

jQuery 1.3 Cheatsheet

<http://www.jquery.com> <http://docs.jquery.com>

Download at: <http://www.gmtaz.com>

SELECTORS

Basics	Content Filters	Forms
#id	:contains(text)	:input
element	:empty	:text
.class	:has(selector)	:password
*	:parent	:radio
selector1, selectorN, ...		:checkbox
	Visibility Filters	:submit
Hierarchy	:hidden	:image
ancestor descendant	:visible	:reset
parent > child		:button
prev + next	Attribute Filters	:file
prev ~ siblings	[attribute]	:hidden
	[attribute=value]	
Basic Filters	[attribute != value]	Form Filters
:first	[attribute\$=value]	:enabled
:last	[attribute\$*=value]	:disabled
:not(selector)	[attribute*=value]	:checked
:even	[attributeFilter1][attributeFilterN]	:selected
:odd		
:eq(index)	Child Filters	
:gt(index)	:nth-child(index/even/odd/equation)	
:lt(index)	:first-child	
:header	:last-child	
:animated	:only-child	

ATTRIBUTES / CSS

Attributes	Value
attr(name)	val()
attr(properties)	val(val)
attr(key, value)	val(val)
attr(key, fn)	
removeAttr(name)	
	Positioning
Class	offset()
addClass(class)	position()
hasClass(class)	scrollTop(val)
removeClass(class)	scrollLeft(val)
toggleClass(class)	scrollLeft(val)
toggleClass(class, switch)	
	CSS
Height and Width	css(name)
height()	css(properties)
height(val)	css(name, value)
width()	
width(val)	

EVENTS

Event Helpers			
blur()	error()	keyup()	mouseover(fn)
blur(fn)	error(fn)	keyup(fn)	mouseup(fn)
change()	focus()	load(fn)	resize(fn)
change(fn)	focus(fn)	mousedown(fn)	scroll(fn)
click()	keydown()	mouseenter(fn)	select()
click(fn)	keydown(fn)	mouseleave(fn)	select(fn)
dblclick()	keypress()	mousemove(fn)	submit()
dblclick(fn)	keypress(fn)	mouseout(fn)	submit(fn)
			upload(fn)
Event Handling	Live Events	Page Load	
bind(type, [data], fn)	live(type, fn)	ready(fn)	
one(type, [data], fn)	die([type], [fn])		
trigger(event, [data])	Interaction Helpers		
triggerHandler(event, [data])	hover(over, out)		
unbind([type], [fn])	toggle(fn, fn2, [fn3], [fn4],...)		

EFFECTS

Basics	Settings
show()	\$.fx.off
show(speed, [callback])	
hide()	
hide(speed, [callback])	
toggle()	
toggle(speed, [callback])	
toggle(switch)	
toggle(speed, [callback])	
Sliding	
slideDown(speed, [callback])	
slideUp(speed, [callback])	
slideToggle(speed, [callback])	
Fading	
fadeIn(speed, [callback])	
fadeOut(speed, [callback])	
fadeTo(speed, opacity, [callback])	
Custom	
animate(params, [duration], [easing], [callback])	
animate(params, options)	
stop([clearQueue], [gotoEnd])	

MANIPULATION

Changing Contents	Replacing
html()	replaceWith(content)
html(val)	replaceAll(selector)
text()	
text(val)	Removing
	empty()
Inserting Inside	remove([expr])
append(content)	
appendTo(selector)	Copying
prepend(content)	clone()
prependTo(selector)	clone(bool)
Inserting Outside	Inserting Around
after(content)	wrap(html)
before(content)	wrap(elem)
insertAfter(selector)	wrapAll(html)
insertBefore(selector)	wrapAll(elem)
	wrapInner(html)
	wrapInner(elem)

TRAVERSING

Filtering	Finding
eq(index)	add(expr)
filter(expr)	children([expr])
filter(fn)	closest([expr])
is(expr)	contents()
map(callback)	find(expr)
not(expr)	next([expr])
slice(start, [end])	nextAll([expr])
	offsetParent()
Chaining	parent([expr])
andSelf()	parents([expr])
end()	prev([expr])
	prevAll([expr])
	siblings([expr])

AJAX

Ajax Request
\$.ajax(options)
load(url, [data], [callback])
\$.get(url, [data], [callback], [type])
\$.getJSON(url, [data], [callback])
\$.getScript(url, [callback])
\$.post(url, [data], [callback], [type])
Ajax Events
ajaxComplete(callback)
ajaxError(callback)
ajaxSend(callback)
ajaxStart(callback)
ajaxStop(callback)
ajaxSuccess(callback)
Misc
\$.ajaxSetup(options)
serialize()
serializeArray()

CORE

The jQuery Function	Data
\$(expression, [context])	data(name)
\$(html, [ownerDocument])	data(name, value)
\$(elements)	removeData(name)
\$(callback)	queue([name])
jQuery Object Accessors	queue([name], queue)
each(callback)	dequeue([name])
size()	
length	Plugins
selector	\$.fn.extend(object)
context	\$.extend(object)
eq(position)	
get()	Interoperability
get(index)	\$.noConflict()
index(subject)	\$.noConflict(extreme)

UTILITIES

Browser and Feature D	Test operations
\$.support	\$.isArray(obj)
\$.browser	\$.isFunction(obj)
\$.browser.version	
\$.boxModel	String operations
	\$.trim(str)
Array and Object operations	URLs
\$.each(object, callback)	\$.param(obj)
\$.grep(array, callback, [invert])	
\$.makeArray(obj)	
\$.map(array, callback)	
\$.inArray(value, array)	
\$.merge(first, second)	
\$.unique(array)	
\$.extend([deep], target, object1, [objectN])	

19. jQuery 플러그인

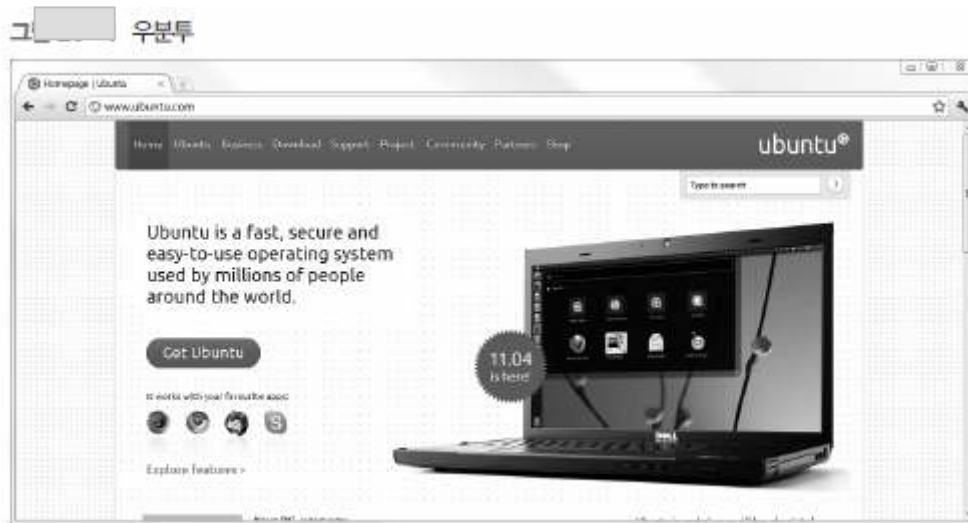
Wallpaper created by Gustavo Tandeciarz and adapted from <http://oscarotero.com/jquery/>

19.1 실제로 jQuery를 사용하는 웹 사이트

❖ jQuery를 가장 효율적으로 사용한 웹 사이트

■ 우분투

- 대부분의 슬라이더와 메뉴바, 툴팁 모두 jQuery로 작성
- <http://www.ubuntu.com/>



19.1 실제로 jQuery를 사용하는 웹 사이트

❖ 국내 소셜 커머스 사이트

- show() 메서드와 hide() 메서드, 라이트 박스
- Ex) 티몬, 쿠팡, 위메프



19.1 실제로 jQuery를 사용하는 웹 사이트

❖ 모든 국가의 웹 사이트에 적용

- 그루폰
- jQuery를 통해 Ajax와 LightBox, gnb 메뉴바 구현



19.1 실제로 jQuery를 사용하는 웹 사이트

❖ 국내 쇼핑몰

- 인터파크
 - 문서 객체 조작 위해 jQuery 사용
- 요소 검사 통해 확인
 - `addClass()`, `removeClass()`, `html()`, `text()`와 같은 jQuery 메서드



19.1 실제로 jQuery를 사용하는 웹 사이트

❖ jQuery를 나름 광범위하게 사용한 국내 사이트

■ CJ 제일제당

- 자체 플러그인을 jQuery 사용해 구현
- 플래시나 실버라이트 이용하지 않음

– 아이패드 포함한 모든 태블릿 PC 기종에서 완벽하게 작동



19.2 jQuery UI 플러그인

❖ jQuery UI 플러그인

- 가장 많이 활용되는 jQuery 플러그인
- jQuery가 공식적으로 지정한 플러그인
- 이름 그대로 UI와 관련된 플러그인
- jQuery UI 플러그인은 내부에서 다음과 같은 네 가지 부류 형성
 - 상호작용 (Interactions)
 - 위젯 (Widgets)
 - 유틸리티 (Utilities)
 - 효과 (Effects)
- <http://jqueryui.com/demos/> 참조



19.3 jQuery UI 기본 설정

❖ jQuery UI 설치와 사용

- <http://jqueryui.com> 접속



19.3 jQuery UI 기본 설정

❖ jQuery UI 설치와 사용 (2)

- Download 탭을 클릭
- 모든 기능 선택한 후 Theme UI lightness로 선택

jQuery UI Download(<http://jqueryui.com/download>)

Components (31 of 31 selected) ✕ Deselect all components

UI Core

A required dependency, contains basic functions and initializers.

☒ Core

The core of jQuery UI, required for all interactions and widgets.

☒ Widget

The widget factory, base for all widgets.

☒ Mouse

The mouse widget, a base class for all interactions and widgets with heavy mouse interaction.

☒ Position

A utility plugin for positioning elements relative to other elements.

Interactions

These add basic behaviors to any element and are used by many components below.

☒ Draggable

Makes any element on the page draggable.

☒ Droppable

Generated drop targets for draggable elements.

☒ Resizable

Makes any element on the page resizable.

☒ Selectable

Makes a list of elements mouse selectable by dragging a box or clicking on them.

☒ Sortable

Makes a list of items sortable.

Theme

Select the theme you want to include or design a custom theme.

UI lightness ▼

Advanced Theme Settings

Version

Select the release version you want to download.

☒ 1.8.13
(Stable, for jQuery 1.7.2+)

☐ 1.7.3
(Legacy release, for jQuery 1.3.2)

Download



19.3 jQuery UI 기본 설정

❖ jQuery UI 설치와 사용 (3)

■ 구성

- CSS 폴더
 - **ui-lightness** 폴더
 - » **images** 폴더
 - » **jquery-ui-1.8.14.custom.css** 파일
- development-bundle 폴더
- js 폴더
 - **jquery-1.5.1.min.js** 파일
 - **jquery-ui.1.8.14.custom.min.js** 파일
- index.html 파일



19.3 jQuery UI 기본 설정

❖ jQuery UI 설치와 사용 (4)

- css 폴더와 js 폴더의 압축을 풀고 HTML 페이지를 구성
- jQuery UI의 폰트 크기가 일반적인 폰트 크기보다 커서 작게 만든 것

코드 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet"href="css/ui-lightness/jquery-ui-1.8.14.custom.css"/>
  <style>
    body { font-size:62.5% }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js">/script>
  <script src="js/jquery-ui-1.8.14.custom.min.js">/script>
  <script>
  </script>
</head>
<body>

</body>
</html>
```



19.4 jQuery UI 플러그인의 기본 사용 방법

❖ Dialog 플러그인

- div 태그에 적용하는 플러그인
- DatePicker 플러그인은 input[type=text] 태그에 적용하는 플러그인
- body 태그, Style 태그 작성

코드 body 태그 구성

```
<body>
  <div id="dialog" title="jQuery UI dialog">
    <h1>Hello jQuery UI .. !</h1>
    <label for="event_name">일정 이름</label>
    <input id="event_name" type="text"/>
    <label for="event_date">날짜</label>
    <input id="event_date" type="text"/>
  </div>
</body>
```

코드 style 태그 구성

```
<style>
  body { font-size:62.5% }
  h1 { text-align:center; }
  input[type=text], label {
    margin-bottom:12px; padding: .4em;
    width:95%;
  }
</style>
```



19.4 jQuery UI 플러그인의 기본 사용 방법

❖ 플러그인 적용

코드 jQuery UI 플러그인 메서드

```
<script>
  $(document).ready(function () {
    $('#dialog').dialog();
    $('#event_date').datepicker();
  });
</script>
```

실행 결과



19.5 jQuery UI 플러그인의 옵션

❖ jQuery UI 플러그인

- 핵심 메서드의 첫 번째 매개 변수에 옵션 객체 넣어줌

표 19-1 jQuery UI Dialog 플러그인의 옵션

옵션 속성 이름	자료형	설명
autoOpen	Boolean	자동으로 실행할지를 지정합니다.
buttons	Object	다이얼로그에 생성할 버튼과 이벤트 핸들러를 지정합니다.
modal	Boolean	모달 다이얼로그의 형태로 생성할지를 지정합니다.

- 버튼 옵션 메서드

코드 19-1 Dialog 플러그인의 옵션 입력

```
<script>
$(document).ready(function () {
    $('#dialog').dialog({
        buttons: {
            submit: function () { },
            reset: function () { },
            cancel: function () { }
        },
        modal: true
    });
    $('#event_date').datepicker();
});
</script>
```

그림 19-1 실행 결과



19.5 jQuery UI 플러그인의 옵션

❖ DatePicker 플러그인 옵션

표 19-2 jQuery UI DatePicker 플러그인의 옵션

옵션 속성 이름	자료형	설명
dateFormat	String	날짜 출력 형식을 지정합니다.
numberOfMonths	Number	달력에 표시할 개월 수를 지정합니다.

코 19-1 DatePicker 플러그인의 옵션 입력

```
// jQuery UI DatePicker 플러그인
$('#event_date').datepicker({
    dateFormat: 'yy년 mm월 dd일',
    numberOfMonths: 3
});
```


그림 19-1 실행 결과



19.6 jQuery UI 플러그인의 메서드

❖ jQuery UI 플러그인의 메서드

- 속성 지정할 때 옵션 객체 사용
- 다이얼로그를 생성한 이후에 “다이얼로그를 띄워라, 닫아라”와 같은 행위는 메서드 사용

 jQuery UI 플러그인의 메서드 형태

`$(selector).dialog('open')`

플러그인 이름 메서드 이름



19.6 jQuery UI 플러그인의 메서드

❖ 예제 코드

- 일정 생성 버튼과 일정 출력할 div 태그 생성

코드 body 태그 구성

```
<body>
  <h1 id="open_button">일정 생성</h1>
  <hr/>
  <div id="output"></div>

  <!-- 다이얼로그 -->
  <div id="dialog" title="jQuery UI dialog">
    <h1>Hello jQuery UI .. !</h1>
    <label for="event_name">일정 이름</label>
    <input id="event_name" type="text"/>
    <label for="event_date">날짜</label>
    <input id="event_date" type="text"/>
  </div>
</body>
```



19.6 jQuery UI 플러그인의 메서드

❖ 예제 코드

- 일정 생성 버튼 클릭 - Dialog 플러그인의 open() 메서드 사용
- 다이얼로그 생성 - dialog() 메서드 옵션에 autoOpen 옵션 속성 입력

코드 Dialog 플러그인의 open() 메서드 사용

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('#open_button').click(function () {
        $('#dialog').dialog('open');
    });

    // jQuery UI Dialog 플러그인
    $('#dialog').dialog({
        autoOpen: false,
        buttons: {
            submit: function () { },
            reset: function () { },
            cancel: function () { }
        },
        modal: true
    });

    // jQuery UI DatePicker 플러그인
    $('#event_date').datepicker({
        dateFormat: 'yy년 mm월 dd일',
        numberOfMonths: 3
    });
});
</script>
```



19.6 jQuery UI 플러그인의 메서드

❖ Datepicker 플러그인의 메서드

표  jQuery UI Dialog 플러그인의 메서드

메서드 이름	설명
destroy	다이얼로그를 파괴합니다.
disable	다이얼로그를 사용 불가능하게 만듭니다.
enable	다이얼로그를 사용 가능하게 만듭니다.
close	다이얼로그를 닫습니다.
isOpen	다이얼로그가 열려 있는지 확인합니다.
open	다이얼로그를 실행합니다.



19.6 jQuery UI 플러그인의 메서드

❖ dialog() 메서드의 buttons 속성 구현

- 어떠한 버튼 눌러도 입력 양식 내용이 지워짐
- submit 버튼 누르면 내용 출력하고 다이얼로그 닫음
- cancel 버튼 누르면 그냥 다이얼로그 닫음
- 입력 양식의 내용을 제거하는 부분 - 별도의 함수

코드 Dialog 플러그인의 메서드 사용

```
// jQuery UI Dialog 플러그인
$('#dialog').dialog({
    autoOpen: false,
    buttons: {

        submit: function () {
            // 변수를 선언합니다.
            var eventName = $('#event_name').val();
            var eventDate = $('#event_date').val();

            // 출력합니다.
            $('<h1X/h1>').html(eventName + ' : ' + eventDate).appendTo('#output');

            // 다이얼로그를 닫습니다.
            $('#event_name').val('');
            $('#event_date').val('');
            $('#dialog').dialog('close');
        },
```



19.6 jQuery UI 플러그인의 메서드

❖ dialog() 메서드의 buttons 속성 구현 (2)

```
    reset: function () {  
        $('#event_name').val('');  
        $('#event_date').val('');  
    },  
    cancel: function () {  
        $('#event_name').val('');  
        $('#event_date').val('');  
        $('#dialog').dialog('close');  
    }  
},  
modal: true  
));
```



19.7 jQuery UI 플러그인의 이벤트

❖ jQuery UI 플러그인의 이벤트

표 19-1 jQuery UI Dialog 플러그인의 이벤트

이벤트 속성	이벤트 타입	설명
create	dialogcreate	다이얼로그를 생성할 때 발생합니다.
beforeClose	dialogbeforeclose	다이얼로그를 닫기 바로 직전에 발생합니다.
open	dialogopen	다이얼로그를 열 때 발생합니다.
focus	dialogfocus	다이얼로그에 초점을 맞추면 발생합니다.
close	dialogclose	다이얼로그를 닫을 때 발생합니다.



19.7 jQuery UI 플러그인의 이벤트

❖ bind() 메서드 사용해 이벤트를 연결하는 방법

- bind() 메서드의 첫 번째 매개 변수에 이벤트 타입 입력
- 두 번째 매개 변수에 이벤트 핸들러 입력

코드 jQuery UI 플러그인의 이벤트 연결 (1)

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('#open_button').click(function () {
        $('#dialog').dialog('open');
    });

    $('#dialog').bind('dialogcreate', function () {
        alert('다이얼로그를 생성했습니다. ');
    });

    // jQuery UI Dialog 플러그인
    $('#dialog').dialog({/* 생략 */});
    // jQuery UI DatePicker 플러그인
    $('#event_date').datepicker({/* 생략 */});
});
</script>
```



19.7 jQuery UI 플러그인의 이벤트

❖ 옵션 사용한 이벤트 연결

- 이벤트 속성을 사용해 옵션 입력해 연결

코드 jQuery UI 플러그인의 이벤트 연결 (2)

```
// jQuery UI Dialog 플러그인
$('#dialog').dialog({
  open: function () {

  },
  /* 생략 */
});
```




19.8 jQuery UI 위젯 플러그인

❖ jQuery UI 위젯 플러그인

표  jQuery UI 플러그인의 위젯 플러그인

플러그인 이름	관련 메서드
Accordion	<code>\$(selector).accordion()</code>
Autocomplete	<code>\$(selector).autocomplete()</code>
Button	<code>\$(selector).button()</code> , <code>\$(selector).buttonset()</code>
Datepicker	<code>\$(selector).datepicker()</code>
Dialog	<code>\$(selector).dialog()</code>
Progressbar	<code>\$(selector).progressbar()</code>
Slider	<code>\$(selector).slider()</code>
Tabs	<code>\$(selector).tabs()</code>

그림  jQuery UI 홈페이지의 Demos & Documentation 탭



19.9 jQuery UI 테마 클래스 (1)

❖ jQuery UI 플러그인

- 기본적으로 24개 테마 제공
- 테마는 CSS 파일의 형태로 제공
- JQuery UI ThemeRoller 사용하면 사용자 지정 테마 작성 가능



19.9 jQuery UI 테마 클래스 (1)


❖ jQuery UI 플러그인의 가장 기본적인 테마 클래스

■ 표] 위젯 클래스

- jQuery UI의 위젯 플러그인에 적용되는 클래스
- 위젯 클래스 활용하면 UI 구성 가능
- 대부분 div 태그에 적용하며, 예제코드처럼 사용

표  jQuery UI 플러그인의 위젯 클래스

클래스 이름	설명
ui-widget	위젯 모양을 형성합니다.
ui-widget-header	위젯 제목을 표시합니다.
ui-widget-content	위젯 내용을 표시합니다.
ui-widget-overlay	위젯을 흐리게 표시합니다.
ui-widget-shadow	위젯 그림자를 표시합니다.

코드  위젯 테마 클래스

```
<body>
  <div id="box" class="ui-widget-content">
    <h1 class="ui-widget-header">jQuery UI Theme</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</body>
```



19.9 jQuery UI 테마 클래스 (1)

❖ 위젯 클래스

- 테마 클래스는 background, border, color, font 속성만 제공
 - 여백이나 크기와 관련된 스타일 제공하지 않음
 - 별도의 margin 속성과 padding 속성 입력 권장

코드 style 태그 구성

```
<style>
  #box { padding: 5px; }
  #box h1 { margin: 0; padding: 5px; text-align: center; }
</style>
```

§ 테마 적용



19.9 jQuery UI 테마 클래스 (1)

❖ border 속성과 관련된 테마 클래스

표  jQuery UI 플러그인의 테두리 클래스


클래스 이름	설명
ui-corner-tl	top-left 위치의 테두리를 둥글게 합니다.
ui-corner-tr	top-right 위치의 테두리를 둥글게 합니다.
ui-corner-bl	bottom-left 위치의 테두리를 둥글게 합니다.
ui-corner-br	bottom-right 위치의 테두리를 둥글게 합니다.
ui-corner-top	top 위치의 테두리를 둥글게 합니다.
ui-corner-bottom	bottom 위치의 테두리를 둥글게 합니다.
ui-corner-right	right 위치의 테두리를 둥글게 합니다.
ui-corner-left	left 위치의 테두리를 둥글게 합니다.
ui-corner-all	모든 위치의 테두리를 둥글게 합니다.



19.9 jQuery UI 테마 클래스 (1)

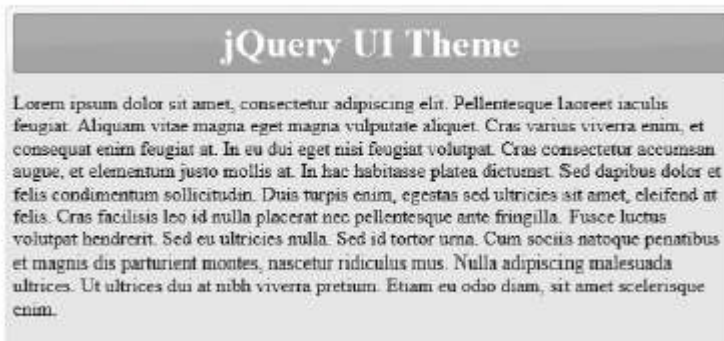
❖ border 속성과 관련된 테마 클래스 활용 예제

- CSS3의 border-radius 속성을 사용해 구현하는 클래스
 - CSS3를 지원하지 않은 브라우저는 테두리가 둥글게 표현되지 않음

코드  테두리 테마 클래스

```
<body>
  <div id="box" class="ui-widget-content ui-corner-all">
    <h1 class="ui-widget-header ui-corner-all">jQuery UI Theme</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</body>
```

그림  테두리 적용



19.10 jQuery UI 테마 클래스 (2)

❖ 상태 클래스

- Button 플러그인과 같이 상호작용 하는 플러그인 적용

jQuery UI 플러그인의 상태 클래스 (1)

클래스 이름	설명
ui-state-default	기본 상태를 표시합니다.
ui-state-hover	마우스가 올려진 상태를 표시합니다.
ui-state-focus	초점이 맞춰진 상태를 표시합니다.
ui-state-active	활성화 상태를 표시합니다.

코드 상태 테마클래스 (1)

```
<body>
  <div id="button" class="box ui-state-default ui-corner-all">
    <p>Lorem ipsum dolor sit amet Button</p>
  </div>
</body>
```

코드 style 태그 구성

```
<style>
  .box { padding: 5px; }
  .box p { margin: 0px; padding: 5px; text-align: center; }
</style>
```



19.10 jQuery UI 테마 클래스 (2)

❖ 상태 클래스

- 마우스 커서가 올라가거나 내려갈 때, 클릭할 때 상태 클래스 적용

코드 상태 테마 클래스의 사용

```
<script>
$(document).ready(function () {
    // hover() 메서드를 적용합니다.
    $('#button').hover(function () {
        $(this).addClass('ui-state-hover');
    }, function () {
        $(this).removeClass('ui-state-hover');
    });

    // click() 메서드를 적용합니다.
    $('#button').click(function () {
        $(this).toggleClass('ui-state-active');
    });
});
</script>
```

테마가 적용된 모습

Lorem ipsum dolor sit amet Button



19.10 jQuery UI 테마 클래스 (2)

❖ 여러 문서 객체 감싸는 Div 태그에 적용하는 클래스

jQuery UI 플러그인의 상태 클래스 (2)

클래스 이름	설명
ui-state-highlight	강조 표시합니다.
ui-state-error	에러 표시합니다.
ui-state-error-text	글자에만 에러 표시합니다.
ui-state-disabled	사용할 수 없는 상태임을 표시합니다.

코드 2 body 태그 구성

```
<body>
  <div id="radio">
    <input type="radio" name="theme" id="highlight" value="highlight"/>
    <label for="highlight">highlight</label>
    <input type="radio" name="theme" id="error" value="error"/>
    <label for="error">error</label>
    <input type="radio" name="theme" id="error_text" value="error-text"/>
    <label for="error_text">error_text</label>
    <input type="radio" name="theme" id="disabled" value="disabled"/>
    <label for="disabled">disabled</label>
    <button>RESET</button>
  </div>
  <div id="output">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</body>
```



19.10 jQuery UI 테마 클래스 (2)

❖ style 태그에 box 클래스 입력

- jQuery 사용하여 클래스 적용

코드 style 태그 구성

```
<style>
  .box { padding: 5px; }
  .box p { margin: 0px; padding: 5px; }
</style>
```



19.10 jQuery UI 테마 클래스 (2)

❖ buttonset() 메서드 사용

■ 버튼 묶어줌

코드 상태 테마 클래스의 사용

```
<script>
$(document).ready(function () {
    // 함수를 선언합니다.
    function reset() {
        var resetClass = 'ui-corner-all ui-widget-content box';
        $('#output').attr('class', resetClass);
    }

    reset();
    // jQuery UI Button 플러그인을 사용합니다.
    $('#radio').buttonset();

    // 이벤트를 연결합니다.
    $('button').click(function () { reset(); });
    $('#radio > input').change(function () {
        reset();
        $('#output').addClass('ui-state-' + $(this).val());
    });
});
</script>
```

그림 실행 결과

highlight	error	error_text	disabled	RESET
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla tempor arcu vel lectus ultricies egestas. Nam ut adipiscing turpis. Donec luctus tempus arcu ut dictum. Sed rhoncus, felis a malesuada egestas, leo est luctus lectus, ut faucibus nunc purus mattis mi. Quisque eget mauris nec tellus imperdiet semper.				



19.11 jQuery UI 테마 클래스 (3)

❖ 헬퍼 클래스

- background, border, color 속성 적용 클래스
- 헬퍼 클래스
 - 위의 클래스에 margin, padding, display, list-style 속성까지 적용

표 19-11 jQuery UI 플러그인의 헬퍼 클래스

클래스 이름	설명
ui-helper-hidden	보이지 않게 만듭니다(display 속성 조작).
ui-helper-hidden-accessible	보이지 않게 만듭니다(clip 속성 조작).
ui-helper-reset	스타일을 초기화합니다.
ui-helper-clearfix	float 스타일 속성을 적용한 문서 객체를 감싸는 객체에 사용합니다.
ui-helper-zfix	iframe 태그가 화면을 꽉 채우게 만듭니다.



19.11 jQuery UI 테마 클래스 (3)

❖ ui-helper-reset 클래스

- margin, padding, border 속성 등 초기화
- ui-helper-reset 클래스는 li 태그에 많이 적용

```
.ui-helper-reset { margin: 0; padding: 0; border: 0; outline: 0; line-height: 1.3; text-decoration: none; font-size: 100%; list-style: none; }
```

코드 헬퍼 테마 클래스

```
<body>
  <ul id="list" class="ui-helper-reset">
    <li><div class="ui-widget-content ui-corner-all"><h1>FULLSCREEN</h1></div></li>
  </ul>
  <iframe width="650" height="510" src="http://www.youtube.com/embed/gBG14Zbk0qk"
    frameborder="0"></iframe>
</body>
```



19.11 jQuery UI 테마 클래스 (3)

❖ 예제

- h1 태그를 클릭할 때 클래스 추가
- window 객체에 keydown 이벤트 연결
 - 전체 화면 상태 벗어날 수 있는 방법 제공

코드 헬퍼 테마 클래스의 사용

```
<script>
$(document).ready(function () {
    // 이벤트를 연결합니다.
    $('h1').click(function () {
        $('iframe').addClass('ui-helper-zfix');
    });

    $(window).keydown(function () {
        $('iframe').removeClass('ui-helper-zfix');
    });
});
</script>
```

그림 26-20 실행 결과

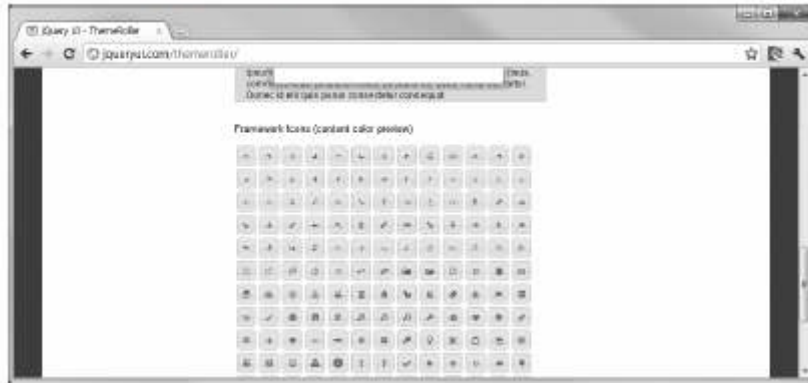


19.11 jQuery UI 테마 클래스 (3)

❖ jQuery UI 플러그인

- 174개의 아이콘 이미지 제공

그림 19.11 jQuery UI 플러그인의 아이콘



19.12 jQuery UI 플러그인의 아이콘

❖ 아이콘 사용법

- ui-icon 클래스 우선 적용
- 아이콘의 이름 클래스 함께 지정
- 각 span 태그에 아이콘 관련 클래스 적용

코드 아이콘 클래스

```
<body>
  <div id="list">
    <div class="ui-state-default ui-corner-all">
      <span class="ui-icon ui-icon-heart">X</span>
    </div>
    <div class="ui-state-default ui-corner-all">
      <span class="ui-icon ui-icon-gear">X</span>
    </div>
    <div class="ui-state-default ui-corner-all">
      <span class="ui-icon ui-icon-search">X</span>
    </div>
    <div class="ui-state-default ui-corner-all">
      <span class="ui-icon ui-icon-wrench">X</span>
    </div>
  </div>
</body>
```



19.12 jQuery UI 플러그인의 아이콘

❖ 아이콘 사용법

■ 스타일 태그 구성

코드 style 태그 구성

```
<style>
  #list > div { padding: 5px; float:left; }
</style>
```

그림 26-22 실행 결과



19.13 jQuery UI 상호작용 플러그인

❖ 상호작용 플러그인

- 사용자가 직접 문서 객체를 드래그해 위치를 이동
- 크기를 변경 가능
- 어떠한 문서 객체에도 적용할 수 있음
 - 일반적으로 div 태그나 li 태그에 적용

표 19-1 jQuery UI 플러그인의 상호작용 플러그인

플러그인 이름	설명
Draggable	문서 객체를 드래그해 위치를 변경할 수 있습니다.
Droppable	문서 객체를 다른 문서 객체 안에 넣을 수 있습니다.
Resizable	문서 객체의 크기를 변경할 수 있습니다.
Selectable	문서 객체를 선택할 수 있습니다.
Sortable	문서 객체의 순서를 변경할 수 있습니다.



19.13 jQuery UI 상호작용 플러그인

❖ Draggable 플러그인

- body 태그 구성
 - 각 div 태그의 box 클래스는 width 속성과 height 속성 지정
- style 태그
 - box 클래스에 width 속성과 height 속성 지정

코드 body 태그 구성

```
<body>
  <div class="box ui-widget-content"></div>
  <div class="box ui-widget-content"></div>
  <div class="box ui-widget-content"></div>
  <div class="box ui-widget-content"></div>
</body>
```

코드 style 태그 구성

```
<style>
  .box { width:100px; height:100px; float:left; }
</style>
```


코드 Draggable 플러그인

```
<script>
  $(document).ready(function () {
    // 변수를 선언합니다.
    $('div.box').draggable();
  });
</script>
```




19.13 jQuery UI 상호작용 플러그인

❖ Draggable 플러그인 옵션과 이벤트

표  jQuery UI Draggable 플러그인의 옵션

속성 이름	자료형	설명
axis	String('x' 또는 'y')	이동 축
grid	Array	이동 거리
helper	String('original' 또는 'clone')	이동 방식
scroll	Boolean	문서 객체를 스크롤할 때 스크롤이 움직일지 지정
zIndex	Number	z-index 속성 지정


표  jQuery UI Draggable 플러그인의 이벤트

이벤트 이름	이벤트 타입	설명
create	dragcreate	드래그 객체를 생성할 때 발생합니다.
start	dragstart	드래그를 시작할 때 발생합니다.
drag	drag	드래그할 때 발생합니다.
stop	dragstop	드래그가 끝날 때 발생합니다.




19.13 jQuery UI 상호작용 플러그인

❖ 옵션과 이벤트 사용 예제

코드  Draggable 플러그인의 옵션과 이벤트

```
<script>
  $(document).ready(function () {
    // jQuery UI Draggable 플러그인을 사용합니다.
    $('#div.box').draggable({
      grid: [102, 102],
      stop: function (event, ui) {

      }
    });
  });
</script>
```

그림  jQuery UI Draggable 플러그인의 Overview 탭

All callbacks (start, stop, drag) receive two arguments: The original browser event and a prepared ui object, view below for a documentation of this object (if you name your second argument 'ui'):

- **ui.helper** - the jQuery object representing the helper that's being dragged
- **ui.position** - current position of the helper as { top, left } object, relative to the offset element
- **ui.offset** - current absolute position of the helper as { top, left } object, relative to page



19.13 jQuery UI 상호작용 플러그인

❖ ui 객체의 offset 속성 출력

코드 상호작용 플러그인 이벤트의 ui 객체

```
<script>
$(document).ready(function () {
    // jQuery UI Draggable 플러그인을 사용합니다.
    $('div.box').draggable({
        grid: [102, 102],
        stop: function (event, ui) {
            alert(ui.offset.top + ' : ' + ui.offset.left);
        }
    });
});
</script>
```

그림 실행 결과



19.14 jQuery UI 테마와 상호작용 연습

❖ 드래그앤 드롭 연습 예제

그림 26-25 폴더 구성



❖ Body 태그 구성

- 드래그할 대상이 있는 div 태그
- 드래그해서 놓을 div 태그

코드 body 태그 구성

```
<body>
  <div id="wrap">

    <div>
      <div id="dock" class="ui-state-error">

        <div>
</body>
```



19.14 jQuery UI 테마와 상호작용 연습

❖ Style 태그 구성

코드 style 태그 구성

```
<style>
  * { margin:0px; padding:0px; }
  body { font-size:62.5%; }

  .item {margin:3px;padding:5px;}
  .item > img { width:100px; height: 100px;}

  #wrap { overflow:hidden; }
  #wrap > div { float:left; }
  #dock { position:fixed; min-height:120px; width:100%; bottom:0px; z-index:1;}
  #dock > div { float:right; }
</style>
```



19.14 jQuery UI 테마와 상호작용 연습

❖ 동적 문서 생성

- 반복문 사용해 문서 객체 생성
- Draggable 플러그인과 Droppable 플러그인 적용

코드 문서 객체 생성과 플러그인 적용

```
<script>
$(document).ready(function () {
    // 문서 객체를 생성합니다.
    for (var i = 0; i < 21; i++) {
        var icon = '';
        icon += '<div class="ui-widget-content ui-corner-all item">';
        icon += '    ';
        icon += '</div>';

        $('#wrap').append(icon);
    }

    // jQuery UI Draggable 플러그인을 사용합니다.
    $('#wrap > div').draggable();

    // jQuery UI Droppable 플러그인을 사용합니다.
    $('#dock').droppable();
    $('body').droppable();
});
</script>
```



19.14 jQuery UI 테마와 상호작용 연습

❖ draggable() 메서드 옵션 객체

- 문서 객체를 이동할 때 문서 객체가 복사된 대상이 이동
- Scroll 속성을 false로 입력해
 - 문서 객체를 드래그할 때 스크롤 만들지 않게 함
 - zIndex 속성
 - style 태그에서 지정한 z-index 속성과 관련
 - 옮겨지는 대상이 옮길 장소보다 앞에 위치하게 더 큰 값 입력

코드 Draggable 플러그인의 옵션 적용

```
// jQuery UI Draggable 플러그인을 사용합니다.  
$('#wrap > div').draggable({  
    helper: 'clone',  
    zIndex: 100,  
    scroll: false  
});
```



19.14 jQuery UI 테마와 상호작용 연습

❖ droppable() 메서드 옵션 객체

- accept 옵션 속성 - 놓을 문서 객체 제한
- Drop 이벤트 - 문서 객체를 놓았을 때 발생하는 이벤트
- Droppable 플러그인은 이동됐음을 확인만 함

코드 Droppable 플러그인의 옵션 및 이벤트 적용

```
// jQuery UI Droppable 플러그인을 사용합니다.
$('#dock').droppable({
  accept: '#wrap > div',
  drop: function (event, ui) { }
});
$('body').droppable({
  accept: '#dock > div',
  drop: function (event, ui) { }
});
```



19.14 jQuery UI 테마와 상호작용 연습

❖ 문서 객체를 옮기는 예제

코드 문서 객체 이동 구현

```
// jQuery UI Droppable 플러그인을 사용합니다.
$('#dock').droppable({
  accept: '#wrap > div',
  drop: function (event, ui) {
    $(ui.draggable).fadeOut(function () {
      $(this).appendTo('#dock').fadeIn();
    });
  }
});

$('#body').droppable({
  accept: '#dock > div',
  drop: function (event, ui) {
    $(ui.draggable).fadeOut(function () {
      $(this).appendTo('#wrap').fadeIn();
    });
  }
});
```

그림 실행 결과

