

# 1. jQuery 개요

---

## 1-1. jQuery란?

- 오픈 소스 자바스크립트 라이브러리 중 한가지 이다.
- 가볍고 빠르며, 간결한 문법으로 구성되어 있다.
- jQuery 셀렉터, 객체 접근자, 데이터 캐시를 제공한다.
- Effects 기능성을 제공하여 풍부한 상호작용 구현이 쉽다.
- 거의 완벽한 크로스 브라우저 지원한다.
- 메서드 체인(연쇄구현)방식의 프로그래밍 패턴을 지원한다.
- DOM 상호작용부터 Ajax까지 지원한다.
- 풍부한 플러그인을 지원한다.
- jQuery 코어 또는 셀렉터를 사용하여 jQuery 객체를 생성함으로써 시작된다.

## 1-2. JavaScript의 함수관련 독특한 문법 스타일

### 1. 함수를 변수에 대입할 수 있다.

```
var myFunc = function( ) {  
    alert("안녕하세요");  
}  
  
function showAlert( myParam ){  
    myParam();  
}
```

### 2. 정의된 함수를 다른 함수의 매개변수로 사용하여 호출 할 수 있다.

```
showAlert(myFunc);
```

### 3. 함수를 매개변수로 바로 설정 할 수 있다.

```
showAlert(function(){  
    alert("안녕하세요2");  
});
```



## 2. jQuery Core

---

## 2. jQuery Core

- 모든 jQuery의 기본이 되는 문법을 정의한 함수로 jQuery()가 있다.
- jQuery()는 축약해서 \$()를 대신 사용할 수 있다.
- jQuery객체를 만들어낸다.

### 1. jQuery() 함수들

#### jQuery( "selector" [, context] )

- selector에는 태그(요소)명, CSS selector, XPath등을 지정하여 일치하는 요소를 찾아낸다.
- context가 지정되면 해당 context안에서 selector에 지정한 요소를 찾아낸다.
- jQuery객체를 반환한다.

```
<script type="text/javascript">  
  var myjQuery1 = jQuery("p");  
  var myjQuery2 = $("p", document.forms[2]);  
</script>
```

## 2. jQuery Core

### 1. jQuery() 함수들 (계속)

`jQuery( "html" [, ownerDocument ] ) , jQuery( "html" , props )`

- 주어진 html을 생성한 후 생성된 요소를 갖는 jQuery객체를 반환한다.
- ownerDocument는 생성된 요소가 위치한 문서를 의미한다.
- props는 새로 만들어진 요소의 속성, 이벤트, 함수(메소드)등을 지정한다.
- 이 함수는 XML에는 적용되지 않고 HTML에만 적용된다.

```
<script type="text/javascript">
    $(document).ready(function(){
        $("<div><p>안녕하세요</p></div>").appendTo("span");
    });
</script>
```

- “<div><p>안녕하세요, 방가요</p></div>” 형식의 HTML을 span 엘리먼트에 추가한다.
- 문서에 span 엘리먼트가 여러 개이면 모든 span 엘리먼트에 HTML내용이 추가된다.

## 2. jQuery Core

### 1. jQuery() 함수들 (계속)

```
<script type="text/javascript">
  $(document).ready(function(){
    $("<div/>", {
      class: "test",
      text: "Click me!",
      click: function(){
        $(this).toggleClass("test");
      }
    }).appendTo("body");
  });
</script>
```

- “<div class='test'>Click me!</div>” 형식의 HTML을 구성하고 여기에 click이벤트를 설정한 후 이 요소를 body 엘리먼트에 추가한다.



## 2. jQuery Core

### 1. jQuery() 함수들 (계속)

#### jQuery( elements )

- elements에 주어진 요소를 갖는 jQuery객체를 반환한다.

```
<script type="text/javascript">
    $(document).ready(function(){
        alert( $("div p").html() );
    });
</script>
```

- div의 자식엘리먼트 중 모든 p태그를 검색한 후 검색된 p태그의 contents를 html형식으로 alert창에 보여 준다.
- 검색된 p태그의 개수에 따라 반환되는 요소들은 0개 이상의 배열로 처리된다.
- 위 예제에서 검색된 요소의 개수가 2개 이상이면 첫 번째(0번) 요소에 대한 html() 함수의 결과를 출력한다.

## 2. jQuery Core

### 1. jQuery() 함수들 (계속)

#### jQuery( fn )

- fn에는 처리될 함수를 지정한다.
- jQuery( fn )은 jQuery(document).ready( fn )와 같다.
- 즉, 문서의 DOM element가 모두 로드된 후 지정한 함수를 호출한다.
- 이 함수는 XML에는 적용되지 않고 HTML에만 적용된다.

```
<script type="text/javascript">  
    $(function(){  
        $("div").css("background-color", "yellow");  
    });  
</script>
```

- 문서상의 모든 div요소들의 배경색을 yellow로 지정한다.

## 2. jQuery Core

### 2. jQuery 객체 조작하기

- jQuery 객체는 0개 이상의 DOM 요소를 가질 수 있다.
- jQuery는 이러한 DOM 요소에 대한 조작을 실시하는 메소드들을 제공한다.

#### size() , length

- 검색된 DOM 요소의 개수를 반환한다.

```
<script type="text/javascript">
    $(document).ready(function(){
        alert( $("p").size() );
        alert( $("p").length );
    });
</script>
```

- jQuery 객체가 내부에 포함하고 있는 p 엘리먼트의 개수를 출력한다.
- 위 2가지 모두 결과가 같다.

## 2. jQuery Core

### 2. jQuery 객체 조작하기 (계속)

#### each( fn )

- 검색된 DOM요소의 개수만큼 지정된 fn함수를 호출한다.
- 지정된 함수 안에서 사용되는 this는 해당되는 각각의 DOM요소를 의미한다.
- fn함수에 전해 줄 수 있는 파라미터는 해당 요소의 index변수와 this와 같은 역할을 하는 변수를 지정할 수 있다.

```
<script type="text/javascript">
    $(function(){
        var str = "";
        $("p").each(function() {
            str += $(this).html() + "Wn";
        });
        alert( str );
    });
</script>
```

## 2. jQuery Core

### 2. jQuery객체 조작하기 (계속)

```
<script type="text/javascript">
    $(function(){
        var str = "";
        $("p").each(function( i , currElement ) {
            str += (i + 1) + "번째 요소 : " + $(currElement).html() + "₩n";
        });
        alert(str);
    });
</script>
```

- 지정된 callback함수에서 this는 jQuery객체에서 반환된 각각의 element를 의미한다.
- 그러므로 만약 해당 엘리먼트의 태그명을 알려면 this.tagName형식으로 사용해야한다.
- 두번째 예제에서 i는 해당 엘리먼트의 index값이, currElement에는 반환되는 각각의 element가 자동으로 할당된다. (지정할 변수명은 임의로 지정할 수 있다.)

## 2. jQuery Core

### 2. jQuery 객체 조작하기 (계속)

#### get( ) , get( N ) , [N]

- get()함수는 매치되는 DOM요소들을 배열로 반환한다.
- get(N)함수는 DOM요소 배열에서 N번째 요소를 반환한다. ([N]와 같다.)
- N이 배열 범위를 넘을 경우 "undefined"를 반환한다.
- 배열의 첨자는 0부터 시작된다.

```
<script type="text/javascript">
    $(function(){
        var str = "";
        var pElement = $("p").get();
        for(var i=0; i<pElement.length; i++){
            str += $(pElement[i]).html() + "Wn";
        }
        alert( str );
    });
</script>
```

## 2. jQuery Core

### 2. jQuery 객체 조작하기 (계속)

```
<script type="text/javascript">
    $(function(){
        alert( $( $("div").get(1) ).html() );
        alert( $( $("div")[1] ).html() );
    });
</script>
<body>
    <p>첫번째 </p>
    <div>
        <p>두번째 </p>
    </div>
    <div>
        <p>세번째 </p><p>네번째 </p>
    </div>
</body>
```

- 요소 집합 중 1번째 div엘리먼트를 검색하므로 <div><p>세번째</p><p>네번째</p></div>를 반환한다.

## 2. jQuery Core

### 3. jQuery의 엘리어스인 \$ 사용금지

#### jQuery.noConflict()

- 다른 종류의 JavaScript Library 중 jQuery의 별칭 행태로 사용되는 "\$"와 충돌할 경우 jQuery의 별칭 예약어를 변경하여 충돌을 방지한다.
- 즉, jQuery() 의 별칭인 \$() 의 사용을 더 이상 하지 않으려 할 때 실행한다.

```
<script type="text/javascript">
    jQuery.noConflict();
    // 이후의 $는 jQuery()의 $()가 아니다.
    ...

    // noConflict()를 호출하고서 "$"를 사용하는 방법
    (function($) {    // 이 안에서는 "$"문자를 jQuery의 alias로 사용 가능
        $(function() {
            ...
        });
    })(jQuery);
</script>
```



## 3. jQuery Selectors

---

## 3-1. jQuery Selectors

### 1. jQuery Selectors의 개요

- HTML 문서 안에는 다양한 엘리먼트들이 포함되어 있는데, 이를 선택하고 컨트롤 하기란 결코 간단한 일이 아니다.
- 하지만 jQuery에서는 \$("셀렉터 문법") 함수를 사용하여 매우 간단하고 다양한 스타일의 엘리먼트들을 쉽게 얻을 수 있다.
- 셀렉터로 얻은 엘리먼트들을 확장 집합이라 하는데, 특수한 배열 형태의 객체로 반환이 된다. (jQuery객체)
- 이렇게 반환된 객체들은 jQuery가 지원하는 매우 다양한 함수들을 사용할 수 있다.
- 셀렉터는 jQuery의 가장 강력한 기능 중 하나 이다.

## 3-1. jQuery Selectors

### 2. 엘리먼트 관련 CSS 셀렉터들

셀렉터	설 명
*	모든 요소
E1	태그명이 E1인 모든 요소
E1.class	E1요소 중 CLASS속성값이 class와 같은 모든 요소
E1#id	E1요소 중 ID속성값이 id와 같은 요소
E1, E2	모든 E1요소와 모든 E2 요소
E1 E2	E1의 자식 요소 중 모든 E2 요소
E1 > E2	E1의 바로 아래 자식요소 중 모든 E2 요소
E1 + E2	E1의 바로 다음에 나오는 형제요소 중 E2 요소
E1 ~ E2	E1의 다음에 나오는 형제요소 중 모든 E2 요소

## 3-1. jQuery Selectors

### 2. 엘리먼트 관련 CSS 셀렉터들 (예제3-1-2)

```
<script type="text/javascript">
$(document).ready(function(){
    $("#list").css("border", "1px solid silver");
    $("p").css("border","2px solid red");
    $("span").css("background-color","pink").css("color","#009999");
    $("span a").css("text-decoration", "underline").css("color","#000000");
    $("span.detail").css("color", "blue");
    $("div~b").css("background", "green");
    $("div>b").css("border", "1px solid red");
    $("div+b").css("border", "1px solid red");});
</script>
<style type="text/css">
* { font-size:12px; font-family:돋움}
</style>
<body>
<div>
<span>It 교육의 요람 <a href="http://ddit.or.kr">(재)대덕 인재 개발원</a>으로 오세요
</span><br><br>
```

## 3-1. jQuery Selectors

### 2. 엘리먼트 관련 CSS 셀렉터들 (예제3-1-2 계속)

```
<a href="mailto:power2c@ddit.or.kr">빠워 에게 메 일 보내기 </a>
<p>
RIA 기 술 : <input type="checkbox" name="ria" value="플렉스"/> 플렉스
<input type="checkbox" name="ria" value="실버라이트"/> 실버라이트
<input type="checkbox" name="ria" value="HTML5"/> HTML5<br>
성 별 : <input type="radio" name="sex" value="남자" /> 남자
<input type="radio" name="sex" value="여자" /> 여자 <br>
<input type="button" value="클릭 ~"?/> <br>
</p>

<div id="list">selector의 종류들입니다.</div>

<b>엘리먼트 : </b><span class="detail"><b>태그명, ID속성값, CLASS속성값 이 용</b>
</span><br>
<b>속성 : </b><span class="detail">각 요소에 설정된 속성명과 속성값 이 용</span><br>
<b>form : </b><span class="detail">input, select, textarea등의 특성을 이 용</span>
</div>
</body>
```

## 3-1. jQuery Selectors

### 3. 속성 관련 CSS 셀렉터들

셀렉터	설 명
E1[attr]	attr속성을 갖는 모든 E1 요소
E1[attr=val]	attr속성값이 val인 모든 E1 요소
E1[attr^=val]	attr속성값이 val 으로 시작하는 모든 E1 요소
E1[attr!=val]	attr속성값이 val 값과 같지 않은 모든 E1 요소
E1[attr\$=val]	attr속성값이 val 으로 끝나는 모든 E1 요소
E1[attr*=val]	attr속성값이 val 을 포함하는 모든 E1 요소
E1[attr =val]	attr속성값이 val 과 같거나 'val-' 로 시작하는 모든 E1 요소
E1[attr~=val]	attr속성값이 공백으로 구분된 값을 가질 경우 구분된 값 중에 val 값과 같은 값을 갖는 모든 E1 요소

## 3-1. jQuery Selectors

### 3. 속성 관련 CSS 셀렉터들 (예제3-1-3)

```
<script type="text/javascript">
$(document).ready(function(){
    $("img[alt]").css("border", "1px solid red");
    $("input[type='button']").css("background-color", "yellow");
    $("a[href^='mailto:']").css("background-color", "lightblue");
    $("img[id!='hibiscus']", $("div")).css("border", "3px solid blue");
    $("img[src$='.jpg']").css("border", "2px solid #00ffff");
    $("a[href*='power2c']").css("text-decoration", "none");
    $("img[id|='little']").css("border", "2px solid green");
    $("img[title~='named']").css("border", "2px solid pink");});
</script>

<style type="text/css">
* { font-size: 12px; font-family: 돋움 }
</style>
```

## 3-1. jQuery Selectors

### 3. 속성 관련 CSS 셀렉터들 (예제3-1-3 계속)

```
<body>
<div>
  <a href="http://ddit.or.kr">(재)대덕 인재 개발원 </a><br><br>
  <a href="mailto:power2c@ddit.or.kr">빠워 에 게 메 일 보 내 기 </a><br><br>

</div>
  
  
  
  
  
  
</div>

<div id="someDiv">이 곳은 ID속성이 <tt>someDiv</tt>인 &lt;div&gt;태그 입니다. </div>
<div title="myTitle1"><span>Hello</span></div>
<div title="myTitle2"><span>Goodbye</span></div>
```



## 3-1. jQuery Selectors

### 3. 속성 관련 CSS 셀렉터들 (예제3-1-3 계속)

```
<form onsubmit="return false;">
  <div>
    <label>Text:</label> <input type="text" id="aTextField" name="someTextField"/>
  </div>
  <div>
    <label>Radio group:</label>
    <input type="radio" name="radioGroup" id="radioA" value="A"/> A
    <input type="radio" name="radioGroup" id="radioB" value="B"/> B
  </div>
  <div>
    <label>Checkboxes:</label>
    <input type="checkbox" name="checkboxes" id="checkbox1" value="1"/> 1
    <input type="checkbox" name="checkboxes" id="checkbox2" value="2"/> 2
  </div>
  <button type="submit" id="submitButton">Submit</button>
  <input type="button" value="클릭~~">
</form>
</body>
```

## 3-1. jQuery Selectors

### 4) Form 관련 셀렉터들

셀렉터	설 명
:input	모든 input, textarea, select, button 요소들과 일치.
:text	text 타입의 모든 input 요소들과 일치.
:password	password 타입의 모든 input 요소들과 일치.
:radio	radio 타입의 모든 input 요소들과 일치.
:checkbox	checkbox 타입의 모든 input 요소들과 일치.
:submit	submit 타입의 모든 input 요소들과 일치.
:image	image 타입의 모든 input 요소들과 일치.
:reset	reset 타입의 모든 input 요소들과 일치.
:button	모든 button 요소들과 button 타입의 input 요소들과 일치.
:file	file 타입의 모든 input 요소들과 일치.

## 3-1. jQuery Selectors

### 4. Form관련 셀렉터들 (예제3-1-4)

```
<script type="text/javascript">
$(document).ready(function(){
    $("input").css("border", "2px solid red");
    $("input:button").css("background-color", "yellow");
    $("button").css("background-color", "yellow");
    $("submit").css("background-color", "pink");
    $("reset").css("background-color", "yellow");
    $("checkbox").css("border", "2px solid red");
    $("radio").css("border", "2px solid red");
    $("text").css("border", "2px solid red");
    $("password").css("border", "2px solid red");
    $("file").css("border", "2px solid red");
    $("image").css("border", "2px solid red");
</script>

<style type="text/css">
* { font-size:12px; font-family:돋움 }
</style>
```

## 3-1. jQuery Selectors

### 4. Form관련 셀렉터들 (예제3-1-4 계속)

```
<body>
<form onsubmit="return false;">
Text : <input type="text"/><br>
Password : <input type="password" /><br><br>
Radio : <input type="radio" name="radioGroup" id="radioA" value="A"/> A
        <input type="radio" name="radioGroup" id="radioB" value="B"/> B<br><br>
Checkbox : <input type="checkbox" name="checkboxes" id="checkbox1" value="1"/> 1
            <input type="checkbox" name="checkboxes" id="checkbox2" value="2"/> 2<br><br>
Textarea : <br>
            <textarea rows="15" cols="50" id="myTextarea" id="myTextarea"></textarea><br><br>
Image : <input type="image" src="images/image.1.jpg"><br><br>
File : <input type="file"><br><br>
Buttons :
        <button type="submit" id="submitButton">Submit</button>
        <input type="button" value="일 반버 튠">
        <input type="submit" value="전 송버 튠">
        <input type="reset" value="초기 화버 튠">
</form></body>
```

## 3-2. 필터들(Filters)

### 1. 기본 필터들

필터	설명
:first	선택된 개체들 중 첫 번째 요소와 일치.
:last	선택된 개체들 중 마지막 요소와 일치.
:not(selector)	괄호에 주어진 셀렉터와 일치되는 모든 요소를 제외.
:even	짝수 요소들과 일치 (0부터 시작)
:odd	홀수 요소들과 일치 (0부터 시작)
:eq(index)	인덱스에 해당하는 단일 요소와 일치.
:gt(index)	주어진 인덱스보다 큰 인덱스를 갖는 모든 요소.
:lt(index)	주어진 인덱스보다 작은 인덱스를 갖는 모든 요소.
:header	모든 헤더 요소들(h1, h2, h3 등)과 일치.
:animated	현재 애니메이션이 동작중인 모든 요소와 일치.

## 3-2. 필터들(Filters)

### 1. 기본 필터들 (예제3-2-1)

```
<script type="text/javascript">
$(function(){
    $("tr").css("text-align", "center");
    $(":header:eq(0)").css("font-weight", "bold").css("color", "blue");
    $("tr:first").css("background", "silver");
    $("tr:last").css("background", "yellow");
    $("tr:odd").css("background", "#efefef");
    $("tr:even").css("background", "#3388cc");
    $("tr:gt(4)").css("background", "silver");
    $("tr:lt(4)").css("background", "silver");
});
</script>

<style type="text/css">
* { font-size: 12px; font-family: 돋움 }
.scoreTable { background: white; border-collapse: collapse }
</style>
```

## 3-2. 필터들(Filters)

### 1. 기본 필터들 (예제3-2-1 계속)

```
<body>
<h3>수학 시험 점수</h3>
<table class="scoreTable" border="1" cellpadding="10">
<tr><td width="100">이 름</td><td width="100">점 수</td></tr>
<tr><td>홍길동</td><td>80</td></tr>
<tr><td>이지매</td><td>98</td></tr>
<tr><td>성춘향</td><td>85</td></tr>
<tr><td>이몽룡</td><td>96</td></tr>
<tr><td>강감찬</td><td>88</td></tr>
<tr><td>이순신</td><td>90</td></tr>
<tr><td>박문수</td><td>92</td></tr>
</table>
</body>
```

## 3-2. 필터들(Filters)

### 2) Content 필터들

필터	설명
:contains(text)	지정한 텍스트를 포함하는 요소들. (대소문자 구분, 자식 요소의 텍스트까지 검색한다.)
:empty	대상 요소 중 자식 요소가 없고 텍스트도 가지지 않는 요소들.
:has(selector)	지정된 셀렉터에 해당하는 요소를 갖는 모든 요소들과 일치.
:parent	대상 요소 중 자식 요소를 갖거나 텍스트를 갖는 요소들이 이에 해당됨.



## 3-2. 필터들(Filters)

### 2) Content 필터들 (예제3-2-2-1)

```
<script type="text/javascript">
$(function(){
    $("span:contains('jQuery')").css("border", "1px solid blue");
    $("span:not(:contains('jQuery'))").css("border", "1px solid red");
});
</script>

<style type="text/css">
* { font-size:12px; font-family:돋움 }
</style>

<body>
<div>Hello, <span>Enjoy developing! </span> </div>
<p>It's another sector! <span>Thanks, jQuery! </span> </p>
<div class="myClass">jQuery! </div>
<span class="notMyClass">It's so easy! </span>
</body>
```

## 3-2. 필터들(Filters)

### 2) Content 필터들 (예제3-2-2-2)

```
<script type="text/javascript">
$(function(){
    $("td:empty").text("새로운값").css("background", "yellow");
});
</script>

<style type="text/css">
* { font-size:12px; font-family:돋움 }
</style>

<body>
<table border="1">
<tr><td>TD 1-1</td><td></td></tr>
<tr><td>TD 2-1</td><td></td></tr>
<tr><td></td><td>TD 3-2</td></tr>
</table>
</body>
```

## 3-2. 필터들(Filters)

### 2) Content 필터들 (예제3-2-2-3)

```
<script type="text/javascript">
$(function(){
    $("div:has(span)").css("border", "1px solid blue");
});
</script>

<style type="text/css">
* { font-size:12px; font-family:돋움 }
</style>

<body>
<div>
    Hello, <span>Enjoy developing!</span> <div class="myClass">jQuery!</div>
</div>
<p>It's another sector! <span>Thanks, jQuery!</span> </p>
<span class="notMyClass">It's so easy!</span>
</body>
```

## 3-2. 필터들(Filters)

### 2) Content 필터들 (예제3-2-2-4)

```
<script type="text/javascript">
$(function(){
    $("span:parent").css("border", "1px solid blue");
});
</script>

<style type="text/css">
* { font-size:12px; font-family:돋움 }
</style>

<body>
<span></span>
<div>
    Hello,<span>Enjoy developing!</span><div class="myClass">jQuery!</div>
</div>
<p>It's another sector! <span>Thanks, jQuery!</span></p>
<span class="notMyClass">It's so easy!</span>
</body>
```

## 3-2. 필터들(Filters)

### 3) 자식(Child) 필터들

필터	설명
:nth-child (index/even/odd/ equation)	자식 중 index로 지정된 위치의 요소들과 일치되거나, even(짝수), odd(홀수)번째에 해당하는 자식들과 일치. (단, index는 1부터 시작.)
:first-child	첫 번째 자식인 모든 요소와 일치.
:last-child	마지막 자식인 모든 요소와 일치.
:only-child	자신이 부모 요소의 유일한 자식인 모든 요소와 일치.

## 3-2. 필터들(Filters)

### 3) 자식(Child) 필터들(예제3-2-3-1)

```
<script type="text/javascript">
$(function(){
    $("td:nth-child(3)").css("backgroundColor", "yellow");
    $("td:nth-child(3n+1)").css("backgroundColor", "pink");
    $("td:nth-child(odd)").css("backgroundColor", "cyan");
    $("td:nth-child(even)").css("backgroundColor", "red");
});
</script>
<body>
<table border="1">
<tr><td>TD #1</td><td>TD #2</td><td>TD #3</td><td>TD #4</td></tr>
<tr><td>TD #5</td><td>TD #6</td><td>TD #7</td><td>TD #8</td></tr>
<tr><td>TD #9</td><td>TD #10</td><td>TD #11</td><td>TD #12</td></tr></table>
<table border="1">
<tr><td>TD #1</td><td>TD #2</td><td>TD #3</td><td>TD #4</td>
<td>TD #5</td><td>TD #6</td><td>TD #7</td><td>TD #8</td>
<td>TD #9</td><td>TD #10</td><td>TD #11</td><td>TD #12</td></tr></table>
</body>
```

## 3-2. 필터들(Filters)

### 3) 자식(Child) 필터들 (예제3-2-3-2)

```
<script type="text/javascript">
$(function(){
    $("td:first-child").css("backgroundColor", "yellow");
    $("td:last-child").css("backgroundColor", "cyan");
    $("td:only-child").css("backgroundColor", "green");
});
</script>

<body>
<table border="1">
<tr><td>TD #1</td><td>TD #2</td><td>TD #3</td></tr>
<tr><td colspan="3">TD #4</td></tr>
<tr><td colspan="2">TD #5</td><td>TD #6</td></tr>
</table>
</body>
```

## 3-2. 필터들(Filters)

### 4) Visibility 필터들

필터	설명
:hidden	hidden 상태인 모든 요소들. (1) display:none;인 것 (2) type="hidden";인 것 (3) width:0; height:0;인 것 (4) 부모 Element가 hidden 인 것 (5) visibility:hidden or opacity: 0 은 보이는 것으로 간주
:visible	visible 상태인 모든 요소들. (1) display:none;가 아닌 것 (2) type="hidden";가 아닌 것 (3) width:0; height:0;가 아닌 것 (4) 부모 Element가 hidden 이 아닌 것



## 3-2. 필터들(Filters)

### 5) Form 필터들

필터	설명
:enabled	현재 enable 상태인 모든 요소와 일치.
:disabled	현재 disable 상태인 모든 요소와 일치.
:checked	체크된 모든 요소들과 일치.
:selected	선택된 모든 요소들과 일치.

## 3-2. 필터들(Filters)

### 4) Visibility 필터들 (예제3-2-4)

```
<script type="text/javascript">
$(function(){
    var result="", input_tag="";
    $(":hidden", "body").each(function() {
        if(this.tagName == "INPUT") {
            input_tag+="input의 hidden객체 [ "+$(this).attr("name")+":"+$(this).val()+" ]\n";
        } else { result+=$(this).html()+"\n"; }
    });
    alert("hidden결과 \n" + input_tag + result);
    result=""; input_tag="";
    $(":visible", "#testForm").each(function() {
        if(this.tagName == "INPUT") {
            input_tag+="input의 visible객체 [ "+$(this).attr("name")+":"+$(this).val()+" ]\n";
        } else { result+=$(this).html()+"\n"; }
    });
    alert("visible결과 \n" + input_tag + result);
});
</script>
```

## 3-2. 필터들(Filters)

### 4) Visibility 필터들 (예제3-2-4 계속)

```
<body>
<form id="testForm">
  <input type="hidden" name="userAge" value="33">
  <input type="hidden" name="userCity" value="대전">
  ID : <input type="text" name="userID">
    <span style="display:none">ID가 중복됩니다. </span> <br>
  이름 : <input type="text" name="userName">
    <span style="visibility:hidden">이름을 입력하세요 </span> <br>
  성별 : <input type="radio" name="sung" value="남" checked>남자
    <input type="radio" name="sung" value="여">여자
</form>
</body>
```

## 3-2. 필터들(Filters)

### 5) Form 필터들 (예제3-2-5-1)

```
<script type="text/javascript">
$(function(){
    var result="";
    $("form :enabled").each(function() {
        if($.trim($(this).html()) == "")
            result += "<" + this.tagName + " type=" + $(this).attr("type") + " name=" +
                $(this).attr("name") + ">";
        result += $(this).css("border", "5px solid #66ff00").html()+"\n";
    });
    alert(result); result="";
    $("form :disabled").each(function() {
        if($.trim($(this).html()) == "")
            result += "<" + this.tagName + " type=" + $(this).attr("type") + " name=" +
                $(this).attr("name") + " disabled="+ $(this).attr("disabled")+ ">";
        result += $(this).css("border", "5px solid blue").html()+"\n";
    });
    alert(result);
}); </script>
```

## 3-2. 필터들(Filters)

### 5) Form 필터들 (예제3-2-5-1 계속)

```
<body>
  <form>
    <input type="button" value="Input Button" name="disButton" disabled/>
    <input type="checkbox" name="enCheckbox"/>
    <input type="file" name="disFile" disabled/>
    <input type="hidden" name="enHidden"/>
    <input type="image" name="disImage" disabled/>
    <input type="password" name="enPassword"/>
    <input type="radio" name="disRadio" disabled/>
    <input type="reset" name="enReset"/>
    <input type="submit" name="disSubmit" disabled/>
    <input type="text" name="enText"/>
    <select name="disSelect" disabled><option>Option</option></select>
    <textarea name="enTextarea"></textarea>
    <button name="disButton2" disabled>Button</button>
  </form>
</body>
```

## 3-2. 필터들(Filters)

### 5) Form 필터들 (예제3-2-5-2)

```
<script type="text/javascript">
$(function(){
    function testChecked() {
        var result="";
        $(":checkbox:checked").each(function() {
            if(result!="") result += ", ";
            result += $(this).val() ;
            $(this).css("border", "5px #009999 solid");
        });
        $(":checkbox:not(:checked)").each(function() {
            $(this).css("border", "0px");
        });

        alert(result == "" ? "취미가 없군요!!" : "취미는 " + result + "이군요!!");
    }
    testChecked();
    $(":checkbox").click(testChecked);
});
```

## 3-2. 필터들(Filters)

### 5) Form 필터들 (예제3-2-5-2 계속)

```
function testSelected() {  
    var result="봄소식 전령사들\n\n";  
    $("select[name='spring'] option:selected").each(function() {  
        result += $(this).val()+"\n";  
        $(this).css("border", "5px #009999 solid") + "\n";  
    });  
    alert($.trim(result) == "" ? "empty!" : $.trim(result));  
}  
testSelected();  
$("select[name='spring']").change(testSelected);  
});  
</script>
```

## 3-2. 필터들(Filters)

### 5) Form 필터들 (예제3-2-5-2 계속)

```
<body>
  <form>
    취미 : <input type="checkbox" name="hobby" value="여행" checked> 여행
          <input type="checkbox" name="hobby" value="장기"/> 장기
          <input type="checkbox" name="hobby" value="바둑"/> 바둑
          <input type="checkbox" name="hobby" value="독서" checked> 독서
          <input type="checkbox" name="hobby" value="낚시"/> 낚시 <br>
    봄소식 :
      <select name="spring" multiple="multiple" size="6">
        <option>개나리 </option>
        <option selected>진달래 </option>
        <option>민들레 </option>
        <option selected>벚꽃 </option>
        <option>목련 </option>
        <option>철쭉 </option>
      </select>
  </form>
</body>
```



## 4. jQuery Methods

---

## 4-1. 조작(Manipulation)관련 메소드들

### 1) 내용 확인 및 변경 메소드들

메소드	설 명
html()	일치된 요소의 html 내용을 가져온다. 이는 요소의 innerHTML 값과 동일하다. 만일, 일치된 요소가 여러 개라면 그 중 첫 번째 요소의 HTML을 가져온다.
html(val)	일치된 요소의 html 본문을 val 값으로 설정한다. 만일, 일치된 요소가 여러 개라면 모든 요소에 이러한 작업을 수행한다.
text()	일치된 모든 요소의 텍스트를 합쳐서 가져온다.
text(val)	모든 일치된 요소의 텍스트를 val 값으로 설정한다.
val()	해당 입력 요소의 value속성의 값을 가져온다.
val(data)	해당 입력 요소의 value속성에 data를 설정한다.

## 4-1. 조작(Manipulation)관련 메소드들

### 2) 요소 내부에 추가하는 메소드들

메소드	설 명
append(content)	일치된 요소 내부의 마지막 위치에 content를 추가한다.
appendTo(selector)	선택된 요소를 selector에 일치된 모든 요소들의 내부 마지막 위치에 추가한다. 만일, 일치된 요소가 본문에 존재하면 그 요소를 제거한 후 복사한다. (즉, 이동)
prepend(content)	append(content)와 동일, 다만, 내부의 처음위치에 추가한다.
prependTo(selector)	appendTo(selector)와 동일, 다만, 내부의 처음위치에 추가한다.

## 4-1. 조작(Manipulation)관련 메소드들

### 3) 요소 외부에 추가하는 메소드들

메소드	설 명
after(content)	일치된 요소 뒤에 content를 삽입한다. 요소 내부가 아닌 외부에 삽입된다.
insertAfter(selector)	선택된 요소를 selector에 의해 일치된 모든 요소들 뒤쪽에 삽입한다. 요소 내부가 아닌 외부에 삽입된다.
before(content)	일치된 요소 앞에 content를 삽입한다. 요소 내부가 아닌 외부에 삽입된다.
insertBefore(selector)	insertAfter(selector)와 유사하나, 요소 앞쪽에 삽입한다. 요소 내부가 아닌 외부에 삽입된다.

### 4) HTML로 감싸는 메소드들

메소드	설 명
wrap(html)	일치된 요소 각각을 html로 감싼다.
wrapAll(html)	일치된 요소 전체를 한꺼번에 html로 감싼다.

## 4-1. 조작(Manipulation)관련 메소드들

### 4) 삭제 메소드들

메소드	설 명
empty()	일치된 모든 요소들의 자식 Content를 삭제한다.
remove() remove(selector)	일치된 모든 요소들을 DOM에서 삭제한다. 일치되는 요소 중 selector에 일치하는 요소를 삭제한다.
detach()	remove()와 같지만 이벤트 핸들러등의 정보는 삭제되지 않는다.

### 5) 복사 메소드들

메소드	설 명
clone()	일치된 요소를 복사하고, 그를 선택한다.
clone(bool)	이벤트 처리기를 포함하여 DOM 요소를 복사하고 그를 선택한다.

## 4-1. 조작(Manipulation)관련 메소드들

### 조작관련 메소드 (예제4-1)

```
<script type="text/javascript">
$(function(){
    $(".annotation").each(function(i) {
        var no = i + 1;
        $(this).after("<sup>" + no + "</sup>");
    });
    $("button").click(function() {
        alert($(this).val());
        $(this).remove();
    });
    var $prev = $("#prevBtn").clone();
    var $next = $("#nextBtn").clone(true);

    $("div#bottomDiv").prepend($prev);
    $("div#bottomDiv").append($next);
});
</script>
```

## 4-1. 조작(Manipulation)관련 메소드들

### 조작관련 메소드 (예제4-1 계속)

```
<style type="text/css">
    * { font-size:12px; font-family:돋움; }
    .annotation { background: pink; }
</style>

<body>
    <button id="prevBtn">이전 페이지로</button>
    <button id="nextBtn">다음 페이지로</button>
    <h3>jQuery란?</h3>
    <div><b>jQuery</b>는 가볍고 빠르며, 간결한 <span class="annotation">오픈소스
    </span> 스크립트 라이브러리입니다.<br>이를 이용하면 Rich 웹 UI를 개발하는데 도움이
    되는 다양한 기능들<br>즉, HTML 문서 <span class="annotation">트래버스</span>, 이
    벤트 처리, 애니메이션, <span class="annotation">Ajax</span> 상호 작용 등을 지원하여
    <br>빠르고 견고하게 리치 웹 애플리케이션 개발을 할 수 있도록 지원합니다.
    </div><br>
    <div id="bottomDiv"><span>기 준</span></div>
</body>
```

## 4-2. 횡단(Traversing) 메소드들

### 1) Filtering하는 메소드들

메소드	설 명
eq(index)	일치된 요소들 중에서 index와 일치하는 단일 요소를 가져 온다. Index가 음수이면 뒤에서부터 카운트한다. (제일 뒤 : -1 )
filter(expr)	해당 요소 중 지정된 표현식과 매치되는 요소들만 가져 온다.
filter(func)	지정된 함수의 반환값이 true인 요소들만 가져 온다.
is(expr)	현재 개체가 표현식에 해당한다면 true, 표현식에 여러 개의 조건이 있다면 그 중 한 개만 맞아도 true가 됨
map(callback)	jQuery 개체 안에 있는 요소들의 집합을 다른 집합으로 변경해서 옮긴다.
not(expr)	지정된 표현식과 매치되지 않는 요소들만을 가져 온다.
slice(start, end)	해당 요소 중 start부터 end이전까지의 요소를 가져 온다. (end가 생략되면 끝까지의 요소를 가져 온다.)



## 4-2. 횡단(Traversing) 메소드들

### 1) Filtering하는 메소드들 (예제4-2-1)

```
<script type="text/javascript">
$(function(){
    $("div").eq(0).addClass("lightblue");
    $("div").filter(":odd")
        .eq(0).css("background", "orange")
        .end() // 방금 실행한 집합 이전으로 되돌린다.
        .eq(1).css("background", "blue")
        .end()
        .css("color", "red");
    var $myDiv = $("div").eq(5);
    if ($myDiv.is("div")) $myDiv.css("border", "4px solid yellow");
    if ($myDiv.is(".orange, .blue, .lightblue")) $myDiv.text("칼라");

    $("div").map(function() {
        return $(this).text($(this).text().toUpperCase());
    });
});
</script>
```

## 4-2. 횡단(Traversing) 메소드들

### 1) Filtering하는 메소드들 (예제4-2-1 계속)

```
<style>
  div { width:60px; height:60px; margin:10px; float:left; border:2px solid blue; }
  .blue { background:blue; }
  .lightblue { background:lightblue; }
  .orange { background:orange; }
</style>

<body>
  <div>a</div>
  <div>b</div>
  <div>c</div>
  <div>d</div>
  <div>e</div>
  <div class="orange">f</div>
  <div>g</div>
</body>
```

## 4-2. 횡단(Traversing) 메소드들

### 2) 찾기 관련 메소드들

메소드	설 명
find(expr)	해당요소의 자식 요소 중 지정된 표현식과 일치하는 요소를 검색한다.
add(expr)	검색된 현재 요소에 expr와 일치하는 요소를 추가한다. 즉, 일치되는 요소가 추가 확장되는 의미이다. (검색된 대상집합에만 추가된다.)
next(expr)	해당 요소 바로 다음에 나오는 형제 요소를 선택한다. expr 을 지정하면 대상이 되는 요소가 expr식과 일치하는 경우에만 리턴한다.
nextAll(expr)	해당 요소 다음에 나오는 모든 형제 요소들을 선택한다. expr 을 지정하면 대상이 되는 요소가 expr식과 일치하는 경우에만 리턴한다.
prev(expr)	해당 요소 바로 앞에 나오는 형제 요소를 선택한다. expr 을 지정하면 대상이 되는 요소가 expr식과 일치하는 경우에만 리턴한다.
prevAll(expr)	해당 요소보다 앞에 나오는 모든 형제 요소들을 선택한다. expr 을 지정하면 대상이 되는 요소가 expr식과 일치하는 경우에만 리턴한다.

## 4-2. 횡단(Traversing) 메소드들

### 2) 찾기 관련 메소드들 (계속)

메소드	설 명
parent(expr)	해당 요소의 부모 요소를 선택한다. expr 을 지정하면 그것으로 필터링을 수행한다.
parents(expr)	해당 요소의 조상 요소들을 모두 선택한다. expr 을 지정하면 그것으로 필터링을 수행한다.
children(expr)	해당 요소의 바로 아래의 자식 요소를 선택한다. expr 을 지정하면 그것으로 필터링을 수행한다.
contents()	해당 요소의 모든 자식 요소들(text값 포함)을 선택한다.
siblings(expr)	해당 요소의 모든 형제 요소들(자신은 제외)을 선택한다. expr 을 지정하면 그것으로 필터링을 수행한다.

## 4-2. 횡단(Traversing) 메소드들

### 2) 찾기 관련 메소드들 (예제4-2-2)

```
<script type="text/javascript">
$(function(){
    $("div:eq(1)")
        .siblings().css("border", "1px solid blue")
        .end()
        .next().text("third")
        .end()
        .nextAll().css("background", "yellow");

    $("div").find("p").css("color", "blue")
        .add("span").css("border", "1px solid red");
});
</script>
```

## 4-2. 횡단(Traversing) 메소드들

### 2) 찾기 관련 메소드들 (예제4-2-2 계속)

```
<style>
```

```
  * { font-size:12px; font-family:돋움; }
```

```
  div { width:60px; height:60px; margin:10px; float:left;
```

```
        border:1px solid silver; padding: 5px }
```

```
</style>
```

```
<body>
```

```
  <div>A <p>p a</p> </div>
```

```
  <div>B <p>p b</p> </div>
```

```
  <div>C <p>p c</p> </div>
```

```
  <div>D <br> <span>span d</span> </div>
```

```
  <div>E <br> <span>span e</span> </div>
```

```
</body>
```

## 4-3. 스타일시트(CSS)관련 메소드들

### 1) CSS 관련 메소드들

메소드	설 명
css( name )	매치되는 첫번째 요소의 스타일 속성을 반환한다. 예 : <code>var color = \$(this).css("color");</code>
css( name, value )	매치된 모든 요소에 대해 단일 스타일 속성을 설정한다. 예 : <code>\$(this).css('color', 'yellow');</code>
css( properties )	매치된 모든 요소들의 스타일 속성에 키/값을 설정한다. 예 : <code>\$(this).css( { 'color':'yellow',                                 'font-weight':'bolder' } );</code>

## 4-3. 스타일시트(CSS) 관련 메소드들

### 2) CSS class 관련 메소드들

메소드	설 명
<code>addClass( class )</code>	매치된 요소들의 각 집합에 지정된 CSS 클래스를 추가한다.
<code>hasClass( class )</code>	지정된 클래스가 매치된 요소 집합 중 최소 한 군데 이상 적용되어 있다면 true를 반환한다.
<code>removeClass( class )</code>	매치된 요소들의 각 집합에서 지정된 CSS 클래스 혹은 모든 클래스를 제거한다.
<code>toggleClass( class )</code>	지정된 클래스가 적용되지 않았다면 적용하고, 이미 적용되어 있다면 제거한다.



## 4-3. 스타일시트(CSS) 관련 메소드들

### CSS 관련 메소드들 (예제4-3-1)

```
<script type="text/javascript">
$(function(){
    $("#Mt").addClass("winter").css("width", "200");

    $("button").click(function() {
        $("#Mt").toggleClass("summer")
    });
});
</script>
<style>
    * { font-size:12px; font-family:돋움; }
    .winter { border:7px solid #dddddd; }
    .summer { border:7px solid lightblue; }
</style>
<body>
    <div></div>
    <button>눌러요~~~</button>
</body>
```

## 4-4. 속성(Attribute)관련 메소드들

### 1) 속성(Attribute) 관련 메소드들

메소드	설 명
attr( name )	메치된 첫 번째 요소의 특정 어트리뷰트에 접근하여 값을 가져온다. 만일, 지정된 어트리뷰트 명이 존재하지 않는다면 undefined가 반환된다.
attr( properties )	모든 매치되는 요소들의 어트리뷰트를 키/값 개체로 설정한다.
attr( key, value )	모든 매치되는 요소들의 단일 속성의 값을 지정한다.
attr( key, fn )	모든 매치되는 요소들의 단일 속성에 대해 계산된 값을 지정한다.
removeAttr( name )	매치된 요소 각각으로부터 해당 속성을 제거한다.

## 4-4. 속성(Attribute)관련 메소드들

### Attr 관련 메소드 (예제 4-4-1)

```
<script type="text/javascript">
$(function(){
    $("#photos img")
        .attr({ border: "1px", height : "100" })
        .attr("title", function(i) {
            return (i + 1) + "번째 이미지입니다";
        });
});
</script>
<body>
<div id="photos">
    
    
    
    
    
</div>
</body>
```

## 4-5. 이벤트(Event)관련 메소드들

### 1) 이벤트(Event) 관련 메소드들

메소드	설 명
<code>bind( type, data, fn )</code>	매치된 요소에 이벤트 처리기를 바인딩한다. type에는 이벤트 명칭을, data에는 부가적으로 전달할 데이터 개체(JSON)를, fn 에는 이벤트 처리기 함수를 작성한다.
<code>unbind( type, fn )</code>	매치된 요소에서 지정된 이벤트와 매핑된 모든 처리기들을 제거한다. bind()와는 상반되는 메서드이다.
<code>one( type, data, fn )</code>	사용 방법은 bind() 함수와 동일하지만 한번만 실행되고 자동으로 이벤트 처리기를 삭제한다.
<code>trigger( type, data )</code>	매치된 요소에 대하여 전달된 이벤트 타입에 해당하는 이벤트 처리기를 모두 실행한다.

## 4-5. 이벤트(Event)관련 메소드들

### 1) 이벤트(Event) 관련 메소드들(계속)

메소드	설 명
<code>toggle( fn1, fn2 )</code>	매치된 요소에서 클릭 이벤트가 발생할 때 마다 처리될 처리기 지정한다. (홀수번째: fn1, 짝수번째: fn2)
<code>hover( fn1, fn2 )</code>	매치된 요소에서 <code>mouseover(fn1)</code> 와 <code>mouseout(fn2)</code> 이벤트가 발생할 때 처리될 처리기 지정한다.
<code>stopPropagation()</code>	DOM 트리의 이벤트 버블링을 막음.
<code>preventDefault()</code>	이벤트의 기본 동작을 막음. 이벤트 버블링과 기본 동작을 동시에 막고 싶다면 리스너 함수에 <code>false</code> 반환.

## 4-5. 이벤트(Event)관련 메소드들

### ★ 이벤트(Event) 처리함수에서 발생하는 event객체의 속성들

속 성	설 명
altKey	이벤트 발생시 Alt 키가 true, 그렇지 않으면 false
ctrlKey	이벤트 발생시 Ctrl 키가 눌리면 true, 그렇지 않으면 false
data	핸들러를 할당할 때, bind() 커맨드의 두 번째 매개변수로 전달된 값
keyCode	keyup이나 keydown 이벤트인 경우, 눌린 키를 반환. (무조건 대문자, Which 프로퍼티 권장)
metaKey	이벤트 발생시 메타 키가 눌리면 true가 설정. (일반 PC는 Ctrl키를 말함)
pageX(clientX)	페이지에서 이벤트가 일어난 X 좌표
pageY(clientY)	페이지에서 이벤트가 일어난 Y 좌표

## 4-5. 이벤트(Event)관련 메소드들

### ★ 이벤트(Event) 처리함수에서 발생하는 event객체의 속성들 (계속)

속 성	설 명
relatedTarget	몇몇 마우스 이벤트의 경우, 이벤트가 발생했을 때, 커서가 들어가거나 나온 엘리먼트를 가리킨다.
screenX	스크린에서 이벤트가 일어난 X 좌표 (마우스 이벤트)
screenY	스크린에서 이벤트가 일어난 Y 좌표 (마우스 이벤트)
shiftKey	이벤트 발생시 shift 키가 눌리면 true
target	이벤트가 발생한 엘리먼트 반환
type	발생한 이벤트 타입을 반환. (예, "click") 다수의 이벤트를 한 이벤트 핸들러로 처리하는 경우에 유용하다.
which	키보드 이벤트에 대한 ASCII코드 반환, 마우스 이벤트 (왼쪽 : 1, 중간 : 2, 오른쪽 : 3) 반환

## 4-5. 이벤트(Event)관련 메소드들

### 이벤트(Event) 관련 메소들 (예제4-5-1)

```
<script type="text/javascript">
$(function(){
    $("#myBtn").bind("click", { name: "홍길동", gender: "남자" },
        function(e) {
            alert("제 이름은 " + e.data.name + "이 고요, " + e.data.gender+ "입니다.");
        });

    $("#myMT").one("click", function() {
        alert("이미지는 클릭하지 마세요. 두 번 말하지 않습니다");
    });

    $("#testTrgg").bind("myTriggerOne", function(){
        $(this).attr("src","./images/셔틀콕.jpg");
    });

    $("#testTrgg").bind("myTriggerTwo", function(){
        $(this).attr("src","./images/풍차.jpg");
    });
});
```



## 4-5. 이벤트(Event)관련 메소드들

### 이벤트(Event) 관련 메소들 (예제4-5-1계속)

```
$("#trggBtn").toggle(
    function(){ $("#testTrgg").trigger("myTriggerOne"); },
    function(){ $("#testTrgg").trigger("myTriggerTwo"); }
);
$("#testHover").hover(
    function(){ $(this).css("border", "2px solid blue"); },
    function(){ $(this).css("border", "0px"); }
);
});
</script>

<body>
    <div>
        <span id="testHover"></span>
        
    </div>
    <input type="button" id="myBtn" value="내 소개!!">
    <input type="button" id="trggBtn" value="Trigger Test">
</body>
```

## 4-5. 이벤트(Event)관련 메소드들

### 2) 이벤트(Event)명과 동일한 이벤트 설정 메소드들

메소드	설 명
이벤트명( fn )	해당 이벤트가 발생했을 때 처리할 처리기를 설정한다. fn에는 이벤트 처리기 함수를 작성한다.

- 이벤트명과 같은 메소드명들  
(blur, focus, change, click, dblclick, mousedown, mouseup, mousemove, mouseover, keydown, keyup, keypress, error, load, unload, resize, scroll, select, submit)

## 4-5. 이벤트(Event)관련 메소드들

### 이벤트(Event) 관련 메소들 (예제4-5-2)

```
<script type="text/javascript">
$(function(){
    $("#images img").attr("src1", function() { return this.src; })
        .mouseout(function() { this.src = this.src1; })
        .mouseover(function() { this.src = this.src2; })
        .dblclick(function() { $(this).hide(); });
</script>

<body>
<div id="images">
    
    
    
</div>
</body>
```



## 5. jQuery Utilities

---

## 5. jQuery Utility

메소드	설 명
<code>\$.browser</code>	Browser의 version과 종류를 Map 형식으로 반환한다. <ul style="list-style-type: none"><li>- version : browser의 버전</li><li>- safari : safari 브라우저면 true, 그렇지 않으면 false</li><li>- opera : opera 브라우저면 true, 그렇지 않으면 false</li><li>- msie : MS의 IE이면 true, 그렇지 않으면 false</li><li>- mozilla : 파이어폭스, 네스케이프 이면 true, 그렇지 않으면 false</li></ul>
<code>\$.trim( str )</code>	문자열 str의 양쪽 공백을 제거한다.
<code>\$.each( obj , fn )</code>	Array 또는 Map 형태의 첫 번째 값( obj )의 각 요소 값 마다 fn 함수를 호출한다. <ul style="list-style-type: none"><li>• fn함수의 형식<ul style="list-style-type: none"><li>- fn(idx) : idx는 obj의 index값</li><li>- fn(idx, val) : idx는 index값, val는 해당 위치의 요소값</li></ul></li></ul>

## 5. jQuery Utility

메소드	설 명
<code>\$.grep(array, fn, invert)</code>	<p>전달된 배열의 각 요소 값 마다 fn함수를 호출하고 fn함수의 반환 값들을 새로운 배열로 수집될지를 결정한다.</p> <ul style="list-style-type: none"><li>• invert가 생략되거나 false인 경우 ☞ fn의 반환 값이 true인 경우에만 수집한다.</li><li>• invert가 true인 경우 ☞ fn의 반환 값이 false인 경우에만 수집한다.</li></ul>
<code>\$.map( array, fn )</code>	<p>전달된 배열의 각 요소 값 마다 fn함수를 호출하고 fn함수의 반환 값들은 새로운 배열로 수집된다.</p>
<code>\$.unique( array )</code>	<p>전달된 배열에서 중복되는 값을 제거한다.</p>
<code>\$.isFunction( obj )</code>	<p>전달된 obj가 function인지 여부를 true, false로 반환한다.</p>

## 5. jQuery Utility

메소드	설 명
<code>\$.inArray( value, array )</code>	array에서 value가 몇 번째 index번째에 존재 하는지 반환한다. 값이 존재하지 않으면 -1을 반환한다. ( index는 0부터 시작 )
<code>\$.makeArray( obj )</code>	Array 형태로 변형 가능한 obj를 Array 객체로 변형한다.
<code>\$.extend( target, obj1, obj2, ... )</code>	첫 번째 target에 두 번째 이후(obj1, obj2, ...)의 값을 추가한다. 만약 추가하는 값에 중복되는 값이 있을 경우 나중 값이 추가된다.
<code>\$.getScript( url , fn )</code>	url에 설정된 주소에서 스크립트 파일을 읽어온 후 fn함수를 실행한다. (사파리는 사용 불가)



## 5. jQuery Utility

### jQuery Utility (예제 5-1)

```
<script type="text/javascript">
$(function(){
    var result = "";
    jQuery.each(jQuery.browser, function(i, val) {    //$$.browser의 모든 속성 출력
        result += i + ":" + val + "Wn";
    });
    alert(result);
    if ($.browser.msie) {    //$$.browser 확인
        alert("IE 웹 브라우저");
    } else {    alert('다른 웹 브라우저'); }
    var s = " Abc Def Fed Cba ";
    alert "[" + s + "]";
    alert(s.length);
    alert "[" + $.trim(s) + "]";
    alert(jQuery.trim(s).length);
}</script>
```

## 5. jQuery Utility

### jQuery Utility (예제 5-2)

```
<script type="text/javascript">
$(function(){
    var empty={ }
    var defaults={ validate: false, limit: 5, name: "foo" };
    var options1=[ "one","two","three","four","five" ];
    var options2=[ "six","seven","eight","nine","ten" ];
    var settings=$.extend(empty, defaults, options1,options2);
    var result="";
    jQuery.each(settings, function(i, val) {
        result += "[name:"+i+"] " + "[value:"+val+"]\n";
    });
    alert(result);
</script>
```

## 5. jQuery Utility

### jQuery Utility (예제 5-3)

```
<script type="text/javascript">
$(document).ready(function() {
    var arr1=[ 0,1,2,3,4,5,6,7,8,9 ];
    var arr2=jQuery.grep(arr1, function(n, i) {
        return (n != 5 && i>3);
    });
    alert("변경 전: " + arr1.join(", ") + "\n" + "변경 후: " + arr2.join(", "));

    var arr3=jQuery.grep(arr1, function(n, i) {
        return (n != 5 && i>3);
    },true);
    alert("변경 전: " + arr1.join(", ") + "\n" + "변경 후: " + arr3.join(", "));
});
</script>
```

## 5. jQuery Utility

### jQuery Utility (예제 5-4)

```
<script type="text/javascript">
$(document).ready(function() {
    var arr=jQuery.makeArray($("#div"));
    var result="";
    jQuery.each(arr,function(i){
        result+="index[" + i + "] element[" + $(this).text() + "]\n";
    });
    alert(result);

    result="";
    var arr=[ "성 춘향", 40, "일 지매", 55, "홍길 동", 40 ];
    result ="홍길 동 위치: " + jQuery.inArray("홍길 동", arr)+"\n";
    result += "40 위치: " + jQuery.inArray(40, arr)
    alert(result);
});
</script>
```

## 5. jQuery Utility

### jQuery Utility (예제 5-4 계속)

```
<body>  
  <div>계룡산</div>  
  <div>백두산</div>  
  <div>한라산</div>  
  <div>지리산</div>  
  <p>태백산</p>  
  <p><div>소백산</div></p>  
</body>
```

## 5. jQuery Utility

### jQuery Utility (예제 5-5)

```
<script type="text/javascript">
var testNull=null;
var testVal="값이 있는 변수";
function testFunc() {
    alert("함수입니다.");
}

$(document).ready(function() {
    var result="";
    var arr=jQuery.unique(jQuery.makeArray($("#div")).concat($("#div").get()));
    jQuery.each(jQuery.unique(arr.concat(arr)),function(i){
        result += "[index: " + i + "] [value: " + $(this).text() + "]\n";
    });
    alert(result);
});
```

## 5. jQuery Utility

### jQuery Utility (예제 5-5 계속)

```
var objs=[ testNull,testVal,testFunc ];
result="";
jQuery.each(objs, function (i) {
    result += "[value: " + objs[i] + "]" + "[" + $.isFunction(objs[i]) + "]" + "\n";
});
alert(result);
});
</script>

<body>
    <div>하나</div>
    <div>둘</div>
    <div>셋</div>
    <div>넷</div>
</body>
```

## 5. jQuery Utility

### jQuery Utility (예제 5-6)

```
<script type="text/javascript">
$(document).ready(function() {
    var arr1=[ "a", "b", "c", "d", "e" ];
    var arr2=jQuery.map(arr1, function(n, i) {
        return (n.toUpperCase());
    });
    alert("변경 전 : " + arr1.join(", ") + "Wn" +      "변경 후 : " + arr2.join(", "));
});
</script>
```



## 6. jQuery Effects

---

## 6. jQuery Effects

### 1) jQuery의 기본 효과

메소드	설 명
<code>hide()</code> <code>hide( duration, fn )</code>	해당 요소를 숨겨준다. duration은 숨기는데 걸리는 소요시간을 설정하고, fn은 효과 실행 후 처리 할 콜백 함수를 구성한다. 소요시간은 <code>slow(600)</code> , <code>normal(400)</code> , <code>fast(200)</code> 와 숫자로 설정할 수 있다.
<code>show ()</code> <code>show (duration, fn )</code>	해당 요소를 보여준다. duration은 보이는데 걸리는 소요시간을 설정하고, fn은 효과 실행 후 처리 할 콜백 함수를 구성한다.
<code>toggle ()</code> <code>toggle ( fn1, fn2 )</code>	해당 요소가 현재 보이는 상태이면 보이지 않게, 보이지 않는 상태이면 보이도록 변경한다. fn1은 홀수 번째, fn2는 짝수 번째 클릭 시 실행 할 콜백 함수를 구성한다.

## 6. jQuery Effects

### jQuery Effects (예제 6-1)

```
<script type="text/javascript">
$(document).ready(function() {
    $("div").css( {width:130,height:130,border:"1px solid black"      }).attr("align",
    "center");

    $("input#imgShow1").click(function(e){
        $("img").show();
    });
    $("input#imgShow2").click(function(e){
        $("img").show(3000);
    });
    $("input#imgShow3").click(function(e){
        $("img").show(1000,function(){
            alert("보이기 작업이 모두 끝났습니다");
        });
    });
});
```

## 6. jQuery Effects

### jQuery Effects (예제 6-1 계속)

```
$( "input#imgHide1" ).click(function(e){  
    $( "img" ).hide();  
});  
$( "input#imgHide2" ).click(function(e){  
    $( "img" ).hide(3000);  
});  
$( "input#imgHide3" ).click(function(e){  
    $( "img" ).hide(1000, function(){  
        alert("감추기 작업이 모두 끝났습니다");  
    });  
});  
$( "input#imgToggle" ).click(function(e){  
    $( "img" ).toggle();  
});  
});  
</script>
```

## 6. jQuery Effects

### jQuery Effects (예제 6-1 계속)

```
<body>
  <input type="button" id="imgShow1" value="이미지 보이기"/>
  <input type="button" id="imgShow2" value="천천히 보이기"/>
  <input type="button" id="imgShow3" value="보이기 후 처리"/> <br>

  <input type="button" id="imgHide1" value="이미지 감추기"/>
  <input type="button" id="imgHide2" value="천천히 감추기"/>
  <input type="button" id="imgHide3" value="감추기 후 처리"/> <br>

  <input type="button" id="imgToggle" value="보이기 <-> 감추기"/>

  <div>  </div>

</body>
```

## 6. jQuery Effects

### 2) jQuery의 Fade 효과

메소드	설 명
<code>fadeIn ()</code> <code>fadeIn ( duration, fn )</code>	해당 요소의 불투명도를 점차 높여서 보여준다. duration은 보이는데 걸리는 소요시간을 설정하고, fn은 효과 실행 후 처리 할 콜백 함수를 구성한다. 소요시간은 <code>slow(600)</code> , <code>normal(400)</code> , <code>fast(200)</code> 와 숫자로 설정할 수 있다.
<code>fadeOut ()</code> <code>fadeOut (duration, fn )</code>	해당 요소의 불투명도를 점차 낮춰서 보이지 않게 한다. duration은 숨기는데 걸리는 소요시간을 설정하고, fn은 효과 실행 후 처리 할 콜백 함수를 구성한다.
<code>fadeTo( duration, opacity)</code> <code>fadeTo( duration, opacity, fn )</code>	해당 요소의 불투명도를 정해진 시간 동안 변화 시켜준다. 불투명도(opacity)는 0 ~ 1사이의 값을 사용한다. fn은 효과 실행 후 처리 할 콜백 함수를 구성한다.

## 6. jQuery Effects

### jQuery Effects (예제 6-2)

```
<script type="text/javascript">
$(document).ready(function() {
    $("div").css( {width:130,height:130,border:"1px solid black" }).attr("align", "center");
    $("input#imgFadeIn").click(function(e){
        $("img").fadeIn(2000);
    });
    $("input#imgFadeOut").click(function(e){
        $("img").fadeOut(2000);
    });
    $("img").mouseover(function(e){
        $(this).fadeTo(2000, 0.2);
    });
    $("img").mouseout(function(e){
        $(this).fadeTo(2000, 1);
    });
});
</script>
```

## 6. jQuery Effects

### jQuery Effects (예제 6-2 계속)

```
<body>
```

```
  <input type="button" id="imgFadeIn" value="이미지 FadeIn"/>
```

```
  <input type="button" id="imgFadeOut" value="이미지 FadeOut"/>
```

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

```
</body>
```



## 6. jQuery Effects

### 3) jQuery의 Sliding 효과

메소드	설 명
<code>slideDown ()</code> <code>slideDown ( duration, fn )</code>	해당 요소를 위에서부터 아래 방향으로 보여준다. duration은 보이는데 걸리는 소요시간을 설정하고, fn은 효과 실행 후 처리 할 콜백 함수를 구성한다. 소요시간은 <code>slow(600)</code> , <code>normal(400)</code> , <code>fast(200)</code> 와 숫자로 설정할 수 있다.
<code>slideUp ()</code> <code>slideUp (duration, fn )</code>	해당 요소를 아래에서부터 위쪽 방향으로 숨겨준다. duration은 숨기는데 걸리는 소요시간을 설정하고, fn은 효과 실행 후 처리 할 콜백 함수를 구성한다.
<code>slideToggle ()</code> <code>slideToggle ( duration, fn )</code>	해당 요소가 보이는 상태이면 <code>slideUp</code> , 보이지 않는 상태이면 <code>slideDown</code> 를 실행시켜 준다. fn은 효과 실행 후 처리 할 콜백 함수를 구성한다.

## 6. jQuery Effects

### jQuery Effects (예제 6-3)

```
<script type="text/javascript">
$(document).ready(function() {
    $("div").css( {width:130,height:130,border:"1px solid black" });

    $("input#imgSlideDown1").click(function(e){
        $("div").attr("align", "center");
        $("img").slideDown(2000);
    });

    $("input#imgSlideDown2").click(function(e){
        $("div").attr("align", "right");
        $("img").slideDown(2000);
    });

    $("input#imgSlideDown3").click(function(e){
        $("div").attr("align", "left");
        $("img").slideDown(2000);
    });
});
```

## 6. jQuery Effects

### jQuery Effects (예제 6-3 계속)

```
$("#input#imgSlideUp1").click(function(e){  
    $("#div").attr("align", "center");  
    $("#img").slideUp(2000);  
});
```

```
$("#input#imgSlideUp2").click(function(e){  
    $("#div").attr("align", "right");  
    $("#img").slideUp(2000);  
});
```

```
$("#input#imgSlideUp3").click(function(e){  
    $("#div").attr("align", "left");  
    $("#img").slideUp(2000);  
});
```

## 6. jQuery Effects

### jQuery Effects (예제 6-3 계속)

```
$( "input#imgSlideToggle" ).click(function(e){
    $( "div" ).attr( "align", "center" );
    $( "img" ).slideToggle( 2000 );
});
});
</script>
<body>
<input type="button" id="imgSlideDown1" value="이미지 SlideDown(가운데 정렬)"/>
<input type="button" id="imgSlideDown2" value="이미지 SlideDown(오른쪽 정렬)"/>
<input type="button" id="imgSlideDown3" value="이미지 SlideDown(왼 쪽 정렬)"/><br>
<input type="button" id="imgSlideUp1" value="이미지 SlideUp(가운데 정렬)"/>
<input type="button" id="imgSlideUp2" value="이미지 SlideUp(오른쪽 정렬)"/>
<input type="button" id="imgSlideUp3" value="이미지 SlideUp(왼 쪽 정렬)"/><br>
<input type="button" id="imgSlideToggle" value="이미지 SlideToggle(가운데 정렬)"/>
<div></div>
</body>
```

## 6. jQuery Effects

### 4) jQuery의 사용자 정의(Custom) 효과

메소드	설 명
<code>animate ( props )</code> <code>animate ( props, duration, easing, fn )</code>	해당 요소에 props에 설정한 style속성값으로 지정된 시간 동안 변경시켜준다. easing은 외부 plug-in을 통해 효과를 적용시킬 수 있다. fn은 효과 실행 후 처리할 콜백 함수를 구성한다.
<code>animate ( props , options )</code>	options { duration:시간, easing:외부plug-in, complete: fn(효과 후 처리할 함수), step:fn(애니메이션 각 단계별로 처리할 함수), queue:false(애니메이션이 동시에 실행된다.) }

## 6. jQuery Effects

### jQuery Effects (예제 6-4)

```
<script type="text/javascript">
$(document).ready(function() {
    $("div").css( {width:330,height:330,border:"1px solid black"    });
    $("input#imgAni1").click(function(e){
        $("img").animate( {width: "100%", opacity:0.4}, 2000    );
    });
    $("input#imgAni2").click(function(e){
        $("img").animate( { marginLeft: 200 },
                           {duration:2000, easing:"easeOutBounce"}    );
    });
    $("input#imgAni3").click(function(e){
        $("img").animate( { marginLeft: 100 }, 1000    );
        // $("img").animate( { marginTop: 100 }, 1000    );
        $("img").animate( { marginTop: 100 }, {duration:1000, queue:false}    );
    });
});
</script>
```

## 6. jQuery Effects

### jQuery Effects (예제 6-4 계속)

```
<body>
  <input type="button" id="imgAni1" value="이미지 Animate1"/>
  <input type="button" id="imgAni2" value="이미지 Animate2"/>
  <input type="button" id="imgAni3" value="이미지 Animate3"/>

  <div></div>

</body>
```

## 6. jQuery Effects

### 4) jQuery의 사용자 정의(Custom) 효과 (계속)

메소드	설 명
<code>delay( duration, queueName )</code>	해당 요소의 효과를 지정한 시간 동안 지연 시킨다. (기본 queueName : fx )
<code>stop ()</code>	해당 요소에 적용된 animate를 중지 시킨다.
<code>queue ()</code> <code>queue ( fn )</code> <code>queue (queue, newQueue)</code>	Queue에 저장되어 있는 함수 배열을 반환한다. Queue에 animate를 추가한다. Queue의 내용을 newQueue로 변경한다.
<code>dequeue ()</code>	dequeue앞에 실행된 queue를 제거하고 실행한다. queue에 쌓인 것을 실행하는 중간에 toggleClass와 같이 animate가 아닌 것이 포함되서 실행하면 queue가 중단된다. 이때 다시 queue에 쌓여 있는 나머지를 실행하기 위해 .dequeue를 호출한다
<code>jQuery.fx.off</code>	효과를 키거나 끈다. ( true , false )



## 7. jQuery Ajax

---

## 7. jQuery Ajax

메소드	설 명
jQuery.ajax( options )	Ajax 요청 관련 정보와, 응답 처리를 담당하는 콜백 함수등을 설정하고 Ajax요청을 전송한다.

### Options에 설정될 속성

속성명	설 명
url	요청 URL
type	사용할 HTTP 메서드, 일반적으로 POST나 GET 사용
data	요청에 전달되는 프로퍼티를 가진 객체
timeout	Ajax 요청의 제한 시간을 밀리초 단위로 설정한다. 제한 시간 안에 요청이 완료되지 않으면 요청을 취소하거나, error 콜백이 정의되어 있다면 호출된다.

## 7. jQuery Ajax

### Options에 설정될 속성 (계속)

속성명	설 명
global	true나 false에 따라 전역 함수를 활성화하거나 비활성화한다. 전역 함수는 엘리먼트에 덧붙일 수 있으며 Ajax 호출 동안 다양한 위치나 조건에서 실행된다.
contentType	요청에 명시되는 콘텐츠 타입. 기본값은 'application/x-www-form-urlencoded'로 설정된다.
success	응답이 성공 상태 코드를 반환하면 호출되는 함수. 응답 본문은 이 함수의 첫 번째 매개변수로 전달되며, dataType 프로퍼티에 명시한 형태로 구성된다. 두 번째 매개변수는 상태 값을 나타내는 문자열이며 이번 경우에는 항상 'success'다
error	응답이 에러 상태 코드를 반환하면 호출되는 함수. 이 함수의 매개변수는 각각 XHR 인스턴스, 상태 값이 항상 'error'인 메시지 문자열, 선택사항으로 XHR 인스턴스가 반환한 예외 객체이다.

## 7. jQuery Ajax

### Options에 설정될 속성 (계속)

속성명	설 명
complete	요청이 완료되면 호출되는 함수. 전달되는 매개변수는 각각 XHR 인스턴스와 'success' 혹은 'error'를 나타내는 상태 메시지 문자열이다. success나 error 콜백을 설정하면 해당 콜백이 호출된 후에 실행된다.
beforeSend	요청이 전송되기 전에 먼저 호출되는 함수. (반환값이 false면 전송취소된다.) 이 함수는 XHR 인스턴스를 전달 받으며, 사용자 정의 헤더를 설정하거나 요청 전에 필요한 연산을 수행하는데 사용할 수 있다.
async	false이면 요청이 동기 호출로 전송된다. 기본값은 true(비 동기 요청)이다.
processData	false이면, URL 인코딩된 형태로 처리되어 전달된 데이터를 금지한다. 기본 값은 데이터가 'application/x-www-form-urlencoded'타입의 요청에 사용하는 형태의 URL로 인코딩 된다.

## 7. jQuery Ajax

### Options에 설정될 속성 (계속)

속성명	설 명
ifModified	<p>현재의 응답이 이전의 응답과 다를 경우에만 처리하도록 한다. true일 때 Last-Modified 헤더를 확인하여 마지막 요청 이후에 응답 콘텐츠가 변경되지 않았다면 요청이 성공한다. 만일 생략하면 헤더를 확인하지 않는다.</p>
dataType	<p>응답의 결과로 반환되는 데이터의 종류를 식별하는 키워드.</p> <ul style="list-style-type: none"><li>• xml - 응답 결과는 XML문서로 파싱되어 XML DOM형식으로 처리함.</li><li>• html - 응답 결과는 HTML형식으로 처리한다. HTML 코드에 있는 모든 &lt;script&gt;블록이 평가된다.</li><li>• json - 응답 결과는 JSON 객체 형식으로 처리한다.</li><li>• jsonp - 요청 DATA에 콜백함수를 추가해서 보내고, 응답은 JSON객체형식으로 받는다.</li><li>• script - 응답 결과를 javascript로 인식한다. 응답은 모든 콜백 호출보다 먼저 javascript구문으로 처리됨.</li><li>• text - 응답 결과는 일반 텍스트로 처리한다.</li></ul>

## 7. jQuery Ajax

메소드	설 명
ajaxSetup (options )	한 페이지에서 동일하게 반복되는 \$.ajax() 메서드에 동일한 설정을 유지하기 위한 global 설정을 한다. \$.ajax()를 여러 번 사용시 반복되는 인자는 생략 가능하고 \$.ajax()에서 사용하는 인자를 그대로 사용할 수 있다.
\$.get ( url, data, fn, dataType )	설정한 URL로 GET방식으로 Ajax 요청을 전송한다. fn은 응답이 성공일 경우에만 호출된다. fn의 형식 : fn(응답결과, 성공여부 문자열, XMLHttpRequest)
\$.post ( url, data, fn, dataType )	설정한 URL로 POST방식으로 Ajax 요청을 전송한다. 기타 나머지는 \$.get()와 같다.
\$.getJSON( url, data, fn )	설정한 URL로 GET방식으로 요청을 보내고 JSON 형식으로 응답을 받는다. 기타 나머지는 \$.get()와 같다.

## 7. jQuery Ajax

메소드	설 명
<code>\$.getScript( url, fn )</code>	설정한 URL에서 javascript파일을 load한 후 사용 가능한 형태로 변경한다. fn은 응답이 성공일 경우에만 호출된다. fn의형식 : fn(응답결과, 성공여부 문자열)
<code>load( url, data, fn )</code>	설정한 URL로 Ajax 요청을 전송한다. url : 요청이 전송될 서버의 URL주소 data : 서버로 전송할 자료 (data가 있으면 POST, 없으면 GET 방식으로 전송) fn 함수는 요청이 완료되면 호출된다. fn의형식 : fn(응답결과, 성공여부 문자열, XHR객체)
<code>ajaxStart (fn )</code>	요청작업이 시작할 때 fn함수를 호출한다. 콜백함수의 매개변수는 없다.
<code>ajaxSend (fn )</code>	요청이 서버로 보내지기 전에 fn함수를 호출한다. fn의 형식 : fn(event객체, XHR객체, 옵션들)

## 7. jQuery Ajax

메소드	설 명
ajaxSuccess (fn )	요청에 대한 응답이 성공일 경우 fn함수를 호출한다. fn의 형식 : fn(event객체, XMLHttpRequest, 옵션들)
ajaxError (fn )	요청에 대한 응답이 실패일 경우 fn함수를 호출한다. fn형식 : fn(event객체, XMLHttpRequest, 옵션들, errorThrown)
ajaxComplete (fn )	요청이 끝났을 경우 항상 fn함수를 호출한다. fn의 형식 : fn(event객체, XMLHttpRequest, 옵션들)
ajaxStop (fn )	ajax작업 완료 후 fn함수를 호출한다. 콜백함수의 매개변수는 없다.
serialize ( )	form엘리먼트에 속한 데이터 속성을 갖는 엘리먼트들의 값의 형식을 &name=value 형식으로 변환한다.
serializeArray ( )	form엘리먼트에 속한 데이터 속성을 갖는 엘리먼트들의 값의 형식을 Array 형식인 [ { name:변수명, value:값 }, ... ]로 변환한다.