

JavaScript의 발전과 jQuery 예제

– jQuery UI 기능 확장, jqGrid 서버 연동 예제 –

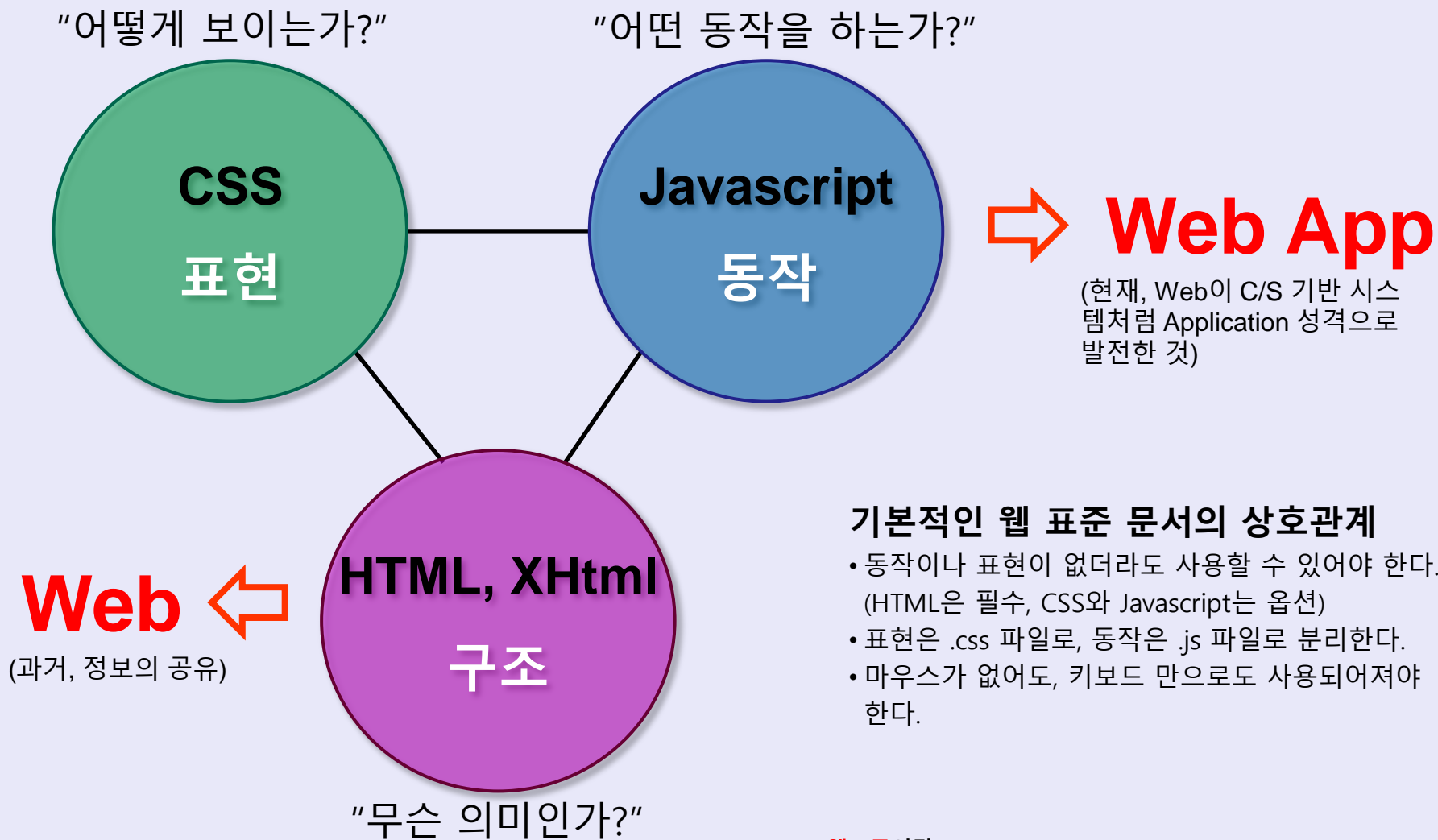
1. 웹 개발과 JavaScript
2. jQuery 와 jQueryUI
3. jQuery UI Tab 기능 확장
4. jqGrid - Spring jsonView 연동
5. 기타

이완근

2013.06.26

1. 웹 개발과 Javascript

웹 페이지 개발은 W3C에서 규정한 **웹표준**을 따라 웹 문서의 **구조와 표현 그리고 동작**을 구분해서 사용해야 한다.(관심사의 분리)



※ **웹표준**이란?

사용자가 어떠한 브라우저를 이용하더라도 같은 결과를 볼 수 있도록 W3C에서 권고한 표준안을 이용하여 웹 페이지를 만드는 것

웹 브라우저에서 동작하는 객체 기반의 **스크립트 프로그래밍 언어**^[1]다
(인터프리터어, 컴파일(compile)을 하지 않고, 작성해서 바로 실행시킬 수 있는 언어)

역사

- Brendan Eich, **넷스케이프 사** (現 Firefox)
- Mocha > LiveScript > JavaScript. 1995.12 넷스케이프 2.0B 3 채택
(넷스케이프 네비게이터 브라우저에 자바 기술 지원을 포함하면서 이름 변경.
썬마이크로시스템사가 공동 개발)

특징

- Compile을 하지 않고 소스를 읽어서 바로 실행하는 인터프리터 언어
(코드 재배치나 최적화 불가, 브라우저에서 소스 노출)
- C 언어의 기본 구문을 바탕으로 만들어 자바와 구문(Syntax)이 유사함
(자바 언어와 직접적인 관련 없음)

스펙

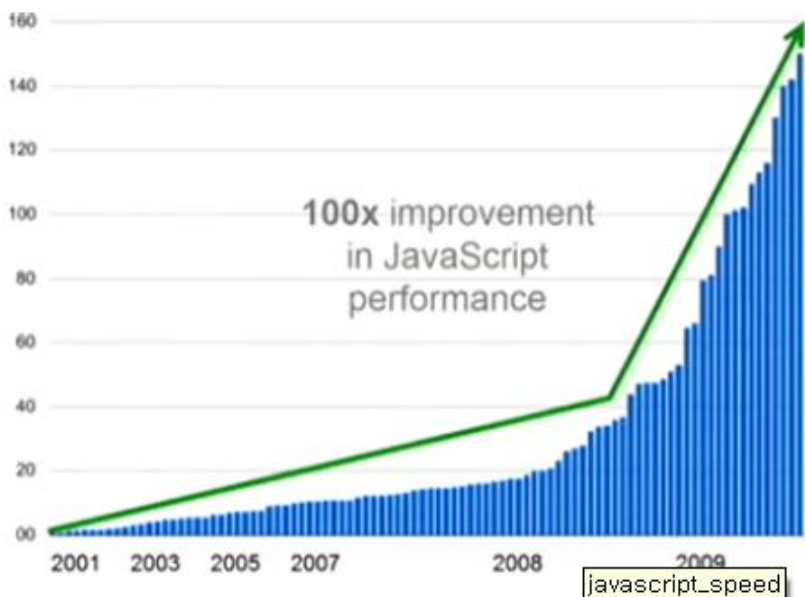
- 최신 버전 1.8.5 (2013.01 기준), Firefox 3 지원
- 자바스크립트는 브라우저마다 지원되는 버전이 다름
(현재 범용적 표준 ECMA-262[2] 3판에 대응하는 자바스크립트 버전은 1.5)

※ **ECMAS크립트** : 자바스크립트의 표준화된 버전

1.3 Javascript의 기술 발전

HTML5와 함께 Multi Device 지원의 해결책으로 Javascript는 급속한 발전을 하고 있다.
최근 브라우저 업체들이 **고성능의 Javascript엔진**을 선보이고 있다

JavaScript 엔진의 획기적인 성능 개선



크롬에 탑재된 V8엔진의 JIT(Just-In-Time)

- runtime 직전 직접적인 기계어 컴파일 (인터프리터 X)
- Inline caching 같은 최적화 기법 적용
- V8(Crome)
- 니트로(Safari)
- 트레이스 몽키(Firefox)

Google V8 JavaScript Engine

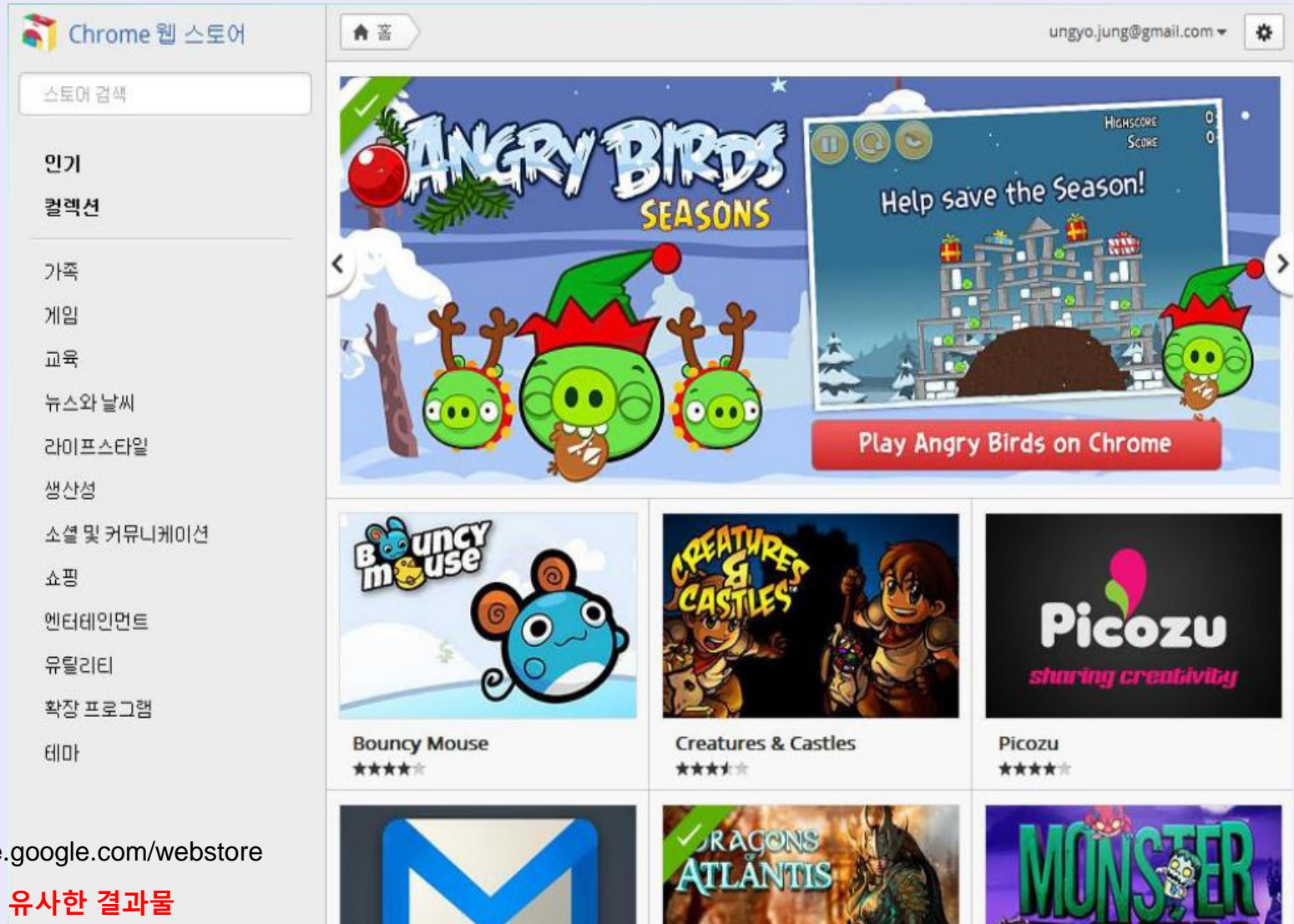
개발자	구글
프로그래밍 언어	C++, 자바스크립트
운영 체제	마이크로소프트 윈도, 맥 OS X, 리눅스, FreeBSD, 안드로이드, 웹OS
플랫폼	x86, x64, ARM
종류	자바스크립트 엔진
라이선스	BSD 라이선스
웹사이트	http://code.google.com/p/v8/

※ 과거 JavaScript는 PC의 성능, 컴파일 Engine의 성능이 느렸고, Device 제어가 불가능한 한계를 지녔음.

1.4 JavaScript의 확산

1. 웹 개발과 Javascript

모바일 크롬 웹 스토어에 2010.12 오픈 후 2만여 Web App이 등록, 빠르게 성장하고 있다



<https://chrome.google.com/webstore>

Native 언어와 유사한 결과물

1.5 jQuery Mobile vs Sencha Touch

Mobile WebApp을 만들때, 사용하는 대표적인 Javascript 프레임워크

구분	jQuery Mobile	Sencha Touch
정의	jQuery진영에서 개발한 모바일 웹앱 프레임워크	ExtJs 진영에서 개발한 모바일 웹앱 프레임워크
개발방식	마크업 기반의 개발 스타일 (HTML 개발과 비슷)	JavaScript 개발 스타일 -> 방대하고 체계적인 API 체계를 갖추고 있음(ExtJs 라이브러리 중심), 객체지향적 기법이 API에 녹아 있음
개발 난이도	낮음	높음
기능의 다양함	보통 (jqGrid 기능 부족(불합리한 구조), jqChart(유료))	높음
테마 특징	CSS, 테마롤러 도구 지원	SaaS 기반 -> 테마를 변경하거나 관리에 이점
기반 스크립트	jQuery (확장 어려움)	ExtJs (확장 자유로움)
지원 플랫폼	iOS, 안드로이드, 블랙베리, 팜, 심비안, 바다, 윈도우 모바일 (가장 안정적이고 많은 기기 지원)	iOS, 안드로이드, 블랙베리 6.0 이상
장점	친숙, Js 몰라도 사용 가능, 기존 사이트를 쉽게 모바일로 변경 가능, 가벼움. 드림위버에서 지원.	터치 이벤트에 특화, 거의 모든 UI 내장. 디자이너 제공.(WYSWYG 개발 가능)
단점	버그 다수 Sencha Touch에 비해 터치 이벤트 부분이 약함	새로운 문법을 익혀야 함 무거움 -> 로딩, 실행 느림(웹 사이트 전체를 하나로 관리함) 안드로이드 1.0에서 HTML5+CSS 가속 안됨

2. jQuery와 jQueryUI

2.1 jQuery 역사와 특징

2. jQuery와 jQueryUI

jQuery는 2006년 John Resig이 개발한 단순화된 클라이언트 사이드 스크립트 언어이다

브라우저 이벤트 처리, 애니메이션 등을 손쉽게 구현할 수 있어서 최근 크로스 플랫폼, HTML5, 크로스 브라우저 등의 이슈로 관심이 급증

개발자	John Resig http://ejohn.org/about/	
개발년도	2006년	
라이선스	자유 오픈 소프트웨어 (MIT, GNU)	
설계 Concept	소스 가독성 향상 쉬운 객체 모델 찾기 쉬운 애니메이션 개발 쉬운 이벤트 제어 쉬운 Ajax 개발	
표준 적용	Microsoft ASP.NET Ajax/MVC 노키아 런타임 웹 위젯 개발 플랫폼 적용 미디어위키 1.16 이후 사용	

- 출처 : 위키피디아 -

특징	설명
CSS 셀렉터	html내의 엘리먼트들을 손쉽게 표현 및 사용이 가능
플러그인 아키텍처	이미 개발된 많은 플러그 인을 쉽고 빠르게 사용
메서드 체인	여러개의 기능을 한줄에 나열하여 불필요한 코드 반복을 줄임
크로스 브라우저	브라우저별 발생 이벤트를 각각 지정 해 줄 필요가 없다.

배포일자	버전	부가정보
2011년 6월 30일	1.6.2	
2011년 5월 12일	1.6.1	
2011년 5월 3일	1.6	attr()와 val() 함수의 큰 성능향상
2011년 3월 31일	1.5.2	
2011년 2월 24일	1.5.1	
2011년 1월 31일	1.5	
2010년 11월 11일	1.4.4	
2010년 10월 16일	1.4.3	
2010년 2월 19일	1.4.2	성능향상에 초점을 둔 버전업
2010년 1월 25일	1.4.1	1.4 첫 번째 버그 픽스
2010년 1월 14일	1.4	메이저 업데이트입니다.
2009년 2월 20일	1.3.2	
2009년 1월 21일	1.3.1	
2009년 1월 14일	1.3	Sizzle Selector엔진을 핵심부에 도입
2008년 5월 24일	1.2.6	
2008년 5월 21일	1.2.5	1.2.4 오류 빌드 수정
2008년 5월 19일	1.2.4	
2008년 2월 8일	1.2.3	
2008년 1월 15일	1.2.2	
2007년 9월 16일	1.2.1	
2007년 9월 10일	1.2	
2007년 8월 24일	1.1.4	
2007년 7월 5일	1.1.3.1	
2007년 7월 1일	1.1.3	
2007년 5월 20일	1.1.3a	Alpha 배포
2007년 2월 27일	1.1.2	
2007년 1월 22일	1.1.1	
2007년 1월 14일	1.1	
2007년 1월 8일	1.1a	Alpha 배포
2006년 12월 12일	1.0.4	Last 1.0 bug fix
2006년 10월 27일	1.0.3	
2006년 10월 9일	1.0.2	
2006년 8월 31일	1.0.1	
2006년 8월 26일	1.0	첫 번째 안정화 배포
2006년 6월 30일	1.0a	Alpha 배포

2.2 jQuery 개요

2. jQuery와 jQueryUI

JavaScript로 구현이 어려운 기능들을 손쉽게 개발할 수 있도록, 단 몇 줄의 코딩만으로 해결할 수 있게 해주는 **JavaScript 라이브러리 모음**이다



<장점>

- 브라우저 호환성 보장
- 액션에 특화
- 뛰어난 가독성과 간결성 (주로 웹표준과 연동)
- HTML5 표준 준수

<제공 기능>

- **공통의 DOM**
- **이벤트**
- **특수 효과**
- **Ajax 함수**

Javascript Library 오픈소스 단체
'jQuery Foundation(www.jquery.com)'

jQuery는 JavaScript에 비해 훨씬 간략한 표현을 사용한다

원래 JavaScript 방식

```
for (i = 0; i <= 4; i++)  
{  
    document.getElementsByTagName("p")[i].innerHTML =  
    "Change the page.";  
}
```

- 문서에게 얘기하고 있다.
(DOM의 D 말이에요)
- 태그 이름이 'p'인 요소를 전부 선택
- 태그 이름이 'p'인 요소 중 i번째 요소를 선택
- 그 요소의 HTML을 ...
- 괄호 안의 내용으로 수정

jQuery 변수 선언

일반적인 자바 스크립트는 var my 이런 식으로 변수를 선언한다면, jQuery는 \$my 이런 표현으로 선언 할 수 있다.

```
var my = "안녕하세요";  
$my = "안녕하세요";
```

jQuery 방식

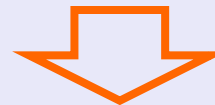
```
$("#p").html("Change the page.");
```

- 문단 요소를 선택
- **jQuery는 '선택자 엔진'이란 걸 사용하기 때문에 CSS에서 사용하던 선택자를 사용할 수 있다.**
- 그 요소의 HTML을 괄호 안에 있는 내용으로 수정

jQuery 단축 표기

jQuery()

: 괄호 안에 넣은 요소를 선택하는 역할



\$()

: jQuery 단축 표기
(\$ 지시자)

www.jquery.com 사이트에서 Download 받을 수 있다

jQuery 실행 파일은 Compressed/Uncompressed 버전을 Download 받을 수 있다.
운영환경 배포용에는 Compressed 버전을 사용하고, 개발에는 Uncompressed 버전을 사용한다.

Downloading jQuery

The jQuery 1.9 line has major changes from previous versions. We *strongly* recommend that you also use the jQuery Migrate plugin if you are upgrading from older versions of jQuery or need to use plugins that haven't yet been updated. Read the [jQuery 1.9 Upgrade Guide](#) and the [jQuery 1.9 release blog post](#) for more information.

[Download the compressed, production jQuery 1.9.1](#)

[Download the uncompressed, development jQuery 1.9.1](#)

[jQuery 1.9.1 release notes](#)

미너파이(minify) 버전

- 주석 없음
- 띄어쓰기 없음
- 줄바꿈 없음
- 변수는 약어 사용

```
if(count == 0){  
    alert("방문자 없음");  
} else if(count == 1){  
    alert("최초 방문자");  
} else if(count > 1){  
    alert("환영합니다.");  
}
```

※ 주석 및 불필요한 공백과 줄바꿈을 모두 제거

파일 크기 최소화

```
if(a==0){alert("방문자 없음");}else if(a==1){alert("최초 방문자");}else if(a>1){alert("환영합니다.");}
```

❖ **\$ 함수** : jQuery 오브젝트의 팩토리 메소드.

- 이 함수들은 "chainable"하다. (Method Chain : 메소드를 여러 개 이어서 소스를 간소화한다.)
- 각각은 jQuery 오브젝트를 반환한다.
- \$ 함수로 시작된다.
- CSS 셀렉터 스트링을 가지고 호출된다.

```
$("div.test").add("p.quote").addClass("blue").slideDown("slow");
```

Chaining : 1. div 태그가 달린 모든 엘리먼트를 찾되, 클래스 애트리뷰트가 test인 것을 찾는다.
2. p 태그를 찾되, 클래스 애트리뷰트가 quote인 것을 찾는다.
3. 찾아낸 각각의 엘리먼트에 대해 클래스 애트리뷰트 blue를 추가한다.
4. 그 뒤 애니메이션 효과를 주어 아래쪽으로 슬라이드(미끄러지게) 시킨다.

❖ **\$. -가 앞에 붙은 함수**

- jQuery 오브젝트 그 자체와 연동되지는 않는다.(유틸리티 메소드)

```
$.each([1,2,3], function() {  
    document.write(this + 1);  
}); // 234를 도큐먼트에 출력한다
```

```
$.ajax({  
    type: "POST",  
    url: "some.php",  
    data: "name=John&location=Boston",  
    success: function(msg){  
        alert( "Data Saved: " + msg );  
    }  
}); //파라미터 name=John, location=Boston을 주면  
서 some.php에 요청을 보낸다. 성공하면 그 응답이  
alert()된다.
```

jQuery는 선택자(Selector)와 함수의 Chaining 관계로 이루어져 있다

\$(document) . ready (function) {

선택자에 의해서 선택
된 DOM Element가

준비가
되면

다음에 오는 괄호 안의
코드를 즉시 실행해라.

\$("#clickMe") . click (function) {

점(.)은 선택자와
메서드를 구분한다.

ID가 clickMe인 버튼에 click 이벤트를
연결하여 버튼을 클릭할 수 있게 한다.

코드 블록

중괄호 안에 있는 내용을 '코드
블록' 이라고 한다.

버튼을 눌렀을 때 일어나는
일은 중괄호({}) 안에 쓴다.

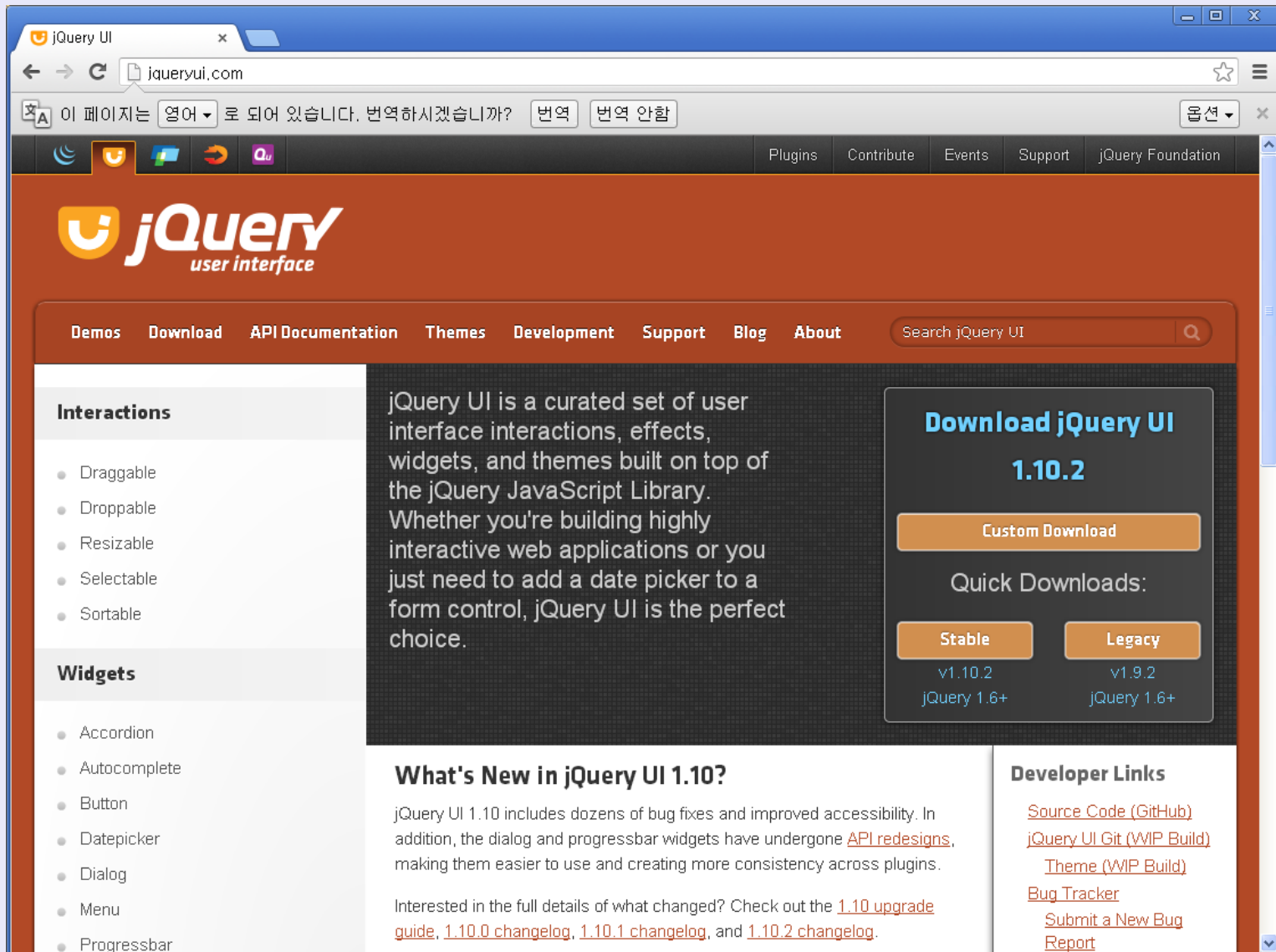
});

세미콜론(;)은 jQuery 문장을 끝맺는 종결자 (click 문장을 닫는다.)

});

ready 함수를 닫는다.

JavaScript와 CSS를 이용하여 UI를 손쉽게 구현 할 수 있도록 제작된 라이브러리



jQuery UI 라이브러리는 jQuery 코어를 보완하는 세 가지 **플러그인**을 제공한다

애니메이션

jQuery UI는 jQuery에 여러가지 애니메이션을 추가합니다. **요소들이 통통 튀고, 폭발하고, 맥동하고, 진동하는 효과를** 낼 수 있습니다. jQuery UI는 복잡한 수학 계산을 통해 애니메이션을 더욱 사실적으로 보이게 하는 이징 함수도 제공합니다.

상호작용

상호작용을 통해 웹 애플리케이션에 복잡한 행동방식을 추가할 수 있습니다. 예를 들면 **요소를 드래그 앤 드롭하거나 요소의 내용을 정렬하는 등의 플러그인이** 이 범주에 들어갑니다.

위젯

위젯이란 웹앱에 기능을 추가하는 인터페이스 구성 요소입니다. 사용하기 쉬운 **반응형 인터페이스**를 만드는 것은 무척 복잡하고 시간도 아주 많이 필요한데 위젯을 사용하면 이런 작업에 필요한 시간을 대폭 줄일 수 있다.

※ jQuery와 마찬가지로 사용할 함수와 Argument(인자, Parameter) 값만 알고 있다면, 복잡한 코딩 없이 각종 이펙트와 위젯, 피커 등을 만들 수 있다.

물론 조금 더 복잡한 기능을 원한다면 자바스크립트와 혼용을 해야 한다.

<http://jqueryui.com> 사이트에서 jQuery UI 가 제공하는 각종 컴포넌트에 대한 예제와 API를 확인할 수 있다.

※ jQuery는 웹 개발자들이 코어 라이브러리를 확장하거나 기능을 추가할 수 있는 **플러그인 구조**를 가지고 있다.

※ jQuery UI가 제공하는 각종 컴포넌트와 API는 jqueryui.com 사이트에서 제공하며, 실제 샘플을 보고 소스를 복사해서 사용할 수 있도록 편리하게 제공하고 있다.

2.9 HTML 화면 구조 Template

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="utf-8" />
```

```
  <title>jQuery UI Draggable - Default functionality</title>
```

```
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css" />
```

```
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
```

```
  <script src="http://code.jquery.com/ui/1.10.1/jquery-ui.js"></script>
```

```
  <link rel="stylesheet" href="/resources/demos/style.css" />
```

```
<style>
```

```
#draggable { width: 150px; height: 150px; padding: 0.5em; }
```

```
</style>
```

```
<script>
```

```
$(function() {
```

```
  $( "#draggable" ).draggable();
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div id="draggable" class="ui-widget-content">
```

```
  <p>Drag me around</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

HTML 선언부

필수 스크립트 Import

: 소스를 순차적으로 읽어서 실행하기 때문에 대상 Element가 메모리 상에 올라 오기 전에 대상 Element를 사용하는 이벤트가 발생할 경우 오류가 발생한다.

개별화면 CSS 정의

개별화면 기능 정의

UI 화면 구현

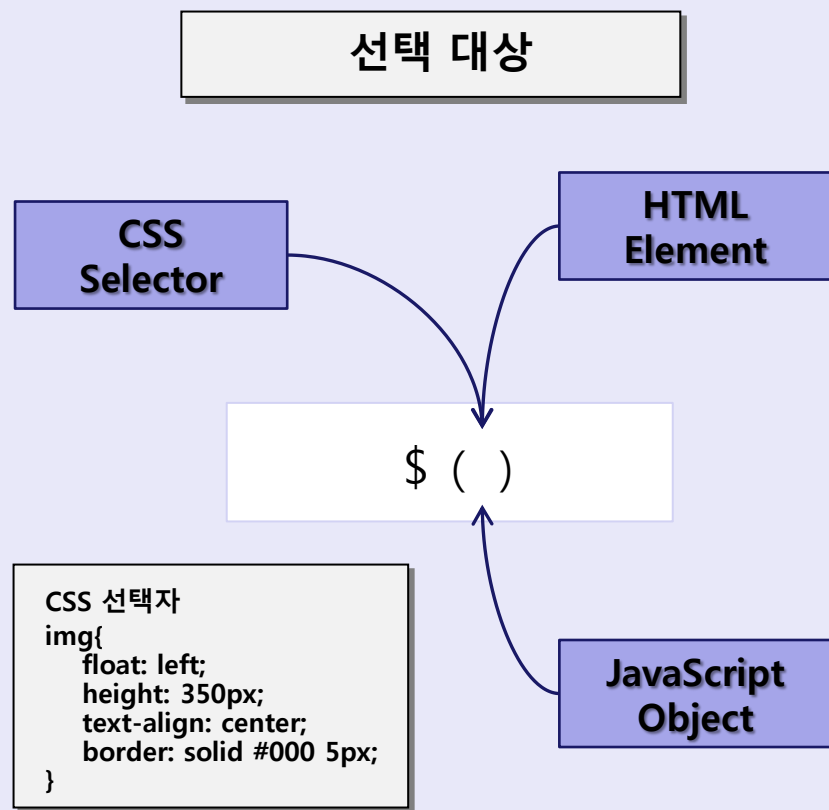
개별화면 JavaScript 정의

■ Script의 위치

보통 페이지에서는 페이지 로딩을 통하여 여러 이미지와 파일들을 순차적으로 한번에 로딩할 필요없이 로딩되는 순으로 페이지에 표시하곤 한다. 이미지 파일은 여러 파일을 동시에 받을 수 있기 때문이다. 그러나 <script> 파일이 있으면 다른 이미지나 script 파일을 받아오지 못한다. 그러므로 script 부분은 페이지의 가장 하단에 위치하는 것이 좋다.

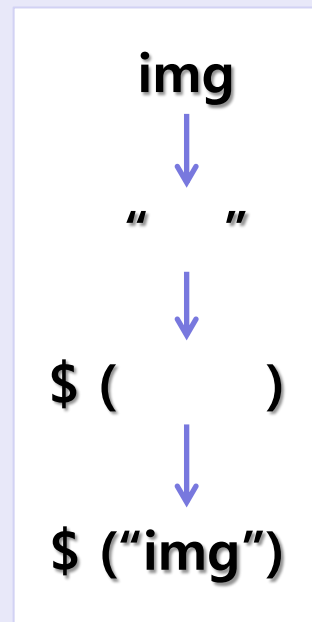
2.10 jQuery 함수의 선택자

jQuery 는 선택자 엔진을 사용하여 CSS, HTML, JavaScript Object를 선택할 수 있다



- CSS Selector : CSS 선택자를 넣으면 jQuery 함수는 그 선택자에 맞는 요소를 반환
- HTML Element : HTML 문자열을 넣으면 브라우저 상의 페이지에 바로 요소를 추가 가능
- JavaScript Object : 자바스크립트 객체도 추가 가능

문법 해석



HTML Element

선택자는 모두 문자열 형태
이므로 따옴표를 사용한다.

선택자를 jQuery 단축
기호로 묶는다.

페이지에 있는 이미지
요소를 모두 반환한다.

jQuery 라이브러리는 CSS와 마찬가지로 요소를 선택하도록 만들어졌다.

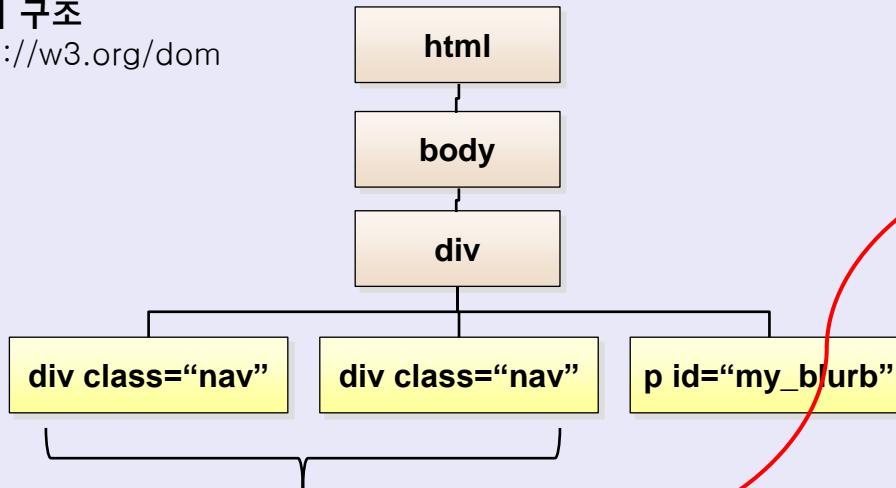
```
$("button").click(function){}
```

```
$("p").hide();
```

```
$("#myTop").css({"color":"blue"});
```

2.11 jQuery 함수의 선택자

웹 페이지의 DOM
트리 구조
<http://w3.org/dom>



클래스 선택자는 그 클래스에 속하는 요소를 모두 선택한다.

CSS 코드 -> 디자인

```
.nav{  
  display: block;  
  border: solid #00f 1px;  
  width: 100%;  
}
```

jQuery 코드 -> 이벤트

```
$(".nav").click( function() {  
  alert("나를 눌렀군요!");  
});
```

자손 선택자

왼쪽에 있는 선택자가 부모

오른쪽에 있는 선택자가 자식

자식의 자식을 선택

\$ ("div p")

\$ ("div div p")

jQuery 선택자에서 DOM Element 상의 자손 Element를 선택할 때는 빈칸을 이용한다

2.12 jQuery vs CSS 선택자 비교

CSS 선택자는 요소를 선택해서 스타일을 입히고, jQuery 선택자는 동작을 추가한다

CSS 선택자

요소 선택자

```
h1 {  
  text-align: left;  
}
```

- h1 : 요소 선택자 or 태그 선택자
(HTML 문서에 있는 h1 요소를 모두 선택)
- text-align : CSS 속성, left : CSS 속성의 값

클래스 선택자

```
.my_class {  
  position: absolute;  
}
```

- .my_class : CSS 클래스는 항상 마침표로 시작함
(클래스 선택자를 사용해서 요소 그룹을 선택)

ID 선택자

```
#my_id {  
  color: #3300FF;  
}
```

- #my_id : CSS ID는 항상 해시(#) 기호로 시작함
(요소를 딱 하나만 선택할 때는 ID 선택자를 사용)

표현식

```
border-style  
border-width
```

- CSS는 CSS 표기법을, jQuery는 DOM 표기법을 사용 : http://www.oxfordu.net/webdesign/dom/straight_text.html

jQuery 선택자

요소 선택자

```
$("h1").hide();
```

- Hide() : 메서드
- 페이지에 있는 h1요소를 모두 숨깁니다.

클래스 선택자

```
$(".my_class").slideUp();
```

- slideUp() : 메서드
- CSS 클래스 my_class가 있는 요소는 모두 슬라이드로 올립니다.

ID 선택자

```
$("#my_id").fadeOut();
```

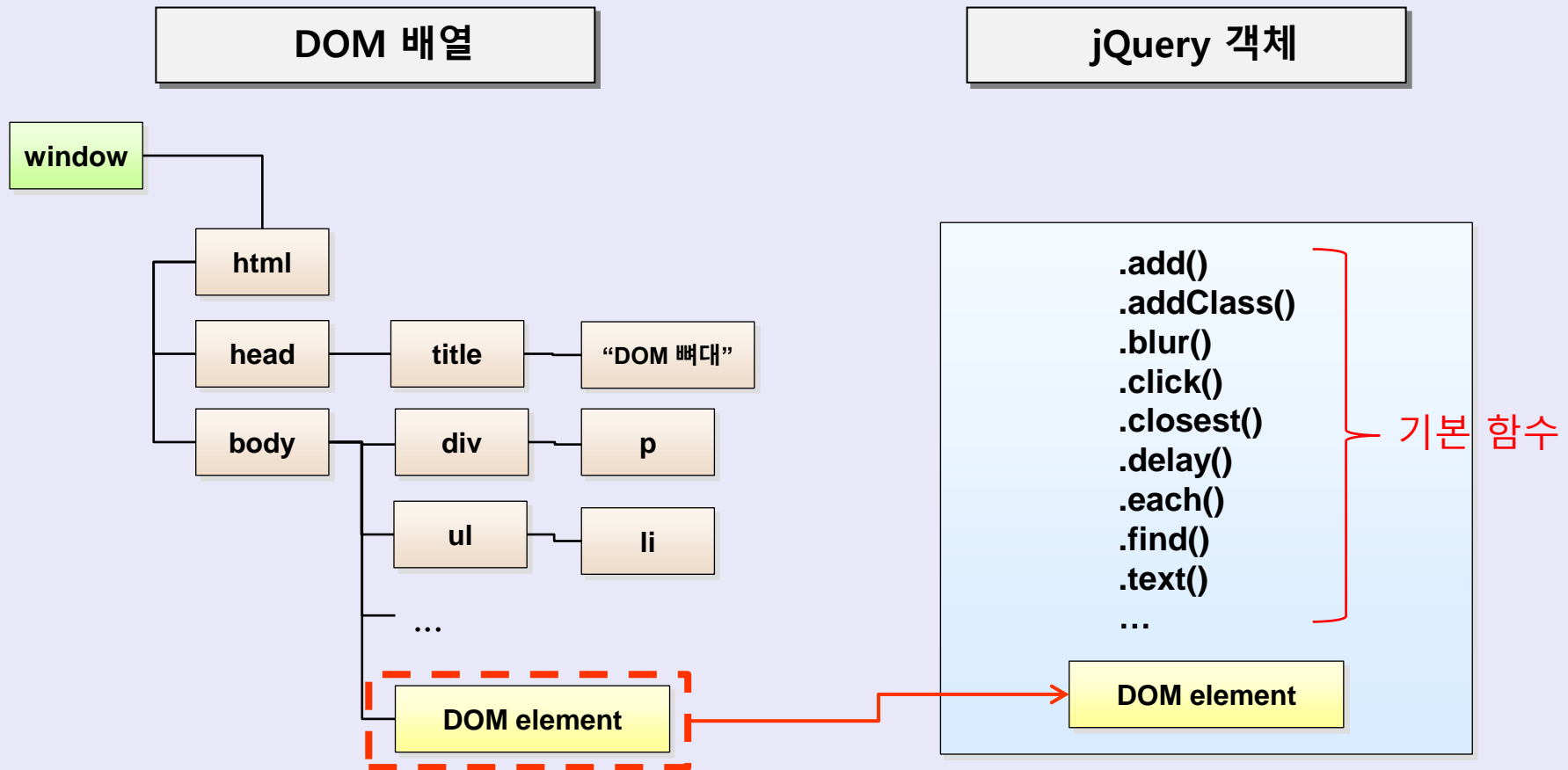
- fadeOut() : 메서드
- CSS ID가 my_id인 요소를 페이드아웃시켜 보이지 않게 합니다.

표현식

```
borderStyle  
borderWidth
```

2.13 DOM 배열 vs jQuery 객체 구조

jQuery는 DOM element를 가져올 때, jQuery 객체로 Wrapping 하여 사용한다



- DOM element 를 직접 관리
- `document.getElementsByTagName("p")`

- DOM element 를 포함한 jQuery 객체를 관리

※ 브라우저마다 DOM 메서드가 일관성 없이 구현되어 있기 때문에 jQuery를 이용하여 호환성 확보.

2.14 jQuery 구성요소와 Plugin

jQuery는 Selector, Filter, API Method, Event, Plugin을 제공한다

❖ jQuery 구성요소

구성요소	문법	예제	설명
Selector	\$(대상)	<code>\$("div p")</code>	필요한 요소나 클래스, 태그를 찾아옴
Filter	\$(대상:필터명)	<code>\$("td:last")</code>	더 강력한 셀렉터 역할을 수행하게 도와줌
API Method	jQuery객체.API명	<code>\$(this).each()</code>	jQuery 객체를 다루기 위한 기본 기능을 제공
Event	\$().이벤트함수명	<code>\$("#button").click()</code>	이벤트 발생을 인지하여 원하는 Function을 실행

❖ jQuery Plugin의 종류

구성요소		예제	설명
jQueryUI	Interactions	<code>\$('#box').draggable();</code>	UI 객체간에 연동하면서 처리되는 이벤트
	Widgets	<code>\$("#calendar").datepicker();</code>	단위 모듈로 구현되는 UI 기능 컴포넌트
	Effects	<code>\$("#combo").effect(selectedEffect, options, 500, callback);</code>	CSS를 통해 표현할 수 있는 화면 효과
	Utilities	<code>\$(".positionable").position(...)</code>	UI에서 사용되는 도구 모음
jqGrid		<code>\$('#jqGrid').jqGrid({ ... });</code>	jQuery 기반의 Grid 플러그인
jqTree		<code>\$('#jsTree').jstree({ ... });</code>	Data를 Tree 형태로 표현해주는 Plugin
jqChart		<code>\$('#jqChart').jqChart({ ... });</code>	Data를 Chart 형태로 표현해주는 Plugin

요소에 이벤트를 등록하는 일을 묶는다(bind)고 한다

단축 메소드

단축 메소드를 사용하여 직접 함수를 연결

```
$("#myElement").click( function() {  
    alert($(this).text());  
});
```

.bind() 함수

DOM 요소가 이미 존재할 때만 쓸 수 있음

```
$("#myElement").bind('click', function() {  
    alert($(this).text());  
});
```

※ 두 방법 모두 ID가 myElement인 요소에 클릭(**click**) 이벤트 리스너를 등록한다.

❖ .bind() 와 .unbind()

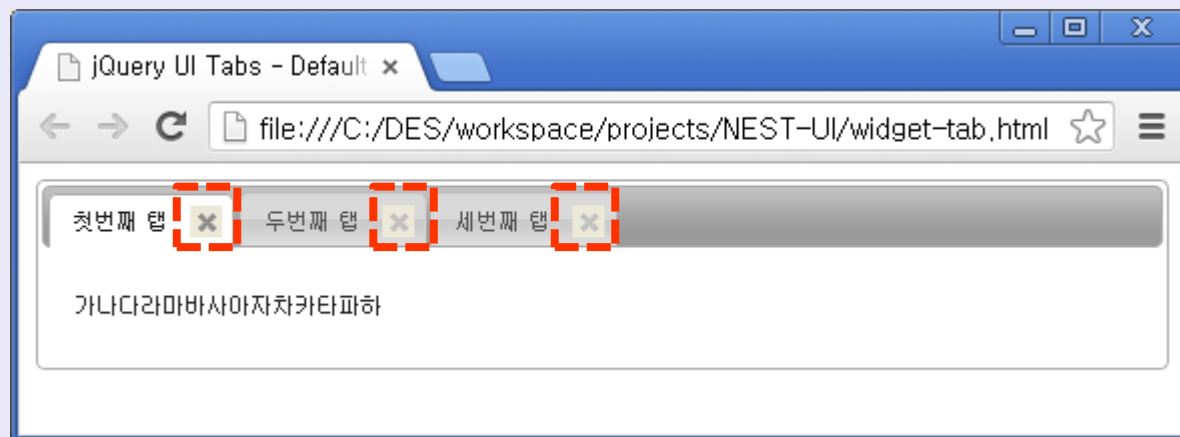
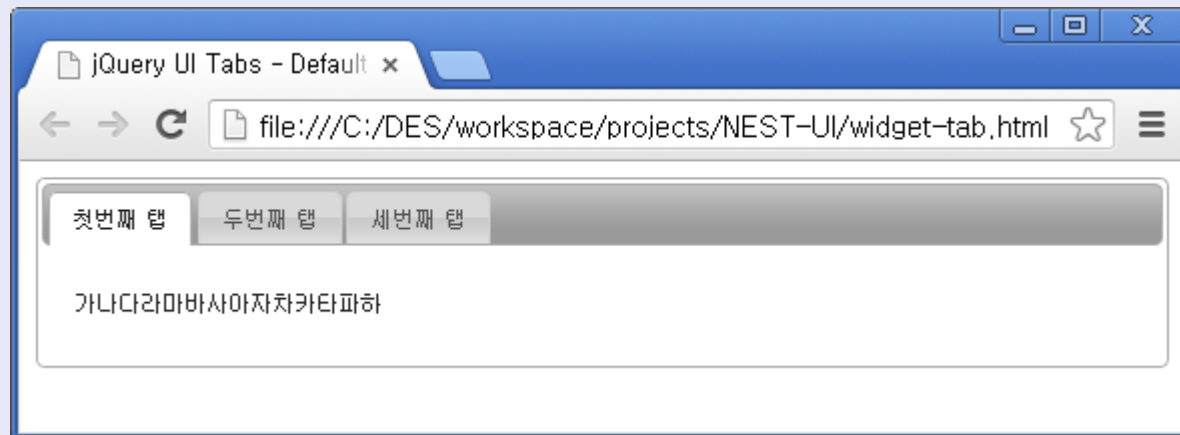
- .bind()는 이벤트를 직접 처리하지 않고, 엘리먼트 집합에 함수를 연결(전달)한다.
- unbind를 하게 되면 연결된 이벤트를 끊어내어 이벤트 연결 이전의 상태로 돌려놓는 효과가 발생된다.

사용 방법	설명
<code>.bind(eventType,[eventData,],handler(eventObject))</code>	eventType 클릭 서브밋 같은 하나 이상의 자바스크립트 이벤트 타입 이름 eventData 이벤트 핸들러로 전달할 데이터를 담은 데이터 맵 handler(eventObject) 이벤트가 발생했을때 실행되는 이벤트 핸들러 함수
<code>.bind(eventType,[eventData,],false)</code>	
<code>.unbind(eventType,handler(eventObject))</code>	
<code>.unbind(eventType,false)</code>	

3. jQuery UI Tab 기능 확장

1. jQuery Function 재정의
2. Options 확장
3. Button 추가
4. 기능 확장 부분 모듈화
5. CSS 적용

jQuery UI에서 제공하는 기본 Tab Widget에 'Tab 닫기 버튼'을 추가하는 기능 구현



❖ jQuery UI 샘플 참조

브라우저에서 한글 깨짐 (utf-8에서 **euc-kr** 로 수정)

jQuery를 사용하기 위한 필수 입력 사항
'전체 Library' or '사용하려는 컴포넌트(기능 그룹)' 입력
디자인 정의 파일(css) 입력

본문에서 사용할 스크립트 함수 정의

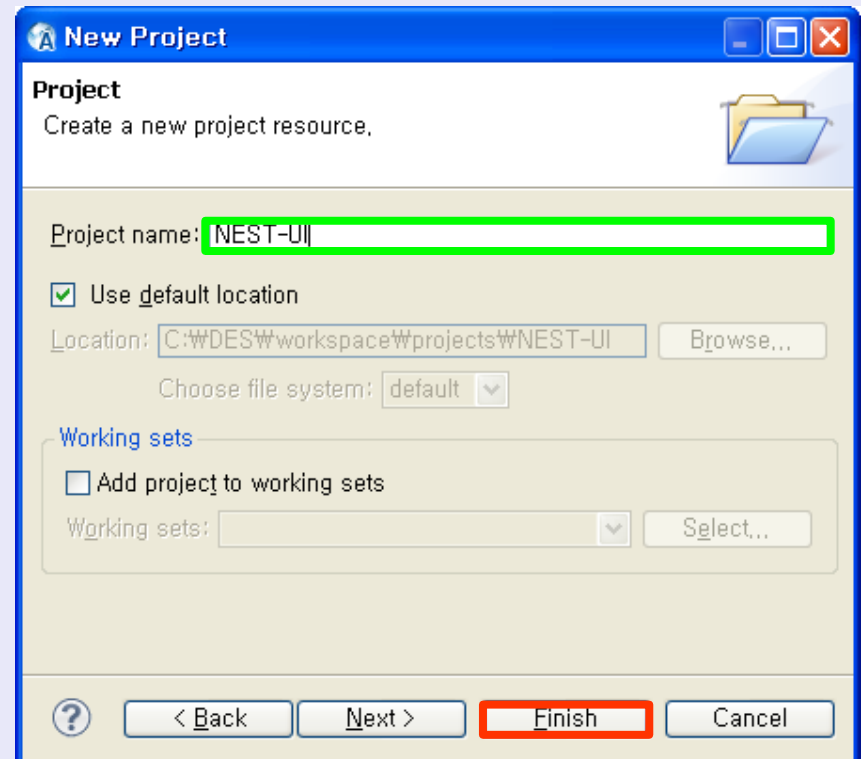
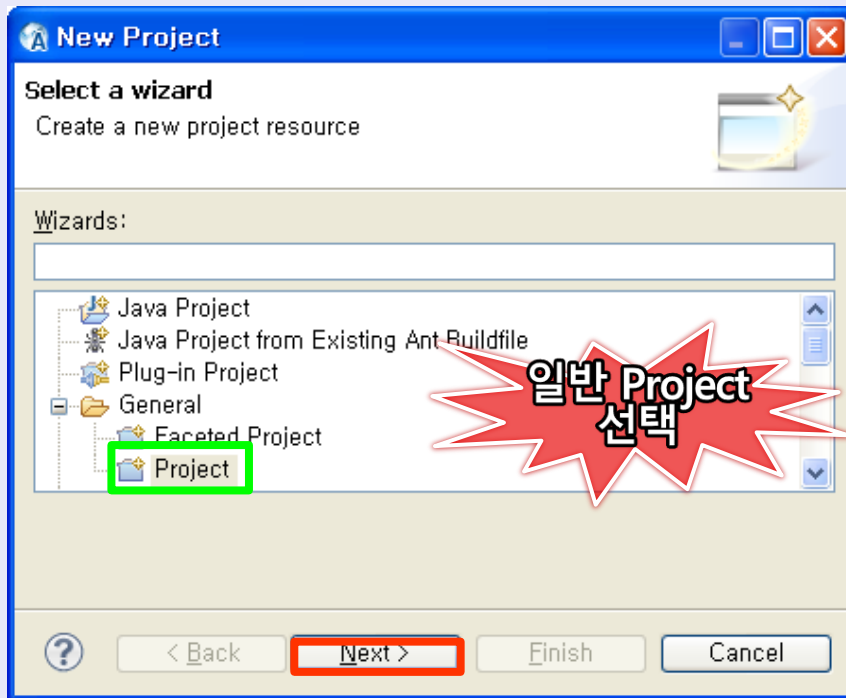
본문 HTML 화면 구현 영역
(표준 : in-line 함수를 사용하지 않음)

```
1) <!doctype html>
2)
3) <html lang="ko">
4) <head>
5)   <meta charset="euc-kr" />
6)   <title>jQuery UI Tabs - Default functionality</title>
7)   <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css" />
8)   <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
9)   <script src="http://code.jquery.com/ui/1.10.1/jquery-ui.js"></script>
10)  <link rel="stylesheet" href="/resources/demos/style.css" />
11)  <script>
12)    $(function() {
13)      $( "#tabs" ).tabs();
14)    });
15)  </script>
16) </head>
17) <body>
18)
19) <div id="tabs">
20)   <ul>
21)     <li><a href="#tabs-1">첫 번째 탭</a></li>
22)     <li><a href="#tabs-2">두 번째 탭</a></li>
23)     <li><a href="#tabs-3">세 번째 탭</a></li>
24)   </ul>
25)   <div id="tabs-1">
26)     <p>가나다라</p>
27)   </div>
28)   <div id="tabs-2">
29)     <p>마바사아</p>
30)   </div>
31)   <div id="tabs-3">
32)     <p>자차카</p>
33)     <p>타파하</p>
34)   </div>
35) </div>
36)
37) </body>
38) </html>
```

</body> 태그를 닫기 전에 이 화면에서만 사용할 이벤트를 정의함
(화면 로딩 지연으로 인한 오류 방지)

1. Eclipse에 신규 Project를 생성한다

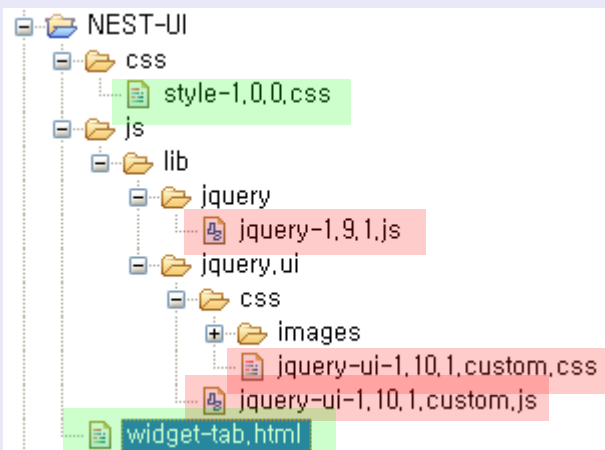
- ① General > Project 선택 후 Next 버튼 클릭
- ② Project Name 에 'NEST-UI' 입력 후 Finish 버튼 클릭



※ Dynamic Web Project로 생성하면 Eclipse가 제공하는 Tomcat 활용 가능

3.3 Project 생성 - Package 구조 정의

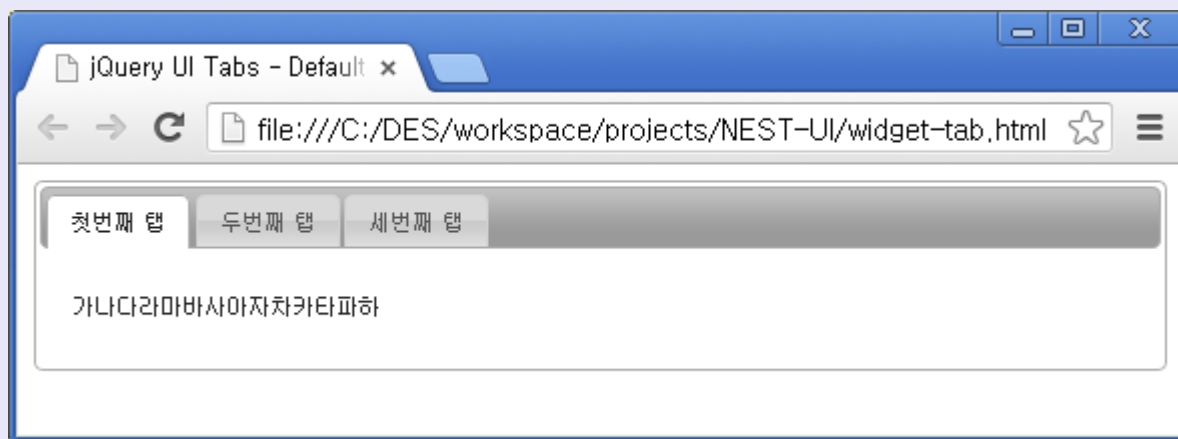
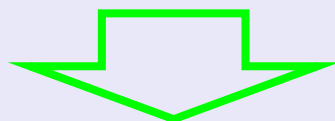
2. Eclipse에서 아래 그림과 같이 Package 구조를 만든다



■ 소스 파일 설명

파일명	용도
style-1.0.0.css	대상 화면에서만 사용하는 CSS 정의
jquery-1.9.1.js	jQuery를 사용하기 위한 Core Library
jQuery-ui-1.10.1.custom.css	jQuery UI에서 사용하는 CSS 정의
jquery-ui-1.10.1.custom.js	jQuery UI를 사용하기 위한 JavaScript (jQuery UI Core Library 내포)
widget-tab.html	대상 화면을 구현한 HTML 정의

※ Tab 위젯만 대상으로 할 경우 jquery-ui-1.10.1.custom.js 대신에 jqueryui.com > download 에서 tab widget만 내려받은 js 파일로 대체 할 수 있다. (jquery.ui.tabs.js)



3.4 Tabs 원본 HTML 소스 수정

/widget-tab.html 파일 내용

```
1) <!doctype html>
2)
3) <html lang="ko">
4) <head>
5)   <meta charset="euc-kr" />
6)   <title>jQuery UI Tabs - Default functionality</title>
7)   <link rel="stylesheet" href="js/lib/jquery.ui/css/jquery.ui.1.10.1.custom.css" />
8)   <link rel="stylesheet" href="js/lib/nest.ui/css/nest.ui.tabs-1.0.0.css" />
9)   <script src="js/lib/jquery/jquery-1.9.1.js"></script>
10)  <script src="js/lib/jquery.ui/jquery-ui-1.10.1.custom.js"></script>
11)  <script src="js/lib/nest.ui/nest.ui.tabs-1.0.0.js"></script>
12)  <link rel="stylesheet" href="css/style-1.0.0.css" />
13)  <script>
14)    $(function() {
15)      $( "#tabs" ).tabs();
16)    });
17)  </script>
18) </head>
19) <body>
20)
21) <div id="tabs">
22)   <ul>
23)     <li><a href="#tabs-1">첫번째 탭</a></li>
24)     <li><a href="#tabs-2">두번째 탭</a></li>
25)     <li><a href="#tabs-3">세번째 탭</a></li>
26)   </ul>
27)   <div id="tabs-1">
28)     <p>가나다라마바사아자차카타파하</p>
29)   </div>
30)   <div id="tabs-2">
31)     <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ</p>
32)   </div>
33)   <div id="tabs-3">
34)     <p>가나다라마바사아자차카타파하</p>
35)     <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ</p>
36)   </div>
37) </div>
38)
39) </body>
40) </html>
```

3. jQuery UI의 Tab Widget의 원본 소스는 그대로 두고 기능을 확장하기 위하여 HTML 소스에 확장하려는 JavaScript, CSS 파일명을 추가한다.

– nest.ui.tabs-1.0.0.js

– nest.ui.tabs-1.0.0.css

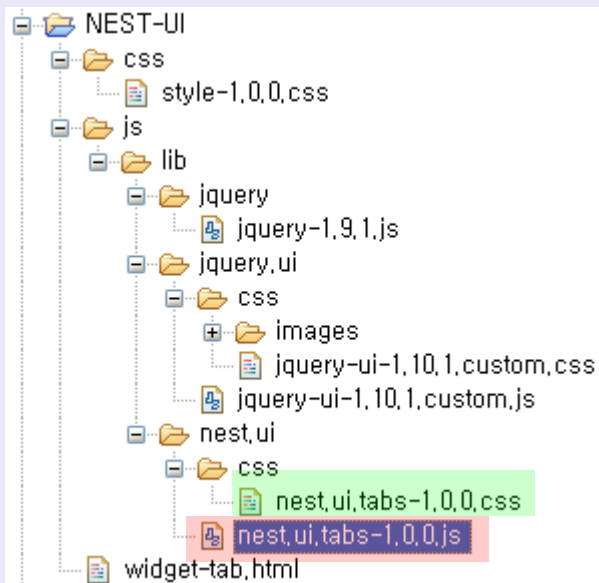
※ CSS 파일을 먼저 정의하는 이유는 JavaScript 파일에서 CSS에 정의된 CSS Tag를 사용할 경우 오류 발생 위험이 있기 때문이다.

Tab 구현 소스

- 웹표준에 의해 in-line Style, in-line Event를 사용하지 않는다.
- id명 / class명을 적절히 구사하여 Widget 컴포넌트를 구현한다.

3.5 기능 확장 js, css 파일 생성

4. 확장 폴더(nest.ui) 아래에 **확장 js 파일(nest.ui.tabs-1.0.0.js)**을 만들고, **alert(0);**을 찍어서 파일이 정상적으로 import 되었는지 확인한다



■ 소스 파일 설명

파일명	용도
nest.ui.tabs-1.0.0.js	확장하려는 내용 (JavaScript 소스)
nest.ui.tabs-1.0.0.css	확장된 Widget에서 사용할 CSS Style

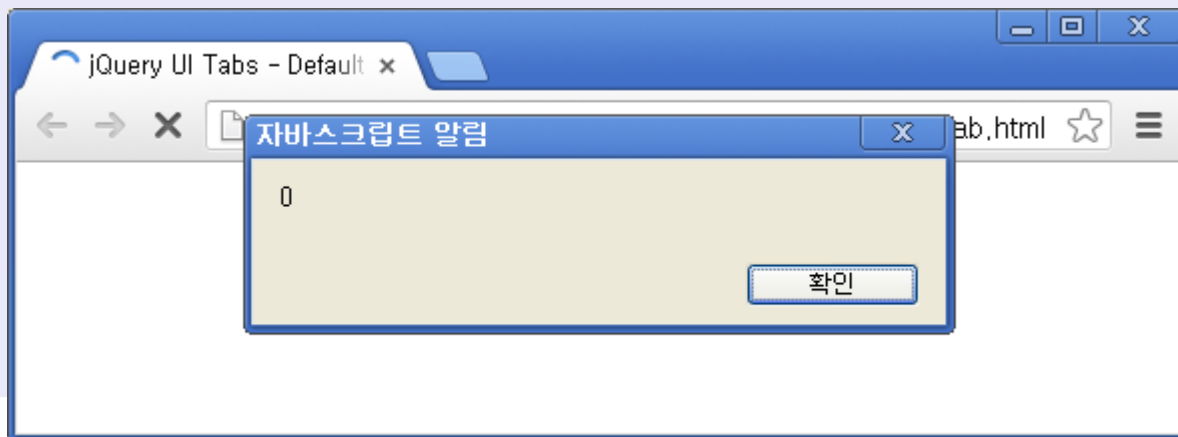
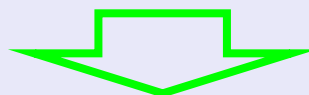
■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js

```
// 확장 스크립트 import 확인  
alert(0);
```

/js/lib/nest.ui/css/nest.ui.tabs-1.0.0.css

```
// 내용 없음
```



5. **확장 js 파일(nest.ui.tabs-1.0.0.js)**에 소스 전체를 익명함수로 감싸고 jQuery(\$)을 인자로 넘긴다. jQuery UI Tab 라이브러리가 있는지 체크한다.

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js

```
// ① 소스 전체를 익명함수로 처리
(function($) {

    // 라이브러리 확인
    if (!$.ui || !$.ui.tabs) { // ② jQuery UI Tab 관련 library가 대상 화면 상에 존재하고 있는지 체크
        return false;
    }

    // 확장 로직 구현 위치
})(jQuery);
```

※ 변수의 전역 등록 방지를 고려하여 로직을 익명 함수로 감싸고, jQuery를 인자로 넘겨 익명 함수 안에서 \$가 jQuery임을 보장시킨다.

함수 이름 정하기 - 함수 선언

```
function myFunc1() {
    $("div").hide();
}
```

함수 이름

함수 이름 정하기 - 함수 표현식

```
var myFunc2 = function() {
    $("div").show();
}
```

함수를 변수에 할당

익명 함수는 코드를 실행하다가 익명 함수가 정의된 위치에서 바로 실행됩니다. 익명 함수 안에서 선언한 변수는 해당 함수 안에서만 사용할 수 있습니다.

jQuery UI의 Widget을 확장할 경우 공통적으로 아래와 같은 패턴의 소스를 만들게 된다.

```

1)  /**
2)   * Supported Browser : MSIE, Chrome, FireFox
3)   * Object      : nest.ui.tabs.js
4)   * Description : jQuery UI Tabs 확장
5)   */
6)  (function($) {
7)
8)    // 라이브러리 존재 여부 확인
9)    if (!$.ui || !$$.ui.tabs) {
10)     return false;
11)    }
12)
13)    // 닫기 버튼
14)    // ADD_OPTION closeButton
15)    // EXT_METHOD _processTabs
16)    // EXT_METHOD _setOption
17)    // ADD_METHOD _setCloseButton
18)    $.widget('ui.tabs', $.ui.tabs, {
19)      options: {
20)        /**
21)         * <p>닫기 버튼 추가 여부 또는 템플릿.</p>
22)         *
23)         * @name jquery.ui.tabs.prototype.closeButton
24)         * @property
25)         * @type Boolean|String
26)         * @default false
27)         */
28)        closeButton: false
29)      },
30)
31)      /**
32)       * <p>_processTabs 메서드 확장.</p>
33)       * <p>closeButton 옵션 처리.</p>
34)       *
35)       * @see jquery.ui.tabs.prototype._processTabs
36)       */

```

```

37)      _processTabs: function() {
38)        // 원본 메서드 호출
39)        this._super.apply(this, arguments);
40)
41)        // 기능 구현
42)      },
43)
44)      /**
45)       * <p>_setOption 메서드 확장.</p>
46)       * <p>closeButton 옵션 처리.</p>
47)       *
48)       * @see jquery.ui.tabs.prototype._setOption
49)       */
50)      _setOption: function(key, value) {
51)        // 원본 메서드 호출
52)        this._super.apply(this, arguments);
53)
54)        // 기능 구현
55)      },
56)
57)      /**
58)       * <p>닫기 버튼 추가.</p>
59)       *
60)       * @name jquery.ui.tabs.prototype._setCloseButton
61)       * @private
62)       * @function
63)       * @param {String|Boolean} [closeButton] 닫기 버튼 추가 여
64)       *   부 또는 템플릿
65)       */
66)      _setCloseButton: function(closeButton) {
67)        // 기능 구현
68)      };
69)
70)    })(jQuery);

```


6. 확장하려는 기능이 Tab이므로 \$.widget으로 등록한다. (확장 js파일에서)
- jQuery 용어 규칙에 따라 Widget에는 Options, Methods, Event 가 API 그룹명으로 올 수 있다.

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

```
$.widget('ui.tabs', $.ui.tabs, {  
    options: {  
  
        /**  
        * <p>닫기 버튼 추가 여부 또는 템플릿.</p>  
        *  
        * @name jquery.ui.tabs.prototype.closeButton  
        * @property  
        * @type Boolean|String  
        * @default false  
        */  
        closeButton: false  
    }  
});
```

// ① \$.widget()을 사용하여 jQuery의 ui.tabs을 상속
// ② 닫기 버튼은 Tab Widget이 생성될 때 만들어져야 하므로, Options를 확장

※ jQueryUI 사이트 API를 보면 Options(생성시 적용됨), Methods(실행 중 적용됨), Event(실행 중 적용됨) 그룹명을 확인 할 수 있다.

7. Tab Widget 원본 소스의 로직을 분석하여 어떻게 확장할지 알아본다.

- ① **jquery-ui-1.10.1.custom.js** 파일을 열어서 Tab Widget이 호출되었을 때, 실행되는 로직을 분석한다.
- ② 맨 처음 `_CreateWidget` 부터 실행된다.
- ③ `_CreateWidget` function의 `_create()`; 메소드를 확장하여 customizing 하기로 결정
(return값 유무 확인이 필요하다. – method chain¹⁾ 에 영향이 있음)

■ 소스 분석 - Tab Widget 로직에 관한 내용 추가

jquery-ui-1.10.1.custom.js

```
(function( $, undefined ) {  
.....  
$.Widget.prototype = {  
.....  
  _createWidget: function( options, element ) { //11450  
    .....  
  },  
  ....  
  $.widget( "ui.tabs", {  
    .....  
    _create: function() {  
      .....  
    },  
    .....  
  }  
})( jQuery );
```

#. jQuery UI 확장 분석 Tips - Interaction

- Interaction은 마우스 이벤트가 중요
- JQuery.ui.mouse.js 파일에서 이벤트 정의됨
- Hover 이벤트는 위젯에서만 존재함
- Droppable 이벤트는 ddmanager 함수에서 객체간 처리를 담당함

jQuery 컴포넌트 플러그인과 jQueryUI 위젯 컴포넌트는 대부분 하나의 실행 메서드만 노출하고, 그 메서드에서 인스턴스 생성 및 메서드 실행을 처리한다

- ❖ <http://docs.jquery.com/Plugins/Authoring>
- ❖ 실행 메서드를 여러개 등록할 경우 다른 jQuery 플러그인의 메서드와 중복될 가능성이 높아진다. (JavaScript는 기본적으로 overloading을 지원하지 않는다.)
- ❖ 실행 메서드의 첫 번째 인자가 객체일 경우 인스턴스 생성으로 간주한다.
ex) `$(selector).dialog({ width: 500 });`
- ❖ 실행 메서드의 첫 번째 인자가 문자열일 경우 메서드 실행으로 간주한다. 두 번째 인자부터는 파미터로 사용된다.

```
ex) $(selector).dialog('close');  
    $(selector).dialog('option', param1, param2, ...);  
    $(selector).dialog('option', 'height');
```

- ❖ 예외적으로 메서드를 여러 개 노출해 놓는 경우도 있다.

```
ex) $(selector).jqGrid();  
    $(selector).jqGrid('setCaption', param1);  
    $(selector).setCaption(param1);
```

```
ex) $.fn.tooltip = function( method ) {  
    if ( methods[method] ) {  
        // 메서드 실행  
        return methods[method].apply( this, Array.prototype.slice.call( arguments, 1 ));  
    } else if ( typeof method === 'object' || ! method ) {  
        // 인스턴스 생성  
        return methods.init.apply( this, arguments );  
    } else {  
        // 아니면 오류  
        $.error( 'Method ' + method + ' does not exist on jQuery.tooltip' );  
    }  
};
```

jQuery 원본 소스 분석 시 여러가지 제약조건을 잘 고려한다

- ❖ button의 type([button|reset|submit])은 기본값이 submit이다.
- ❖ input text가 하나만 있을 때 text안에서 엔터를 누르면 submit이 일어난다.
- ❖ 객체의 property 참조 시 object.propertyName 또는 object['propertyName'] 으로 참조할 수 있다.
- ❖ 객체에 length(number), splice(function) 프로퍼티가 있을 경우 console은 이 객체를 마치 array처럼 출력한다.
 - ✓ <http://stackoverflow.com/questions/7261670/what-makes-a-jquery-object-show-up-as-an-array-in-chromes-developer-tools>
- ❖ 즉시실행함수 작성시 인자를 전달하면 함수 내부에서 전달된 인자의 값을 보장할 수 있고, 지역변수로 사용되므로 좀 더 빠르다.

ex) // 이 익명함수 내에서 사용되는 \$는 jQuery임이 보장된다. (단, 지역변수로 \$를 재정의 하지 않은 경우)

```
$ = 'test';
(function($) {
    console.log($);
})(jQuery);
```

// 이 익명함수 내에서 사용되는 \$는 window임이 보장된다. (단, 지역변수로 \$를 재정의 하지 않은 경우)

```
$ = 'test';
(function($) {
    console.log($);
})(window);
```

jQuery 원본 소스 분석 시 메소드의 특징을 잘 알고 사용해야 한다

- ❖ \$.fn.extend 메서드는 객체 복사 기능을 한다. 첫 번째 인자에 두 번째 인자의 내용이 복사(replace)된다.

```
ex) var a = { v: 10 }, b = { v: 11 }, c { v: 12 };  
    var d = $.extend({}, a, b, c); // d === { v: 12 };  
    $.extend(a, b, c); // a === { v: 12 };
```

- ❖ Function은 내부에 arguments라는 시스템 변수를 갖고 있다.

- ✓ arguments는 인자의 개수, 인자의 값 등의 정보를 갖고 있다.
- ✓ 인자는 배열과 같이 인덱스로 꺼내올 수 있지만 arguments가 배열인 것은 아니다.

```
ex) console.log(arguments.constructor);  
    console.log([].constructor);
```

- ✓ arguments를 배열로 변환할 때는 Array의 slice 함수를 이용하도록 한다.
 - [oArray=]Array.slice(start[, end])
 - Array.prototype.slice는 두개의 인덱스를 인자로 받으며 인덱스 부분의 요소를 Array로 복사하여 반환한다.
 - 인자가 없을 경우 복사된 Array를 반환한다.
 - 두번째 인자인 end를 생략할 경우 length - 1까지의 요소를 반환한다.

```
ex) var args = Array.prototype.splice.call(arguments);
```

jQuery 원본 소스 분석 시 Function vs Function의 Instance 차이를 잘 알고 사용해야 한다

❖ Function과 Function의 인스턴스의 차이

▪ Function

- ✓ Java, C++ 등의 OOP 언어에서의 Class와 유사한 역할을 한다.
- ✓ new 키워드를 사용하여 실행하면 인스턴스가 반환된다.
ex) `var obj = new Object();`
- ✓ Function은 prototype이라는 특별한 객체 공간을 갖고 있다.
- ✓ prototype에 등록된 프로퍼티와 메서드는 이 Function의 모든 인스턴스가 참조, 공유한다.
- ✓ 브라우저 내장 console 객체에서 prototype의 내용을 확인할 수 있다.

ex) `console.dir($);`

`console.log($.prototype);`

▪ Function 인스턴스

- ✓ new 키워드와 함께 실행된 Function의 결과이다.
- ✓ Function의 prototype을 참조, 사용할 수 있다.
- ✓ prototype에 등록된 메서드 실행시 자동으로 자신을 컨텍스트로 전달하여 this에 바인딩된다.
- ✓ 브라우저 내장 console 객체에서 인스턴스가 사용할 수 있는 메서드(프로퍼티)를 `__proto__`에서 확인할 수 있다.

<!-- __proto__는 prototype chain을 보여주는 브라우저 고유의 변수이며, 각 브라우저마다 구현체가 다를 수 있으니 로직에서 사용은 하지 않도록 한다.

ex) `console.log($('body'));`

❖ \$.extend, \$.trim, \$.each와 같이 jQuery 인스턴스 \$(selector)에서 사용할 수 있는 모든 메서드는 jQuery.fn(=== jQuery.prototype)에 등록되어 있다.

❖ jQuery를 실행하면 jQuery.fn.init 함수의 인스턴스를 생성하여 반환하는데, jQuery.fn.init.prototype은 jQuery.fn을 참조하고 있다.

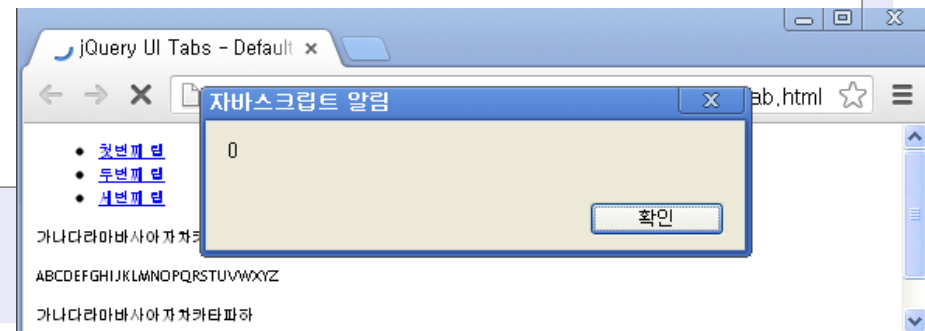
- ✓ <https://groups.google.com/forum/?fromgroups=#!topic/jquery-en/K9aNKgW4PtM>

8. _CreateWidget function의 **_create();** 메소드를 확장하여 customizing 한다

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

```
(function($) {  
    // 라이브러리 확인 .....  
  
    // 닫기 버튼 구현  
    $.widget('ui.tabs', $.ui.tabs, {  
        options: .....  
  
        // ① _create();를 추가  
        _create: function() {  
            // 컨텍스트(this === 위젯객체)를 지역변수로 참조할 수 있게 선언  
            var self = this;  
  
            // ② alert();로 함수 호출 여부 확인  
            alert(0);  
  
            // ③ 원본 _create 함수 호출 (로직 확장 후 원본 method를 호출해주기 위함)  
            this._super();  
        }  
    });  
})(jQuery);
```



3.10 _createCloseButton 메서드 호출 위치 결정 jQuery UI Tab 기능 확장

9. 추가할 _createCloseButton 메서드 호출 위치를 결정한다

- ① this._super() 전/후 어디에 **확장 로직을 넣을지 결정**하기 위하여 **원본 Tab Widget 로직을 분석**한다.
- ② **this._super() 후에 로직을 넣기로 결정**

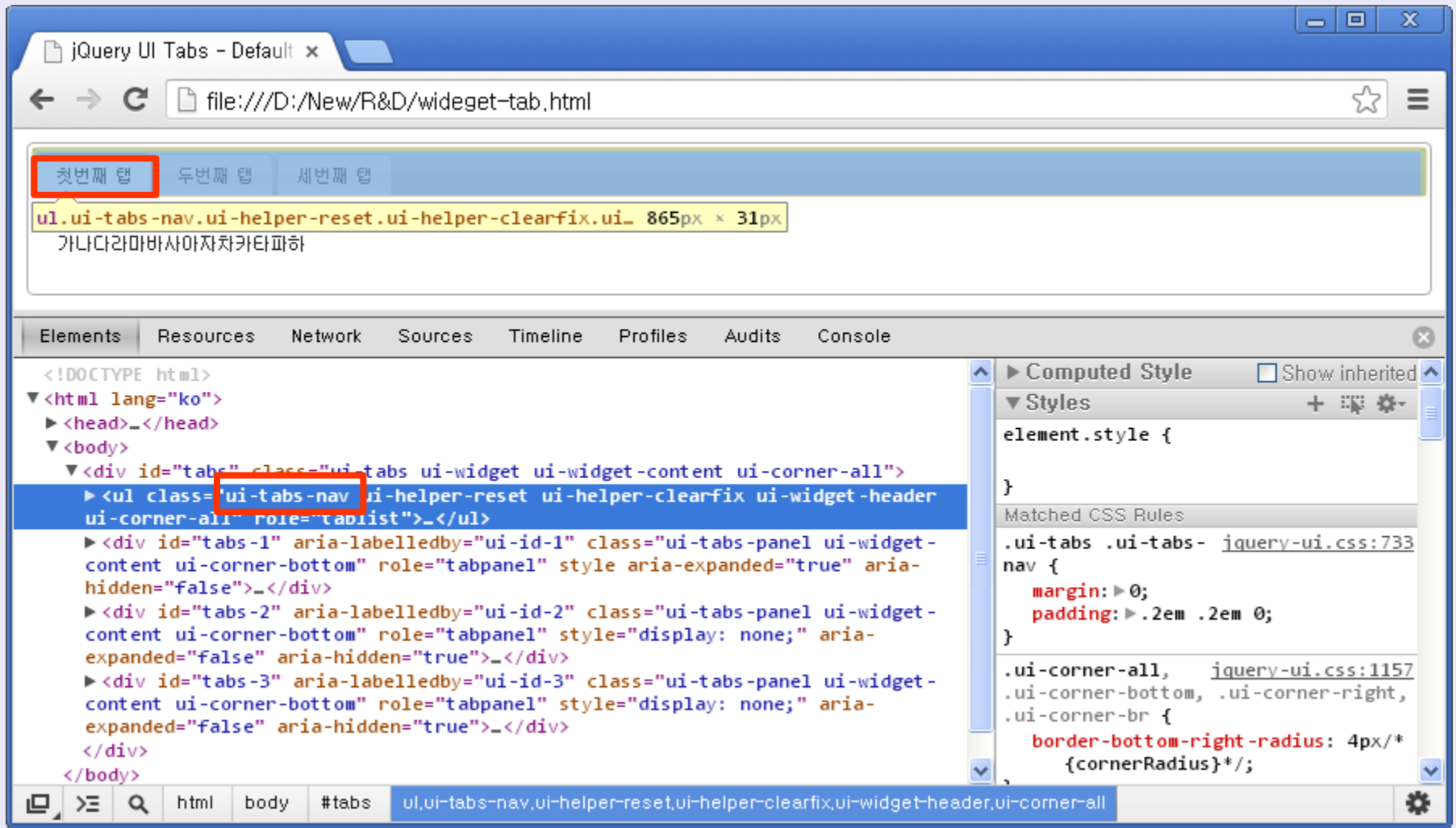
■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

```
(function($) {  
    // 라이브러리 확인 .....  
  
    // 닫기 버튼 구현  
    $.widget('ui.tabs', $.ui.tabs, {  
        options: .....  
  
        _create: function() {  
            // 컨텍스트(this === 위젯객체)를 지역변수로 참조할 수 있게 선언  
            var self = this;  
  
            // 원본 _create 함수 호출  
            this._super();  
  
            // 추가할 닫기 버튼 로직 구현 위치  
        }  
    });  
})(jQuery);
```


10. 개발자도구로 **Close Button**을 넣을 대상 **Element**를 판단하여 기능 확장(메소드 추가)

① id, class명으로 판단하여 "ui-tabs-nav" 클래스를 채택



11. **소스를 분석**하여 _processTabs를 수정하기로 결정한다.

■ 소스 분석

jquery-ui-1.10.1.custom.js

```
(function( $, undefined ) {
.....
$.Widget.prototype = {
.....
$.widget( "ui.tabs", {
.....
_create: function() {
.....
    this._processTabs(); //11510 line
.....
},
// ① jquery-ui-1.10.1.custom.js 파일 내 원본 소스를 분석하여 _processTabs 을 확장하기로 결정
_processTabs: function() { //11809 line
.....
// ② li 엘리먼트에 확장하기로 결정 (this.tabs)
this.tabs = this.tablist.find( "> li:has(a[href])" ) //11816 line
    .addClass( "ui-state-default ui-corner-top" )
    .attr({
        role: "tab",
        tabIndex: -1
    });
},
.....
}
})( jQuery );
```

11. 소스를 분석하여 _processTabs를 수정하기로 결정한다.

③ 개발자도구로 li 엘리먼트의 구조를 확인한다.

The screenshot shows a web browser window with a jQuery UI Tabs widget. The browser's address bar shows the file path: `file:///D:/New/R&D/wideget-tab.html`. The tabs widget has three tabs: "첫번째 탭", "두번째 탭", and "세번째 탭". The first tab is active, and its content is "가나다라마바사아자차카타파하".

The browser's developer tools are open, showing the "Elements" panel. The HTML structure is as follows:

```
<!DOCTYPE html>
<html lang="ko">
  <head>_</head>
  <body>
    <div id="tabs" class="ui-tabs ui-widget ui-widget-content ui-corner-all">
      <ul class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-all" role="tablist">
        <li class="ui-state-default ui-corner-top ui-tabs-active ui-state-active" role="tab" tabindex="0" aria-controls="tabs-1" aria-labelledby="ui-id-1" aria-selected="true">_</li>
        <li class="ui-state-default ui-corner-top" role="tab" tabindex="-1" aria-controls="tabs-2" aria-labelledby="ui-id-2" aria-selected="false">_</li>
        <li class="ui-state-default ui-corner-top" role="tab" tabindex="-1" aria-controls="tabs-3" aria-labelledby="ui-id-3" aria-selected="false">_</li>
      </ul>
      <div id="tabs-1" aria-labelledby="ui-id-1" class="ui-tabs-panel ui-widget-content ui-corner-bottom" role="tabpanel" style="aria-expanded="true" aria-hidden="false">_</div>
    </div>
  </body>
</html>
```

The "Computed Style" panel on the right shows the styles for the selected element, `li.ui-state-default.ui-corner-top.ui-tabs-active.ui-state-active`. The styles include:

- `margin-bottom: -1px;`
- `padding-bottom: 1px;`
- `list-style: none;`
- `float: left;`
- `position: relative;`

12. 대상 엘리먼트(tab)가 여러 개이므로 **each** 함수로 닫기 버튼 소스를 구현한다.

- ① this.tabs.each(function) 형태로 each 함수를 사용하여 소스 로직을 구현한다.
- ② 브라우저에서 Element가 정상적으로 추가되었는지 확인한다.
- ③ 버튼 이벤트 구현 로직을 등록하기 전에 함수 안에서 alert(0)을 찍어 이벤트 정상작동 확인 : \$b.click(function())
- ④ jQueryUI 사이트 Tab Widget API를 찾아서 구현하려고 하는 함수(method)의 존재 유무 확인 (비슷한거 있으면 복사해서 활용)

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

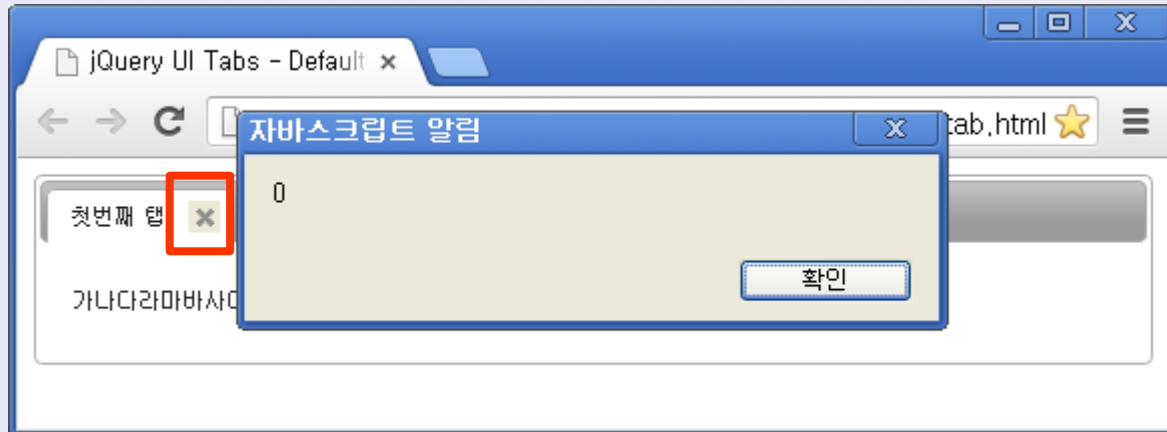
```
.....
// 각각의 탭에 닫기 버튼 추가 로직 구현 위치
this.tabs.each(function() {
    // 버튼 생성
    var $b = $('<button class="custom-tabs-close ui-icon ui-icon-closethick">close</button>');

    // 탭에 버튼 추가
    $(this).append($b);    //소스 추가는 append 메소드로 처리한다.

    // 버튼에 클릭 이벤트 등록
    $b.click(function() {
        // 버튼 클릭 여부 확인
        alert(0);
    });
});
.....
```

12. 대상 엘리먼트(tab)가 여러 개이므로 each 함수로 닫기 버튼 소스를 구현한다.

① 버튼 이벤트 구현 로직을 등록하기 전에 함수 안에서 **alert(0)**을 찍어 이벤트 정상작동 확인



13. 닫기 버튼 구현을 위해 **원본 소스를 분석**하여 유사한 기능처리 패턴을 참조한다.

① remove 메소드를 사용하기로 결정

■ 소스 분석

jquery-ui-1.10.1.custom.js

```
(function( $, undefined ) {  
.....  
$.Widget.prototype = {  
.....  
  $.widget( "ui.tabs", {  
    .....  
    _create: function() {  
      .....  
      this._processTabs(); //11510 line  
      .....  
    },  
    _processTabs: function() { //11809 line  
      .....  
      this.tabs = this.tablist.find( "> li:has(a[href])" ) //11816 line  
      .addClass( "ui-state-default ui-corner-top" )  
      .attr({  
        role: "tab",  
        tabIndex: -1  
      });  
    },  
    .....  
  }  
})( jQuery );
```

14. 닫기 버튼 이벤트 로직을 구현한다. (Coding & Debugging)

- ① 컨텍스트(this === tab 위젯)를 지역변수로 미리 선언 : self 변수 (이벤트를 등록하기 위한 this가 tab 위젯이 아니었음)
- ② remove()를 self에 구현
- ③ each()에서 던져주는 index를 인자로 사용. => index값이 오류 => index 값을 받아오기로 함
- ④ 부모의 (tab) 몇 번째 인덱스인지 알아봄
- ⑤ 로직 삽입 후 test : remove 에 index 값 적용

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

```
.....
this.tabs.each(function() {
    .....
    // 버튼에 클릭 이벤트 등록
    $b.click(function() {
        // 부모 tabs을 찾아서 몇번째 인덱스에 위치하는지 확인
        var index = $(this).parent().index();
        console.log(index); // 개발자도구 console 화면에서 확인

        // 탭 삭제
        var panelId = $( this ).closest( "li" ).remove().attr( "aria-controls" );
        console.log(panelId);
        $( "#" + panelId ).remove();

        // 탭 삭제
        //self.remove(index); // self Object의 remove를 못 찾고 있음
    });
});
.....
```

15. 닫기 버튼 확장 소스 영역을 묶어서 **함수로 처리**한다.

① 확장 소스 영역을 함수로 처리 : `this._createCloseButton()`; // 함수명에 `_`는 private 함수를 의미함

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

```
(function($) {  
    .....  
    $.widget('ui.tabs', $.ui.tabs, {  
        .....  
        _create: function() {  
            var self = this;    // 컨텍스트(this === 위젯객체)를 지역변수로 참조할 수 있게 선언  
            this._super();    // 원본 _create 함수 호출  
            this._createCloseButton();    // 추가한 _createCloseButton 메서드 호출  
        },  
  
        _createCloseButton: function() {  
            this.tabs.each(function() {  
                var $b = $('                $(this).append($b);  
                $b.click(function() {  
                    var index = $(this).parent().index();    // 부모 tabs을 찾아서 몇번째 인덱스에 위치하는지 확인  
                    console.log(index);    // 개발자도구 console 화면에서 확인  
                    // 탭 삭제  
                    var panelId = $( this ).closest( "li" ).remove().attr( "aria-controls" );  
                    console.log(panelId);  
                    $( "#" + panelId ).remove();  
                });  
            });  
        }  
    });  
})(jQuery);
```


최종적인 **nest.ui.tabs-1.0.0.js** 파일의 소스 내용은 아래와 같다.

```

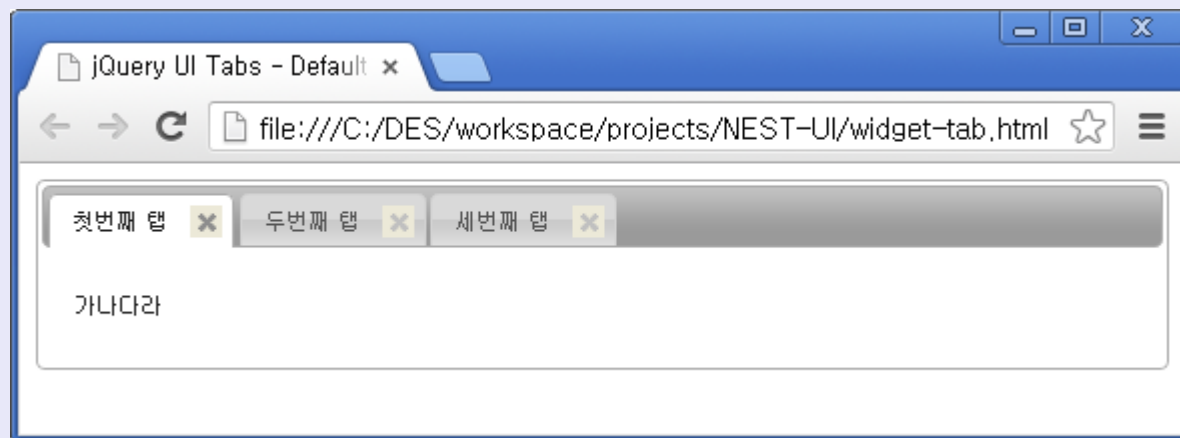
1) // 확장 스크립트 임포트 확인
2) // alert(0);

3) (function($) {
4)
5)     // 라이브러리 확인
6)     if (!$.ui || !$$.ui.tabs) {
7)         return false;
8)     }
9)
10)    // 닫기 버튼 구현
11)    // ADD_OPTION closeButton
12)    // EXT_METHOD _create
13)    // ADD_METHOD _createCloseButton
14)    $.widget('ui.tabs', $.ui.tabs, {
15)        options: {
16)            closeButton: false
17)        },
18)
19)        _create: function() {
20)            // 컨텍스트(this === 위젯객체)를 지역변수로 참조할 수
            있게 선언
21)            var self = this;
22)
23)            // 함수 호출 여부 확인
24)            // alert(0);
25)
26)            // 원본 _create 함수 호출
27)            this._super();
28)
29)            // 추가한 _createCloseButton 메서드 호출
30)            this._createCloseButton();
31)        },
32)
33)        // 추가한 _createCloseButton 메서드
34)        // 이 메서드에서 닫기 버튼을 추가한다.
35)        _createCloseButton: function() {
36)            // 각각의 탭에 닫기 버튼 추가
37)            this.tabs.each(function() {
38)                // 버튼 생성
39)                var $b = $('<button class="custom-tabs-close ui-
                    icon ui-icon-closethick">close</button>');
40)
41)                // 탭에 버튼 추가
42)                $(this).append($b);
43)
44)                // 버튼에 클릭 이벤트 등록
45)                $b.click(function() {
46)                    // 버튼 클릭 여부 확인
47)                    // alert(0);
48)
49)                    // 부모 tabs을 찾아서 몇번째 인덱스에 위치하는지
                    확인
50)                    var index = $(this).parent().index();
51)                    console.log(index);
52)
53)                    // 탭 삭제
54)                    var panelId =
                    $( this ).closest( "li" ).remove().attr( "aria-controls" );
55)                    console.log(panelId);
56)                    $( "#" + panelId ).remove();
57)
58)                    // 탭 삭제
59)                    //self.remove(index);
60)                });
61)            }
62)        });
63)
64)    })(jQuery);

```

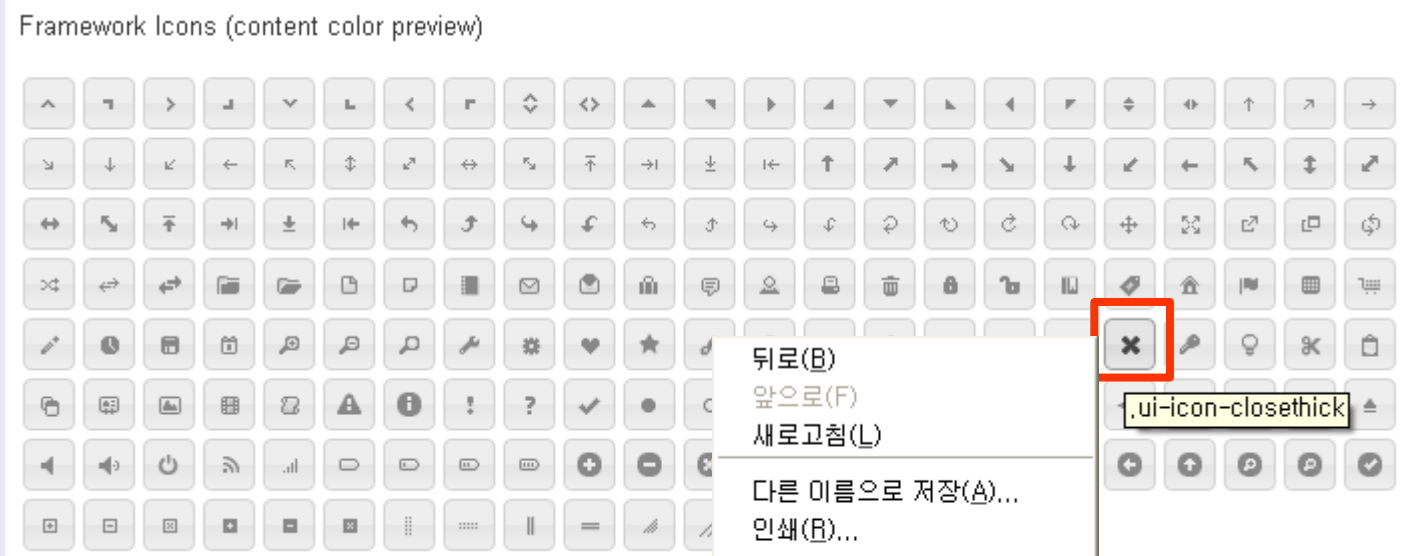
16. 브라우저에서 구현 **기능 Test**

- ① Tab 버튼 누르기
- ② Tab의 닫기 버튼 누르기
- ③ 이미지 확인(모양/위치/색상)



17. 닫기 버튼의 **디자인**을 원하는 모양으로 바꿔준다.

- ① jQueryUI 사이트(jQueryui.com)의 Themes 메뉴에서 Framework icon을 선택한다.
: .ui-icon-closethick 클래스를 선택
- ② Class 명을 복사하려면, 대상 아이콘 위에서 오른쪽 버튼 클릭 > 요소검사 > class 명 복사



```
><li class="ui-state-default ui-corner-all" title=".ui-icon-close">_</li>
▼<li class="ui-state-default ui-corner-all" title=".ui-icon-closethick">
  <span class="ui-icon ui-icon-closethick"></span>
</li>
▶<li class="ui-state-default ui-corner-all" title=".ui-icon-key">_</li>
```

17. 닫기 버튼의 디자인을 원하는 모양으로 바꿔준다.

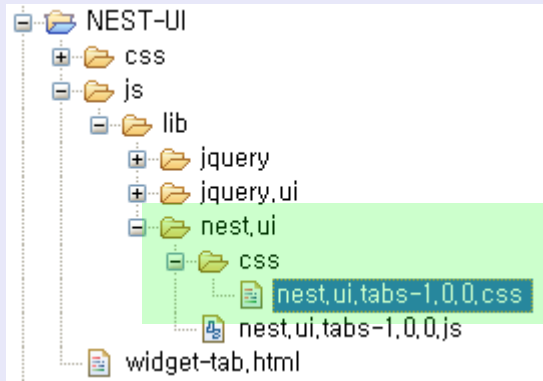
③ 소스에 class명을 inline으로 삽입한다.

■ 소스 내용

/js/lib/nest.ui/nest.ui.tabs-1.0.0.js 의 일부

```
_createCloseButton: function() {  
    this.tabs.each(function() {  
        // 버튼 생성  
        var $b = $('<button class="custom-tabs-close ui-icon ui-icon-closethick        .....  
    });  
}
```

④ CSS 파일을 별도의 파일(nest.ui.tabs-1.0.0.css)로 생성한다. (6.5 기능 확장 js, css 파일 생성 참고)



17. 닫기 버튼의 디자인을 원하는 모양으로 바꿔준다.

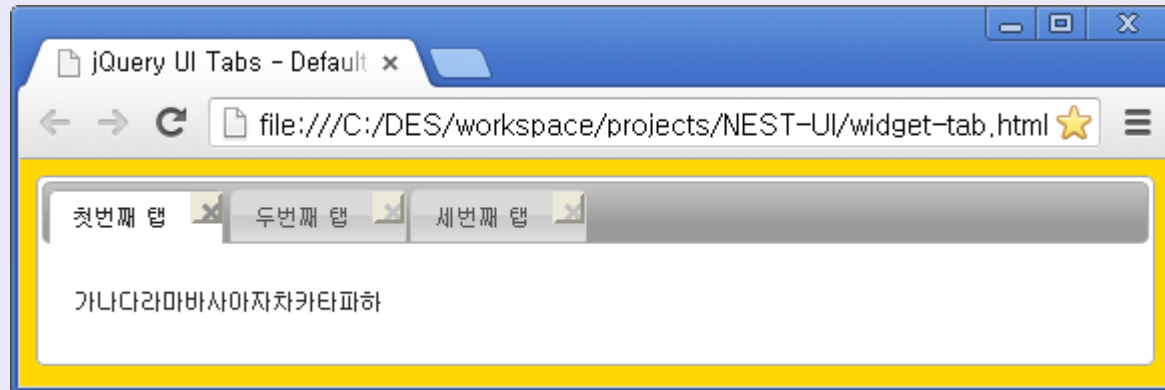
⑤ 신규 CSS 파일에 `body { background: gold; }` 를 입력하여 파일이 정상 등록되었는지 확인한다.

■ 소스 내용

/js/lib/nest.ui/css/nest.ui.tabs-1.0.0.css 의 일부

```
/* CSS import 확인 */  
body { background: gold; }
```

⑥ 브라우저에 아래와 같이 표현되면, 정상적으로 CSS파일이 import된 것이다.



17. 닫기 버튼의 디자인을 원하는 모양으로 바꿔준다.

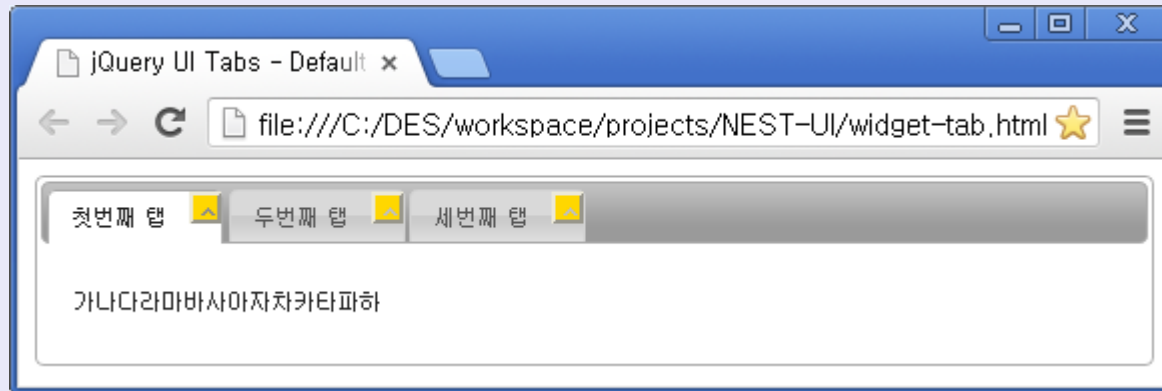
⑦ 대상 객체에 CSS 적용되는지 확인한다.

■ 소스 내용

/js/lib/nest.ui/css/nest.ui.tabs-1.0.0.css 의 일부

```
/* 대상 객체 CSS 적용 확인 */  
.ui-icon-closethick { background: gold; }
```

⑧ 브라우저에 아래와 같이 표현되면, 정상적으로 대상 객체에 적용된 것이다.



17. 닫기 버튼의 디자인을 원하는 모양으로 바꿔준다.

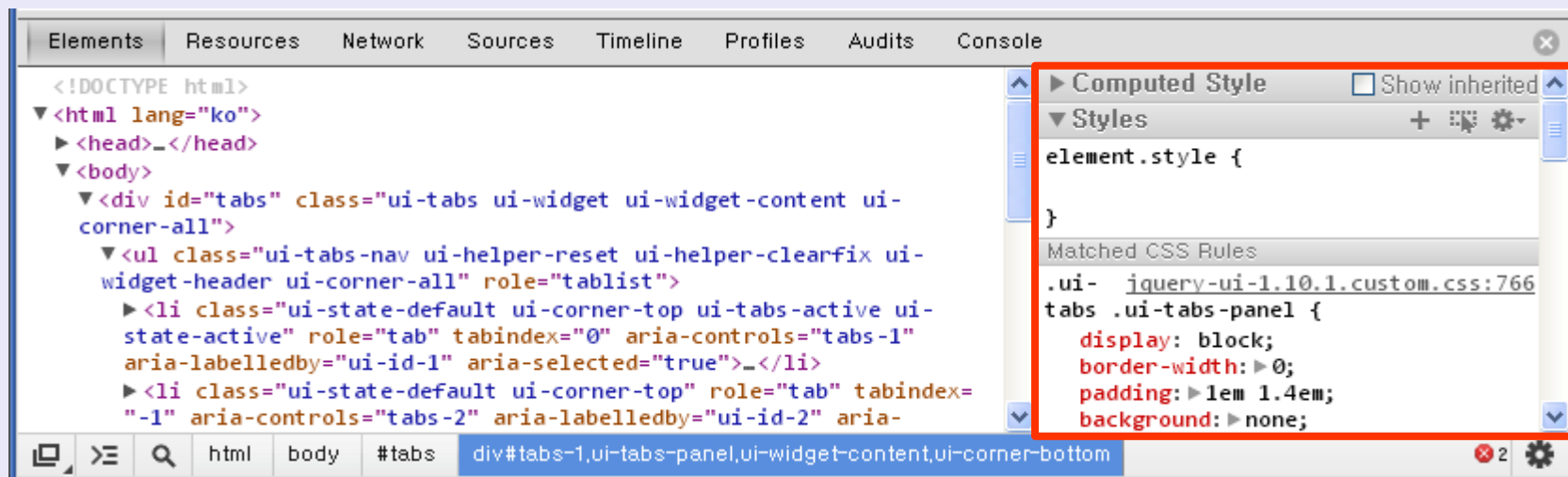
⑨ 닫기 버튼의 위치/테두리 등을 조절한다.

: 개발자 도구에서 CSS값을 직접 변경해서 확인 후 소스에 등록해준다.(margin 값으로 위치 수정)
 element.style { }

■ 소스 내용

/js/lib/nest.ui/css/nest.ui.tabs-1.0.0.css 의 일부

```
/* jQuery UI 에서 사용하는 흔한 클래스명을 확장할 경우 그 영향력이 너무 커진다. */
.ui-icon-closethick { margin-top: 5px; margin-right: 5px; border: 0; }
```



17. 닫기 버튼의 디자인을 원하는 모양으로 바꿔준다.

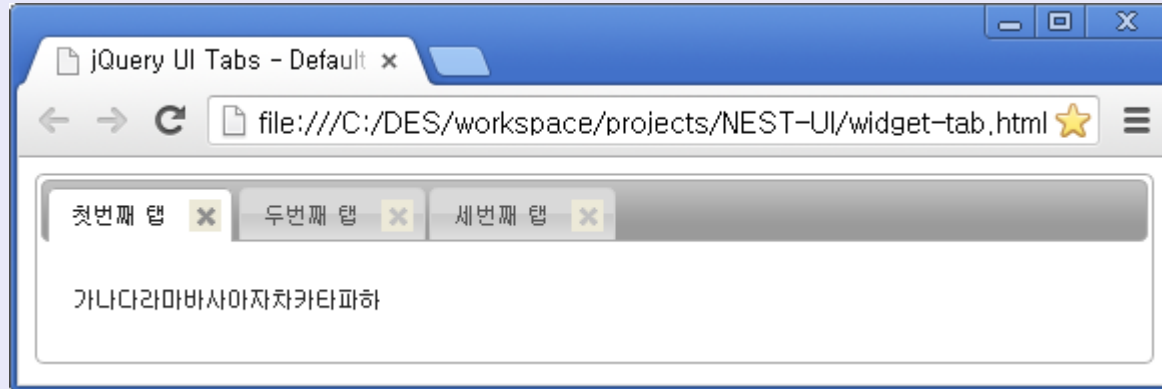
⑩ 기존의 클래스명(.ui-icon-closethick)은 jQuery가 확장되었을 경우 변경 위험이 있으므로 **Class명을 변경**해준다.

: **CSS 클래스명 충돌을 피하기 위해 수정 부분만을 별도 클래스로 정의**한다. (공백으로 클래스명을 여러 개 나열 시 여러 개 클래스가 중복으로 적용된다.)

■ 소스 내용

/js/lib/nest.ui/css/nest.ui.tabs-1.0.0.css 의 일부

```
.custom-tabs-close { margin-top: 5px; margin-right: 5px; border: 0; }
```



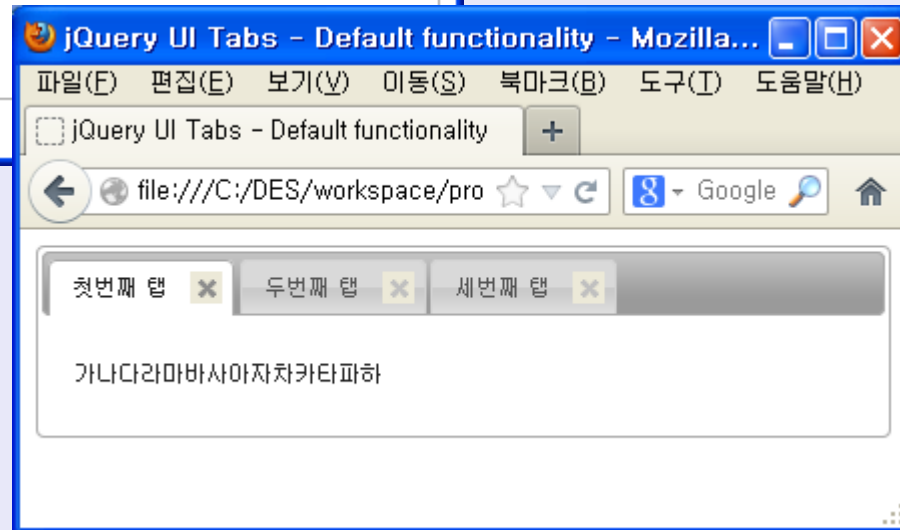
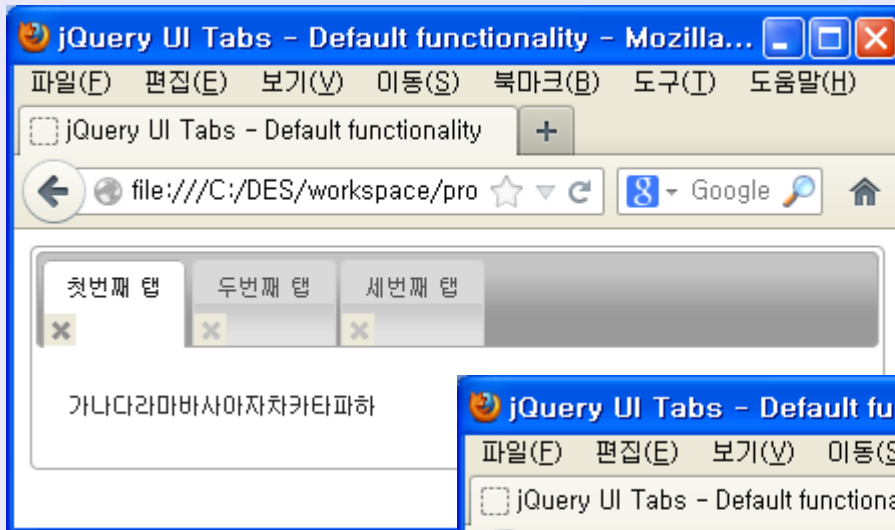
최종적인 nest.ui.tabs-1.0.0.css 파일의 소스 내용은 아래와 같다.

- 1) /* CSS import 확인
- 2) body { background: gold; }
- 3) */
- 4) /* 대상 객체 CSS 적용 확인
- 5) .ui-icon-closethick { background: gold; }
- 6) */
- 7) /* jQuery UI 에서 사용하는 흔한 클래스를 확장할 경우 그 영향력이 너무 커진다.
- 8) .ui-icon-closethick { margin-top: 5px; margin-right: 5px; border: 0; }
- 9) */
- 10) .custom-tabs-close { margin-top: 5px; margin-right: 5px; border: 0; }

Firefox에서는 Element 배열에 대한 **상세한 정의**를 추가해주어야 한다.

- css에 float: right; 옵션을 추가한다.

```
.custom-tabs-close { float: right; margin-top: 5px; margin-right: 5px; border: 0; }
```



IE에서는 Debug 모드가 실행 되어야 **console 객체**가 생성된다

– js 소스의 console.log를 주석처리 한다. (캐시 파일이 남아 있을 때는 오류 창이 보이지 않는다.)

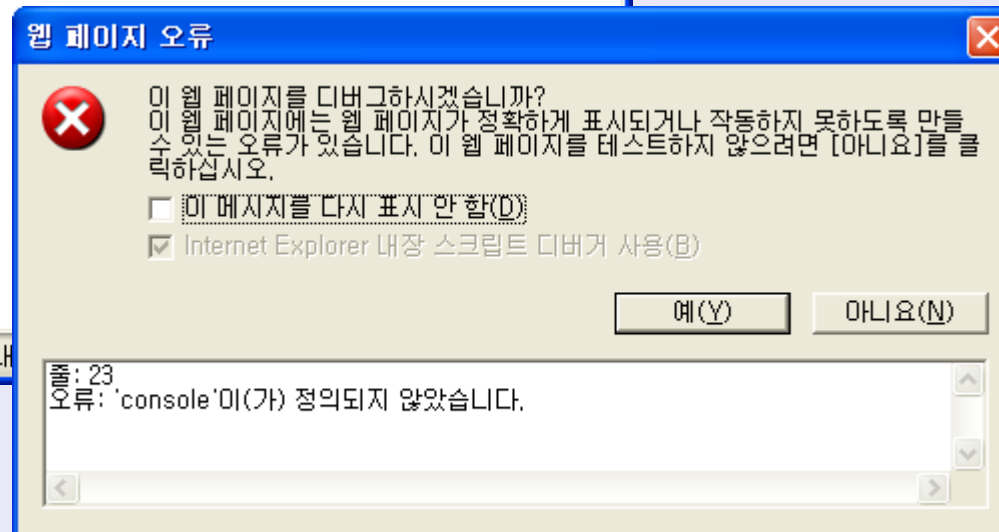
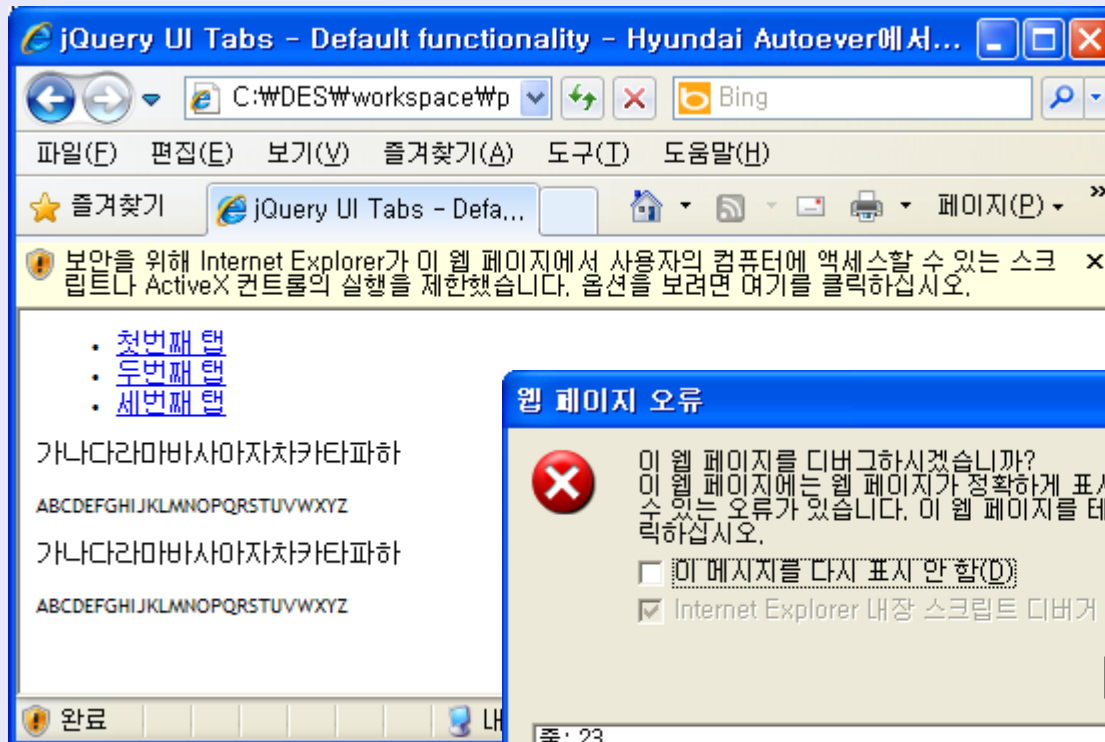
※ 주석처리

```
//console.log(self);
```

```
//console.log(index);
```

```
//console.log(self);
```

```
//console.log(panelId);
```



4. jqGrid - Spring jsonView 연동

1. 서버단 jsonView 설정
2. 서버단 리턴값 셋팅
3. jqGrid 설정
4. jqGrid 구현

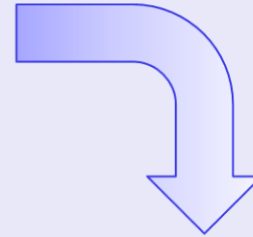
4.1 jsonView 적용 예제

4. jsonView 연동

List Sample

| No | 카테고리ID | 카테고리명 | 사용여부 | Description | 등록자 |
|----|--------------|-------------|------|-----------------------|-----|
| 1 | SAMPLE-00028 | testtest | Y | testtest | 보너스 |
| 2 | SAMPLE-00027 | Grid Test15 | Y | test complete | 종결자 |
| 3 | SAMPLE-00026 | Grid Test13 | Y | Grid Test13 땃 ㄷ러 ㄷ러 | 갑동이 |
| 4 | SAMPLE-00025 | Grid Test11 | Y | Grid Test11 adfsfds | 갑순이 |
| 5 | SAMPLE-00016 | Grid Test14 | Y | Grid Test14 | 김갑동 |
| 6 | SAMPLE-00015 | Grid Test12 | Y | Grid Test12 오래오래 | 빙튀기 |
| 7 | SAMPLE-00014 | Grid Test10 | Y | 잘 들어가고 있음 | 유관순 |
| 8 | SAMPLE-00013 | Grid Test9 | Y | 많이 한 듯 | 이성계 |
| 9 | SAMPLE-00012 | Grid Test8 | Y | Grid Test 2페이지 패 | 태권도 |
| 10 | SAMPLE-00011 | Grid Test7 | | | |

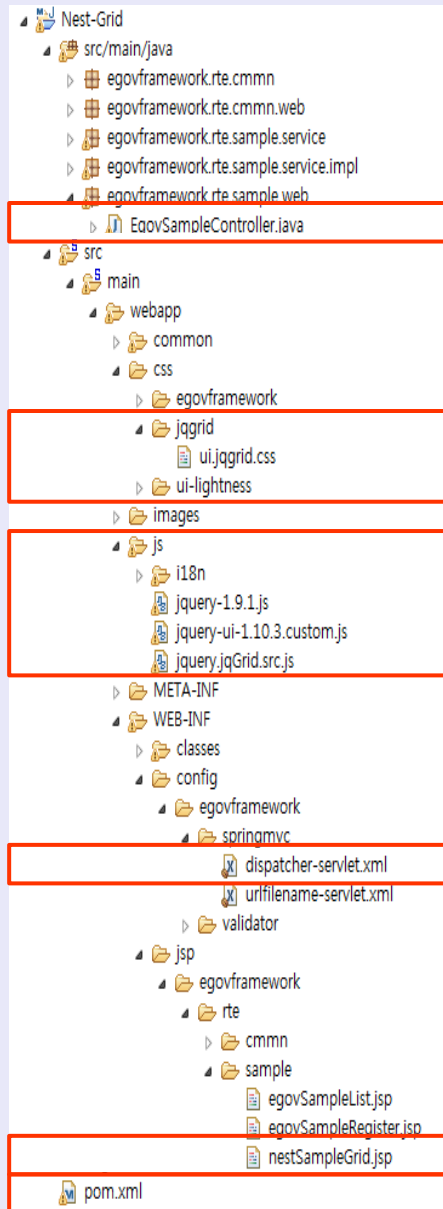
easyCompany 프로젝트의 List 예제를 jqGrid를 적용하여 변환(with jsonView)



Data Grid Sample

| Nest Grid Caption | | | | | | |
|-------------------|--------------|-------------|------|-----------------------|-----|--|
| No | 카테고리ID | 카테고리명 | 사용여부 | Description | 등록자 | |
| 1 | SAMPLE-00028 | testtest | Y | testtest | 보너스 | |
| 2 | SAMPLE-00027 | Grid Test15 | Y | test complete | 종결자 | |
| 3 | SAMPLE-00026 | Grid Test13 | Y | Grid Test13 땃 ㄷ러 ㄷ러 | 갑동이 | |
| 4 | SAMPLE-00025 | Grid Test11 | Y | Grid Test11 adfsfds | 갑순이 | |
| 5 | SAMPLE-00016 | Grid Test14 | Y | Grid Test14 | 김갑동 | |

1. jsonView 설정 : json data 통신 설정
2. 서버단 업무 구현 : URL 요청 처리
3. jqGrid 설정 : jqGrid 사용 환경 설정
4. 그리드 업무 구현 : 업무 화면 개발



2. EgovSampleController.java

3. jqGrid 의 CSS 파일

3. jqGrid 의 Library 파일

1. jsonView 설정

4. 업무 화면 개발

1. jsonView 설정

C:\WeGovFrameDev-2.5.0-FullVer\workspace\W-Nest-Grid\pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project .....>
  <modelVersion>4.0.0</modelVersion>
  <groupId>Nest-Grid</groupId>
  <artifactId>Nest-Grid</artifactId>
  <packaging>war</packaging>
  <version>1.0.0</version>
  <name>Nest-Grid</name>
  <url>http://maven.apache.org</url>
  <properties>.....</properties>
  <repositories>.....</repositories>
  <dependencies>
```

<!-- jsonView 설정 -->

<dependency>

<groupId>net.sf.json-lib</groupId>

<artifactId>json-lib-ext-spring</artifactId>

<version>1.0.2</version>

</dependency>

.....

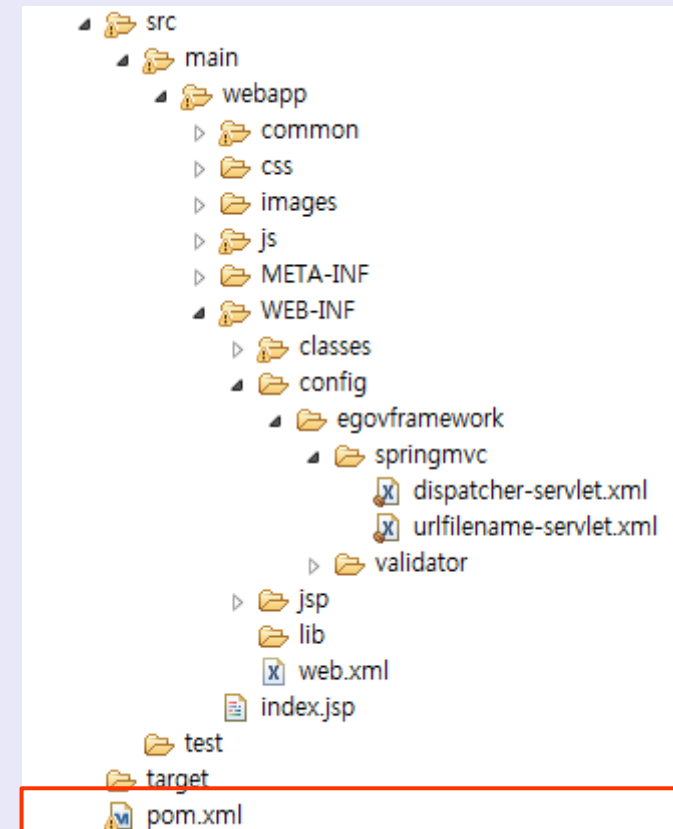
</dependencies>

<build>.....</build>

<reporting>.....</reporting>

</project>

jsonView Library 추가



4.3 jsonView 설정 – dispatcher-servlet.xml

WEB-INF>config>egovframework>springmvc>dispatcher-servlet.xml

WEB-INF>config>egovframework>springmvc>dispatcher-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DispatcherServlet application context for Application's web tier. -->
<beans ..... >
```

```
.....

<!-- jsonView 설정 -->
```

```
<bean id="jsonView" class="net.sf.json.spring.web.servlet.view.JsonView">
```

```
  <property name="contentType" value="application/json;charset=UTF-8" /> <!-- 한글 -->
```

```
</bean>
```

```
<bean id="beanNameResolver" class="org.springframework.web.servlet.view.BeanNameViewResolver">
```

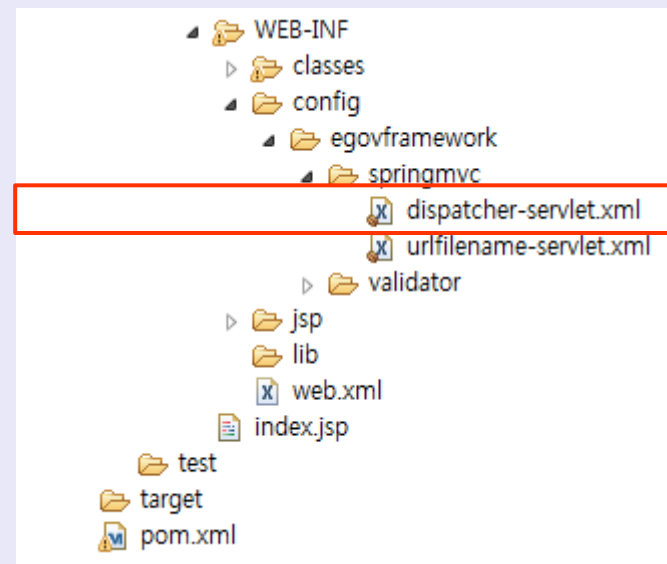
```
  <property name="order" value="0"/>
```

```
</bean>
```

```
.....
```

```
</beans>
```

BeanNameViewResolver에
리턴값이 jsonView인 값들을
Mapping



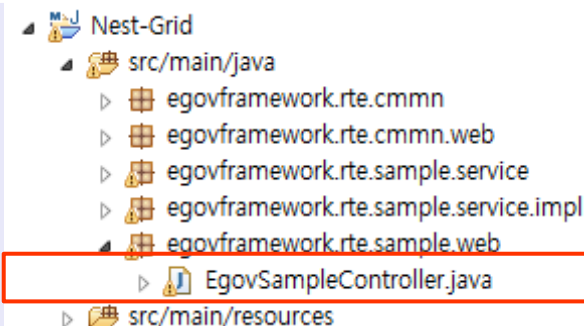
그리드 화면에서 호출하는 URL에 대한 응답 로직 구현

```
package egovframework.rte.sample.web;
import .....
@Controller
@SessionAttributes(types=SampleVO.class)
public class EgovSampleController {
```

```
.....
@RequestMapping(value="/sample/egovSampleList.do") // 글 목록 조회
public String selectSampleList(.....) throws Exception {
    글 목록 조회
    return "/sample/egovSampleList";
}
```

```
@RequestMapping(value="/sample/nestSampleGridList.do") // 그리드 화면 호출
public String selectSampleGridList(@ModelAttribute("searchVO") SampleDefaultVO searchVO, ModelMap model)
    throws Exception {
    return "/sample/nestSampleGrid";
}
```

```
@RequestMapping(value="/sample/nestSampleGrid.do") // 그리드 데이터 조회
public String selectSampleGrid(@ModelAttribute("searchVO") SampleDefaultVO searchVO,
    ModelMap model,
    @RequestParam(value = "page", required = false, defaultValue = "1") int page,
    @RequestParam(value = "rows", required = false, defaultValue = "5") int rows,
    @RequestParam(value = "sidx", required = false, defaultValue = "5") String sidx,
    @RequestParam(value = "sord", required = false, defaultValue = "5") String sord)
    throws Exception {
    .....
    return "jsonView";
}
```

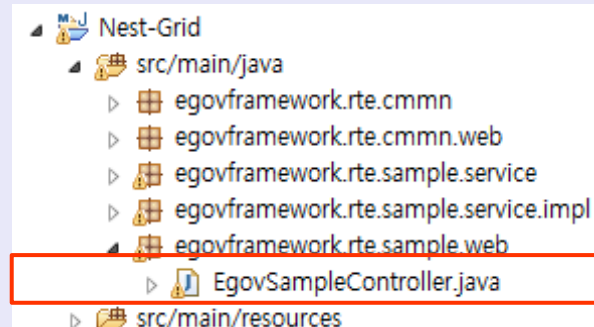


그리드로 구현될 화면을 호출하는 URL 구현

```
/**
 * Grid Data 화면을 호출한다.
 * @param
 * @return "/sample/nestSampleGridList"
 * @exception Exception
 */
@RequestMapping(value="/sample/nestSampleGridList.do")
public String selectSampleGridList(@ModelAttribute("searchVO") SampleDefaultVO searchVO,
                                   ModelMap model)
    throws Exception {

    return "/sample/nestSampleGrid";
}

.....
```



그리드에 뿌려줄 데이터를 읽어오는 URL 구현

```
/**
 * jqGrid Grid Data를 조회한다.
 * @param searchVO - 조회할 정보가 담긴 SampleDefaultVO
 * @param model
 * @return "jsonView"
 * @exception Exception
 */
```

```
@RequestMapping(value="/sample/nestSampleGrid.do") // 데이터 조회
```

```
public String selectSampleGrid(@ModelAttribute("searchVO") SampleDefaultVO searchVO,
```

```
    ModelMap model,
```

```
    @RequestParam(value = "page", required = false, defaultValue = "1") int page, // 몇번째 페이지
```

```
    @RequestParam(value = "rows", required = false, defaultValue = "5") int rows, // 페이지당 행 수
```

```
    @RequestParam(value = "sidx", required = false, defaultValue = "5") String sidx, // Sort 기준 컬럼
```

```
    @RequestParam(value = "sord", required = false, defaultValue = "5") String sord) // desc/asc
```

```
    throws Exception {
```

```
// Query문 Paging 처리
```

```
searchVO.setPageUnit(rows); // easyCompany 소스의 Paging 변수
```

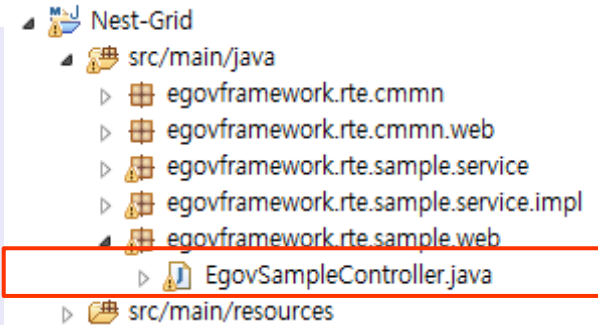
```
searchVO.setFirstIndex((page - 1) * rows); // easyCompany 소스의 Paging 변수
```

```
searchVO.setRecordCountPerPage(page * rows); // easyCompany 소스의 Paging 변수
```

```
// 데이터 조회 : selectSampleList() 메소드와 동일
```

```
final List<SampleVO> sampleList = sampleService.selectSampleList(searchVO);
```

```
.....
```



그리드에 뿌려줄 데이터를 읽어오는 URL 구현

```
.....

// 총 건수
int totalSize = sampleService.selectSampleListTotCnt(searchVO);

// Paging 관련 연산
int pageSize = 0;
if (totalSize != 0) {
    if ((totalSize % rows) == 0) {
        pageSize = (totalSize / rows);
    } else {
        pageSize = (totalSize / rows + 1);
    }
}

// 그리드에 뿌려줄 데이터 셋팅
model.addAttribute("page", page);           // 몇번째 페이지
model.addAttribute("total", pageSize);      // 전체 페이지수
model.addAttribute("records", totalSize);   // 전체 행수
model.addAttribute("rows", sampleList);     // 그리드 데이터

return "jsonView";
}
```

서버에서 만들어주는 json data의 형태에 따라 단말단에서 읽는 방법이 다름

■ 서버단 리턴

```
model.addAttribute("page", page);
model.addAttribute("total", pageSize);
model.addAttribute("records", totalSize);
model.addAttribute("rows", sampleList);
```

■ json data

```
{
  "total": "4",
  "page": "1",
  "records": "20",
  "rows": [
    {"id": "SAMPLE-00028", "name": "testdata28", ... },
    {"id": "SAMPLE-00027", "name": "testdata27", ... },
    ...
  ],
  "searchVO": "....."
}
```

※ Debugging – Data 분석

■ Grid 소스

```
jsonReader: {
  repeatitems: false,
  root:'rows',
  page:'page',
  total:'total',
  records:'records',
  id:'id'
},
```

The screenshot shows the Chrome DevTools Network tab. A request to `/Nest-Grid/sample` is highlighted. The 'Preview' sub-tab displays the JSON response:

```
{total:4, records:20, page:1,...}
page: 1
records: 20
rows: [{id:SAMPLE-00028, name:testtest, description:testte
       0: {id:SAMPLE-00028, name:testtest, description:testtest.
       1: {id:SAMPLE-00027, name:Grid Test15, description:test (
       2: {id:SAMPLE-00026, name:Grid Test13, description:Grid 1
       3: {id:SAMPLE-00025, name:Grid Test11, description:Grid 1
       4: {id:SAMPLE-00016, name:Grid Test14, description:Grid 1
searchVO: {firstIndex:0, lastIndex:1, pageIndex:1, pageSiz
total: 4
```

서버에서 만들어주는 json data의 형태에 따라 단말단에서 읽는 방법이 다름

■ 서버단 리턴

```
model.addAttribute("resultList", new GridPageVO<SampleVO>(sampleList, page, pageSize, totalSize));
```

■ json data

```
{
  "resultList": {
    "total": "4",
    "page": "1",
    "records": "20",
    "rows": [
      {"id": "SAMPLE-00028", "name": "testdata28", ... },
      {"id": "SAMPLE-00027", "name": "testdata27", ... },
      ...
    ]
  },
  "searchVO": "....."
}
```

■ Grid 소스

```
jsonReader: {
  repeatitems: false, //서버에서 받은 data와 Grid 상의 column 순서를 맞추것인지?
  root: function (obj) { return obj.resultList.rows; }, //root는 default로 "rows"를 찾아서 읽는다.
  page: function (obj) { return obj.resultList.page; },
  total: function (obj) { return obj.resultList.total; },
  records: function (obj) { return obj.resultList.records; },
  id:'id' // 키값, 꼭 화면 전체에서 겹치는 값이 없는 column을 설정할 것, 안그러면 에러 발생
}, //페이징 처리에 관련된 파라미터 등
```

Grid의 리턴 클래스를 하나로 만들어서 보낼 때 예시

```
package egovframework.rte.sample.service;
import java.util.List;
public class GridPageVO<T> {
    private List<T> rows;
    private int page;
    private int total;
    private int records;

    public GridPageVO() {
    }

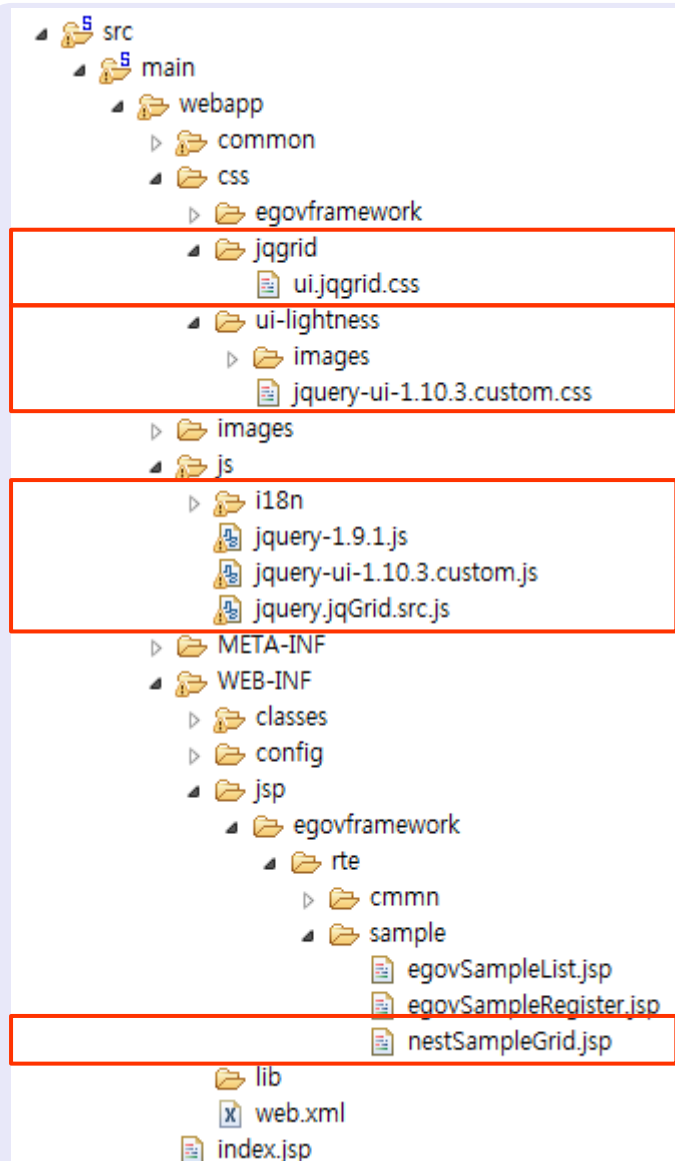
    public GridPageVO(List<T> rows, int page, int total, int records) {
        this.rows = rows;
        this.page = page;
        this.total = total;
        this.records = records;
    }

    public List<T> getRows() {
        return rows;
    }

    public void setRows(List<T> rows) {
        this.rows = rows;
    }

    ..... //일반적인 Value Object와 동일함
}
```

화면에서 jqGrid를 사용할 수 있도록 Library와 CSS 파일을 import 한다.



※ 다운로드 받은 jqGrid Library(jQuery.jqGrid-4.4.4.zip)를 압축을 풀어 js 폴더의 파일들은 js 폴더에, css 폴더의 파일들은 css 폴더에 집어 넣는다.
※ js/i18n 폴더에는 다국어를 지원하기 위한 파일들이 들어간다. (jQuery UI도 동일함. DatePicker 참조)

3. jqGrid 제공 CSS

2. jQueryUI 제공 CSS

i18n : jqGrid 제공 다국어 문자 Set

1. jQuery Library
2. jQueryUI Library
3. jqGrid Library

jqGrid를 적용할 JSP 파일

4.7 jqGrid 실행 Library 다운로드

jqGrid 압축 파일을 다운로드 받는다. (jquery.jqGrid-4.4.4.zip)

http://www.trirand.com/blog/?page_id=6 사이트에서 다운로드 받는다.

The image shows two screenshots of the jqGrid website. The left screenshot displays the 'Downloads' section, which includes a table for selecting modules to download. The right screenshot shows the 'Other modules' section, where the 'Download' button is highlighted with a red box.

Downloads
January 31st, 2013
jqGrid 4.4.4 Downloads (Free, Open Source package)
Customize your jqGrid download by selecting the specific modules you need.
Last development can be get from [GitHub](#) and [Demo files \(4.x\)](#)

Base		
Grid base	<input checked="" type="checkbox"/> jqGrid	Base grid. In order to use this file should be loaded before language files are located for download
Formatter	<input checked="" type="checkbox"/> Formatter	This module allows a pre

Other modules

Table to Grid	<input checked="" type="checkbox"/> Table to Grid	This function convert existing table to grid
---------------	---	--

jQuery UI addon methods

jQuery UI addons	<input checked="" type="checkbox"/> Additional methods using jQuery UI library	Methods that using jQuery UI. These includes sortable columns, sortable rows, drag and drop rows between tables, column chooser and grid resizing
------------------	--	---

Download

jqGrid Older Release
jQuery Grid Plugin 4.4.3 – last version which support IE6
jQuery Grid Plugin 3.8.2
jQuery Grid Plugin 3.7.2
jQuery Grid Plugin 3.6.5
jQuery Grid Plugin 3.5.3
Patch for 3.4.4
jQuery Grid Plugin 3.4.4
jQuery Grid Plugin 3.4.3 jqGrid 3.4.x Documentation (pdf format)
Demo files for 3.4.x

Blogroll
[jqGrid in GitHub](#)
[jqGrid Wiki](#)
[jQuery](#)

Categories
[jQuery](#)

Archives
[January 2013](#)
[November 2012](#)
[August 2012](#)
[June 2012](#)
[May 2012](#)
[April 2012](#)
[February 2012](#)
[December 2011](#)
[October 2011](#)
[July 2011](#)
[June 2011](#)
[April 2011](#)

4.7 jqGrid 실행 Library 다운로드

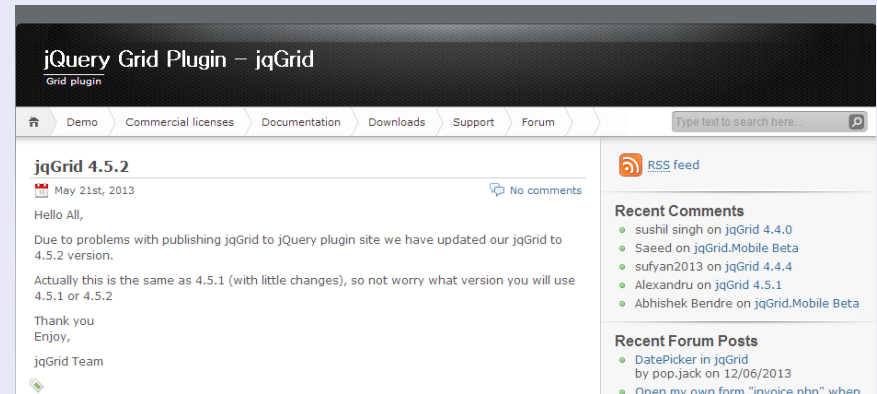
4. jsonView 연동

jqGrid 적용 시 참고 할 사이트

<http://jquery.com/>

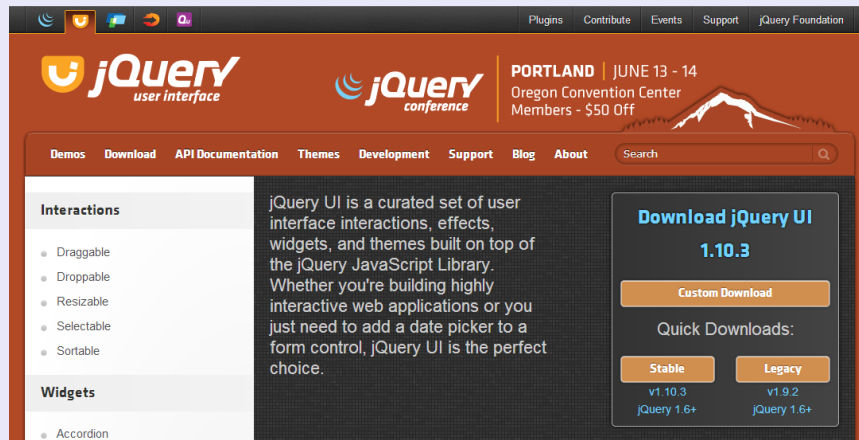


<http://www.trirand.com/blog/>



Download, Wiki, Demo 제공

<http://jqueryui.com/>



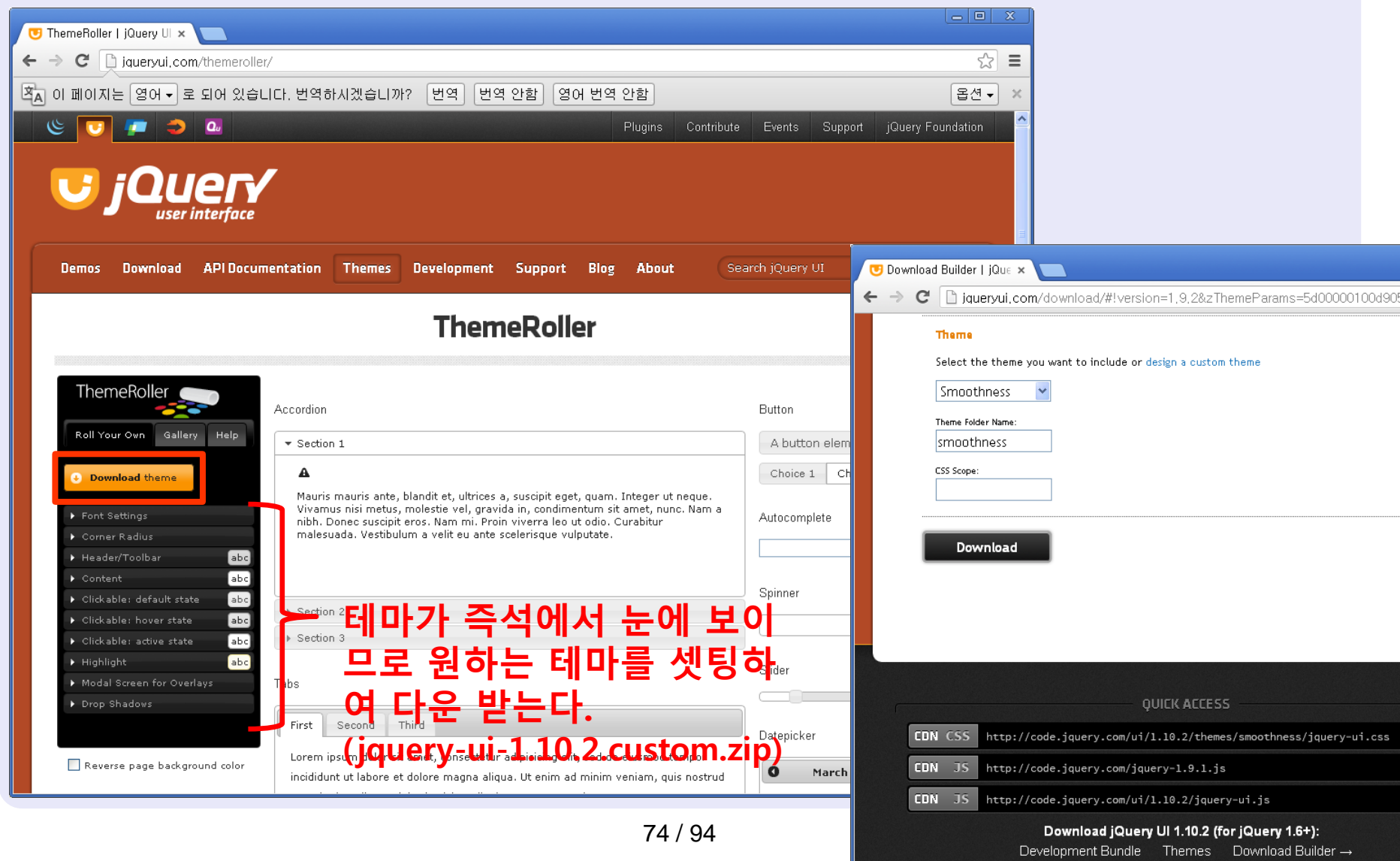
<http://www.anyframejava.org/>



업무 적용 샘플 제공(IAM)

jqGrid 테마 파일을 다운로드 받는다.

<http://jqueryui.com/themeroller/> 사이트에서 테마 파일을 다운로드 받는다. (jQuery UI 동일)



테마가 즉석에서 눈에 보이므로 원하는 테마를 선택하여 다운 받는다.
(jquery-ui-1.10.2.custom.zip)

egovSampleList.jsp 파일을 복사하여 nestSampleGrid.jsp 파일을 만들고 아래와 같이 수정

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="ui" uri="http://egovframework.gov/ctl/ui"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<% ... %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Basic Board List</title>
<link type="text/css" rel="stylesheet" href="<c:url value='/css/egovframework/sample.css' />" />

<script type="text/javascript" language="javascript" defer="defer">
<!--
function fn_egov_select(id) {...} // 글 수정 화면 function
function fn_egov_addView() {...} // 글 등록 화면 function
function fn_egov_selectList() {...} // 글 목록 화면 function
function fn_egov_link_page(pageNo){...} // pagination 페이지 링크 function
-->
</script>

</head>
```

} 모두 삭제

egovSampleList.jsp 파일을 복사하여 nestSampleGrid.jsp 파일을 만들고 아래와 같이 수정

```
<body style="text-align:center; margin:0 auto; display:inline; padding-top:100px;">
<form:form commandName="searchVO" name="listForm" method="post">
<input type="hidden" name="selectedId" />
<div id="content_pop">
  <!-- 타이틀 -->
  <div id="title">
    <ul>
      <li> List Sample </li>
    </ul>
  </div>
  <!-- // 타이틀 -->
  <div id="search">
    <ul>
      <li>
        <form:select path="searchCondition" cssClass="use">
          <form:option value="1" label="Name" />
          <form:option value="0" label="ID" />
        </form:select>
      </li>
      <li><form:input path="searchKeyword" cssClass="txt"/> </li>
      <li><span class="btn_blue_l"> <a href="javascript:fn_egov_selectList();"> <spring:message code="button.search" /> </li>
    </ul>
  </div>
  <!-- List -->
  <div id="table">
    Grid로 대체
  </div>
  아래 부분 삭제
</div>
</form:form>
</body>
</html>
```

nestSampleGrid.jsp 파일 Head 부분 (import 순서 중요)

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="ui" uri="http://egovframework.gov/ctl/ui"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Data Grid 실습(jqGrid)</title>

<link type="text/css" rel="stylesheet" href="<c:url value='/css/egovframework/sample.css'/>" />
<link type="text/css" rel="stylesheet" href="<c:url value='/css/jqgrid/ui.jqgrid.css'/>" />
<link type="text/css" rel="stylesheet" href="<c:url value='/css/ui-lightness/jquery-ui-1.10.3.custom.css'/>" />
<script src="<c:url value='/js/jquery-1.9.1.js'/>" type="text/javascript"></script>
<script src="<c:url value='/js/jquery-ui-1.10.3.custom.js'/>" type="text/javascript"></script>
<script src="<c:url value='/js/i18n/grid.locale-en.js'/>" type="text/javascript"></script>
<script src="<c:url value='/js/jquery.jqGrid.src.js'/>"></script>

</head>
```

Google Ajax API CDN (Also supports SSL via HTTPS)

<https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js>
<http://code.google.com/apis/ajaxlibs/documentation/index.html#jquery>

Microsoft CDN (Also supports SSL via HTTPS)

<http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js>

jQuery CDN (via Media Temple)

<http://code.jquery.com/jquery-1.7.2.min.js>
<http://code.jquery.com/jquery-1.7.2.js>

jQuery 파일은 Site Link 보다는 Local 저장을 권장 (Site 다운이 잦음)

```
<script type="text/javascript" src="http://code.jquery.com/jquery-1.7.2.min.js"></script>
```

nestSampleGrid.jsp 파일 HTML 태그 부분

```
<body>
<form:form commandName="searchVO" name="listForm" method="post">
<input type="hidden" name="selectedId" />
<div id="content_pop">
<!-- 타이틀 -->
<div id="title">
    <ul>
        <li>" Data Grid Sample </li>
    </ul>
</div>
<!-- // 타이틀 -->
<div id="search">
    <ul>
        <li>
            <form:select id="searchCondition" path="searchCondition" cssClass="use">
                <form:option value="1" label="Name" />
                <form:option value="0" label="ID" />
            </form:select>
        </li>
        <li><form:input id="searchKeyword" path="searchKeyword" cssClass="txt"/> </li>
        <li><span class="btn_blue_l"><a href="javascript:fn_egov_selectGrid();"><spring:message code="button.search" /></a>
    </div>
<!-- List -->
<div id="table">
    <table id="list"></table>
    <div id="pager"></div>
</div>
<!-- /List -->
</div>
</form:form>
```

} **Grid 구현부**
(Grid와 Paging이 들어갈 위치)

nestSampleGrid.jsp 파일 JavaScript 부분 (해당 화면에서만 사용하는 JavaScript는 HTML 태그 아래에 위치)

```
<script>
$(function() {
    var lastsel;

    $('#list').jqGrid({
        url: "<c:url value='/sample/nestSampleGrid.do'/>", // 서버주소
        postData : {
            searchCondition:$("#searchCondition").val(),
            searchKeyword:$("#searchKeyword").val()
        }, // URL에서 검색 조건으로 가져갈 파라미터
        mtype:'POST', // 전송 타입
        datatype : "json", // 받는 데이터 형태
        colNames:[ 'No', '카테고리ID', '카테고리명', '사용여부', 'Description', '등록자' ], //테이블의 제목
        colModel:[
            { name: 'rownum', index: 'rownum', width: 40 },
            { name: 'id', index: 'id', width: 100 },
            { name: 'name', index: 'name', width: 150, editable:true, edittype:'text' },
            { name: 'useYn', index: 'useYn', width: 80, align:'center' },
            { name: 'description', index: 'description', width: 200, editable:true, edittype:'text' },
            { name: 'regUser', index: 'regUser', width: 60, align:'center' }
        ], //서버에서 받은 데이터 설정
        jsonReader: {
            repeatitems: false, //서버에서 받은 data와 Grid 상의 column 순서를 맞출것인지?
            root:'rows', //서버의 결과 내용에서 데이터를 읽어오는 기준점
            page:'page', // 현재 페이지 currentPage
            total:'total', // 총 페이지 수 totalPage
            records:'records', // 보여지는 데이터 갯수(레코드) totalRecord
            id:'id' // 키값, 꼭 화면 전체에서 겹치는 값이 없는 column을 설정할 것, 안그러면 에러 발생
        }, //페이징 처리에 관련된 파라미터 등
    });
});
```

컬럼명

컬럼양식

nestSampleGrid.jsp 파일 JavaScript 부분 (해당 화면에서만 사용하는 JavaScript는 HTML 태그 아래에 위치)

```
pager: '#pager', //navigation bar가 들어갈 위치..
//toppager: true, //상단에 페이저 추가, 밑에랑 2개가 된다.
pagerpos: 'center', //페이징 위치
pgbuttons: true, //페이징 버튼 유무
pginput: true, //페이징 인풋입력 유무
rownumbers:true, //행 번호 표시 true/false, 전체 row의 줄번호(서버 데이터는 아니다..)를 보여줄까??
rownumWidth:40, //줄번호 column의 사이즈..
height:180, //테이블의 세로 크기, Grid의 높이
loadtext:'Loading...',//서버 연동시 보여주는 문자열..
emptyrecords: "Nothing to display",
rowNum: 5, //표시되는 행의 갯수, 초기화면에 찍을 row 갯수..
rowList: [5, 10, 15, 20], //유저에 의해서 row 갯수 지정하는 콤보박스 문자열..
sortname: 'id',
sortorder: "desc",
viewrecords: true, //페이징 처리시 전체 행의 개수 표시, '총 몇건 중에서 몇건을 보고 있다' 등의 문자열 출력 여부
gridview : true, // 선표시 true/false
//loadonce: true, //DB 조회를 하지 않고 현재 불러온 데이터로 처리(기본값 false)
loadComplete: function(data){
    var couponuseStats = data.dataList;
    alert("Grid Data 출력 완료 couponuseStats = " + data.records);
},
loadError:function(xhr,status,error){
    alert("서버 통신 실패 >> " + error + ", 통신응답 = " + xhr.responseText);
},
onPaging:function(){
    alert("페이징 시작 시 함수 작동");
},
```

※ 만약 loadComplete를 타고 들어갔다면, 서버에 요청은 제대로 됐지만, json이 잘못 됐을 것이고, loadError는 서버를 못탄 것이다.

nestSampleGrid.jsp 파일 JavaScript 부분 (해당 화면에서만 사용하는 JavaScript는 HTML 태그 아래에 위치)

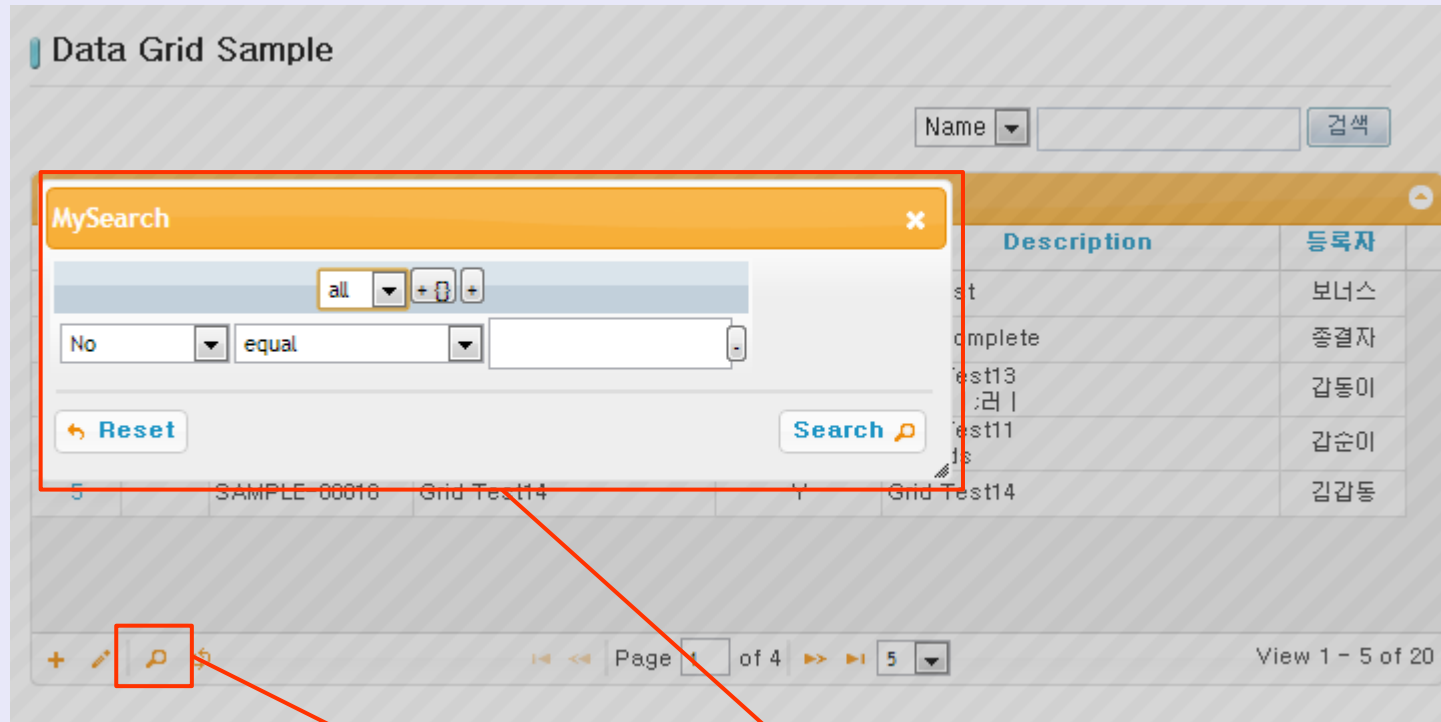
```
onSelectRow:function(id){
    if(id && id != lastsel){
        $('#list').jqGrid('restoreRow',lastsel);
        $('#list').jqGrid('editRow',id,true);
        lastsel=id;
    }
},
editurl:"<c:url value='/sample/nestSampleGridUpdate.do'/>", //edit 버튼을 눌렀을 때 적용되는 URL(미구현)
caption: "Nest Grid Caption" //테이블 제목, title 문자열..
}).navGrid('#pager',{edit:true,add:true,del:false,search:true,refresh:true,position:'left'}, // pager area에 버튼 추가
{, // edit 버튼 옵션
{, // add 버튼 옵션
{, // del 버튼 옵션
{
    caption:'MySearch',//dialog title
    Find:'Search',
    Reset:'Reset',
    groupOps:[{op:"AND",text:"all"},{op:"OR",text:"any"}],
    multipleSearch:true, //여러 컬럼 검색 가능
    multipleGroup:true //여러 조건을 조합 가능하게..
},
{}); //jqGrid의 하단의 추가, 수정, 삭제, 검색, 조회 버튼들의 활성화 여부
});
```

nestSampleGrid.jsp 파일 JavaScript 부분 (해당 화면에서만 사용하는 JavaScript는 HTML 태그 아래에 위치)

```
function fn_egov_selectGrid() { // ※ 검색 버튼을 눌렀을 때 처리방법
    $("#list").clearGridData(); // 이전 데이터 삭제
    jQuery("#list").jqGrid('setGridParam', {
        postData: {
            searchCondition:$("#searchCondition").val(),
            searchKeyword:$("#searchKeyword").val()
        },
        datatype:"json"
    }).trigger("reloadGrid");
}
</script>
</body>
</html>
```

4.11 Grid 안에서 검색

Paging Navigation에서 제공하는 Search 버튼은 Grid Data 에서 검색 기능을 제공



Search 버튼과 검색 팝업

jqGrid 로 데이터를 한번 뿌려준 후 다시 호출하면 반응이 없다.

jqGrid는 Library 소스 안에 기존 그리드가 존재하면 멈추는 로직이 있기 때문에, 동일 그리드에 데이터를 다시 뿌려주기 위해서는 **먼저 이전 데이터를 삭제**해주어야 한다.

■ 소스 내용

/js/jquery.jqGrid.src.js

```
641 $.fn.jqGrid = function( pin ) {
642     if (typeof pin == 'string') {
643         var fn = $.jgrid.getAccessor($.fn.jqGrid,pin);
644         if (!fn) {
645             throw ("jqGrid - No such method: " + pin);
646         }
647         var args = $.makeArray(arguments).slice(1);
648         return fn.apply(this,args);
649     }
650     return this.each( function() {
651         if(this.grid) {return;} <----- 기존에 그리드가 존재 하면 텅김
652
653         var p = $.extend(true,{
654             url: "",
655             .....
656         }
657     )
658 }
```



`$("#list").clearGridData();` // 이전 데이터 삭제

5. 기타

5.1 브라우저별 특징

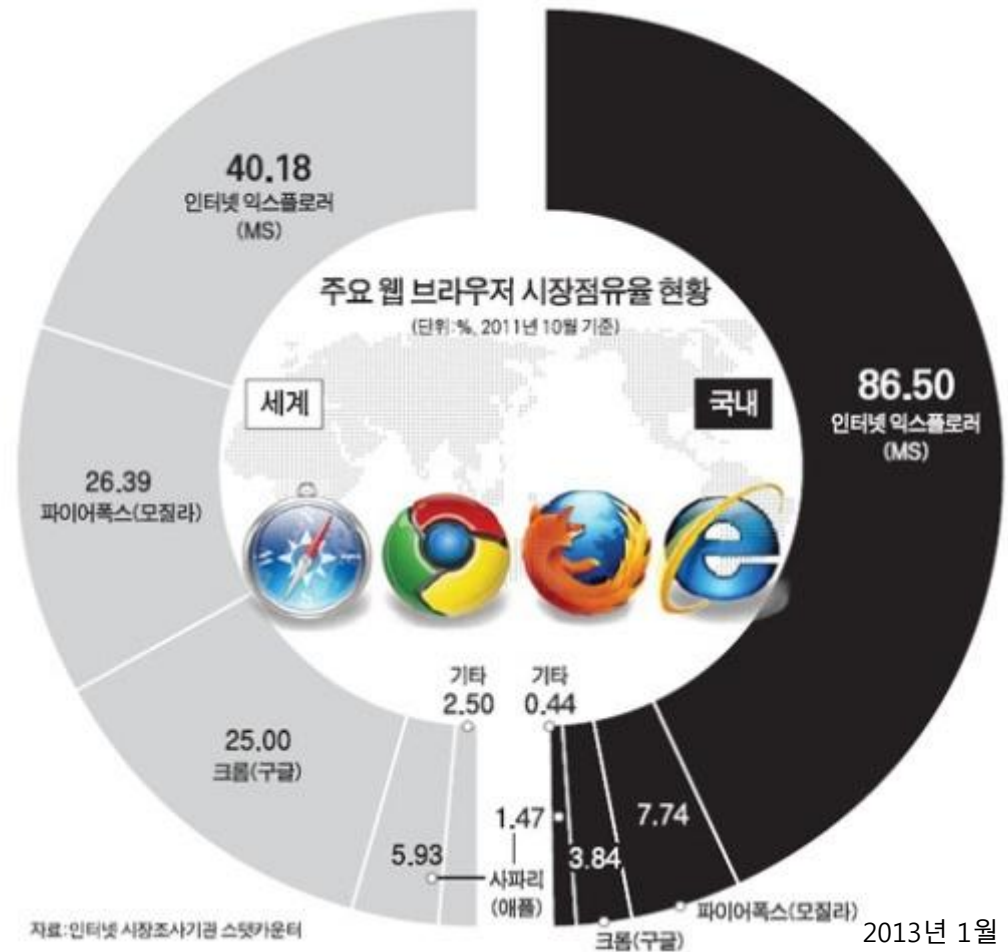
5. 기타

웹 브라우저의 대표 제품은 Chrome, Opera, Firefox, Safari, IE 가 있다

- ❖ 주요 브라우저들이 모두 HTML5를 앞다투어 지원하면서 Active-X 구현 기능들을 JavaScript로 대체
- ❖ 과거 JavaScript는 디버깅의 어려움으로 개발자들이 꺼려하는 언어.(막강한 기능의 개발자도구 제공)



2012년 4월



2013년 1월

Crome 브라우저에서 디버깅 방법 및 개발자 도구 활용

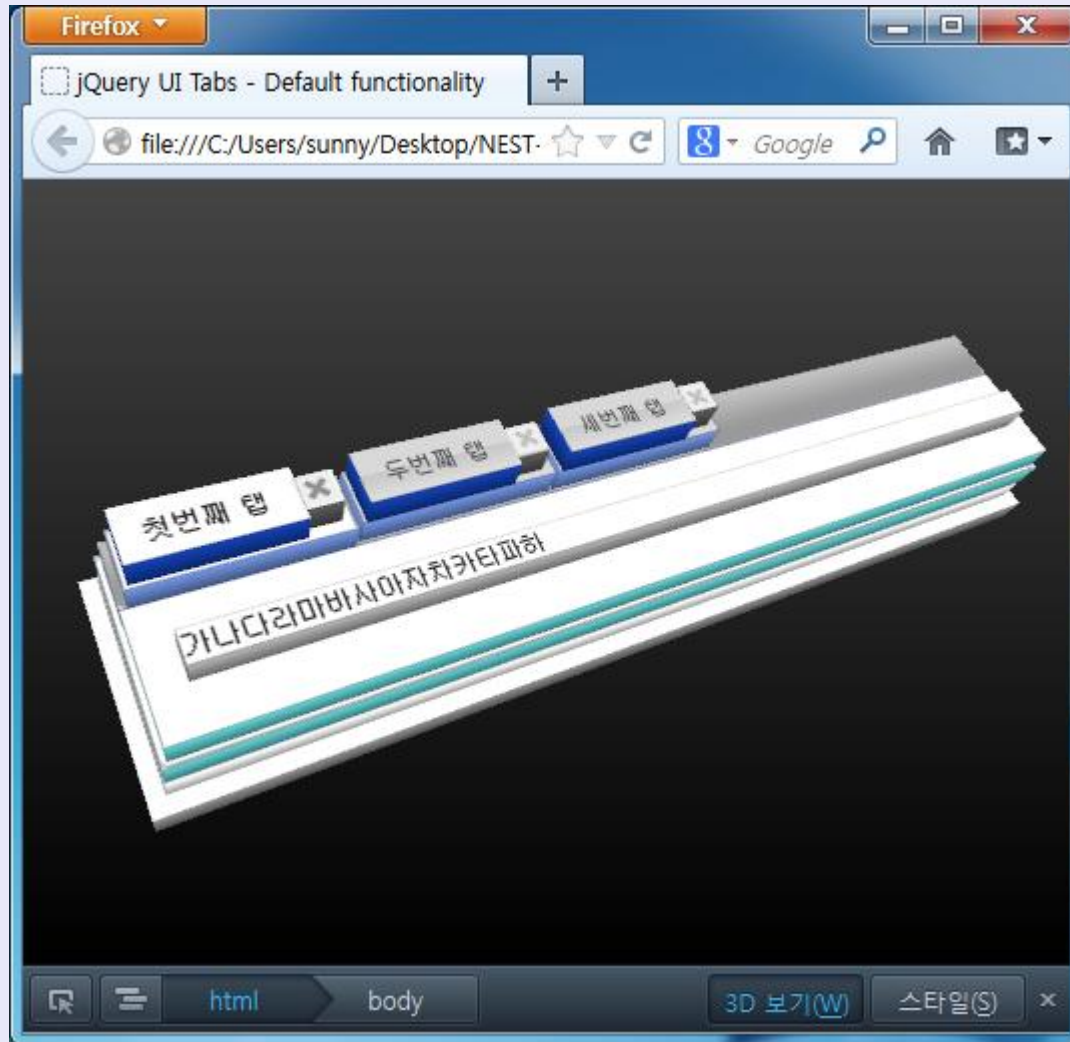
- ❖ jQuery로 애플리케이션을 개발할 때에 디버깅하는 방법들이야 파이어폭스를 사용하기는 하는데, <http://plugins.jquery.com/tag/jquery/> 여기에 보면 각종 플러그인 들이 업데이트되고 있다.
- ❖ 개발자도구 > source 에서 break point 사용 가능 : step 넘어가는 아이콘은 점에 오른쪽 회전 화살표
- ❖ Consolelog 에러 표시 옆 파일명을 클릭하면 해당 소스로 자동 이동한다.
- ❖ 개발자도구 Network 창에서 XHR을 보면 Ajax 통신에 대한 결과를 확인할 수 있다. (Headers와 Response 영역을 확인한다.)
- ❖ File을 못불러와서 에러가 날 경우 개발자도구의 consolelog와 Network에서 확인할 수 있다.
- ❖ UI 소스 변경이 브라우저에 반영이 안될 때는 Cache를 의심해봐라. (브라우저의 Cache 파일을 지운 후 다시 테스트 해본다.)
- ❖ 파일 import 구문의 변경 사항이 브라우저에 반영이 안될때는 파일명을 바꿔봐라. (jquery-ui-1.10.1.custom -> js/lib/jquery.ui/css/jquery.ui-1.10.1.custom.css)
- ❖ 브라우저 별 개발자도구 실행 단축키
 - ✓ 크롬 F12
 - ✓ Firefox Shift+F2
 - ✓ IE F12
 - ✓ Opera Ctrl+Shift+O

5.3 DOM 계층 구조를 이용한 요소 접근 하기

5. 기타

Firefox 브라우저의 3D 영상으로 본 DOM 계층 구조 (Tilt 기능)

- ❖ DOM 계층은 html문서 내에서 부모-자식관계,형제관계,조상-후손 관계를 정의한 것



<http://www.trirand.com/blog/>

jqGrid 컴포넌트를 이용한 그리드(GRID) 생성 순서

```
<html>
...
<body>
...
<table id="first_grid"></table>
...
</body>
</html>
```

그리드를 생성할
<table id="**grid_name**">
정의

```
$("#first_grid").jqGrid(options);
```

렌더링 명령, **options**에 상세한 세부
설정(컬럼명, 사이즈 등 설정)



- Grid 활성화/비활성화
- Column 숨기기, 활성화/비활성화
- 틀 고정
- Row 이동
- Paging
- Multi Row Select
- 컬럼 & Object 연동
- 멀티 헤더/Row
- 소계/합계
- Validation
- Mask
- Size,
- 행 추가 삭제
- Record Data CRUD
- Excel 연동

	Inv No	Date	Client	Amount	Tax	Total	Notes
+	13	2007-10-06	Client 3	1000.00	0.00	1000.00	
+	12	2007-10-06	Client 2	700.00	140.00	840.00	
+	11	2007-10-06	Client 1	600.00	120.00	720.00	
+	10	2007-10-06	Client 2	100.00	20.00	120.00	
-	9	2007-10-06	Client 1	200.00	40.00	240.00	
+	No	Item		Qty	Unit	Line Total	
	1	item 6		1.00	140.00	140.00	
	2	item 3		1.00	60.00	60.00	
+	8	2007-10-06	Client 3	200.00	0.00	200.00	

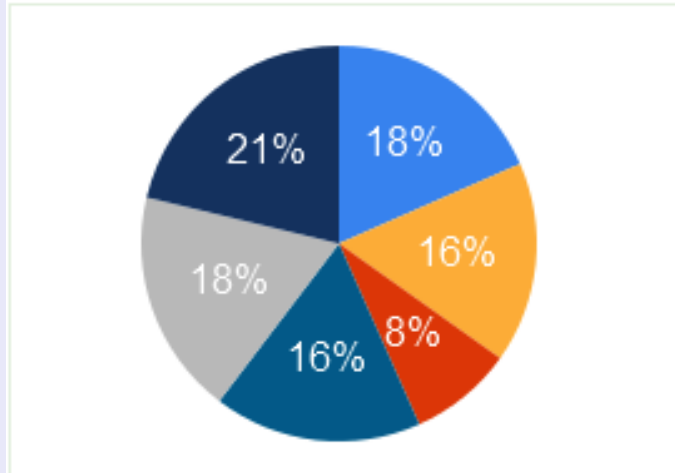
Page 1 of 2 View 1 - 10 of 13

5.5 jqChart

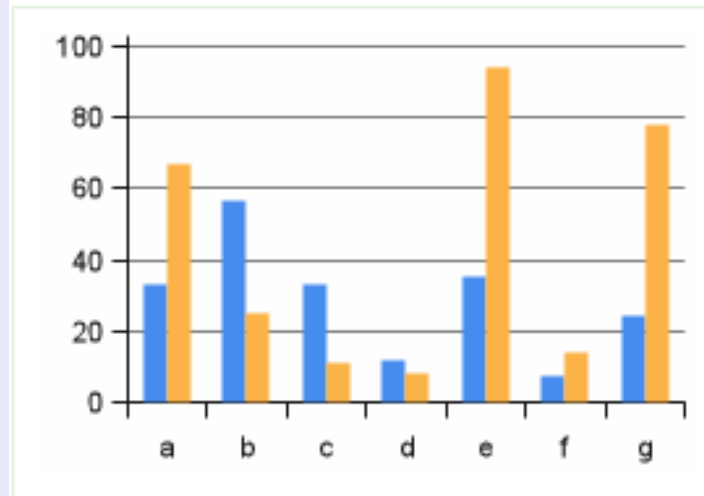
5. 기타

<http://www.jqchart.com/>

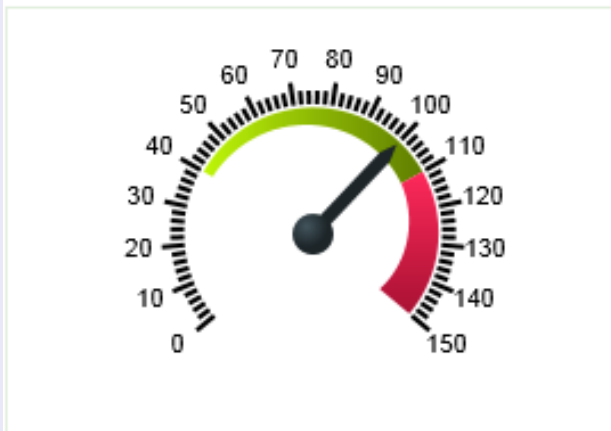
jqChart for jQuery Samples



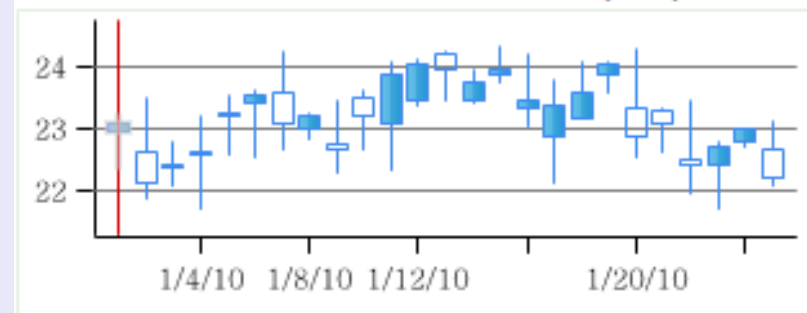
jqChart for ASP.NET MVC Samples



jqGauges for jQuery Samples



O:23.07 H:23.23 L:22.31 C:22.97 1/01/10



<http://www.jqplot.com/index.php>

상용 Software

5.6 jsTree

5. 기타

<http://www.jstree.com/>



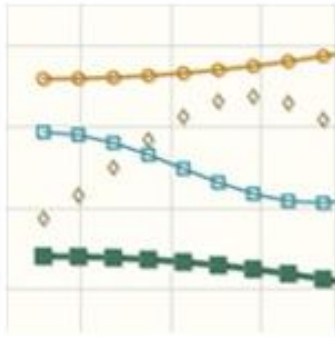
Currently supported browsers are:

- Internet Explorer 6+ *
- Mozilla Firefox 2+
- Safari 3+
- Opera 9+
- Google Chrome

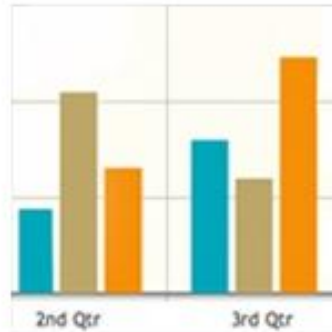
5.7 jqPlot

5. 기타

<http://www.jqplot.com/>



Numerous **line style options** with 6 built in marker styles!



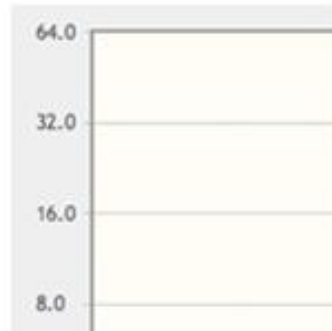
Horizontal and vertical **Bar charts!**



Shadow control on lines, markers, the grid, everything!



Drag and drop points with auto updating of data!



Log Axes with flexible tick marks!



Trend lines computed automatically!

jqPlot has been tested on IE 7, IE 8, Firefox, Safari, and Opera. You can see jqPlot in action on the [tests & examples page](#).

[참조사이트]

1. HTML

- HTML 4.01 명세: <http://www.w3.org/TR/html401/>
- HTML 4.01 명세(한글): <http://trio.co.kr/webrefer/html/cover.html>
- 브라우저 호환성: <http://www.quirksmode.org/compatibility.html>
- Event(DOM Level 2) 명세: <http://www.w3.org/TR/DOM-Level-2-Events/>
- Event(DOM Level 3) 명세: <http://www.w3.org/TR/DOM-Level-3-Events/>
- 추천 튜토리얼: <http://www.w3schools.com/html/default.asp>

2. JavaScript

- ECMA-262 3판: <http://www.ecma-international.org/publications/files/ECMA-ST-ARCH/ECMA-262,%203rd%20edition,%20December%201999.pdf>
- XmlHttpRequest 명세: <http://www.w3.org/TR/XMLHttpRequest/>
- Simple JavaScript Inheritance: <http://ejohn.org/blog/simple-javascript-inheritance/>
- 추천 레퍼런스: <http://koxo.com/>
- 추천 튜토리얼: <http://www.w3schools.com/js/default.asp>
- 추천 블로그: <http://blog.jidolstar.com/category/JavaScript>

3. CSS

- CSS2 명세: <http://www.w3.org/TR/CSS2/>
- CSS2 명세(한글): <http://trio.co.kr/webrefer/css2/cover.html>
- 추천 튜토리얼: <http://www.w3schools.com/css/default.asp>
- 추천 포럼: <http://cssdesign.kr/forum/>

End of Document