

융합형 IT 기술 전문가 과정 - 2010년 8월 16일~27일

6일차. jQuery 소개

주요 JS 라이브러리

■ jQuery

- <http://jquery.com>
- 가장 많이 쓰이는 JS 라이브러리

■ Modernizr

- <http://modernizr.com>
- HTML5 기본 속성들의 호환성을 보완하는 라이브러리

■ Raphaël

- <http://dmitrybaranovskiy.github.io/raphael/>
- 자바스크립트를 통한 그래픽 라이브러리

■ Sencha

- <http://sencha.com>
- Ext JS + jQTouch + Raphaël

JQuery

- 가장 강력한 자바스크립트 라이브러리
 - 간단한 자바스크립트 작업 수행

- 문서 객체 처리 기능
 - 웹 문서 내 객체 접근
 - CSS나 Xpath 같은 직관적 방법
 - 웹 문서 내 객체를 쉽게 변경 가능
 - 웹 문서 내 사용자 액션을 변경 가능

- 다양한 기능 제공
 - 애니메이션 기능 추가 가능
 - Ajax 기능 사용 가능
 - 브라우저 호환성 유지



jQuery의 장점

- 작은 용량- 14kb (Minified and Gzipped)
- 웹 브라우저 지원 (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)
- CSS 같은 문법 방식 - 누구나 이해하기 쉽다!
- 강력한 개발자 커뮤니티
- 확장 기능 제공 - 플러그인

jQuery 사용처

- 35% of all sites that use JavaScript, use jQuery
- 1 out 5 sites, of all sites, use jQuery

reddit.com
espn.com
ibm.com
stackoverflow.com
kickball.com
boxee.tv
bit.ly
twitpic.com

whitehouse.gov
wikipedia.org
microsoft.com
amazon.com
netflix.com
bing.com
monster.com
tv.com

overstock.com
time.com
capitalone.com
usatoday.com
ning.com
wordpress.com
dell.com
twitter.com

jQuery의 기초

동작 실행

```
$ ( '#menu' ) . addClass ( 'hello' )
```

객체 선택

■ 효율적 문서 객체 선택

- `$()` function jQuery 객체를 만드는 기본 메소드
- `$("dt")`는 문서에 `<dt>` 요소를 만드는 방법

■ 쉬운 객체 동작 제어

- `$.addAction()`으로 동작 추가
- `$('img.photo').attr('src','/default.png');`

```
<html>
  <head>
    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $('a').addClass('link');
      });
    </script>
  </head>
  <body>
    <a href="http://jquery.com/">jQuery</a>
  </body>
</html>
```

```
<div id="head">
<h1 id='heading'>Cities of the World</h1>
<dl>
<dt>Paris</dt>
<dd>Chic, fashionable, expensive rude</dd>
<dt>Sydney</dt> <dd>Opera house but no
culture, Mardi Gras, fireworks</dd>
</dl>
</div>
```

```
h1 {font-size: 2.5em; margin-bottom:
0;}
```

```
.emphasize {font-style: italic;
color:red;}
```

Cities of the World

Paris

Chic, fashionable, expensive rude

Sydney

Opera house but no culture, Mardi Gras, fireworks

DOM

```
$( 'div#head h1' );
```

CSS

```
$( 'dt' ).addClass( 'emphasize' );
```


jQuery 선택자

■ CSS

- p 요소명
- #id ID
- .class 클래스
- p.class 요소속 클래스
- p a 자식 요소
- p > a 자식 직접 선택

■ XPath

- /html/body//div DOM 상속 경로
- a[@href] href 속성을 가진 a 요소
- div[ol] ol 요소를 가진 div 요소
- //a[@ref='nofollow'] ref를 가진 모든 a 요소

■ 필터

- p:first 첫번째 자식
- li:last 마지막 자식
- a:nth(3) 네번째 링크
- a:eq(3)
- p:even or p:odd 홀짝수
- a:gt(3) or a:lt(4) 3보다 크거나 4보다 작은 경우
- a:contains('click') click이라는 글자 포함한 경우

JQuery Filters

■ 필터

- p:first 첫번째 자식
- li:last 마지막 자식
- a:nth(3) 네번째 링크
- a:eq(3)
- p:even or p:odd 홀짝수
- a:gt(3) or a:lt(4) 3보다 크거나 4보다 작은 경우
- a:contains('click') click이라는 글자 포함한 경우

개념 1. Find something, do something

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li><a>home</a></li>
    <li><a>about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    jQuery("#nav li");
  </script>
</body>
</html>
```

개념 1. Find something, do something

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li class="navListItem"><a>home</a></li>
    <li class="navListItem"><a>about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery("#nav li").addClass("navListItem");
</script>
</body>
</html>
```

개념 1. Find something, do something

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li><a>home</a></li>
    <li><a>about</a></li>
  </ul>
  <script src="jquery.js"></script>
  <script>
    jQuery("#nav a");
  </script>
</body>
</html>
```

개념 1. Find something, do something

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li><a href="/home">home</a></li>
    <li><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery("#nav a").each(function(){
    jQuery(this).attr("href", "/" + jQuery(this).text());
  });
</script>
</body>
</html>
```

개념 2. Create something, do something

```
<!DOCTYPE html>
<html>
<body>
<ul id="nav">
</ul>
<script src="jquery.js"></script>
<script>
    jQuery("<li>home</li>");
    jQuery("<li>about</li>");
</script>
</body>
</html>
```


개념 2. Create something, do something

```
<!DOCTYPE html>
<html>
<body>
<ul id="nav">
</ul>
<script src="jquery.js"></script>
<script>
    jQuery("<li>home</li>").wrapInner("a");
    jQuery("<li>about</li>").wrapInner("a");
</script>
</body>
</html>
```

개념 2. Create something, do something

```
<!DOCTYPE html>
<html>
<body>
<ul id="nav">
  <li><a>home</a></li>
  <li><a>about</a></li>
</ul>
<script src="jquery.js"></script>
<script>
  jQuery("<li>home</li>").wrapInner("a").appendTo("#nav");
  jQuery("<li>about</li>").wrapInner("a").appendTo("#nav");
</script>
</body>
</html>
```

개념 3. Chaining

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li class="navListItem"><a href="/home">home</a></li>
    <li class="navListItem"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery("ul").attr("id", "nav");
  jQuery("#nav li").addClass("navListItem");
  jQuery("#nav a").each(function(){
    jQuery(this).attr("href", "/" + jQuery(this).text());
  });
</script>
</body>
</html>
```

개념 3. Chaining

```
<!DOCTYPE html>
<html>
<body>
  <ul id="nav">
    <li class="navListItem"><a href="/home">home</a></li>
    <li class="navListItem"><a href="/about">about</a></li>
  </ul>
<script src="jquery.js"></script>
<script>
  jQuery("ul").attr("id","nav")
  .find("li").addClass("navListItem")
  .find("a").each(function(){
    jQuery(this).attr("href","/" + jQuery(this).text());
  });
</script>
</body>
</html>
```

개념 4. jQuery parameter types

- CSS Selectors & custom CSS expressions

`jQuery("#nav")` and `jQuery(":first")`

- HTML

`jQuery("link")`

- DOM Elements

`jQuery(document)` or `jQuery(this)` or `jQuery(event.target)`

- A function (shortcut for jQuery DOM ready event)

`jQuery(function(){}) = jQuery(document).ready(function(){})`

사용법 1: 값 선택

■ 원하는 속성 값 선택

```
var height = $('div#intro').height();  
var src = $('img.photo').attr('src');  
var lastP = $('p:last').html();
```

■ DOM 선택

```
$('div.section').next()  
$('div.section').prev()  
$('div.section').prev('a')  
$('div.section').parent()  
$('div.section').parents()
```

사용법 2: 객체 속성 변경

- HTML attribute 변경

```
$( 'img.photo' ).attr( 'src', ' /default.png' )
```

- HTML 문법 추가(innerHTML)

```
$( 'a.foo' ).html( '<em>Click me now!</em>' );
```

- CSS class 추가

```
$( 'div.section' ).addClass( 'highlighted' );
```

- CSS property 변경

```
$( 'p:odd' ).css( 'color', '#ccc' );
```

사용법 3: 동작 추가

■ 자동 루프

```
$('#div').hide();
```

```
var divs = document.getElementsByTagName('div');  
for(var i=0; i<divs.length;i++) {  
    divs[i].style.display = 'none';  
}
```

■ 연속적인 동작 실행 가능

```
$('#div.section').hide().addClass('gone');  
$('#div.section').html(function(i){  
    return '<p>value' + i + '</p>';});
```


jQuery API 개요

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

- **Core**
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

`jQuery()`

`each()`

`map()`

`size()`

`length`

`selector`

`context`

`index()`

`get()`

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>Gold</p>
<p>Silver</p>

<script
src="http://ajax.googleapis.com/ajax/
libs/jquery/1.3.2/jquery.min.js"></sc
ript>
<script>
alert(jQuery("p").map(function() {
    return $(this).text();
}).get()));
</script>
</body>
</html>
```

<http://jsbin.com/orani/edit#html>

- Core
- **Selectors**
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
jQuery("#nav li.contact")
```

```
jQuery(":visible")
```

```
jQuery(":radio:enabled:checked")
```

```
jQuery("a[title]")
```

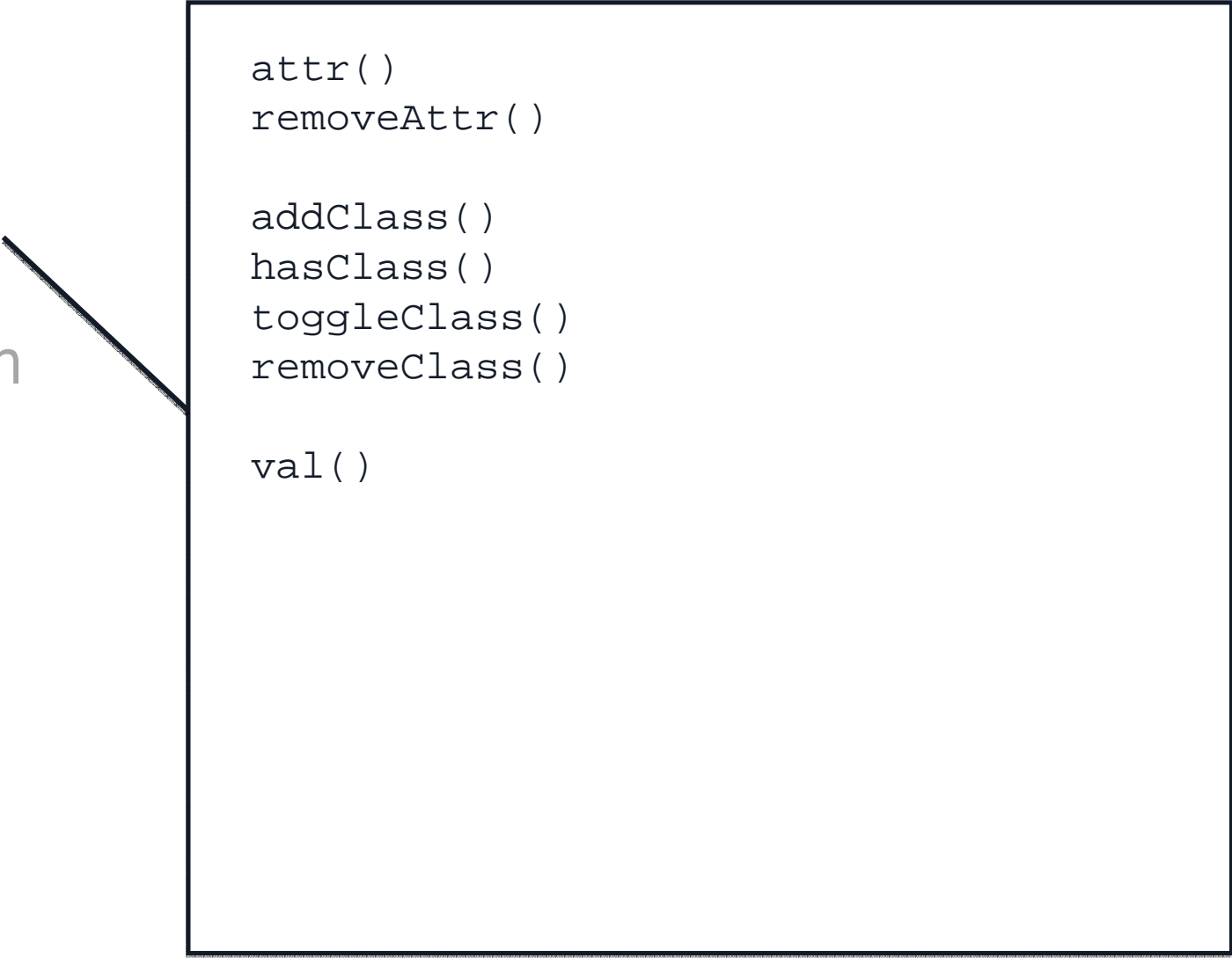
```
jQuery("a[title][href='foo']")
```

```
jQuery("a:first[href*='foo']")
```

```
jQuery("#header, #footer")
```

```
jQuery("#header, #footer").filter(":visible")
```

- Core
- Selectors
- **Attributes**
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities



```
attr()  
removeAttr()  
  
addClass()  
hasClass()  
toggleClass()  
removeClass()  
  
val()
```

- Co
- Se
- Att
- Tra
- Ma
- CS
- Ev
- Eff
- Aja
- Uti

```
<!DOCTYPE html><html><body>  
  
<input type="text" value="search" />  
  
<script src="jquery.js"></script>  
<script>  
  
$("input").focus(function(){  
    var v = $(this).val();  
    $(this).val( v == this.defaultValue ? "" : v);  
}).blur(function(){  
    var v = $(this).val();  
    $(this).val( !v ? this.defaultValue : v);  
});  
  
</script></body></html>
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
add()
children()
closest()
find()
next()
nextAll()
prev()
prevAll()
siblings()

parent()
parents()

andSelf()
end()

eq()
filter()
is()
not()
slice()
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<h3>Quotes</h3>
<div>
  <p>Now or never</p>
  <p>Because he could</p>
</div>

<script src="jquery.js"></script>
<script>
$("h3").click(function() {
  $(this).next().toggle();
}).next().hide();
</script>
</body>
</html>
```

<http://jsbin.com/uhole/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- **Manipulation**
- CSS
- Events
- Effects
- Ajax
- Utilities

```
html()
text()

append()
appendTo()
prepend()
prependTo()

after()
insertAfter()
before()
insertBefore()

wrap()
wrapAll()
wrapInner()

replaceWith()
replaceAll()

empty()
remove()

clone()
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>Make me bold!</p>

<script src="jquery.js"></script>
<script>

jQuery( "p" ).wrapInner( "<b></b>" );

</script>
</body>
</html>
```

<http://jsbin.com/esuwu/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- **CSS**
- Events
- Effects
- Ajax
- Utilities

```
css()
```

```
offset()
```

```
offsetParent()
```

```
position()
```

```
scrollTop()
```

```
scrollLeft()
```

```
height()
```

```
width()
```

```
innerHeight()
```

```
innerWidth()
```

```
outerHeight()
```

```
outerWidth()
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- **CSS**
- Events
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<head>
<style>div{background-color:#ccc; width:100px;
margin:0 20px; float:left;}</style>
</head>
<body>

<div></div>
<div></div>
<div></div>

<script src="jquery.js" ></script>
<script>

jQuery("div").height(jQuery(document).height());

</script>
</body>
</html>
```

<http://jsbin.com/iseti/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

```
ready()  
  
bind()  
one()  
trigger()  
triggerHandler()  
unbind()  
  
live()  
die()  
  
hover()  
toggle()  
  
blur()  
change()  
click()  
dblclick()  
error()  
focus()  
keydown()  
keypress()  
keyup()  
mousedown()  
mouseenter()  
mouseleave()  
mouseout()  
mouseup()  
resize()  
scroll()  
select()  
submit()  
unload()
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>click me</p>
<p>click me</p>

<script src="jquery.js"></script>
<script>

jQuery("p").bind("click", function(){
    $(this).after("<p>click me</p>");
});

</script>
</body>
</html>
```

<http://jsbin.com/ehuki/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- **Events**
- Effects
- Ajax
- Utilities

```
<!DOCTYPE html>
<html>
<body>

<p>click me</p>
<p>click me</p>

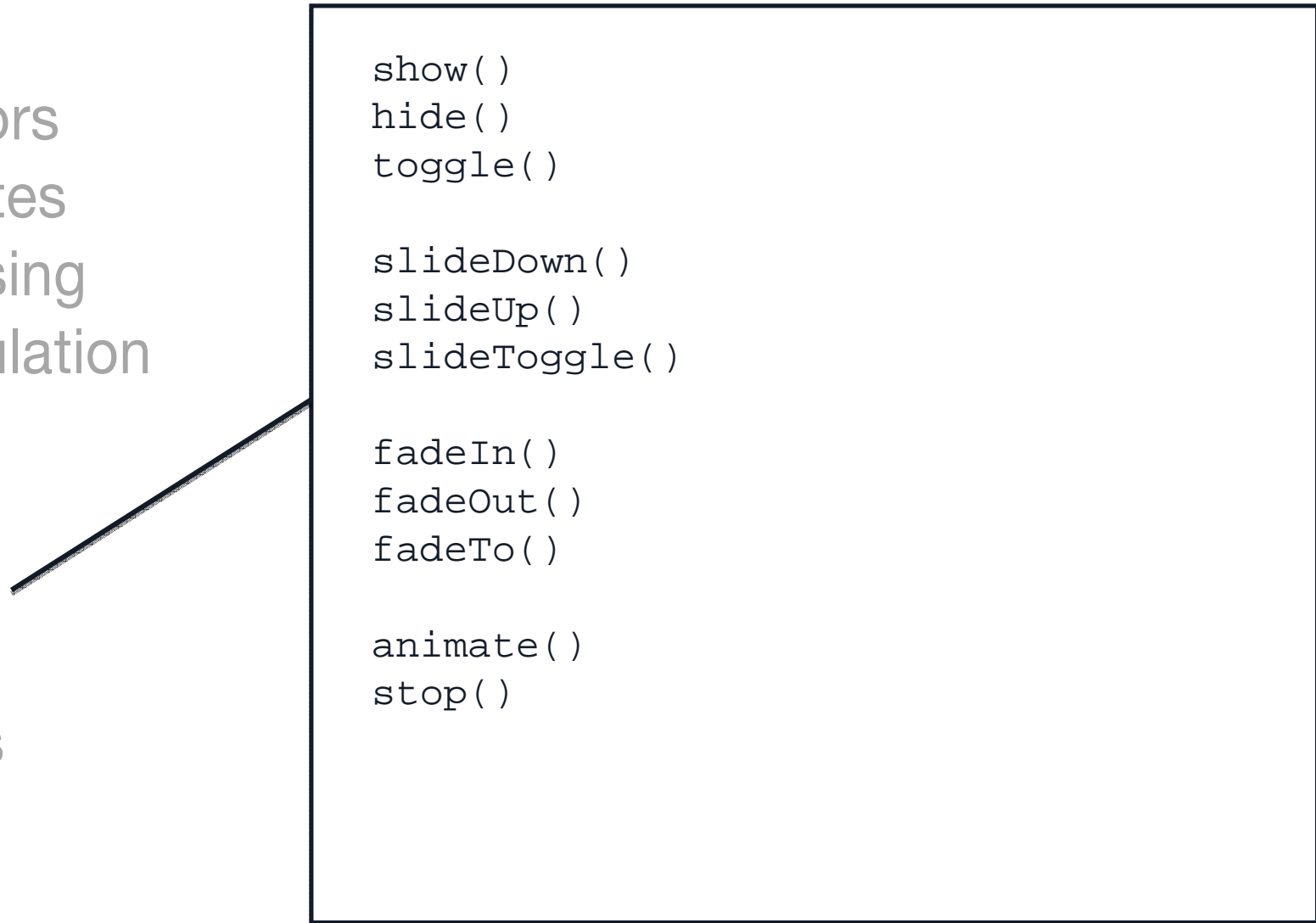
<script src="jquery.js"></script>
<script>

jQuery("p").live("click", function(){
    $(this).after("<p>click me</p>");
});

</script>
</body>
</html>
```

<http://jsbin.com/epeha/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- **Effects**
- Ajax
- Utilities



```
show()  
hide()  
toggle()  
  
slideDown()  
slideUp()  
slideToggle()  
  
fadeIn()  
fadeOut()  
fadeTo()  
  
animate()  
stop()
```


- Co
- Se
- At
- Tr
- M
- CS
- EV
- Ef
- Aj
- Ut

```
var tooltip = $("")
.addClass("tooltip")
.appendTo("body");
$("input").mouseover(function(event) {
    tooltip.text(this.title).css({
        left: event.pageX,
        top: event.pageY
    }).fadeIn();
}).mouseout(function() {
    tooltip.fadeOut();
});
```

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- **Effects**
- Ajax
- Utilities

```
<!DOCTYPE html><html><head>
<style>
div{background-
color:#bca;width:100px;border:1px solid green;}
</style>
</head>
<body>
<div id="block">Hello!</div>
<script
src="http://ajax.googleapis.com/ajax/libs/jquer
y/1.3.2/jquery.min.js" ></script>
<script>

jQuery("#block").animate({
    width: "70%",
    opacity: 0.4,
}, 1500);

</script></body></html>
```

<http://jsbin.com/ulexu/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- **Ajax**
- Utilities

```
jQuery.ajax()           jQuery.ajaxSetup()  
jQuery.get()           serialize()  
jQuerygetJSON()       serializeArray()  
jQuery, getScript()  
jQuery.post()  
  
load()  
  
ajaxComplete()  
ajaxError()  
ajaxSend()  
ajaxStart()  
ajaxStop()  
ajaxSuccess()
```

- Co
- Sel
- Att
- Tra
- Ma
- CS
- Eve
- Eff
- Aja
- Util

```
<!DOCTYPE html><html><body>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jq
uery.min.js" ></script>
<script>

jQuery.getJSON("http://api.flickr.com/services/feeds/phot
os_public.gne?tags=jquery&tagmode=all&format=json&jsoncal
lback=?",
function(data){
    jQuery.each(data.items, function(i,item){
        jQuery("<img/>")
        .attr("src", item.media.m)
        .appendTo("body");
        if ( i == 30 ) return false;
    });
});
</script></body></html>
```

<http://jsbin.com/ivifa/edit#html>

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
jQuery.support           jQuery.trim()  
jQuery.boxModel         jQuery.param()  
  
jQuery.each(),  
jQuery.extend(),  
jQuery.grep(),  
jQuery.makeArray(),  
jQuery.map(),  
jQuery.inArray(),  
jQuery.merge(),  
jQuery.unique()  
  
jQuery.isArray(),  
jQuery.isFunction()
```

Overview of jQuery API

- Core
- Selectors
- Attributes
- Traversing
- Manipulation
- CSS
- Events
- Effects
- Ajax
- Utilities

```
// returns "hello"  
jQuery.trim(" hello ");  
  
//  
returns "name=John&url=ejohn.org"  
jQuery.param({ name: "John",  
url: "ejohn.org" });
```

플러그인 만들기

Step 1. create a private scope for \$ alias

```
<!DOCTYPE html><html><body>  
<script  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min  
.js"></script>  
<script>  
(function($){  
  
})(jQuery);  
</script></body></html>
```

Step 2. attach plugin to fn alias (aka prototype)

```
<!DOCTYPE html><html><body>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        $(this).text($(this).text().replace(/hate/g, "love"));
    };
})(jQuery);
</script></body></html>
```


Step 2. attach plugin to fn alias (aka prototype)

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        $(this).text($(this).text().replace(/hate/g, "love"));
    };
})(jQuery);
jQuery("p").loveNotHate();
</script></body></html>
```

<http://jsbin.com/asice/edit>

Step 3. add implicit iteration

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($){
    $.fn.loveNotHate = function(){
        this.each(function(){
            $(this).text($(this).text().replace(/hate/g, "love"));
        });
    };
})(jQuery);
jQuery("p").loveNotHate();
</script></body></html>
```

<http://jsbin.com/asice/edit>

Step 3. add implicit iteration

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        this.each(function() {
            $(this).text($(this).text().replace(/hate/g, "love"));
        });
    };
})(jQuery);
jQuery("p").loveNotHate();
</script></body></html>
```

Step 4. enable chaining

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        return this.each(function() {
            $(this).text($(this).text().replace(/hate/g, "love"));
        });
    };
})(jQuery);
jQuery("p").loveNotHate();
</script></body></html>
```

Step 4. enable chaining

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        return this.each(function() {
            $(this).text($(this).text().replace(/hate/g, "love"));
        });
    };
})(jQuery);
jQuery("p").loveNotHate().hide();
</script></body></html>
```

Step 5. add default options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        return this.each(function() {
            $(this).text($(this).text().replace(/hate/g, $.fn.loveNotHate.defaults.text));
        });
    };
    $.fn.loveNotHate.defaults = {text: "love"};
})(jQuery);
jQuery("p").loveNotHate();
</script></body></html>
```

<http://jsbin.com/asice/edit>

Step 6. add custom options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function() {
        return this.each(function() {
            $(this).text($(this).text().replace(/hate/g, $.fn.loveN
otHate.defaults.text));
        });
    };
    $.fn.loveNotHate.defaults = {text: "love"};
})(jQuery);
jQuery("p").loveNotHate({text: "love-love-love"});
</script></body></html>
```

<http://jsbin.com/asice/edit>

Step 6. add custom options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function(options) {
        return this.each(function() {
            $(this).text($(this).text().replace(/hate/g, $.fn.loveN
otHate.defaults.text));
        });
    };
    $.fn.loveNotHate.defaults = {text: "love"};
})(jQuery);
jQuery("p").loveNotHate({text: "love-love-love"});
</script></body></html>
```

<http://jsbin.com/asice/edit>

Step 6. add custom options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($) {
    $.fn.loveNotHate = function(options) {
        options = $.extend({}, $.fn.loveNotHate.defaults, options);
        return this.each(function() {
            $(this).text($(this).text().replace(/hate/g, $.fn.loveNotHate.defaults.text));
        });
    };
    $.fn.loveNotHate.defaults = {text: "love"};
})(jQuery);
jQuery("p").loveNotHate({text: "love-love-love"});
</script></body></html>
```

<http://jsbin.com/asice/edit>

Step 6. add custom options

```
<!DOCTYPE html><html><body>
<p>I hate jQuery!</p>
<p>I mean really hate jQuery!</p>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min
.js"></script>
<script>
(function($){
    $.fn.loveNotHate = function(options){
        options = $.extend({},$.fn.loveNotHate.defaults, options);
        return this.each(function(){
            $(this).text($(this).text().replace(/hate/g,options.te
                xt));
        });
    };
    $.fn.loveNotHate.defaults = {text:"love"};
})(jQuery);
jQuery("p").loveNotHate({text:"love-love-love"});
</script></body></html>
```

<http://jsbin.com/asice/edit>

실습 내용: jQuery Playground

<http://code.google.com/apis/ajax/playground/#jquery>

Google code Code Playground

[Tell us what you think](#) | [Sign in](#)

The screenshot displays the jQuery Playground interface. On the left, the 'Pick an API' panel lists various services, with 'jQuery' selected under the 'Libraries' section. The main 'Edit Code' area contains the following JavaScript code:

```
1 /*
2  * How to load jQuery and then use the Search API with it.
3  */
4
5 google.load("jquery", "1");
6
7 // on page load complete, fire off a jQuery json-p query
8 // against Google web search
9 function OnLoad(){
10  var url = "http://ajax.googleapis.com/ajax/services/search/web?q=google&v=1.
11  $.getJSON(url, function (data) {
12    if (data.responseData.results &&
13        data.responseData.results.length > 0) {
14      var results = data.responseData.results;
15      var content = document.getElementById('content');
16      content.innerHTML = '';
17      for (var i=0; i < results.length; i++) {
18        // Display each result however you wish
19        content.innerHTML += results[i].titleNoFormatting + "<br/>";
20      }
21    }
22  }
```

At the bottom, the 'Output' panel shows the results of the search:

```
Google
Google Maps
Google.org - Google Technology-Driven Philanthropy
Google Videos
```