fred	name
10000	score
1	rank

- 자바스크립트의 객체  
생성: 결합 배열 방식

```
var player1 = { name: "Fred", score: 10000, rank: 1 };  
var player2 = { name: "Sam", score: 10000000, rank: 5 };
```



예제: Chapter 4> jsbin.tisetat.1.html

## Break

- 누가 더 큰소리로 우는가?



암컷 Natterjack 두꺼비는 수컷이 얼마나 큰 소리를 낼 수 있는지에 근거를 두고 짝짓기 상대를 결정한다. 큰 소리는 큰 수컷에 의해서만 만들어질 수 있으므로 작은 수컷은 속일 수 없다. 따라서 큰 소리는 정확한 정보를 운반하는 정직한 신호이다.

## 4. 문서 객체 모델(DOM) 이해하기

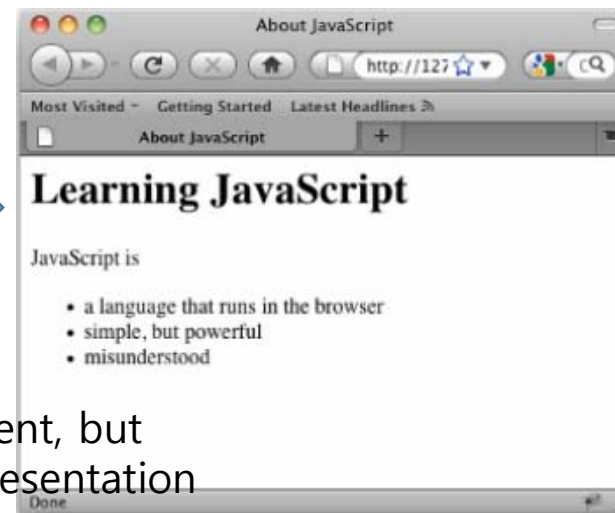
### • 4.1 DOM 이란?

문서 객체 모델(Document Object Model) 은 HTML 혹은 XML과 같은  
마크업 언어로 프로그램적으로 작업하는 표준 API이다.

DOM 은 기본적으로 문서의 객체 지향 표현이다.

HTML의 각 문서는 문서의 DOM 트리안에서 객체로 표현된다.

```
<html>
  <head>
    <title>About JavaScript</title>
  </head>
  <body>
    <h1>Learning JavaScript</h1>
    <p>JavaScript is: </p>
    <ul>
      <li>a language that runs in the browser</li>
      <li>simple, but powerful</li>
      <li>misunderstood</li>
    </ul>
  </body>
</html>
```



Same document, but  
Different representation

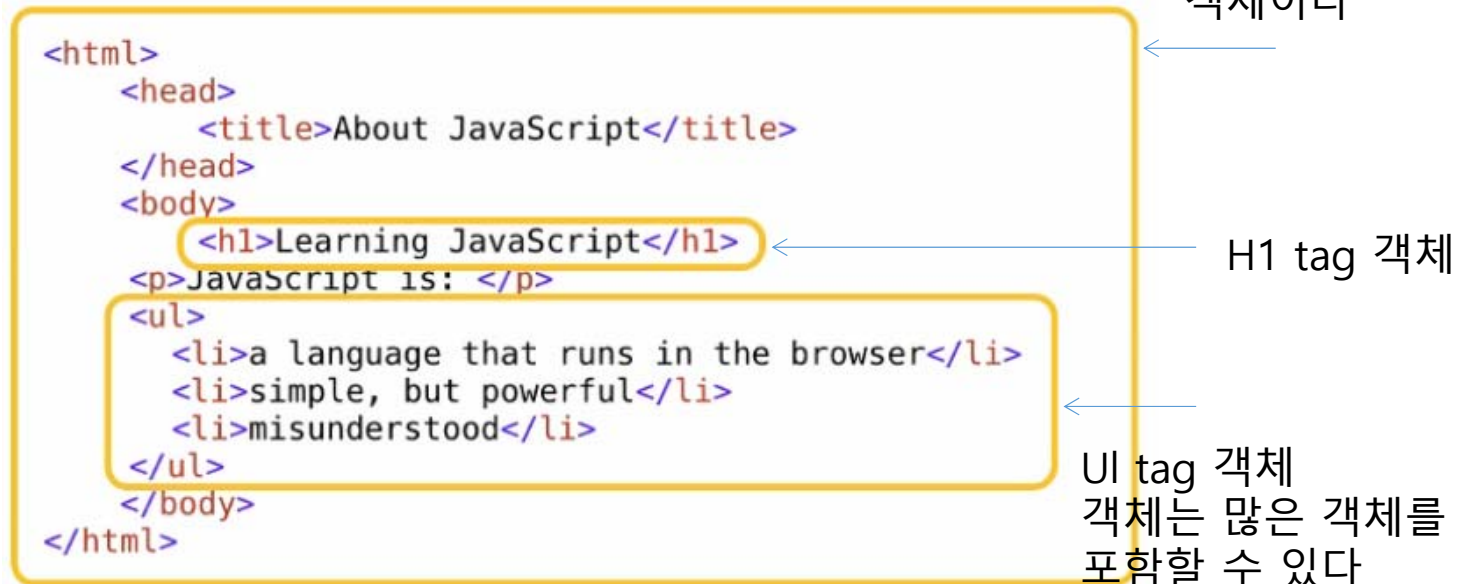
# DOM이란 ? What is the DOM?

- 객체란 무엇인가? →

객체란 간단히 말해서 어떤 것을 말하는데 다  
른 어떤 것도 포함될 수 있는 개별 엔터티로  
취급될 수 있는 의미를 갖는 어떤 것을 나타  
낸다.

- 객체는 속성과 메소드를 갖는다.

전체 문서도  
객체이다



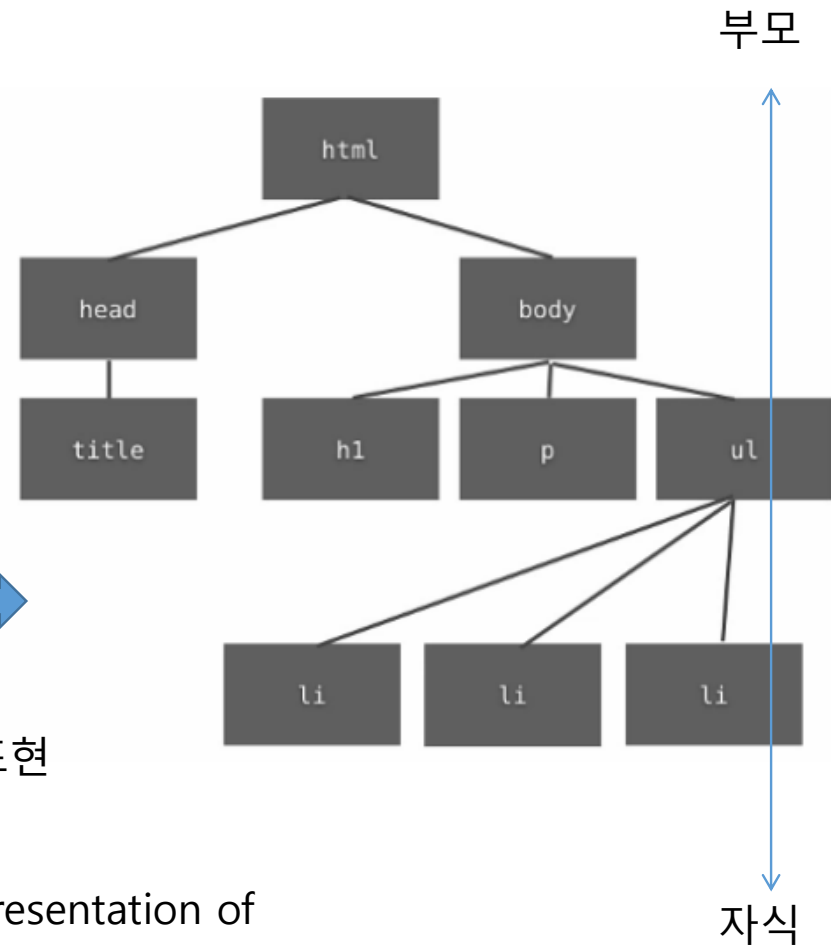
## • 모델이란 무엇인가?

노드간의 관계

```
<html>
  <head>
    <title>About JavaScript</title>
  </head>
  <body>
    <h1>Learning JavaScript</h1>
    <p>JavaScript is: </p>
    <ul>
      <li>a language that runs in the browser</li>
      <li>simple, but powerful</li>
      <li>misunderstood</li>
    </ul>
  </body>
</html>
```



계층적 관계 표현  
Hierarchical  
(or structural)  
Relational representation of  
The document

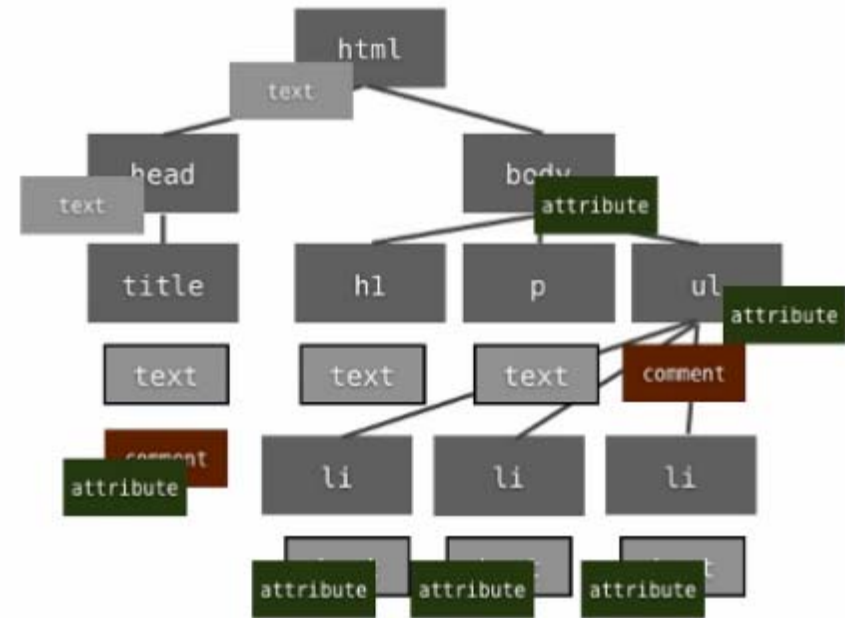


- DOM으로 무엇을 할 수 있는가?
  - **Get the title text**
  - **Get the second paragraph**
  - **Get the third link in the menu and set its CSS to display:none**
  - **Change the background color of all paragraphs with a class of "important"**
  - **Get all the <li> elements in the last unordered list**
  - **Find the image with an id of "logo" and move it 40 pixels to the right**
  - **Change a link so it performs a JavaScript function when clicked**
  - **Create a new unordered list and insert between first and second paragraphs**

## 4.2 노드와 원소로 작업하기

- 노드와 원소 Nodes and Elements
  - 가장 작은 원소라 할지라도 많은 노드를 갖는다.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Come Explore California</title>
<link href="_css/main.css" rel="stylesheet" type="text/css" />
</head>
<!-- here is a comment -->
<body id="home">
<div id="wrapper">
  <div id="header">
    <h1><a href="index.htm">Welcome to Explore California!</a></h1>
  </div>
  <div id="mainNav">
    <ul id="homeNav">
      <li><a href="tours.htm" title="Explore our tours" class="tours">Tours</a></li>
      <li><a href="mission.htm" title="What we think" class="mission">Mission</a></li>
      <li><a href="contact.htm" title="Contact and support" class="contact">Contact
a></li>
      <li><a href="resources.htm" title="Guidance and planning"
class="resources">Resources</a></li>
      <li><a href="explorers.htm" title="Join our community"
class="explorers">Explorers</a></li>
    </ul>
  </div>
</body>
</html>
```





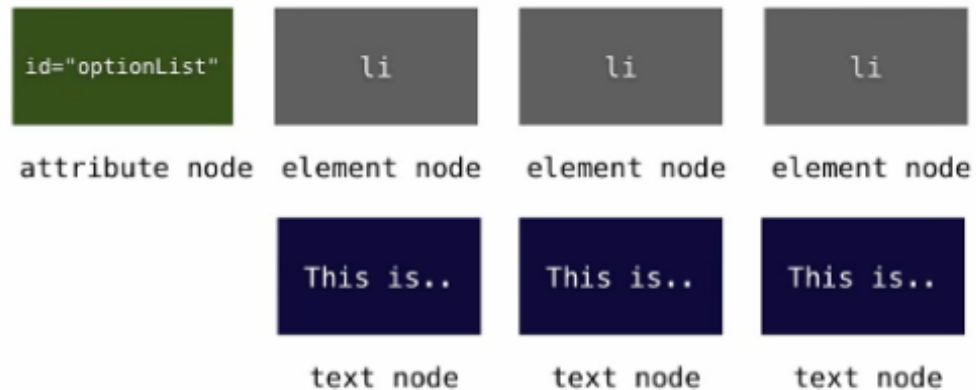
- 노드 유형 : 공식적으로는 12개의 노드 유형 존재, 하지만 3개의 노드가 대부분 사용된다.

```
Node.ELEMENT_NODE == 1
Node.ATTRIBUTE_NODE == 2
Node.TEXT_NODE == 3
Node.CDATA_SECTION_NODE == 4
Node.ENTITY_REFERENCE_NODE == 5
Node.ENTITY_NODE == 6
Node.PROCESSING_INSTRUCTION_NODE == 7
Node.COMMENT_NODE == 8
Node.DOCUMENT_NODE == 9
Node.DOCUMENT_TYPE_NODE == 10
Node.DOCUMENT_FRAGMENT_NODE == 11
Node.NOTATION_NODE == 12
```

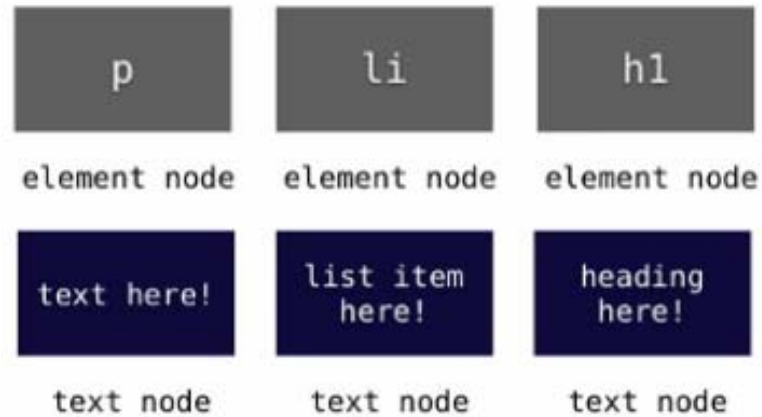
```
Node.ELEMENT_NODE == 1
Node.ATTRIBUTE_NODE == 2
Node.TEXT_NODE == 3
```

- 원소, 속성, 텍스트 노드 Elements, Attributes, and Text Nodes

```
<ul id="optionList">  
  <li>This is the first option</li>  
  <li>This is the second</li>  
  <li>This is the third</li>  
</ul>
```



- 원소 노드는 텍스트를 포함하지 않는다

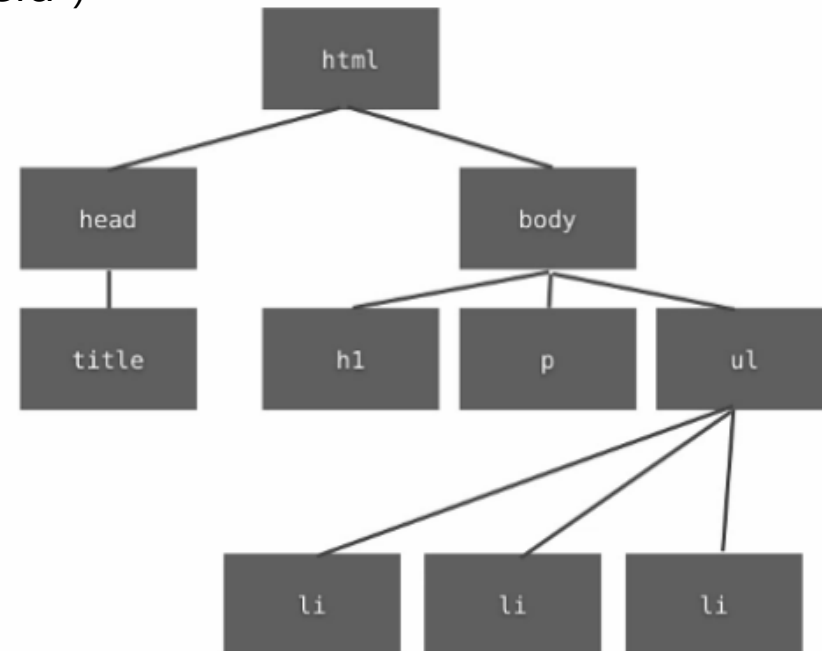


## 5. 문서 객체 모델에서의 작업 Working with the Document Object Model

- DOM 원소 접근 : 원소 노드를 어떻게 얻는가?

Document.getElementById("someId")

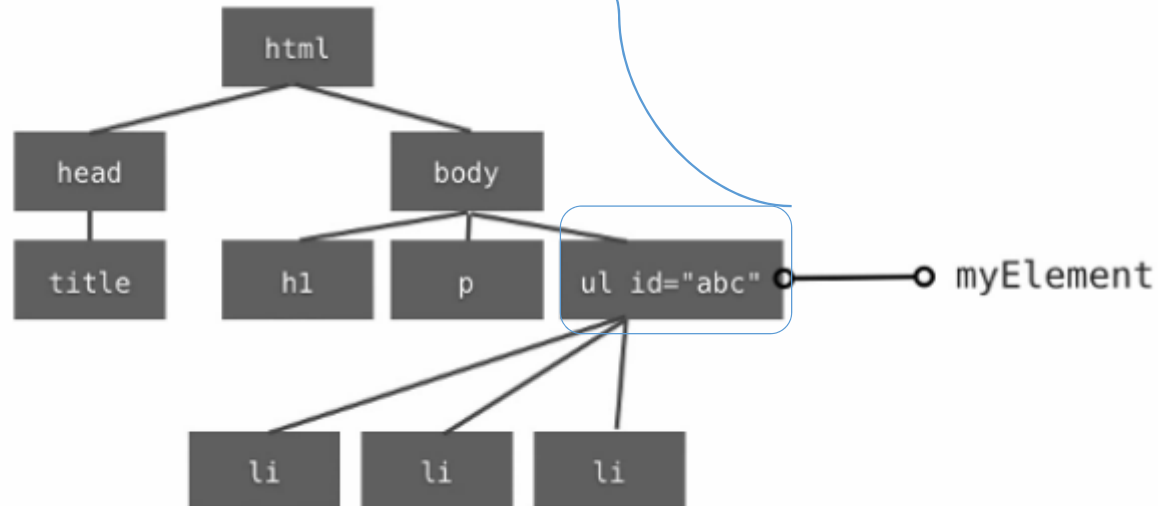
```
<html>
  <head>
    <title>About JavaScript</title>
  </head>
  <body>
    <h1>Learning JavaScript</h1>
    <p>JavaScript is: </p>
    <ul>
      <li>a language that runs in the browser</li>
      <li>simple, but powerful</li>
      <li>misunderstood</li>
    </ul>
  </body>
</html>
```



- Id에 의해 원소 가져오기 Retrieving an element by id

```
var myElement = document.getElementById("abc")
```

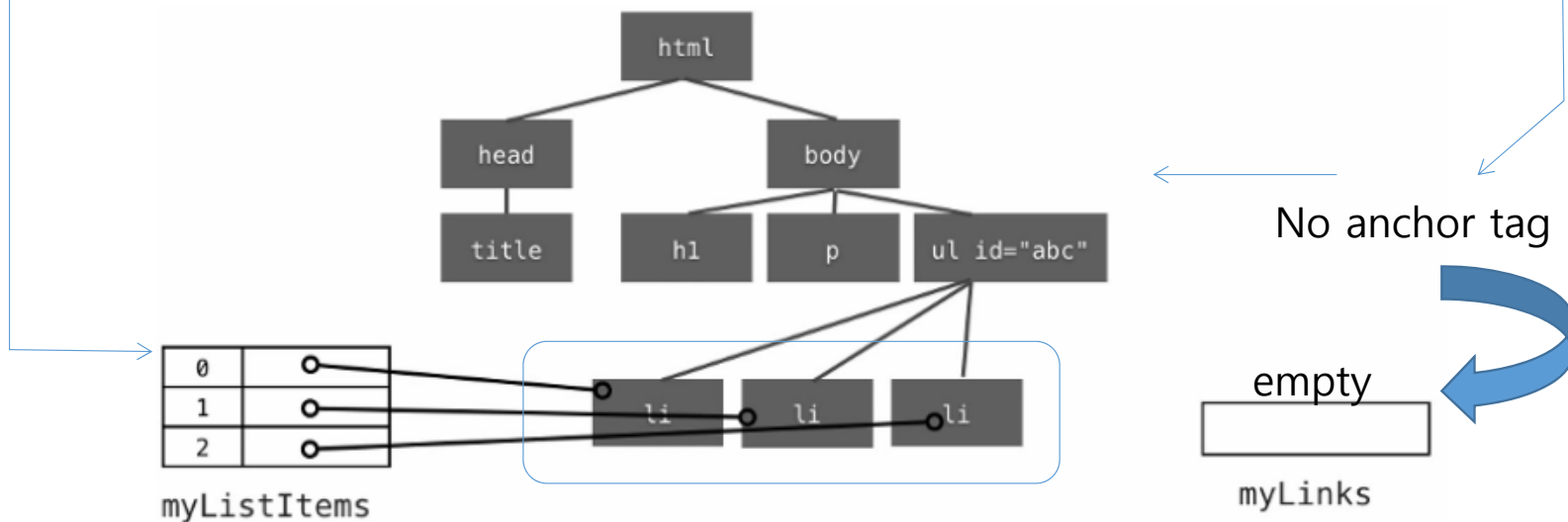
myElement is  
a handle to that place  
in the DOM



- 태그 이름으로 원소 가져오기 Retrieving an element by Tag Name

```
var myListItems = document.getElementsByTagName("li")
```

```
var myLinks = document.getElementsByTagName("a")
```



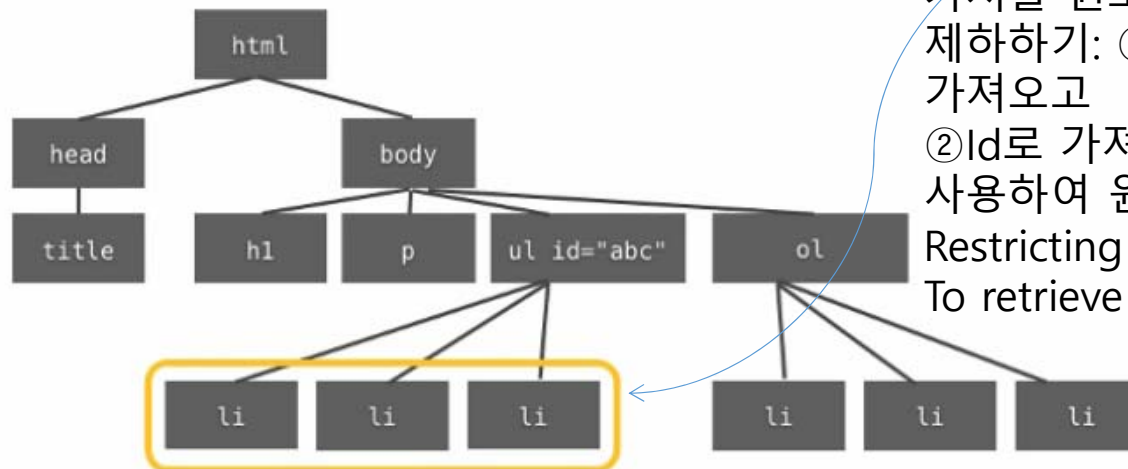
## 실습 practices

```
var mainTitle = document.getElementById("mainTitle");  
console.log("This is an element of type: ",  
    mainTitle.nodeType);  
console.log("The inner HTML is ", mainTitle.innerHTML);  
Console.log("Child nodes: ", mainTitle.childNodes.length);
```

```
var myLinks = document.getElementsByTagName("a");  
console.log("Links: ", myLinks.length);
```

- 가져올 원소 범위를 제한 하기 Restricting elements to retrieve

→ ① `var myFirstList = document.getElementById("abc");`  
② `var limitedList = myFirstList.getElementsByTagName("li");`

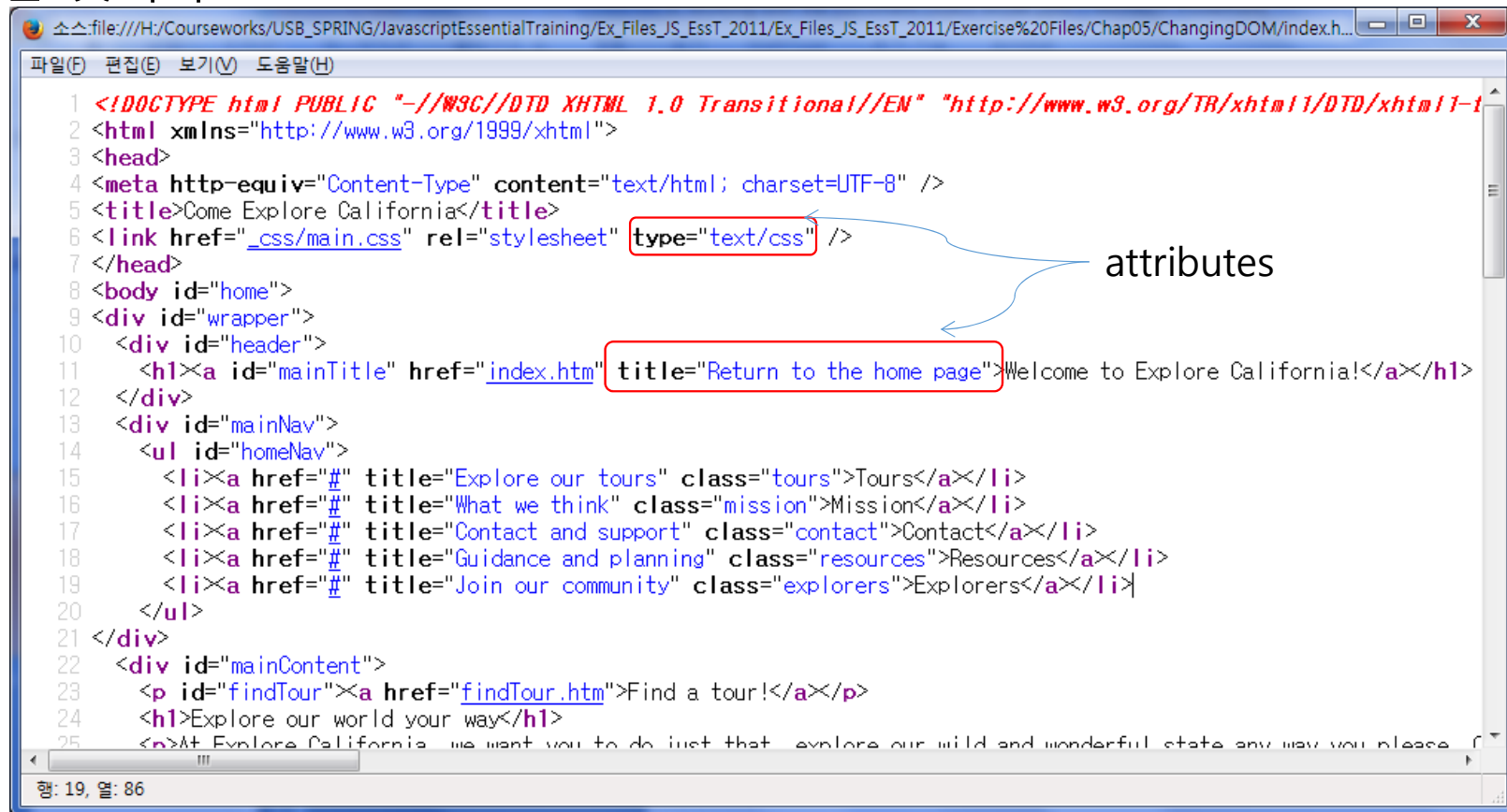


가져올 원소를  
제한하기: ① 먼저 id로  
가져오고  
② id로 가져온 변수를  
사용하여 원소를 가져온다  
Restricting elements  
To retrieve



## DOM 원소 변경 Changing DOM elements

- 취할 수 있는 가장 간단한 변경중의 하나는 원소의 속성을 변경하는 것이다.



The screenshot shows a web browser window displaying an HTML document. The document's source code is visible, and two specific attributes are highlighted with red boxes and arrows pointing to the word "attributes".

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-1
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5 <title>Come Explore California</title>
6 <link href="_css/main.css" rel="stylesheet" type="text/css" />
7 </head>
8 <body id="home">
9 <div id="wrapper">
10 <div id="header">
11 <h1><a id="mainTitle" href="index.htm" title="Return to the home page">Welcome to Explore California!</a></h1>
12 </div>
13 <div id="mainNav">
14 <ul id="homeNav">
15 <li><a href="#" title="Explore our tours" class="tours">Tours</a></li>
16 <li><a href="#" title="What we think" class="mission">Mission</a></li>
17 <li><a href="#" title="Contact and support" class="contact">Contact</a></li>
18 <li><a href="#" title="Guidance and planning" class="resources">Resources</a></li>
19 <li><a href="#" title="Join our community" class="explorers">Explorers</a></li>
20 </ul>
21 </div>
22 <div id="mainContent">
23 <p id="findTour"><a href="findTour.htm">Find a tour!</a></p>
24 <h1>Explore our world your way</h1>
25 <p>At Explore California, we want you to do just that... explore our wild and wonderful state any way you please.
```

Attributes highlighted in the image:

- `type="text/css"` in the `<link>` tag (line 6).
- `title="Return to the home page"` in the `<a>` tag (line 11).

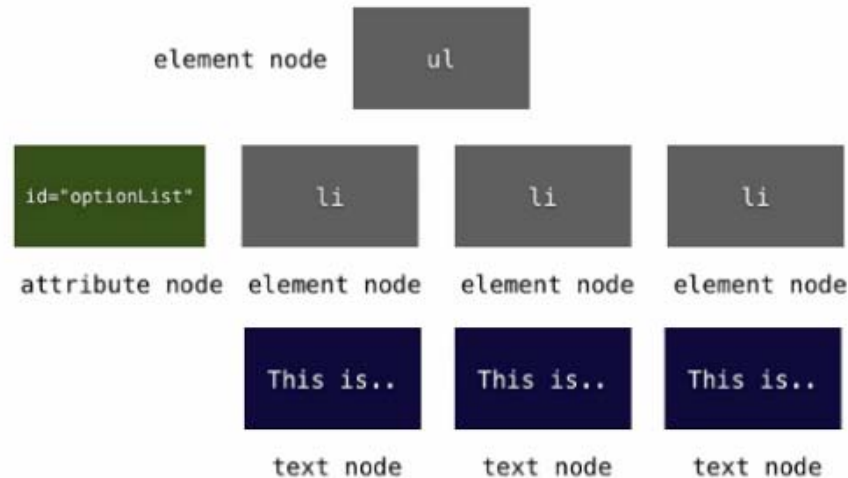
## • 속성 작업하기 Working with attributes

**name - in quotes**

```
myElement.getAttribute("align");
```

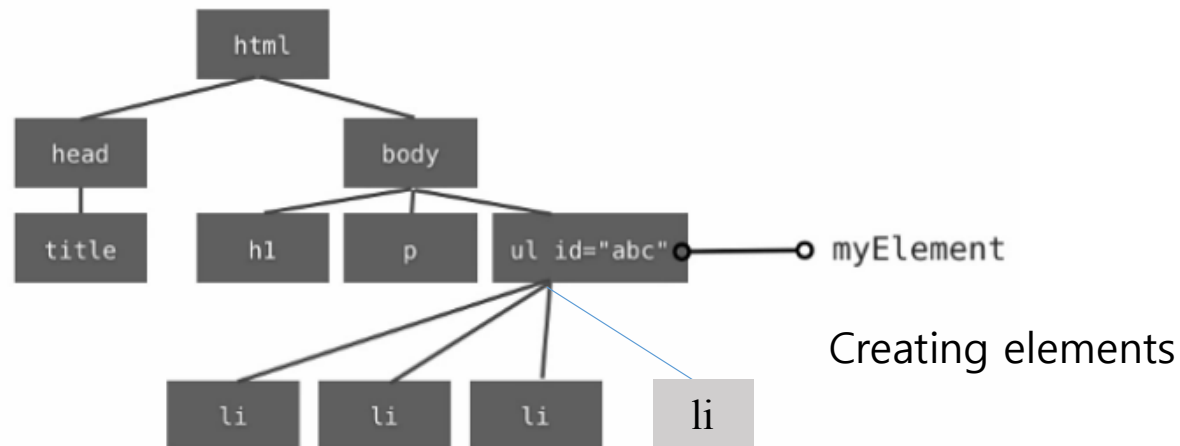
**name      value**

```
myElement.setAttribute("align", "left");
```



## DOM 원소 생성 Creating DOM elements

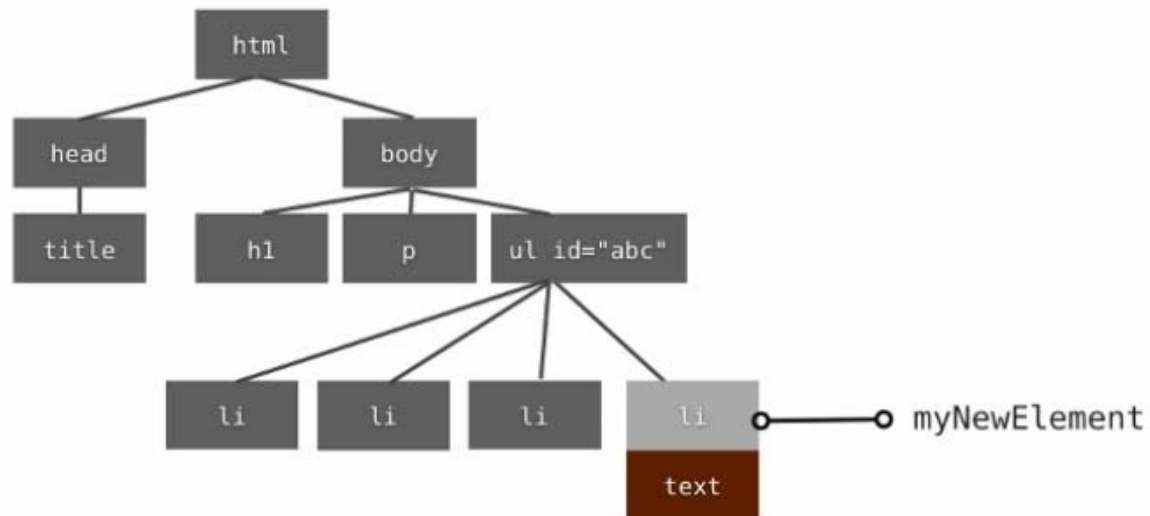
- Step 1 : 원소를 생성한다 create the element
- Step 2 : 문서에 추가한다 add it to the document



```
var myElement = document.getElementById("abc");  
var myNewElement = document.createElement("li");  
myElement.appendChild(myNewElement);  
myNewElement.innerHTML = "New Item Text";
```

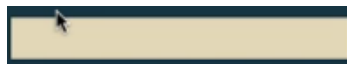
## • 텍스트 노드 원소 생성 Creating Text Node Elements

```
var myText = document.createTextNode("New list item text");  
myNewElement.appendChild(myText);
```



## 실습 A Practice A

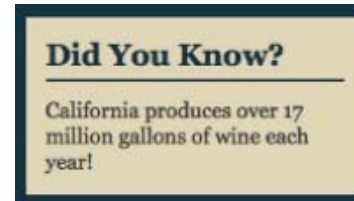
```
<div id="trivia" class="callOut">  
  <!-- new content should go here -->  
</div>
```



Blank area



Script.js



```
1 //create the elements  
2 var newHeading = document.createElement("h1");  
3 var newParagraph = document.createElement("p");  
4  
5 // to add content, either use innerHTML  
6 newHeading.innerHTML = "Did You Know?";  
7 newParagraph.innerHTML = "California produces over 17 million gallons of wine each year!";  
8  
9 // OR create child nodes manually  
10 //var hlText = document.createTextNode("Did You Know?");  
11 //var paraText = document.createTextNode("California produces over 17 million gallons of wine each year!");  
12 // and add them as child nodes to the new elements  
13 //newHeading.appendChild(hlText);  
14 //newParagraph.appendChild(paraText);  
15  
16 // and we still need to attach them to the document!  
17 document.getElementById("trivia").appendChild(newHeading);  
18 document.getElementById("trivia").appendChild(newParagraph);  
19
```

## 실습 B Practice B

- 텍스트 노드 원소 생성 사용하기 Using Creating text node element

```
//create the elements
var newHeading = document.createElement("h1");
var newParagraph = document.createElement("p");

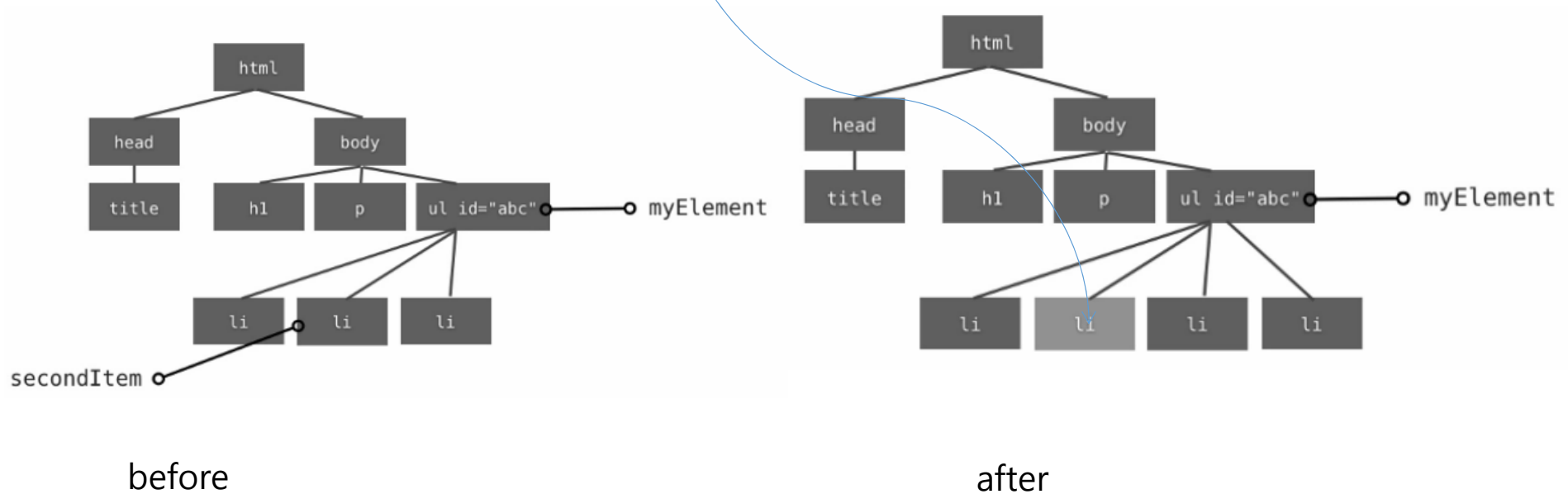
// to add content, either use innerHTML
//newHeading.innerHTML = "Did You Know?";
//newParagraph.innerHTML = "California produces over 17 million gallons of wine each year!";

// OR create child nodes manually
var h1Text = document.createTextNode("Did You Know?");
var paraText = document.createTextNode("California produces over 17 million gallons of wine each year!");
// and add them as child nodes to the new elements
newHeading.appendChild(h1Text);
newParagraph.appendChild(paraText);

// and we still need to attach them to the document!
document.getElementById("trivia").appendChild(newHeading);
document.getElementById("trivia").appendChild(newParagraph);
```

## • Appendchild에 대한 대안들 Alternatives to appendchild

```
var myElement = document.getElementById("abc");  
var myNewElement = document.createElement("li");  
var secondItem = myElement.getElementsByTagName("li")[1]  
myElement.insertBefore( myNewElement, secondItem);
```



## Break

유전자 암호의  
손상:

손상의 다양한 원  
인과 효과

