

삼성 청년 SW 아카데미

jQuery

Confidential

<알림>

본 강의는 삼성 청년 SW아카데미의 컨텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

jQuery

1. jQuery 소개

1. jQuery 소개.

1-1. jQuery 소개

- ✓ jQuery는 John Resig이 2006년 발표한 크로스 플랫폼을 지원하는 경량 javascript library.
- ✓ 복잡하고 방대해진 UI 코드는 관리를 어렵게 한다.
- ✓ 이러한 문제는 웹 페이지의 구조(HTML)와 행위(JS) 그리고 표현(CSS)을 분리하면 어느 정도 해결할 수 있다.
- ✓ HTML 문서의 탐색이나 조작, 이벤트 핸들링, 애니메이션, Ajax 등을 멀티 브라우저를 지원하는 API를 통해 간편하게 사용.
- ✓ 2019년 기준 상위 100만개의 웹페이지 중 84%가 jQuery 사용.

Confidential

Top In JavaScript Library Usage Distribution in the Top 1 Million Sites

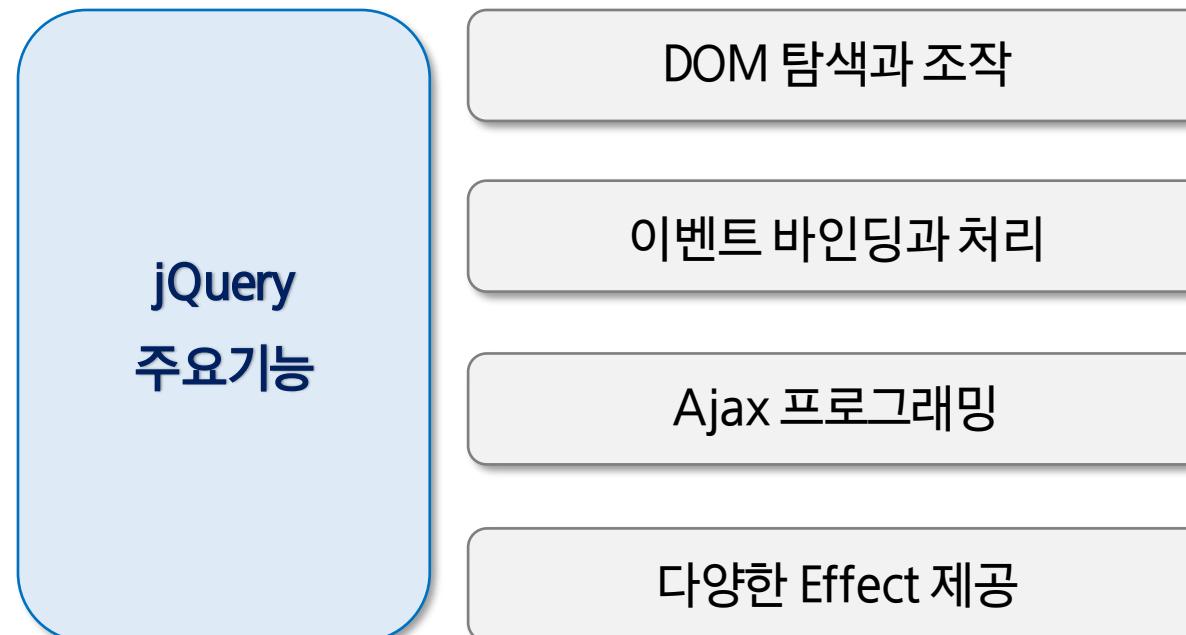
Technology	Websites	%
jQuery	842,979	84.3
Bootstrap.js	140,215	14.02
Moment JS	133,901	13.39
React	123,109	12.31
lodash	93,838	9.38
Vue	46,380	4.64
Yahoo User Interface	40,808	4.08
Backbone.js	26,005	2.6
Prototype	21,847	2.18
script.aculo.us	17,354	1.74
MooTools	17,037	1.7
Socket.IO	13,663	1.37
Enquire JS	12,261	1.23
KnockoutJS	11,963	1.2
Polyfill IO	10,676	1.07
Preact JS	9,576	0.96
jsTimezoneDetect	9,522	0.95
mOxie	3,092	0.31

<https://trends.builtwith.com/javascript/javascript-library>

1-2. jQuery 특징.

Confidential

- ✓ **크로스 플랫폼을 지원하는** jQuery는 어떠한 브라우저에서도 동일하게 동작.
- ✓ **브라우저 호환성을 고려하여 대체코드를 작성할 필요가 없다.**
- ✓ **네이티브 DOM API(DOM Query, Traversing, Manipulation 등)보다 직관적이고 편리한 API 제공하여 개발 속도를 향상.**
- ✓ **Event 처리, Ajax, Animation 효과를 쉽게 사용.**
- ✓ **다양한 Effect 함수를 제공하여 동적인 페이지를 쉽게 구현.**



1. jQuery 소개.

1-3. jQuery 설치(1/2).

Confidential

- ✓ jQuery는 1.x과 2.x 두가지 버전 제공. 동일한 API를 제공하지만 2.x는 IE8 이하를 지원 X, HTML5 추가 지원.
- ✓ 2016년 6월 jQuery 3.0이 release되면서 두버전을 3.x으로 통일.

The screenshot shows the official jQuery website at <https://jquery.com/>. The main navigation bar includes 'Download', 'API Documentation', 'Blog', 'Plugins', and 'Browser Support'. Below the navigation, there are three key features highlighted: 'Lightweight Footprint' (only 30kB minified and zipped), 'CSS3 Compliant' (supports CSS3 selectors), and 'Cross-Browser' (works across Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more). A prominent orange button labeled 'Download jQuery v3.5.1' is circled in red, indicating the latest version. To the right of the download button, a note states: 'The 1.x and 2.x branches no longer receive patches.' The page also features sections for 'What is jQuery?' (explaining it's a fast, feature-rich JavaScript library), 'Other Related Projects' (listing jQuery UI, QUnit, and Sizzle), and 'A Brief Look' (with a code example: `$("button.continue").html("Next Step...")`).

<https://jquery.com/>

1-3. jQuery 설치(2/2).

Confidential

- ✓ jQuery는 공식 사이트에서 다운로드 받아 사용하거나, CDN을 사용.
 - Download : <http://jquery.com/download/>
 - jquery-3.XX.X.min.js 파일 다운로드 시 웹 특정 경로에 저장.
 - <script type="text/javascript" src="../js/jquery-3.xx.x.min.js"></script>
 - CDN(Content Delivery Network)
 - [Google CDN](#) : <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
 - [Microsoft CDN](#) : <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.5.1.min.js"></script>
 - [CDNJS CDN](#)
 - [jsDelivr CDN](#)
- ✓ jQuery API : <http://api.jquery.com>

1. jQuery 소개.

1-4. jQuery 기본구문.

Confidential

- ✓ 기본 구문은 Selector를 사용하여 DOM 객체를 탐색하고, 반환된 래퍼세트를 통해 함수를 수행한다.
- ✓ Selector 표현식과 Action 메소드를 조합한 형태로 구문을 작성한다. >> ex) `$(selector).action();`
- ✓ jQuery로 DOM을 탐색하기 전에, 웹 브라우저에 문서가 모두 로드되어 있어야 한다.
- ✓ jQuery는 Document Ready 이후 처리할 수 있는 두 가지 방법을 제공한다.

jQuery 기본 구문

```
$(selector).action();
```

jQuery 기본 구문 설명.

`$(selector)`는 탐색한 DOM 객체들을 담은 래퍼세트(WrapperSet)를 반환.
래퍼세트 객체를 통해 기능을 처리하는 액션 함수 호출.

Document Rady - 1

```
$(document).ready(function() {  
  
    // TODO...  
  
});
```

Document Rady - 2

```
$(function() {  
  
    // TODO...  
  
});
```

1. jQuery 소개.

1-5. jQuery 맛보기1.

Confidential

- ✓ jQuery를 사용하여 DOM 객체에 이벤트를 바인딩하고 DOM 객체를 조작.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jQuery 기본.</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("button").click(function() {
                $("#p1").css("background", "orange");
            });
        });
    </script>
</head>
<body>
    <h2>jQuery 맛보기</h2>
    <p>안녕하세요 싸피여러분!!!</p>
    <p id="p1">안효인교수입니다. 참여 많이 해주세요 ^^</p>
    <button>클릭</button>
</body>
</html>
```

jQuery 맛보기

안녕하세요 싸피여러분!!!

안효인교수입니다. 참여 많이 해주세요 ^^

클릭

클릭 시 id가 p1인 배경색 변경.

jQuery 맛보기

안녕하세요 싸피여러분!!!

안효인교수입니다. 참여 많이 해주세요 ^^

클릭

1. jQuery 소개.

1-5. jQuery 맛보기2.

Confidential

- ✓ jQuery를 사용하여 DOM 객체에 이벤트를 바인딩하고 DOM 객체를 조작.

```
<html>
<head>
    <meta charset="UTF-8">
    <title>jQuery 기본.</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("tr:eq(0)").css("background", "gray");
            $("td:nth-child(3n+1)").css("background", "orange");
            $("td:nth-child(3n+2)").css("background", "pink");
            $("td:nth-child(3n)").css("background", "cyan");
            $("td:contains('익수')").css("background", "green");
        });
    </script>
</head>
<body>
    <table>
        <tr><th>팀명</th><th>이름</th><th>포지션</th></tr>
        <tr><td>두산</td><td>박건우</td><td>우익수</td></tr>
        <tr><td>삼성</td><td>강민호</td><td>포수</td></tr>
        <tr><td>키움</td><td>박병호</td><td>1루수</td></tr>
        <tr><td>롯데</td><td>손아섭</td><td>좌익수</td></tr>
    </table>
</body>
</html>
```

팀명	이름	포지션
두산	박건우	우익수
삼성	강민호	포수
키움	박병호	1루수
롯데	손아섭	좌익수

1-02.html

1-6. 정리.

Confidential

- ✓ jQuery는 가벼운 자바스크립트 라이브러리로 DOM 조작, Ajax 지원 등을 쉽고 빠른 개발을 지원한다.
- ✓ jQuery 선택자는 DOM을 탐색하고, 다양한 메소드를 제공하는 래퍼세트(WrapperSet) 객체를 반환한다.
- ✓ 래퍼세트 객체는 메소드 체인을 제공하여, 메소드 호출에서의 반복적인 코딩을 줄여준다.
 - `$(selector).func1().func2().func3()…;`
- ✓ jQuery는 공식사이트에서 다운로드 하여 사용하거나, CDN(권장)에서 직접 사용하는 방법이 있다.

2. DOM 요소 선택

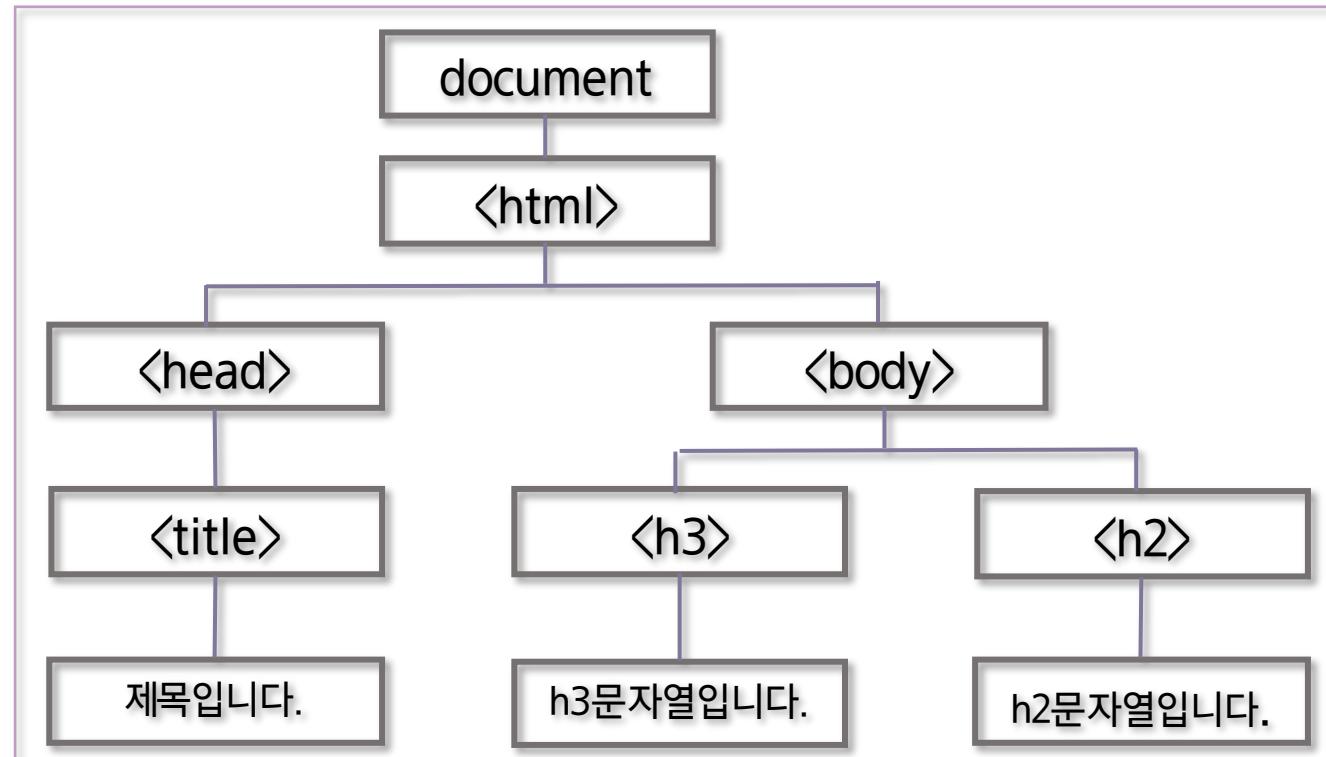
2. DOM 요소 선택.

2-1. DOM 탐색 (1/3) - DOM(Document Object Model).

Confidential

- ✓ DOM(Document Object Model)은 HTML과 XML문서의 구조를 정의하고 API를 제공.
- ✓ DOM은 문서 요소 집합을 트리 형태의 계층 구조로 HTML을 표현.
- ✓ HTML 계층 구조의 제일 위에는 document 노드가 있다.
- ✓ 그 아래로 HTML 태그나 요소(element)들을 표현하는 노드와 문자열을 표현하는 노드가 있다.

```
<!DOCTYPE html>
<html>
<head>
<title>제목입니다.</title>
</head>
<body>
    <h3>h3문자열입니다.</h3>
    <h2>h2문자열입니다.</h2>
</body>
</html>
```



2. DOM 요소 선택.

2-1. DOM 탐색 (2/3) - JavaScript에서 DOM 탐색.

Confidential

- ✓ DOM 객체는 JavaScript를 사용하여 DOM을 탐색하고 조작하는 API를 제공.
- ✓ 주로 document 객체를 사용하여 요소의 id, class, name 또는 태그 이름을 이용하여 DOM 요소를 탐색.
- ✓ 탐색한 DOM 요소 객체를 통해, 필요한 정보를 조회하거나 조작할 수 있다.
- ✓ 그러나, DOM 객체에서 제공하는 API를 사용하면, 단순한 작업임에도 적지 않은 코드를 작성해야 한다.

```
<span id="div1">id가 div1인 span 영역</span>
<div>div 영역</div>
<span name="div3">name이 div3인 span 영역</span>
<span class="div4">class가 div4인 span 영역</span>
<script type="text/javascript">
var element1 = document.getElementById("div1");
var element2 = document.getElementsByTagName("div")[0];
var element3 = document.getElementsByName("div3")[0];
var element4 = document.getElementsByClassName("div4")[0];

console.log(element1.innerHTML);
console.log(element2.innerHTML);
console.log(element3.innerHTML);
console.log(element4.innerHTML);
</script>
```

실행화면

id가 div1인 span 영역
div 영역
name이 div3인 span 영역
class가 div4인 span 영역

2. DOM 요소 선택.

2-1. DOM 탐색 (3/3) - jQuery에서 DOM 탐색.

Confidential

- ✓ jQuery는 DOM 탐색을 위하여 CSS에서 사용하는 Selector 표현 방식을 사용.
- ✓ 브라우저가 표준 CSS 선택자를 올바르게 구현하지 않았더라도, jQuery는 W3C 표준에 맞게 요소를 탐색.
- ✓ jQuery Selector는 CSS Selector와 jQuery에서 정의한 다양한 방법으로 DOM 요소를 탐색.
- ✓ jQuery는 DOM 탐색 결과로 래퍼세트(Wrapper Set)라는 DOM을 래핑 한 객체를 반환.

```
<span id="div1">id가 div1인 span 영역</span>
<div>div 영역</div>
<span name="div3">name이 div3인 span 영역</span>
<span class="div4">class가 div4인 span 영역</span>
<script type="text/javascript">
var element1 = $("#div1");
var element2 = $("div").eq(0);
var element3 = $("[name=div3]").eq(0);
var element4 = $(".div4").eq(0);

console.log(element1.html());
console.log(element2.html());
console.log(element3.html());
console.log(element4.html());
</script>
```

실행화면

id가 div1인 span 영역
div 영역
name이 div3인 span 영역
class가 div4인 span 영역

2. DOM 요소 선택.

2-2. 선택자 (Selector).

Confidential

- ✓ Selector를 이용한 DOM Element의 검색.
 - CSS 문법을 확장해서 태그를 찾는 selector 제공
 - 특정 브라우저에 의존적인 스크립팅을 벗어 날 수 있음
- ✓ jQuery function의 기본 형식

```
$      ("h1") . css("color", "blue");  
jQuery   selector           function
```

2. DOM 요소 선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

- ✓ jQuery 선택자(Selector)는 DOM 객체를 탐색하여 래퍼세트 객체로 반환.
- ✓ jQuery 선택자에는 요소, ID, 클래스, 다중, 복합 그리고 모두 선택자 등이 있다.

selector 종류	selector 표현방법
All selector	<code>\$("*")</code>
ID selector	<code>\$("#id")</code>
Element selector	<code>\$(“elementName”)</code>
Class selector	<code>\$(“.className”)</code>
Muliple selector	<code>\$(“selector1, selector2, elector3, …”)</code>

2. DOM 요소 선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ All Selector

- CSS의 가장 기본 선택자.
- HTML page에 있는 모든 문서객체를 선택.
- 주로 HTML Layout을 확인하기 위한 용도로 사용.
- Selector : *

```
$("*").css('color', 'blue');
```

2. DOM 요소선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ All Selector 예제.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $(function() {
                $("*").css("border", "1px solid blue");
            });
        });
    </script>
</head>
<body>
    <h2>제목이 들어갈 자리입니다.</h2>
    <p>간단 요약이 들어갈 자리입니다.</p>
    <p>또 다른 요약이 들어갈 자리입니다.</p>
    <div>
        <div id="content">상세 내용이 들어갈 자리입니다.</div>
    </div>
</body>
</html>
```

제목이 들어갈 자리입니다.

간단 요약이 들어갈 자리입니다.

또 다른 요약이 들어갈 자리입니다.

상세 내용이 들어갈 자리입니다.

2-03.html

2. DOM 요소 선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ Element Selector

- 특정 element를 선택.
- 요소 선택자(Element Selector)는 DOM 요소 명(태그 이름)이 일치하는 것들을 모두 찾는다.
- DOM API 중 document.getElementsByTagName()과 같다.
- Selector : **element name**

```
$('element').css('color', 'blue');
```

2. DOM 요소선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ Element Selector 예제.

```
<html>
  <head>
    <meta charset="UTF-8" />
    <title>jquery</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function () {
        $(function () {
          $('h1').css('color', 'orange');
        });
      });
    </script>
  </head>
  <body>
    <h1>JavaScript.</h1>
    <p>JavaScript 언어는 web에서 상당히 중요합니다.</p>
    <h1>jQuery.</h1>
    <p>jQuery는 JavaScript의 Library입니다.</p>
  </body>
</html>
```

2-04.html

JavaScript.

JavaScript 언어는 web에서 상당히 중요합니다.

jQuery.

jQuery는 JavaScript의 Library입니다.

2. DOM 요소 선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ ID Selector

- 특정 id 속성을 가지고 있는 요소를 선택..
- 웹 표준에서 id는 html page에서 단 하나의 태그에만 적용.(선택자중 사용빈도 높음).
- 만약 ID가 중복된 경우, 첫 번째 요소를 반환.
- ID 선택자는 다양한 선택자 중에서 DOM객체를 가장 빨리 탐색.
- DOM API 중 document.getElementById()와 같다.
- Selector : **#idname**

```
$('">#idname').css('color', 'blue');
```

or

```
$('tagname#idname').css('color', 'blue');
```

2. DOM 요소선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

- ✓ ID Selector 예제.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $(function() {
                $("h1#target").css("color", "orange");
            });
        });
    </script>
</head>
<body>
    <h1>JavaScript.</h1>
    <h1 id="target">jQuery.</h1>
    <h1>Servlet.</h1>
    <h1>JSP.</h1>
</body>
</html>
```

2-05.html

JavaScript.

jQuery.

Servlet.

JSP.

2. DOM 요소 선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ Class Selector, Complex(복합) Selector

- 특정 class 속성을 가지고 있는 요소를 선택.
- DOM API 중 document.getElementsByClassName()와 같다.
- Selector : `.classname`

```
$('.classname').css('color', 'blue');
```

or

```
$('#tagname.classname').css('color', 'blue');
```

- class 속성이 2개인 경우. AND 조건을 표현.
- 복합 선택자는 주로 Selector + Class Selector 조합으로 사용.

```
$('.cn1.cn2').css('color', 'blue');
```

2. DOM 요소선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

- ✓ Class Selector, Complex(복합) Selector 예제.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/Libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $(function() {
                $(".item").css("color", "orange");
                $(".item.select").css("color", "magenta");
            });
        });
    </script>
</head>
<body>
    <h1 class="item">JavaScript.</h1>
    <h1 class="item select">jQuery.</h1>
    <h1 class="item">Servlet.</h1>
    <h1 class="item">JSP.</h1>
</body>
</html>
```

2-06.html

JavaScript.

jQuery.

Servlet.

JSP.

2. DOM 요소 선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

✓ Multiple(다중) Selector

- 여러 element를 선택.
- 선택할 element가 여러 개일 경우 (콤마(),)로 구분.
- 나열된 선택자를 하나라도 만족하는 DOM 객체를 반환.
- Selector : **element_name, element_name, ...**

```
$('en1, en2, en3').css('color', 'blue');
```

2. DOM 요소선택.

2-3. DOM 요소 탐색 - 요소선택자 (Element Selector).

Confidential

- ✓ Multiple(다중) Selector 예제.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $(function() {
                $("h1, p").css("color", "orange");
            });
        });
    </script>
</head>
<body>
    <h1>JavaScript.</h1>
    <p>JavaScript 언어는 web에서 상당히 중요합니다.</p>
    <h1>jQuery.</h1>
    <p>jQuery는 JavaScript의 Library입니다.</p>
</body>
</html>
```

2-07.html

JavaScript.

JavaScript 언어는 web에서 상당히 중요합니다.

jQuery.

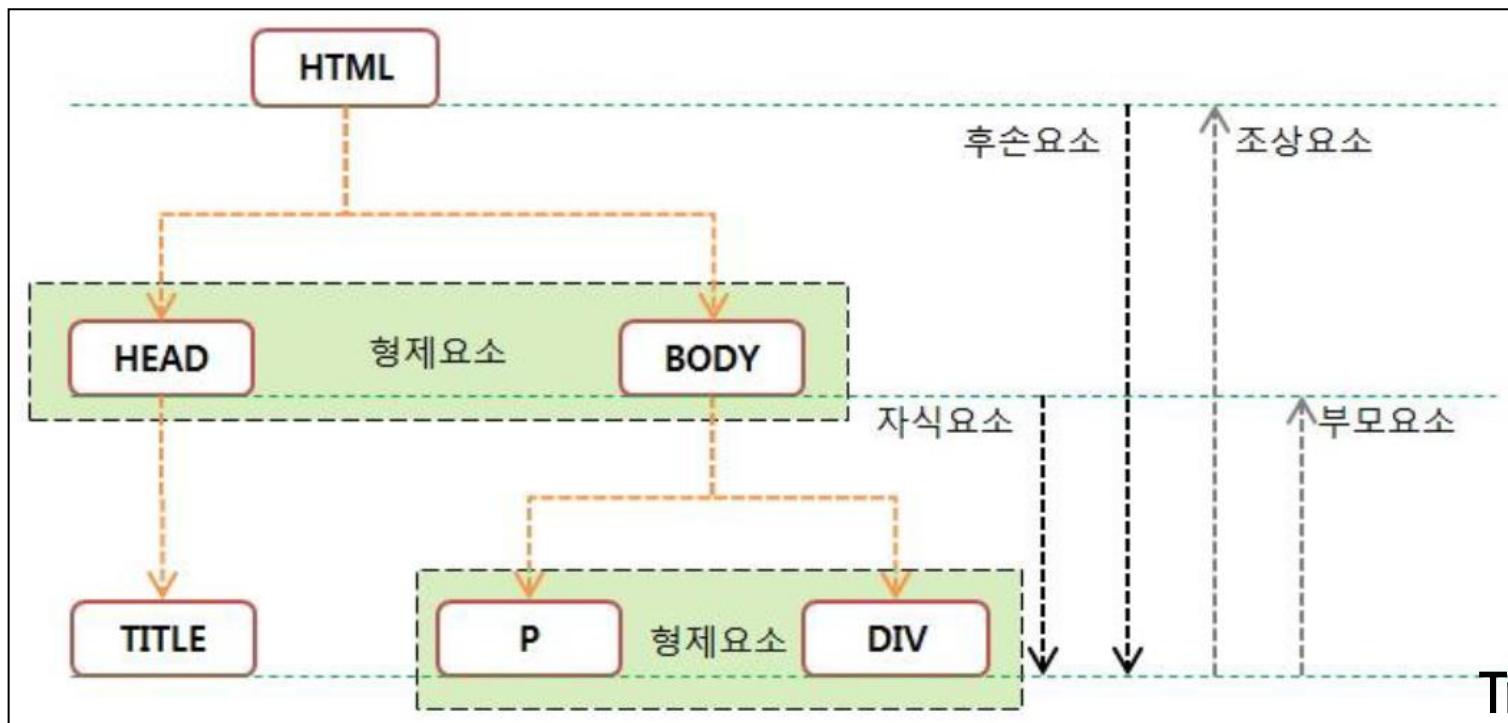
jQuery는 JavaScript의 Library입니다.

2. DOM 요소 선택.

2-4. DOM 계층구조 탐색 - DOM Hierarchy Selector.

Confidential

- ✓ jQuery는 HTML DOM 계층 구조에 접근하고 제어하는 쉬운 방법을 제공.
- ✓ DOM을 탐색할 때, 트리 형태의 DOM 계층구조를 이해하는 것은 중요.
- ✓ DOM 계층 구조가 나타내는 다양한 관계를 이해하면, DOM을 탐색하는데 도움.
- ✓ DOM 요소들은 수평적으로 형제요소, 수직적으로는 부모, 자식, 자손, 조상요소로 관계를 구분.



2. DOM 요소 선택.

2-4. DOM 계층구조 탐색 - DOM Hierarchy Selector.

Confidential

- ✓ \$("Parent > Child")는 자식(Child) 선택자로, 바로 인접한 하위 자식들을 탐색.
- ✓ \$("Ancestor Descendant")는 자손(Descendant) 선택자로, 모든 하위 자식들을 탐색.
- ✓ \$("Previous + Next")는 인접(Next Adjacent) 선택자로, 인접한 바로 다음 형제를 탐색.
- ✓ \$("Previous ~ Next")는 형제(Next Siblings) 선택자로, Next에 대한 모든 형제들을 탐색.

selector 종류	selector 표현방법	
Child selector 부모 요소 바로 아래 자식	\$("parent > child")	상하 관계
Descendant selector 조상과 일치하는 요소에 포함된 모든 후손	\$("ancestor descendant")	
Next Adjacent selector Previous 요소 바로 다음에 나오는 형제수준의 next요소	\$("previous + next")	평행 관계
Next Siblings selector Previous 요소 이후에 형제 요소 중 siblings와 동일한 모든 요소	\$("previous ~ siblings")	

2. DOM 요소 선택.

2-4. DOM 계층구조 탐색 - DOM Hierarchy Selector.

Confidential

✓ DOM 계층구조 탐색 예제1.

```
<script type="text/javascript">
$(document).ready(function() {
    $("div > ul").css("background", "#cc66ff");
    $("ul.site > li").css("border", "1px solid blue");
});
</script>
```

jQuery DOM Hierarchy Selector

검색을 도와주는 사이트는 여러가지가 있습니다.

- 다음 : www.daum.net
- 네이트 : www.nate.com
- ◦ 네이버블로그 : blog.naver.com
 - 네이버카페 : cafe.naver.com
- CGV : www.cgv.co.kr
- Megabox : www.megabox.co.kr

- 1.네이버 : www.naver.com
2.네이버블로그 : blog.naver.com
3.네이버카페 : cafe.naver.com

```
<h3>jQuery DOM Hierarchy Selector</h3>
<p>검색을 도와주는 사이트는 여러가지가 있습니다.</p>
<div>
    <ul class="site">
        <li>다음 : www.daum.net</li>
        <li>네이트 : www.nate.com</li>
        <li>
            <ul>
                <li>네이버블로그 : blog.naver.com</li>
                <li>네이버카페 : cafe.naver.com</li>
            </ul>
        </li>
        <li>CGV : www.cgv.co.kr</li>
        <li>Megabox : www.megabox.co.kr</li>
    </ul>
    <div>
        <p>
            1.네이버 : www.naver.com<br>
            2.네이버블로그 : blog.naver.com<br>
            3.네이버카페 : cafe.naver.com<br>
        </p>
    </div>
</div>
```

2. DOM 요소 선택.

2-4. DOM 계층구조 탐색 - DOM Hierarchy Selector.

Confidential

✓ DOM 계층구조 탐색 예제2.

```
$(document).ready(function() {
    $("div > ul").css("background", "#cc66ff");
    $("ul.site1 > li").css("color", "cyan");
    $("ul.site2 li").css("color", "pink");
});
```

jQuery DOM Hierarchy Selector

검색을 도와주는 사이트는 여러가지가 있습니다.

- 다음 : www.daum.net
- 네이트 : www.nate.com
- ◦ 네이버블로그 : blog.naver.com
- 네이버카페 : cafe.naver.com
- CGV : www.cgv.co.kr

- 다음 : www.daum.net
- 네이트 : www.nate.com
- ◦ 네이버블로그 : blog.naver.com
- 네이버카페 : cafe.naver.com
- CGV : www.cgv.co.kr

```
<h3>jQuery DOM Hierarchy Selector</h3>
<p>검색을 도와주는 사이트는 여러가지가 있습니다.</p>
<div>
    <ul class="site1">
        <li>다음 : www.daum.net</li>
        <li>네이트 : www.nate.com</li>
        <div>
            <li>
                <ul>
                    <li>네이버블로그 : blog.naver.com</li>
                    <li>네이버카페 : cafe.naver.com</li>
                </ul>
            </li>
        </div>
        <li>CGV : www.cgv.co.kr</li>
    </ul>
    <ul class="site2">
        <li>다음 : www.daum.net</li>
        <li>네이트 : www.nate.com</li>
        <div>
            <li>
                <ul>
                    <li>네이버블로그 : blog.naver.com</li>
                    <li>네이버카페 : cafe.naver.com</li>
                </ul>
            </li>
        </div>
        <li>CGV : www.cgv.co.kr</li>
    </ul>
</div>
```

2. DOM 요소 선택.

2-4. DOM 계층구조 탐색 - DOM Hierarchy Selector.

Confidential

- ✓ DOM 계층구조 탐색 예제3.

jQuery DOM Hierarchy Selector

이름 :

나이

아이디 :

```
$(document).ready(function() {
    $("label + input").css("border", "2px solid steelblue");
});
```

```
<h3>jQuery DOM Hierarchy Selector</h3>
<div>
    <form>
        <label>이름 : </label>
        <input name="name"/>
        <fieldset>
            <label>나이</label>
            <input name="age"/>
        </fieldset>
        아이디 : <input name="id"/>
    </form>
    <input name="none"/>
</div>
```

2. DOM 요소 선택.

2-4. DOM 계층구조 탐색 - DOM Hierarchy Selector.

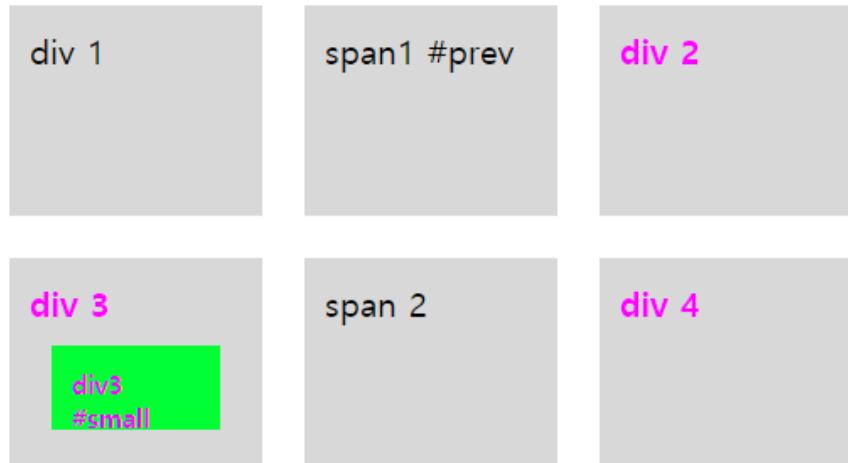
Confidential

- ✓ DOM 계층구조 탐색 예제4.

```
$(document).ready(function() {
    $("#prev ~ div").css("color", "magenta").css("font-weight", "bold");
});
```

jQuery DOM Hierarchy Selector

#prev ~ div



```
<h3>jQuery DOM Hierarchy Selector</h3>
<p id="center">#prev ~ div</p>
<div>div 1</div>
<span id="prev">span1 #prev</span>
<div>div 2</div>
<div>
    div 3 <div id="small">div3 #small</div>
</div>
<span>span 2</span>
<div>div 4</div>
```

2-11.html

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

- ✓ 속성 선택자(Attribute Selector)를 사용하여 DOM 요소가 가진 속성 값으로 DOM 객체를 탐색.
- ✓ 속성 선택자 표현식은 기본 선택자 바로 뒤에 나오는 대괄호([])안에 표현.
- ✓ 기본 선택자는 선택사항입니다. 기본 선택자를 생략하면, 모든 요소에 대해 주어진 속성 선택자를 비교.
- ✓ 입력 양식과 관련된 요소를 선택할 때 사용.

선택자 형식	요 약
<code>\$(selector[attr])</code>	attr을 속성값으로 가지는 문서객체 선택
<code>\$(selector[attr="value"])</code>	attr의 속성값이 value와 같은 문서객체 선택
<code>\$(selector[attr!="value"])</code>	attr의 속성값이 value와 같지 않은 문서객체 선택
<code>\$(selector[attr~="value"])</code>	attr의 속성값이 공백과 함께 value를 포함하는 문서객체 선택
<code>\$(selector[attr^="value"])</code>	attr의 속성값이 value로 시작하는 문서객체 선택
<code>\$(selector[attr\$="value"])</code>	attr의 속성값이 value로 끝나는 문서객체 선택
<code>\$(selector[attr*="value"])</code>	attr의 속성값이 value를 포함하는 문서객체 선택

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

✓ attribute select 예제.

```
<h2>jQuery Attribute Selector</h2>


<input type="text" name="case1" value="jQuery" /><br/>
    <input type="text" name="case2" value="jQuery 셀렉터" /><br/>
    <input type="text" name="case3" value="jQuery 셀렉터 테스트" /><br/>
    <input type="text" name="case4" value="셀렉터 테스트" /><br/>
    <input type="text" name="case5" value="셀렉터테스트" /><br/>
    <input type="text" name="case6" value="테스트" /><br/>



<input type="button" id="button1" value="name 속성을 가짐" />
    <input type="button" id="button2" value="jQuery 일치" />
    <input type="button" id="button3" value="jQuery 미일치" />
    <input type="button" id="button4" value="셀렉터 단어 포함" />
    <input type="button" id="button5" value="셀렉터 포함" />
    <input type="button" id="button6" value="셀렉터로 시작" />
    <input type="button" id="button7" value="셀렉터로 끝남" />
    <input type="button" id="btn_clear" value="clear" />


```

jQuery Attribute Selector

jQuery
jQuery 셀렉터
jQuery 셀렉터 테스트
셀렉터 테스트
셀렉터테스트
테스트

name속성을 가짐 jQuery' 일치 'jQuery' 미일치 '셀렉터' 단어 포함 '셀렉터' 포함 '셀렉터'로 시작 '셀렉터'로 끝남 clear

```
$function () {
    $("#button1").click(function () {
        clear(); $("input[name]").css("background", "silver");
    });
    $("#button2").click(function () {
        clear(); $("input[value='jQuery']").css("background", "silver");
    });
    $("#button3").click(function () {
        clear(); $("#targets > input[value!='jQuery']").css("background", "silver");
    });
    $("#button4").click(function () {
        clear(); $("input[value~='셀렉터']").css("background", "silver");
    });
    $("#button5").click(function () {
        clear(); $("#targets > input[value*='셀렉터']").css("background", "silver");
    });
    $("#button6").click(function () {
        clear(); $("input[value^='셀렉터']").css("background", "silver");
    });
    $("#button7").click(function () {
        clear(); $("input[value$='셀렉터']").css("background", "silver");
    });
    $("#btn_clear").click(function () {
        clear();
    });
}
function clear() {
    $("input[type='text']").css("background", "white");
}
```

2-12.html

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

✓ attribute select 예제.

jQuery Attribute Selector

검색을 도와주는 사이트는 여러가지가 있습니다.

Goooooogle
Naver(target)
Daum
Nate(target)

2-13.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>jquery</title>
  <script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <style type="text/css">
    div {background: #ccffff; padding: 10px; margin: 10px;}
  </style>
  <script type="text/javascript">
    $(document).ready(function() {
      $("div > a[target]").css("background", "#ff99ff");
    });
  </script>
</head>
<body>
  <h3>jQuery Attribute Selector</h3>
  <p>검색을 도와주는 사이트는 여러가지가 있습니다.</p>
  <div>
    <a href="http://www.google.com">Goooooogle</a><br>
    <a href="http://www.naver.com" target="_blank">Naver(target)</a><br>
    <a href="http://www.daum.net">Daum</a><br>
    <a href="http://www.nate.com" target="_blank">Nate(target)</a><br>
  </div>
</body>
</html>
```

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

✓ attribute select 예제.

jQuery Attribute Selector

검색을 도와주는 사이트는 여러가지가 있습니다.

href의 속성값이 <http://www.naver.com>인 요소의 배경색을 지정합니다.

[Goooooogle](http://www.google.com)
[Naver](http://www.naver.com)
[Daum](http://www.daum.net)
[Nate](http://www.nate.com)

2-14.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div {background: #ccffff; padding: 10px; margin: 10px;}
    </style>
    <script type="text/javascript">
        $(document).ready(function() {
            $("div > a[href='http://www.naver.com']").css("background", "#00cc33");
        });
    </script>
</head>
<body>
    <h3>jQuery Attribute Selector</h3>
    <p>검색을 도와주는 사이트는 여러가지가 있습니다.</p>
    <div>
        <p>href의 속성값이 http://www.naver.com인 요소의 배경색을 지정합니다.</p>
        <a href="http://www.google.com">Goooooogle</a><br>
        <a href="http://www.naver.com" target="_blank">Naver</a><br>
        <a href="http://www.daum.net">Daum</a><br>
        <a href="http://www.nate.com" target="_blank">Nate</a><br>
    </div>
</body>
</html>
```

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

✓ attribute select 예제.

jQuery Attribute Selector

검색을 도와주는 사이트는 여러가지가 있습니다.

href의 속성값이
http://www.naver.com이 아닌 요소의
배경색을 지정합니다.

[Goooooogle](http://www.google.com)
[Naver](http://www.naver.com)
[Daum](http://www.daum.net)
[Nate](http://www.nate.com)

2-15.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div {background: #ccffff; padding: 10px; margin: 10px;}
    </style>
    <script type="text/javascript">
        $(document).ready(function() {
            $("div > a[href!='http://www.naver.com']").css("background", "#00cc33");
        });
    </script>
</head>
<body>
    <h3>jQuery Attribute Selector</h3>
    <p>검색을 도와주는 사이트는 여러가지가 있습니다.</p>
    <div>
        <p>href의 속성값이 <br>http://www.naver.com이 아닌 요소의<br> 배경색을 지정합니다.</p>
        <a href="http://www.google.com">Goooooogle</a><br>
        <a href="http://www.naver.com" target="_blank">Naver</a><br>
        <a href="http://www.daum.net">Daum</a><br>
        <a href="http://www.nate.com" target="_blank">Nate</a>
    </div>
</body>
</html>
```

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

- ✓ attribute select 예제.

jQuery Attribute Selector

id가 subject로 시작하는 요소의 배경색을 지정합니다.

subject-1

content-1

content-2

subject-2

subject-3

content-3

2-16.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>jquery</title>
  <script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <style type="text/css">
    div {background: #ccffff; padding: 10px; margin: 10px;}
  </style>
  <script type="text/javascript">
    $(document).ready(function() {
      $("div[id^='subject']").css("background", "#cc66ff");
    });
  </script>
</head>
<body>
  <h3>jQuery Attribute Selector</h3>
  <div>
    <p>id가 subject로 시작하는 요소의 배경색을 지정합니다.</p>
    <div id="subject-1">subject-1</div>
    <div id="content-1">content-1</div>
    <div id="content-2">content-2</div>
    <div id="subject-2">subject-2</div>
    <div id="subject-3">subject-3</div>
    <div id="content-3">content-3</div>
  </div>
</body>
</html>
```

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

✓ attribute select 예제.

jQuery Attribute Selector

id가 1로 끝나는 요소의 배경색을 지정합니다.

subject-1

content-1

content-2

subject-2

subject-3

content-3

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>jquery</title>
  <script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <style type="text/css">
    div {background: #ccffff; padding: 10px; margin: 10px;}
  </style>
  <script type="text/javascript">
    $(document).ready(function() {
      $("div[id$='1']").css("background", "#cc66ff");
    });
  </script>
</head>
<body>
  <h3>jQuery Attribute Selector</h3>
  <div>
    <p>id가 1로 끝나는 요소의 배경색을 지정합니다.</p>
    <div id="subject-1">subject-1</div>
    <div id="content-1">content-1</div>
    <div id="content-2">content-2</div>
    <div id="subject-2">subject-2</div>
    <div id="subject-3">subject-3</div>
    <div id="content-3">content-3</div>
  </div>
</body>
</html>
```

2. DOM 요소 선택.

2-5. DOM 속성 탐색 - 속성선택자 (Attribute Selector).

Confidential

✓ attribute select 예제.

jQuery Attribute Selector

man.

name에 공백을 구분자로 단어가 man인
요소의 배경색을 지정합니다.

name is superman
name is iron man
name is hobbit
name is cat woman
name is batman

2-18.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div {background: #ccffff; padding: 10px; margin: 10px;}
    </style>
    <script type="text/javascript">
        $(document).ready(function() {
            $("input[name~='man']").css("background", "#cc66ff");
        });
    </script>
</head>
<body>
    <h3>jQuery Attribute Selector</h3>
    <p>man.</p>
    <div>
        <p>name에 공백을 구분자로 단어가 man인<br> 요소의 배경색을 지정합니다.</p>
        <input name="superman" value="name is superman"><br>
        <input name="iron man" value="name is iron man"><br>
        <input name="hobbit" value="name is hobbit"><br>
        <input name="cat woman" value="name is cat woman"><br>
        <input name="batman" value="name is batman"><br>
    </div>
</body>
</html>
```

2. DOM 요소 선택.

2-6. DOM 요소 필터링 - 필터선택자(Filter Selector).

Confidential

- ✓ 필터 선택자는 DOM 요소를 탐색한 결과에서 원하는 요소를 걸러내기 위하여 사용.
- ✓ 기본 선택자는 선택사항입니다. 필터 선택자는 기본 선택자 뒤에 콜론(:) 기호와 함께 표기.
- ✓ 필터 선택자에는 입력 폼 유형을 선택하는 필터 선택자와, 요소의 특성으로 선택하는 필터 선택자가 있다.
- ✓ 이 외에도, jQuery에는 위치 기반 필터 선택자와 함수 기반 필터 선택자가 있다.

필터 선택자	요 약	필터 선택자	요 약
el:button	type이 button인 input 요소 또는 button 요소 선택.	el:checked	체크된 입력 폼 선택.
el:checkbox	type이 checkbox인 input 요소 선택.	el:disabled	비활성화된 입력 폼 선택.
el:file	type이 file인 input 요소 선택.	el:enabled	활성화된 입력 폼 선택.
el:image	type이 image인 input 요소 선택.	el:focus	초점이 맞추어진 입력 폼 선택.
el:password	type이 password인 input 요소 선택.	el:input	모든 입력 폼 선택(input, textarea, select, button..).
el:radio	type이 radio인 input 요소 선택.	el:selected	option 객체 중 선택된 요소 선택.
el:reset	type이 reset인 input 요소 선택.	el:hidden	감춰진 요소 선택.
el:submit	type이 submit인 input 요소 선택.	el:visible	보이는 요소 선택.
el:text	type이 text인 input 요소 선택.	el:header	헤더 엘리먼트만 선택(<h1> ~ <h6>).

2. DOM 요소 선택.

2-6. DOM 요소 필터링 - 위치기반 필터선택자.

Confidential

- ✓ 위치 기반 필터 선택자는 선택한 요소들의 위치를 기반으로 필터를 수행.

필터 선택자	요약
:first	첫 번째 요소 선택.
:last	마지막 요소 선택.
:first-child	첫 번째 자식 요소 선택.
:last-child	마지막 자식 요소 선택.
:only-child	형제가 없는 모든 요소 선택.
:even	짝수 번째 요소 선택.
:odd	홀수 번째 요소 선택.

2. DOM 요소 선택.

2-6. DOM 요소 필터링 - 위치기반 필터선택자.

Confidential

✓ 위치 기반 필터 선택자 예제.

- 홀수행은 파란색. 짝수행은 블루색
- 첫 줄의 배경은 검정색, 글자는 흰색.

```
$(document).ready(function() {  
    // #1. 짝수 번째 행 배경색(#ff66ff)  
    $("#domTable tr:even").css("background", "#ff66ff");  
    // #2. 홀수 번째 행 배경색(#33ffff)  
    $("#domTable tr:odd").css("background", "#33ffff");  
    // #3. 첫 번째 행 배경색(black), 글자색(white)  
    $("#domTable tr:first").css("background", "black")  
        .css("color", "white");  
});
```

2-19.html

제목	내용
A1	A2
B1	B2
C1	C2
D1	D2

```
<table id="domTable">  
    <thead>  
        <tr>  
            <th>제목</th><th>내용</th>  
        </tr>  
    </thead>  
    <tbody>  
        <tr>  
            <td>A1</td><td>A2</td>  
        </tr>  
        <tr>  
            <td>B1</td><td>B2</td>  
        </tr>  
        <tr>  
            <td>C1</td><td>C2</td>  
        </tr>  
        <tr>  
            <td>D1</td><td>D2</td>  
        </tr>  
    </tbody>  
</table>
```

2. DOM 요소 선택.

2-6. DOM 요소 필터링 - 함수기반 필터선택자.

Confidential

- ✓ jQuery는 함수 형태의 필터 선택자를 제공.

필터 선택자	요약
:not(filter)	주어진 선택자와 일치하지 않는 모든 요소를 선택.
:contains(str)	텍스트 str을 포함하는 요소만 선택.
:nth-child(n)	n번째 자식 요소를 선택. (n이 2이면 2번째 자식, n이 3n+1이면 4, 7, 10..)
:eq(n)	n번째로 일치하는 요소를 선택.
:gt(n)	n번째(포함 X) 이후 요소를 선택.
:lt(n)	n번째(포함 X) 이전 요소를 선택.
:has(f)	주어진 선택자와 일치하는 하나 이상의 요소를 포함하는 요소를 선택.

- nth-* 계열 필터는 CSS에서 개념을 가져온 것으로 n값은 1부터 시작하는 index를 사용.
- eq와 같은 필터는 n값이 0부터 시작하는 JavaScript의 0기반 index를 사용.

2. DOM 요소 선택.

2-6. DOM 요소 필터링 - 함수기반 필터선택자.

Confidential

✓ 함수기반필터 선택자 예제1.

```
$("tr:eq(0)").css("background", "#000000").css("color", "white");
$("td:nth-child(3n)").css("background", "#F9F9F9");
$("td:nth-child(3n+1)").css("background", "#565656");
$("td:nth-child(3n+2)").css("background", "#A9A9A9");
```

jQuery필터 셀렉터

jQuery필터 셀렉터의 종류를 나열하였습니다.

이름	표현식	설명
button 필터	:button	모든 버튼 선택
checkbox 필터	:checkbox	체크박스 엘리먼트 선택
체크상태 필터	:checked	선택된 체크박스
비활성화 필터	:disabled	비활성화 엘리먼트 선택
활성화 필터	:enabled	활성화 엘리먼트 선택
파일 필터	:file	파일 엘리먼트 선택

```
<h2>jQuery필터 셀렉터</h2>
<p>jQuery필터 셀렉터의 종류를 나열하였습니다.</p>
<div>
  <table>
    <tr>
      <th>이름</th>
      <th>표현식</th>
      <th>설명</th>
    </tr>
    <tr>
      <td>button 필터</td>
      <td>:button</td>
      <td>모든 버튼 선택</td>
    </tr>
    <tr>
      <td>checkbox 필터</td>
      <td>:checkbox</td>
      <td>체크박스 엘리먼트 선택</td>
    </tr>
    <tr>
      <td>체크상태 필터</td>
      <td>:checked</td>
      <td>선택된 체크박스</td>
    </tr>
  </table>
</div>
```

2. DOM 요소 선택.

2-6. DOM 요소 필터링 - 함수기반 필터선택자.

- ✓ 함수기반필터 선택자 예제2.

```
$("td:not(.myClass)").css("background", "#A9A9A9");
```

jQuery필터 셀렉터

jQuery필터 셀렉터의 종류를 나열하였습니다.

이름	표현식	설명
button 필터	:button	모든 버튼 선택
checkbox 필터	:checkbox	체크박스 엘리먼트 선택
체크상태 필터	:checked	선택된 체크박스
비활성화 필터	:disabled	비활성화 엘리먼트 선택
활성화 필터	:enabled	활성화 엘리먼트 선택
파일 필터	:file	파일 엘리먼트 선택

2-21.html

Confidential

```
<h2>jQuery필터 셀렉터</h2>
<p>jQuery필터 셀렉터의 종류를 나열하였습니다.</p>
<div>
  <table>
    <tr>
      <th>이름</th>
      <th>표현식</th>
      <th>설명</th>
    </tr>
    <tr>
      <td>button 필터</td>
      <td>:button</td>
      <td>모든 버튼 선택</td>
    </tr>
    <tr>
      <td>checkbox 필터</td>
      <td>:checkbox</td>
      <td>체크박스 엘리먼트 선택</td>
    </tr>
    <tr>
      <td>체크상태 필터</td>
      <td>:checked</td>
      <td>선택된 체크박스</td>
    </tr>
    <tr>
      <td>비활성화 필터</td>
      <td>:disabled</td>
      <td>비활성화 엘리먼트 선택</td>
    </tr>
    <tr>
      <td>활성화 필터</td>
      <td>:enabled</td>
      <td>활성화 엘리먼트 선택</td>
    </tr>
    <tr>
      <td>파일 필터</td>
      <td>:file</td>
      <td>파일 엘리먼트 선택</td>
    </tr>
  </table>
</div>
```

2. DOM 요소 선택.

2-7. jQuery method - 래퍼세트와 메소드.

Confidential

- ✓ jQuery는 선택자를 통해 탐색한 DOM 객체들을 래퍼세트라는 특별한 배열 객체에 담아 반환합니다.
- ✓ jQuery는 선택된 DOM 객체가 없는 경우에도, 비어있는 래퍼세트 객체를 반환합니다.
- ✓ 래퍼세트 객체에는 내포된 DOM 객체들을 처리하는 다양한 메소드가 있습니다.
- ✓ 이러한 jQuery 메소드는 플러그-인 확장을 통해 추가할 수 있습니다.

메소드	반환값	요약
size()	요소 개수	래퍼세트의 요소 개수 반환.
get(index)	DOM 요소	래퍼세트에서 인덱스 번호에 위치하는 DOM 객체 반환.
index(element)	인덱스 번호	래퍼세트에서 해당 요소의 인덱스 번호 반환
add(expr)	래퍼세트	expr으로 명시한 요소를 래퍼세트에 추가.
not(expr)	래퍼세트	expr으로 명시한 요소를 래퍼세트에서 제거.
each(function(index, element))	이전 래퍼세트	래퍼세트의 각 요소마다 function을 수행.
filter(expr)	래퍼세트	expr에 명시한 요소를 필터링.
slice(begin, end)	래퍼세트	현재 래퍼세트의 일부분으로 새로운 래퍼세트를 생성하여 반환.

2-7. jQuery method- 계층 구조 탐색.

Confidential

- ✓ jQuery는 선택자를 통해 탐색한 DOM 객체들을 래퍼세트라는 특별한 배열 객체에 담아 반환합니다.
- ✓ jQuery는 선택된 DOM 객체가 없는 경우에도, 비어있는 래퍼세트 객체를 반환합니다.
- ✓ 래퍼세트 객체에는 내포된 DOM 객체들을 처리하는 다양한 메소드가 있습니다.
- ✓ 이러한 jQuery 메소드는 플러그-인 확장을 통해 추가할 수 있습니다.

메소드	요약
parent([selector])	래퍼세트에 포함된 각 요소의 부모요소로 구성된 래퍼세트 반환.
parents([selector])	래퍼세트에 포함된 각 요소의 조상요소로 구성된 래퍼세트 반환.
children([selector])	래퍼세트에 포함된 각 요소의 자식요소로 구성된 래퍼세트 반환.
prev([selector])	래퍼세트에 포함된 각 요소의 바로 이전 형제요소로 구성된 래퍼세트 반환.
prevAll([selector])	래퍼세트에 포함된 각 요소의 이전에 나오는 형제요소들로 구성된 래퍼세트 반환.
next([selector])	래퍼세트에 포함된 각 요소의 바로 이후 형제요소로 구성된 래퍼세트 반환.
nextAll([selector])	래퍼세트에 포함된 각 요소의 이후에 나오는 형제요소들로 구성된 래퍼세트 반환.
siblings([selector])	래퍼세트의 각 요소를 제외한 모든 형제요소들로 구성된 래퍼세트 반환.

2. DOM 요소 선택.

2-7. jQuery method- 요소 반복.

Confidential

- ✓ `jQuery.each()` 함수는 배열이나 객체를 반복적으로 처리할 때 사용.
- ✓ 첫 번째 인자에는 자바스크립트 배열이나 래퍼세트 객체가 올 수 있다.
- ✓ 두 번째 인자에는 각 요소를 반복하면서 처리할 콜백(call back)함수를 정의.
- ✓ 콜백 함수는 두 개의 매개변수(`index`: 배열 인덱스, `item`: 반복하는 요소 객체)를 갖는다.

```
$.each(array, function(index, item) {});
```

SSAFY
Naver
Google

```
var array = [  
    { name: 'SSAFY',  
      link: 'http://www.ssafy.com' },  
    { name: 'Naver',  
      link: 'http://www.naver.com' },  
    { name: 'Google',  
      link: 'http://www.google.com' }  
];  
var outHtml = '';  
// array 배열을 탐색하며 outHtml 생성  
$.each(array, function (index, item) {  
    outHtml += '<a href="' + item.link + '">';  
    outHtml += ' <h2>' + item.name + '</h2>';  
    outHtml += '</a>';  
});  
document.body.innerHTML = outHtml;
```

2. DOM 요소 선택.

2-7. jQuery method- 요소 반복.

Confidential

- ✓ jQuery 선택자의 결과인 래퍼세트는 탐색한 DOM 요소들을 반복하는 each() 메소드를 제공.
- ✓ \$.each() 함수와 달리 반복대상을 의미하는 첫 번째 인자가 없으며, 래퍼세트의 DOM 요소들이 반복 대상.
- ✓ 래퍼세트의 each() 메소드는 유일한 인자인 콜백 함수가 있다.
- ✓ 콜백 함수는 두 개의 매개변수(index: 배열 인덱스, item: 반복하는 요소 객체)를 갖는다.

```
$(selector).each(function(index, item) {});
```



```
<h2>첫번째</h2>
<h2>두번째</h2>
<h2>세번째</h2>
<h2>네번째</h2>
<h2>다섯번째</h2>
```

```
<style type="text/css">
  .high-light-0 { background: yellow; }
  .high-light-1 { background: orange; }
  .high-light-2 { background: blue; }
  .high-light-3 { background: green; }
  .high-light-4 { background: red; }
</style>
```

```
// h2 태그마다 high-light-0~4 클래스로 CSS 적용
$("h2").each(function (index, item) {
  // item 대신 this 사용 가능
  $(item).addClass("high-light-" + index);
});
```

2-23.html

2. DOM 요소 선택.

2-7. jQuery method - 필터 함수.

Confidential

- ✓ filter() 는 래퍼세트에 포함된 DOM 요소를 주어진 조건으로 걸러내는 메소드.
- ✓ filter() 메소드는 기존 래퍼세트에서 DOM 요소를 축소한 새로운 래퍼세트를 반환.
- ✓ filter() 메소드 인자에는 필터링 조건을 나타내는 선택자나 함수가 올 수 있다.

```
$(selector).filter(selector);
$selector.filter(function(index, item) {});
```

첫번째

두번째

세번째

네번째

다섯번째

여섯번째

```
/*
$("h2:even").css({
    background: "black",
    color: "white"
});
*/
// h2 태그의 홀수 또는 짝수에 CSS 색상 적용
$("h2").filter(":even").css({
    background: "black",
    color: "white"
});
```

2-24.html

2. DOM 요소 선택.

2-7. jQuery method - 필터 함수.

Confidential

- ✓ 필터 선택자를 사용하여 필터링 조건을 나타내기 어려운 경우, 함수를 사용.
- ✓ 필터링 조건을 함수로 사용한 경우, filter() 메소드는 함수 반환 값이 true인 요소만 새로운 래퍼세트에 포함.
- ✓ filter() 메소드를 호출하여 필터링한 래퍼세트에서 end() 메소드를 호출하면 이전 래퍼세트로 돌아간다.

첫번째

두번째

세번째

네번째

다섯번째

여섯번째

```
// 필터 Selector 적용이 어려운 경우  
// 3n번째 h2 엘리먼트에 CSS 색상 적용  
$("h2").filter(function (index) {  
    return index % 3 == 0;  
}).css({  
    background: "black",  
    color: "white"  
});
```

2-25.html

2-26.html

```
// 필터 Selector 적용한 경우  
/*  
$("h2").css("background", "silver");  
$("h2:even").css("color", "white");  
$("h2:odd").css("color", "red");  
*/  
// 메소드 체인 적용  
// 위의 내용을 filter 메소드를 사용하여 구현  
$("h2").css("background", "silver")  
    .filter(":even").css("color", "white").end()  
    .filter(":odd").css("color", "red");
```

첫번째

두번째

세번째

네번째

다섯번째

여섯번째

2. DOM 요소 선택.

2-7. jQuery method- 위치기반 함수.

Confidential

- ✓ 위치 기반 함수는 HTML DOM 객체의 특정 위치를 선택할 수 있는 함수.
- ✓ `eq()` 메소드는 래퍼세트에서 주어진 인덱스번호에 해당하는 DOM 요소를 감싼 새로운 래퍼세트를 반환.
- ✓ `first()` 메소드는 래퍼세트의 첫 번째 DOM 요소를 감싼 새로운 래퍼세트를 반환.
- ✓ `last()` 메소드는 래퍼세트의 마지막 DOM 요소를 감싼 새로운 래퍼세트를 반환.

```
$(selector).eq(index);  
$(selector).first();  
$(selector).last();
```

첫번째
두번째
세번째
네번째
다섯번째
여섯번째

```
// first() 함수를 사용하여 첫번째 h2 태그 색상 변경  
// last() 함수를 사용하여 마지막 h2 태그 색상 변경  
// eq() 함수를 사용하여 세번째 h2 태그 색상 변경  
$("h2").first().css("background", "blue")  
    .css("color", "white");  
$("h2").last().css("background", "black");  
$("h2").eq(2).css("background", "silver");  
$("h2").eq(-1).css("color", "white");
```

- `eq()` 메소드에 -1과 같이 음수 값(-n)이 들어가는 경우 확장집합요소의 맨 뒤에서 부터 n번째 요소를 선택.

2-27.html

2. DOM 요소 선택.

2-7. jQuery method - 래퍼세트에 요소 추가/삭제.

Confidential

- ✓ 필터 함수와 위치 기반 함수는 래퍼세트에 포함된 요소를 축소하기 위해 사용합니다.
- ✓ jQuery는 래퍼세트에 요소를 추가하거나 특정 요소를 삭제하는 메소드를 제공합니다.
- ✓ `add()` 메소드는 래퍼세트에 주어진 조건을 만족하는 요소를 추가한 새로운 래퍼세트를 반환합니다.
- ✓ `not()` 메소드는 래퍼세트에서 주어진 조건에 해당하는 요소를 제거한 새로운 래퍼세트를 반환합니다.

```
$("element");
.add("element");
.not("element");
```

2. DOM 요소 선택.

2-7. jQuery method - DOM 요소 판별.

Confidential

- ✓ `is()` 메소드는 기존 래퍼세트가 주어진 선택자와 일치하는지 여부를 반환.
- ✓ 다른 메소드들이 새로운 래퍼세트 객체를 반환하는 반면, `is()` 메소드는 boolean 값을 반환.
- ✓ 비교할 조건을 선택자로 표현하기 어려운 경우, 함수를 사용할 수 있다.
- ✓ 비교 조건을 나타내는 함수 호출 결과가 `true`인 경우, `is()` 메소드는 `true`를 반환.

```
$(selector).is(selector);
$(selector).is(function(index, item) {});
```

```
// 체크박스 중 name이 checkedFalse이고
// 체크되어 있으면 체크를 해제함
$("input[type='checkbox']").each(function () {
    if ($(this).is("[name='checkedFalse']") && $(this).is(":checked")) {
        $(this).attr("checked", false);
    }
});
```

<input type="checkbox"/> 체크박스1
<input checked="" type="checkbox"/> 체크박스2
<input checked="" type="checkbox"/> 체크박스3

```
<h3><input type="checkbox" name="checkedFalse" checked="checked"/>체크박스1</h3>
<h3><input type="checkbox" checked="checked"/>체크박스2</h3>
<h3><input type="checkbox" checked="checked"/>체크박스3</h3>
```

2-28.html

2. DOM 요소 선택.

2-7. jQuery method- 자손 요소 탐색.

Confidential

- ✓ `find()` 메소드는 래퍼세트의 모든 요소들에 대하여 주어진 선택자를 만족하는 모든 자손 요소를 선택.
- ✓ `find()` 메소드는 선택자를 통해 탐색한 DOM 요소들을 새로운 래퍼세트로 반환.
- ✓ 래퍼세트의 `filter()` 메소드는 래퍼세트 요소를 걸러내기 위해 사용.
- ✓ 그러나, `find()` 메소드는 래퍼세트 요소들의 하위 자손들을 탐색하기 위하여 사용.

`$(selector).find(selector);`

```
// findArea 하위 요소 중  
// class가 myClass인 요소들의 배경색을 silver로 변경  
$("#findArea").find(".myClass") .css("background", "silver");
```

```
<div id="findArea">  
  <h1>첫번째</h1>  
  <h2 class="myClass">두번째</h2>  
  <h3 class="myClass">세번째</h3>  
  <h4>네번째</h4>  
  <div><h4 class="myClass">div태그안</h4></div>  
</div>
```

2-29.html

첫번째

두번째

세번째

네번째

div태그안

2-8. 정리.

Confidential

- ✓ jQuery는 DOM 탐색을 위하여 CSS에서 사용하는 Selector 표현 방식을 사용.
- ✓ jQuery는 선택자를 통해 HTML DOM 계층 구조에 접근하고 제어하는 쉬운 방법을 제공.
- ✓ 필터 선택자는 DOM 요소를 탐색한 결과에서 원하는 요소를 걸러내기 위하여 사용.
- ✓ jQuery 래퍼세트 객체에는 내포된 DOM 객체들을 처리하는 다양한 메소드가 있다.

3. DOM 객체

3. DOM 객체.

3-1. DOM 객체 제어.

Confidential

- ✓ 순수 자바스크립트만을 이용하여 DOM 객체 구조를 처리하는 것은 어렵고 성가신 작업이다.
- ✓ jQuery 메소드를 사용하면, DOM 객체를 보다 쉽게 다룰 수 있다.
- ✓ jQuery는 DOM 요소의 속성이나 class, style을 제어하는 DOM 특성제어 메소드를 제공.
- ✓ 이 밖에도, DOM 내부제어 메소드, DOM 추가/삭제 메소드와 DOM 객체를 삽입하는 메소드가 있다.

DOM 특성 제어	DOM 내부 제어	DOM 추가 및 삭제	DOM 객체 삽입
attr()	html()	\$()	append()
removeAttr()	text()	remove()	appendTo()
addClass()		empty()	prepend()
removeClass()		clone()	prependTo()
toggleClass()			after()
css()			insertAfter()
			before()
			insertBefore()

3. DOM 객체.

3-2. DOM 특성 제어 - 속성제어 메소드.

Confidential

- ✓ `attr(name)` 메소드는 래퍼세트의 첫 번째 요소에 대한 속성(attribute) 값을 반환.
- ✓ `attr(name)` 메소드는 해당 속성이 없으면 `undefined`를 반환.
- ✓ `attr(name, value), attr(object), attr(name, function)` 메소드는 확장 집합의 모든 요소에 속성 값을 설정.
- ✓ `removeAttr(name)` 메소드는 DOM 요소에서 해당 속성을 제거.

속성 값 조회 메소드

```
$(selector).attr(name);
```

속성 값 설정 메소드

```
$(selector).attr(name, value);
$(selector).attr(name, function(index, attr) {});
$(selector).attr(object);
```

속성 삭제 메소드

```
$(selector).removeAttr(name);
```

3. DOM 객체.

3-2. DOM 특성 제어 - 속성제어 메소드.

Confidential

```
<script type="text/javascript">
$(document).ready(function() {
    var srcValue = $("img").attr("src");
    alert("src1 : " + srcValue);

    $("img").each(function(i, element) {
        var value = $(this).attr("src");
        alert("src2 : " + value);
    });

    $("img").attr({"width" : 100, "height" : 150});
    alert("width와 height를 200으로 설정!!");

    $("img").removeAttr("height");
    alert("height 속성 제거!!!");

    $("img").attr("width", function(i) {
        return (i + 1) * 100;
    });
});
</script>
</head>
<body>





```



3. DOM 객체.

3-2. DOM 특성 제어 - Class 속성제어.

Confidential

class 속성 값 추가 및 삭제 메소드

```
$(selector).addClass(className);
$(selector).removeClass(className);
$(selector).toggleClass(className);
```

안녕하세요1.

클릭시마다 변경

안녕하세요1.

안녕하세요2.

안녕하세요2.

안녕하세요3.

안녕하세요3.

안녕하세요4.

안녕하세요4.

```
<style type="text/css">
.col1 { color : orange }
.col2 { color : magenta }
.col3 { color : steelblue }
</style>
<script type="text/javascript">
$(document).ready(function() {
    $("#col1").addClass("col1");
    $("h3[name='col2']").addClass("col2");
    $(".col3").removeClass(".col3");
    $("h1").click(function() {
        $(this).toggleClass("col3");
    })
});
</script>
<head>
<title>DOM Class 속성 제어</title>
</head>
<body>
<h1>안녕하세요1.</h1>
<h2 id="col1">안녕하세요2.</h2>
<h3 name="col2">안녕하세요3.</h3>
<h4 class="col3">안녕하세요4.</h4>
</body>
</html>
```

3-02.html

3. DOM 객체.

3-2. DOM 특성 제어 - Style 속성제어.

Confidential

- ✓ `css(name)` 메소드는 래퍼세트의 첫 번째 요소에 대한 `style` 속성 값을 반환합니다.
- ✓ `css(name, value)` 메소드는 래퍼세트 내 모든 DOM 요소에 `style`을 설정합니다.
- ✓ `css(name, function)` 메소드는 `style` 속성 값을 바로 지정하는 대신, 함수를 통해 `style` 속성 값을 설정합니다.
- ✓ 여러 `style` 속성 값을 한번에 설정하려면 `css(object)`와 같이 객체를 인자로 사용합니다.

속성 값 조회 메소드

```
$(selector).css(styleName);
```

속성 삭제 메소드

```
$(selector).css(name, value);
$(selector).css(name, function(i, style) {});
$(selector).css(name, value);
```

3. DOM 객체.

3-2. DOM 특성 제어 - Style 속성제어.

Confidential

```
<style type="text/css">
.col1 { color : orange; }
.col2 { color : magenta; }
.col3 { color : steelblue; }
.col4 { color : green; }
</style>
<script type="text/javascript">
$(document).ready(function() {
    var color = $("h2").css("color");
    alert(color);

    $("h2").css("color", "blue");

    var colors = ["silver", "lime"];
    $("h2").css("color", function(index) {
        return colors[index % 2];
    });

    $("h2").css({
        color: function(index){
            return colors[index % 2];
        },
        background: "black"
    });
});
</script>
```

첫 실행화면

함수를 사용한 style
속성 값 설정

3-03.html

```
<h2 class="col1">안녕하세요1.</h2>
<h2 class="col2">안녕하세요2.</h2>
<h2 class="col3">안녕하세요3.</h2>
<h2 class="col4">안녕하세요4.</h2>
```

안녕하세요1.
안녕하세요2.
안녕하세요3.
안녕하세요4.

안녕하세요1.
안녕하세요2.
안녕하세요3.
안녕하세요4.

안녕하세요1.
안녕하세요2.
안녕하세요3.
안녕하세요4.

안녕하세요1.
안녕하세요2.
안녕하세요3.
안녕하세요4.

선택한 모든 h2 요소들에
대하여 style 속성 값 설정

객체를 사용하여 한번에
여러 속성 값 설정

3. DOM 객체.

3-3. DOM 내부 제어 - HTML과 텍스트 조회.

Confidential

- ✓ `html()` 과 `text()` 는 DOM 객체 내부를 조회하는 메소드.
- ✓ 자바스크립트의 `innerHTML`이나 `textContent` 속성과 관련이 있다.
- ✓ `html()` 함수는 HTML 태그를 인식하지만, `text()` 함수는 HTML 태그를 인식하지 않는다.
- ✓ 래퍼세트에서 `text()` 를 호출하면 태그를 인식하지 못하므로 태그를 제외한 텍스트만 반환.

객체내부 조회 메소드

```
$(selector).html();
$(selector).text();
```

```
<div id="div1">
    아이디 : <input type="text" placeholder="아이디입력.." /><br/>
    <span>사용자 설정.</span><br/>
```

```
var html = $("#div1").html();
var text = $("#div1").text();
alert(html);
alert(text);
$("#div1").append(html).append(text);
```

3-04.html

localhost:8080 내용:

```
~~아이디 : <input type="text" placeholder="아이디입력.." />
~~<span>사용자 설정.</span>
```

localhost:8080 내용:

```
~~아이디 :
~~사용자 설정.
```

아이디 : 아이디입력..

사용자 설정.

아이디 : 아이디입력..

사용자 설정.

아이디 : 사용자 설정.

3. DOM 객체.

3-3. DOM 내부 제어 - HTML과 텍스트 설정.

Confidential

- ✓ `html(value)`와 `text(value)`는 DOM 객체 내부를 설정하는 메소드.
- ✓ `html(value)` 메소드는 HTML 태그를 인식하므로, DOM 객체 형태로 지정한 내용을 설정.
- ✓ 그러나, `text(value)`는 태그를 인식하지 못하므로, 텍스트 형태로 지정한 내용을 설정.
- ✓ `text(value)`로 태그가 포함된 내용을 설정하면, 이 내용은 단순 텍스트로 인식되어 태그 내용이 고스란히 보인다.

객체내부 조회 메소드

```
$(selector).html(value);  
$(selector).text(value);
```

안녕하세요1.

안녕하세요2.

```
<h2>안녕하세요1.</h2><h2>안녕하세요2.</h2>
```

```
var html = "<h2>안녕하세요1.</h2>";  
html += "<h2>안녕하세요2.</h2>";  
  
$("div.divHtml").html(html);  
$("div.divText").text(html);
```

```
<div class="divHtml"></div>  
<div class="divText"></div>
```

3. DOM 객체.

3-4. DOM 추가 및 삭제- DOM 객체 추가.

Confidential

- ✓ `html()` 메소드는 문자열로 된 HTML을 DOM 객체로 추가.
- ✓ 그러나, 문자열을 이어 붙여서 HTML을 구성하는 작업은 복잡하고 지저분한 코드를 유발.
- ✓ `$()` 함수는 선택자를 수행하는 기능 뿐만 아니라, DOM 객체를 바로 생성하는 기능을 제공.
- ✓ 메소드 체인을 사용하면, `$()` 함수로 DOM 객체를 생성하여 객체 특성을 설정하는 작업을 간편하게 할 수 있다.

DOM 객체 생성 메소드

```
$();
```

안녕하세요

[싸피 공식 사이트](#)

```
$( "<h2>안녕하세요</h2>" )
.css("color", "steelblue")
.appendTo("body");
$( "<a/>" )
.attr("href", "http://www.ssafy.com")
.text("싸피 공식 사이트")
.appendTo("body");
```

3-06.html

3. DOM 객체.

3-4. DOM 추가 및 삭제- DOM 객체 삭제.

Confidential

- ✓ jQuery는 DOM 객체를 삭제하거나, DOM 객체 내부를 비우는 메소드를 제공.
- ✓ remove() 메소드는 래퍼세트의 모든 요소를 HTML 문서에서 삭제하고, 삭제한 내용을 반환.
- ✓ empty() 메소드는 래퍼세트의 모든 요소에 대해 하위 자식요소들을 삭제.
- ✓ empty() 메소드는 하위 자식요소들을 삭제한 결과를 반환.

객체내부 조회 메소드

```
$(selector).remove();  
$(selector).empty();
```

```
<div>  
  <h2 id="h1">안녕하세요1</h2>  
  <h2 id="h2">안녕하세요2</h2>  
  <h2 id="h3">안녕하세요3</h2>  
</div>  
<input type="button" id="btn1" value="두번째 h2 삭제"/>  
<input type="button" id="btn2" value="div 비움"/>
```

```
$("#btn1").click(function() {  
  $("#h2").remove();  
});  
$("#btn2").click(function() {  
  $("div").empty();  
});
```

안녕하세요1

안녕하세요2

안녕하세요3

두번째 h2 삭제

div 비움

안녕하세요1

안녕하세요3

두번째 h2 삭제

div 비움

두번째 h2 삭제

div 비움

3. DOM 객체.

3-5. DOM 객체 삽입- 개요.

Confidential

- ✓ DOM 객체 삽입은 이미 존재하는 DOM 객체에 다른 DOM 객체를 삽입하는 것.
- ✓ Target 객체를 기준으로 어느 위치에 삽입하는지에 따라 다양한 메소드가 있다.
- ✓ `append()`, `prepend()` 메소드는 Target 객체의 자식요소로 DOM 객체를 삽입.
- ✓ `before()`, `after()` 메소드는 Target 객체의 인접한 형제요소로 DOM 객체를 삽입.

메소드	요약
<code>\$(Target).append(B)</code>	Target 객체의 하위 자식요소로 맨 뒤에 B 객체를 추가.
<code>\$(Target).prepend(B)</code>	Target 객체의 하위 자식요소로 맨 앞에 B 객체를 추가.
<code>\$(Target).after(B)</code>	Target 객체 바로 이전 요소로 B 객체를 추가.
<code>\$(Target).before(B)</code>	Target 객체 바로 이후 요소로 B 객체를 추가.

3. DOM 객체.

3-5. DOM 객체 삽입 - jQuery 메소드.

Confidential

- ✓ DOM 객체 삽입 메소드는 파라미터에 어떤 객체를 사용하는지에 따라 두 가지 유형으로 구분.
- ✓ `appendTo()`, `prependTo()`, `insertAfter()`, `insertBefore()` 메소드는 삽입 대상 객체를 인자로 받는다.
- ✓ `append()`, `prepend()`, `after()`, `before()` 메소드는 삽입하려는 내용(또는 객체)을 인자로 받는다.
- ✓ 삽입하려는 내용이 DOM 객체인 경우, 해당 객체는 이전 위치에서 새로운 위치로 이동.

```
$(document).ready(function() {
    var student = {
        "no"      : "2012987",
        "name"    : "김싸피",
        "phone"   : "010-1234-9876",
        "email"   : "kimssafy@ssafy.com"
    };
    var newStudent = '<tr>';
    newStudent += '<td>' + student.no + '</td>';
    newStudent += '<td>' + student.name + '</td>';
    newStudent += '<td>' + student.phone + '</td>';
    newStudent += '<td>' + student.email + '</td>';
    newStudent += '</tr>';

    $("tbody").append(newStudent);
});
```

학번	이름	전화번호	이메일
2012985	홍싸피	010-5577-9988	hong@naver.com
2012986	박싸피	010-9977-5432	hong@naver.com
2012987	김싸피	010-1234-9876	kimssafy@ssafy.com

```
<tbody>
<tr>
    <td>2012985</td>
    <td>홍싸피</td>
    <td>010-5577-9988</td>
    <td>hong@naver.com</td>
</tr>
<tr>
    <td>2012986</td>
    <td>박싸피</td>
    <td>010-9977-5432</td>
    <td>hong@naver.com</td>
</tr>
</tbody>
```

3. DOM 객체.

3-5. DOM 객체 삽입 - jQuery 메소드.

Confidential

- ✓ 4개의 이미지를 2초에 한번씩 순서를 바꾸는 예제

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("img").css("width", 200);

            setInterval(function() {
                $("img").first().appendTo("body");
            }, 2000);
        });
    </script>
</head>
<body>
    
    
    
    
</body>
</html>
```

3-09.html



`$(A).appendTo(B)` : A가 B의 마지막에 추가.



2초마다 순서 변경.



3. DOM 객체.

3-6. jQuery Effect - Effect 메소드.

Confidential

- ✓ jQuery Effect는 화면에서 보여주는 시각 효과를 구현하는 방법.
- ✓ 사용자가 직접 애니메이션 효과를 만들 수 있다.
- ✓ jQuery 플러그-인으로 시각 효과를 내는 메소드를 구현할 수 있다.
- ✓ jQuery에서 기본으로 제공하는 Effect 메소드는 아래와 같다.

[jQuery Effect API](#)

메소드	요약
show()	DOM 요소를 화면에 보이도록 함.
hide()	DOM 요소를 화면에서 숨김.
toggle()	show()와 hide()를 번갈아 실행.
slideDown()	DOM 요소를 슬라이드 효과와 함께 나타나게 함.
slideUp()	DOM 요소를 슬라이드 효과와 함께 사라지게 함.
slideToggle()	slideDown()과 slideUp()을 번갈아 실행.
fadeIn()	DOM 요소가 서서히 나타남.
fadeOut()	DOM 요소가 서서히 사라짐.
fadeToggle()	fadeIn()과 fadeout()이 번갈아 실행.
animate()	css 속성 값을 이용하여 위치, 크기, 속성 값을 서서히 변경하는 애니메이션 효과를 줌.

3. DOM 객체.

3-6. jQueryEffect - Effect 매개변수.

Confidential

- ✓ jQueryEffect 모든 메소드는 공통적으로 세 개의 매개변수를 제공.
- ✓ speed는 효과를 진행하는 속도를 지정.
- ✓ callback은 효과를 완료한 후 실행할 함수를 지정.
- ✓ easing은 애니메이션 easing 형태 지정합니다. 플러그-인을 사용하지 않으면 linear와 swing만 입력 가능.

객체내부 조회 메소드

```
$(selector).method();
$(selector).method(speed);
$(selector).method(speed, callback);
$(selector).method(speed, easing, callback);
```

```
$(document).ready(function() {
    $("button").click(function() {
        $(".div1").toggle("slow", function() {
            alert("완료!!!");
        });
    });
});
</script>
<head>
<dy>
<button>토글 보여주기</button>
<div class="div1">
    <h2>토글 효과</h2>
    <p>토글 함수를 이용하여 간단한 jQuery효과를 볼 수 있습니다.</p>
</div>
```

3. DOM 객체.

3-6. jQuery Effect - 사용자 정의 효과.

Confidential

- ✓ jQuery에서 기본으로 제공하는 간단한 효과만으로도 시각적인 효과를 얻을 수 있다.
- ✓ 조금 더 수준 높은 효과를 만들려면 효과를 개별적으로 정의해야 한다.
- ✓ animate() 함수는 사용자 정의 효과를 만들 수 있는 방법을 제공.
- ✓ 첫 번째 인자인 객체에 속성 값을 설정하여 세밀하게 효과를 조정할 수 있다.

객체내부 조회 메소드

```
$(selector).animate(object);
$(selector).animate(object, speed);
$(selector).animate(object, speed, callback);
$(selector).animate(object, speed, easing, callback);
```

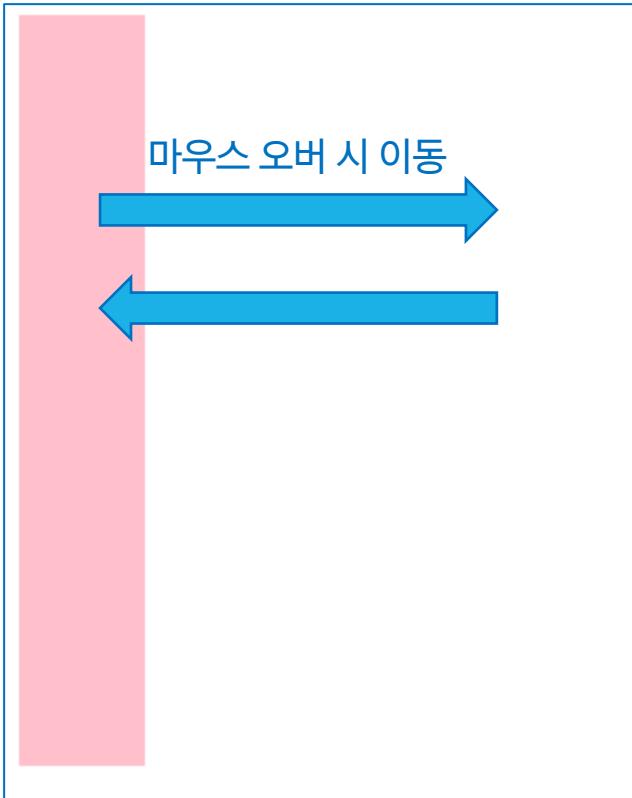
object 속성의 종류

opacity	top
height	bottom
width	margin
left	padding
right	

3-6. jQuery Effect - 사용자 정의 효과.

Confidential

- ✓ 마우스over시 div 이동.



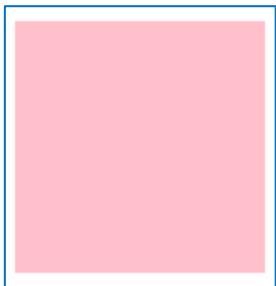
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div {
            width: 50px; height: 50px;
            background-color: pink;
            position: relative;
        }
    </style>
    <script type="text/javascript">
        $(document).ready(function() {
            $("div").hover(function() {
                $(this).animate({ left: 500 }, 500);
            }, function() {
                $(this).animate({ left: 0 }, 100);
            });
        });
    </script>
</head>
<body>
    <div></div><div></div><div></div><div></div>
    <div></div><div></div><div></div><div></div>
</body>
</html>
```

3. DOM 객체.

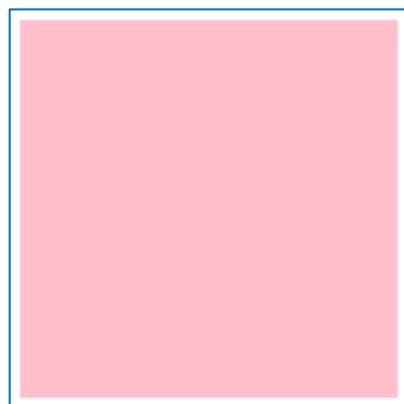
3-6. jQuery Effect - 사용자 정의 효과.

Confidential

- ✓ 클릭하면 커지는 사각형 효과



클릭시마다
50px씩 커짐



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <style type="text/css">
        div {
            width: 50px; height: 50px;
            background-color: pink;
        }
    </style>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("div").click(function() {
                $(this).animate({
                    width: "+=50",
                    height: "+=50"
                }, "slow");
            });
        });
    </script>
</head>
<body>
    <div></div>
</body>
</html>
```

3-7. 정리.

Confidential

- ✓ 순수 자바스크립트만을 이용하여 DOM 객체 구조를 처리하는 것은 쉽지 않은 작업이다.
- ✓ jQuery 메소드를 사용하면, DOM 객체를 다루는 작업을 간단하게 처리할 수 있다.
- ✓ jQuery 앤티패턴은 성능에 좋지 않는 코딩 패턴을 말합니다. 이러한 코딩을 피하여 성능을 개선한다.
- ✓ jQuery Effect를 사용하여 화면에서 발생하는 시각 효과를 구현할 수 있다.

4. jQuery Event

4. jQuery Event.

4-1. jQuery DOM Event 처리.

Confidential

✓ jQuery Event 처리

- jQuery Event로 기존 JavaScript DOM Event를 간편하게 처리, 연결 가능.
- 이벤트 핸들러를 할당, 해제할 수 있는 통합 메소드 제공.
- DOM Element의 이벤트 타입마다 여러 핸들러 할당 가능.
- click, mouseover 등과 같은 표준 이벤트 타입명을 사용.
- 핸들러의 매개변수로 이벤트 인스턴스를 사용 가능.
- 자주 사용하는 이벤트 인스턴스의 프로퍼티 등에 일관된 이름 제공.
- 하나의 API로 표준 호환 브라우저와 IE 동시 지원.

✓ jQuery Event 기능

- click 함수로 클릭 이벤트 처리.
- bind, on 함수를 이용한 모든 이벤트 처리.
- unbind, off 함수를 이용하여 이벤트 제거.

4-2. jQuery Event Binding - bind() 함수.

Confidential

- ✓ 선택된 DOM 객체의 이벤트에 지정한 핸들러를 연결하는 함수.
- ✓ 동적으로 생성한 DOM 객체에는 적용 X.
- ✓ `bind(eventType, data, listener)`
 - `eventType` : 핸들러를 할당할 이벤트 타입의 이름.
 - `data` : 핸들러 함수에서 사용할 데이터, 이벤트 인스턴스에 `data`라는 프로퍼티로 제공됨,
생략 시 두 번째 인자는 `listener`
 - `listener` : 이벤트 발생시 수행되는 핸들러 함수.

4-2. jQuery Event Binding - unbind() 함수.

Confidential

- ✓ **unbind()** 함수는 객체 이벤트에서 지정한 핸들러를 지운다.
 - `unbind(eventType, handler)`
 - `unbind(eventType)`
 - 확장 집합의 모든 엘리먼트에서 매개변수에 따라 이벤트 핸들러를 삭제.
 - 매개변수가 없을 경우 모든 핸들러를 제거.
 - **파라미터 종류**
 - `eventType` : 핸들러를 할당할 이벤트 타입의 이름.
 - `handler` : 이벤트 발생시 수행되는 핸들러 함수.

4. jQuery Event.

4-2. jQuery Event Binding - 예제.

Confidential

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jQuery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function() {
            $("#bound").bind("click", function() {
                $("#show").bind("click", function() {
                    $("p").show();
                });
                $("#hide").bind("click", function() {
                    $("p").hide();
                });
            });

            $("#unbound").bind("click", function() {
                $("#show,#hide").unbind("click");
            });
        });
    </script>
</head>
<body>
    <h2>안녕하세요</h2>
    <p>jQuery입니다.</p>
    <p>Event입니다.</p>
    <button id="show">show</button>
    <button id="hide">hide</button>
    <button id="bound">bound</button>
    <button id="unbound">unbound</button>
</body>
</html>
```

4. jQuery Event.

4-2. jQuery Event Binding - on() 함수.

Confidential

- ✓ on() 함수는 bind() 함수와 마찬가지로 DOM 객체에 이벤트 핸들러를 연결.
- ✓ delegate 방식으로 사용할 경우 현재 존재하는 DOM 객체 뿐만 아니라 동적으로 생성한 DOM 객체에도 적용 가능.
- ✓ 1.7.X 버전부터 생겨났으며 이벤트 연결에 가장 기본이 되는 함수로 **권장**하고 있다.

```
<style type="text/css">
.reverse { background: black; color: white; }
</style>
<script type="text/javascript">
$(document).ready(function() {
    $("h2").on("click", function() {
        $(this).html(function(index, html) {
            return html + "*";
        });
    });

    $("h2").on({
       mouseenter: function() {
            $(this).addClass("reverse");
        },
       mouseleave: function() {
            $(this).removeClass("reverse");
        }
    });
});
</script>
```

```
<h2>안녕하세요1.</h2>
<h2>안녕하세요2.</h2>
<h2>안녕하세요3.</h2>
```

안녕하세요1.*****

안녕하세요2.****

안녕하세요3.***

4. jQuery Event.

4-2. jQuery Event Binding - on() 함수.

Confidential

- ✓ 이벤트 바인딩에 있어서 문제는 **동적으로 생성된 DOM 객체**이다.
- ✓ 즉, 기존 이벤트 연결 방식은 **동적으로 생성된 DOM 객체**에는 적용되지 않은 문제점이 있다.
- ✓ Event Delegate 방식은 부모 DOM 객체에 이벤트를 연결한 후 이를 하위 DOM 객체에 전달하는 **방식으로 동적으로 적용 가능**.

```
$(selector).on(eventType, delegate selector, function(event) {});
```

```
$("#div1 > h2").on("click", function() {
    var origin = $(this).html();
    $("<h2>" + origin + "</h2>").appendTo("#div1");
});

$("#div2").on("click", "h2", function() {
    var origin = $(this).html();
    $("<h2>" + origin + "</h2>").appendTo("#div2");
});
```

4-03.html

싸피 강좌
싸피 강좌
싸피 과목
싸피 과목
싸피 과목
싸피 과목

4. jQuery Event.

4-2. jQuery Event Binding - off() 함수.

Confidential

- ✓ off() 함수는 on()과 반대로 DOM 객체의 이벤트를 제거.
- ✓ 선택된 DOM 객체의 특정 이벤트 또는 모든 이벤트를 제거할 수 있다.

```
$(selector).off();  
$(selector).off(eventType);  
$(selector).off(eventType, function(event) {});
```

```
<h2>안녕하세요1.</h2>  
<h2>안녕하세요2.</h2>  
<h2>안녕하세요3.</h2>  
<button id="btn1">클릭제거</button>
```

4-04.html

```
<style type="text/css">  
.reverse { background: black; color: white; }  
</style>  
<script type="text/javascript">  
$(document).ready(function() {  
    $("h2").on({  
        click: function() {  
            $(this).html(function(index, html) {  
                return html + "*";  
            });  
        },  
        mouseenter: function() {  
            $(this).addClass("reverse");  
        },  
        mouseleave: function() {  
            $(this).removeClass("reverse");  
        }  
    });  
  
    $("#btn1").click(function() {  
        $("h2").off("click");  
    });  
});  
</script>
```

4. jQuery Event.

4-2. jQuery Event Binding - Simple Event Bind.

Confidential

- ✓ jQuery는 DOM 객체에 이벤트를 간단하게 연결할 수 있는 다양한 함수를 제공.
- ✓ 예를 들어, 클릭 이벤트를 연결할 때 `on()`을 사용하는 대신, `click()` 함수를 사용할 수 있다.
- ✓ Simple Event 함수의 종류는 다음과 같다.

.blur	.focus	.focusin	.mousedown	.resize
.change	.keydown	.focusout	.mousemove	.scroll
.click	.keypress	.mouseenter	.mouseout	.select
.dblclick	.keyup	.mouseleave	.mouseover	.submit
.error	.load	.ready	.mouseup	.unload

```
$(selector).on("click", function(event) {});
```

```
$(selector).click(function(event) {});
```

4. jQuery Event.

4-2. jQuery Event Binding - one() 함수.

Confidential

- ✓ one() 함수는 이벤트를 연결하고 한번 실행 후 삭제하는 기능.
- ✓ on() -> 이벤트 실행 -> off() 의미와 동일.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
$(document).ready(function() {
    $("h2").one("click", function() {
        $(this).html(function(index, html) {
            return html + "*";
        });
    });
    </script>
</head>
<body>
    <h2>안녕하세요1.</h2>
    <h2>안녕하세요2.</h2>
    <h2>안녕하세요3.</h2>
</body>
</html>
```

안녕하세요1.*

안녕하세요2.*

안녕하세요3.*

한번만 발생.

4-3. Mouse Event

Confidential

- ✓ jQuery에서는 마우스 클릭 및 움직임과 관련된 이벤트 처리가 가능.
- ✓ mouseenter 이벤트는 경계 외부에서 내부로 접근 시 발생.
- ✓ mouseover 이벤트는 마우스가 요소 안에 들어올 때 발생하며 이벤트 버블링이 적용.

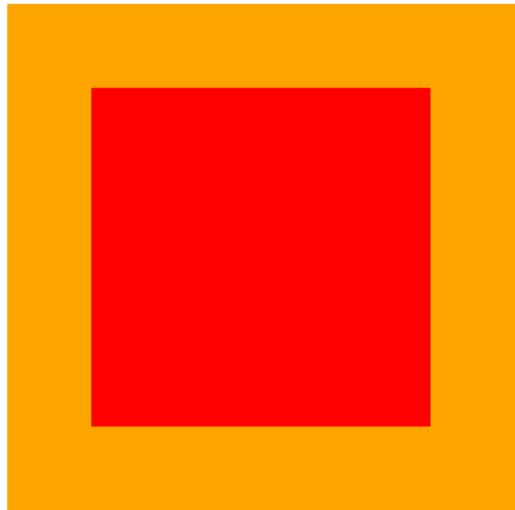
이벤트	요약
click	마우스를 클릭할 때 발생.
dblclick	마우스를 더블 클릭할 때 발생.
mousedown	마우스 버튼을 누를 때 발생.
mouseup	마우스 버튼을 뗄 때 발생.
mouseenter	마우스가 경계의 외부에서 내부로 이동할 때 발생.
mouseleave	마우스가 경계의 내부에서 외부로 이동할 때 발생.
mousemove	마우스가 움직일 때 발생.
mouseout	마우스가 요소를 벗어날 때 발생.
mouseover	마우스가 요소 안에 들어올 때 발생.

4. jQuery Event.

4-3. Mouse Event - 예제.

Confidential

4-06.html



MOUSEOVER Xpos : 316, Ypos : 126

MOUSEENTER Xpos : 316, Ypos : 126

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <style type="text/css">
        .outer {
            width: 200px; height: 200px; background: orange;
            padding: 50px; margin: 10px;
        }

        .inner {
            width: 100%; height: 100%; background: red;
        }
    </style>
    <script
        src="https://ajax.googleapis.com/ajax/Libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            // TODO: 바깥 div 태그에 mouseover와 mouseenter 핸들러를 등록 한다.
            // 핸들러에서는 이벤트가 발생한 마우스의 x,y 좌표를 출력한다.
            $(".outer").mouseover(function (event) {
                $("body").append("<h2>MOUSEOVER Xpos : " + event.pageX + ", Ypos : " + event.pageY + "</h2>");
            }).mouseenter(function (event) {
                $("body").append("<h2>MOUSEENTER Xpos : " + event.pageX + ", Ypos : " + event.pageY + "</h2>");
            });
            // 이벤트 전파를 막을 경우
            /* $(".inner").mouseover(function(event) { event.stopPropagation(); }); */
        });
    </script>
</head>
<body>
    <div class="outer">
        <div class="inner"></div>
    </div>
</body>
</html>
```

4-4. Keyboard Event

Confidential

- ✓ 키보드에서 입력할 때, 키보드 이벤트 처리 가능.

이벤트	요약
keydown	키보드를 누를 때 발생.
keypress	글자가 입력될 때 발생.
keyup	키보드를 뗄 때 발생.



4-4. Keyboard Event - 예제.

Confidential

메모장

194

안녕하세요 .

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
$(document).ready(function() {
    // TODO: textarea 태그에 키가 올라왔을 때 핸들러 등록
    // 핸들러 내용은 textarea 안의 글자수를 계산하여 출력.
    $("textarea").keyup(function() {
        var inputSize = $(this).val().length, remain = 200 - inputSize;
        $("h2").html(remain);
    });
});
</script>
</head>
<body>
    <div>
        <p>메모장</p>
        <h2>200</h2>
        <textarea rows="6" cols="70"></textarea>
    </div>
</body>
</html>
```

4-5. Window Event

Confidential

- ✓ 원도우에서 발생한 변화를 이벤트로 처리 가능.

이벤트	요약
ready	DOM 객체가 준비되면 발생.
load	원도우를 불러들일 때 발생.
unload	원도우가 닫을 때 발생.
resize	원도우 크기가 변화될 때 발생.
scroll	원도우 스크롤을 할 때 발생.
error	에러가 있을 때 발생.

4. jQuery Event.

4-5. Window Event - 예제.

Confidential

- ✓ 무한 스크롤링.

4-08.html

무한 스크롤 0

무한 스크롤 1

무한 스크롤 2

무한 스크롤 3

무한 스크롤 4

무한 스크롤 5

무한 스크롤 6

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>jquery</title>
<script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    var count = 0;
    $(window).scroll(function() {
        var scrollHeight = $(window).scrollTop() + $(window).height();
        var documentHeight = $(document).height();
        //console.log(scrollHeight + " : " + documentHeight);
        if (scrollHeight === documentHeight) {
            for (var i = 0; i < 10; i++) {
                $("<h2>무한 스크롤 " + count++ + "</h2>").appendTo("body");
            }
        });
        for (var i = 0; i < 20; i++) {
            $("<h2>무한 스크롤 " + count++ + "</h2>").appendTo("body");
        }
    });
    </script>
</head>
<body>
</body>
</html>
```

4-6. Input Event

Confidential

- ✓ 일반적으로 HTML에서 입력 양식으로 데이터를 전송하는 작업을 많이 한다.
- ✓ 이러한 입력 양식에서도 발생하는 이벤트가 존재하며 jQuery에서는 이를 지원.
- ✓ 주로 <form>, <input> 등에서 발생하는 입력 이벤트를 처리한다.

이벤트	요약
change	입력 양식의 내용이 변경할 때 발생.
focus	입력 양식에 포커스가 가면 발생.
focusin	입력 양식에 포커스가 가기 전에 발생.
focusout	입력 양식에 포커스가 사라지기 전에 발생.
blur	입력 양식에 포커스가 사라지면 발생.
select	입력 양식을 선택할 때 발생 (text, textarea 제외)
submit	submit 버튼 클릭 시 발생.
reset	reset 버튼 클릭 시 발생.

4. jQuery Event.

4-6. Input Event

Confidential

- ✓ submit event.

이름:

비밀번호:

4-09.html

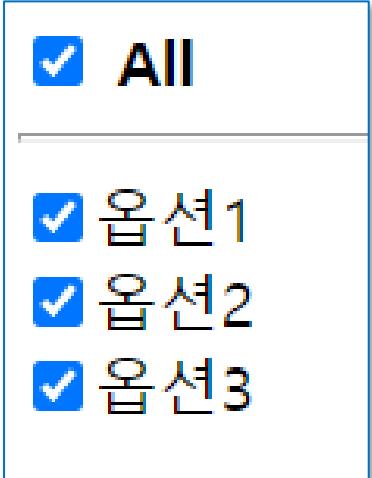
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>jquery</title>
<script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    $("#myForm").submit(function(event) {
        var name = $("#name").val(), password = $("#password").val();
        alert(name + " : " + password);
        event.preventDefault();
    });
});
</script>
<body>
<form id="myForm">
    <table>
        <tr>
            <td>이름:</td>
            <td><input type="text" name="name" id="name" /></td>
        </tr>
        <tr>
            <td>비밀번호:</td>
            <td><input type="password" name="password" id="password" /></td>
        </tr>
    </table>
    <input type="submit" value="제출" />
</form>
</body>
</html>
```

4. jQuery Event.

4-6. Input Event

Confidential

- ✓ change event1.



4-10.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
$(document).ready(function () {
    // TODO : All 체크박스를 체크/해제 하면
    // 다른 모든 체크박스도 체크/해제
    $("#allCheck").change(function () {
        if ($("#this").is(":checked")) {
            $("#checkItem").children().prop("checked", true);
        } else {
            $("#checkItem").children().prop("checked", false);
        }
    });
    </script>
</head>
<body>
    <input type="checkbox" id="allCheck" />
    <label><b>All</b></label>
    <hr />
    <div id="checkItem">
        <input type="checkbox"/><label>옵션1</label><br/>
        <input type="checkbox"/><label>옵션2</label><br/>
        <input type="checkbox"/><label>옵션3</label><br/>
    </div>
</body>
</html>
```

4. jQuery Event.

4-6. Input Event

Confidential

✓ change event2.



4-11.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery</title>
    <style type="text/css">
        div {
            width: 500px;
            height: 500px;
            background-color: orange;
        }
    </style>
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            // TODO: textArea의 텍스트가 변경되면 내용을 divArea에 복사
            $("#textArea").change(function() {
                $("#divArea").text($(this).val());
            });
        });
    </script>
</head>
<body>
    <h5>입력 :</h5>
    <textarea id="textArea" rows="5" cols="68"></textarea>
    <hr />
    <div id="divArea"></div>
</body>
</html>
```

4. jQuery Event.

4-6. Input Event

Confidential

✓ 참고 - 아코디언

텍스트1

텍스트 2

텍스트 3

```
<script type="text/javascript">
$(function() {
    $("dd:not(:first)").css("display", "none");
    $("dl dt").click(function() {
        if ($("+dd", this).css("display") == "none") {
            $(this).siblings("dd").slideUp("slow");
            $($"+dd", this).slideDown("slow");
        }
    });
});
</script>
```

```
<div id="container">
  <dl>
    <dt>텍스트1</dt>
    <dd>
      <p>텍스트 텍스트 텍스트</p>
    </dd>
    <dt>텍스트 2</dt>
    <dd>
      <p>텍스트 텍스트 텍스트</p>
    </dd>
    <dt>텍스트 3</dt>
    <dd>
      <p>텍스트 텍스트 텍스트</p>
    </dd>
  </dl>
</div>
```

4-7. 정리.

Confidential

- ✓ JavaScript는 DOM에서 발생하는 다양한 이벤트에 반응하여 원하는 작업을 수행할 수 있다.
- ✓ jQuery는 기존 JavaScript DOM Event를 간편하게 처리하고 연결할 수 있는 메소드를 제공.
- ✓ bind, delegate, live, on과 같은 함수는 이벤트 핸들러를 DOM 요소에 적용하는 jQuery함수이다.
- ✓ 계층구조를 가진 DOM 객체에서 정확한 이벤트 처리를 하려면, 기본이벤트와 이벤트 전파방식 이해가 필요.

내일
방송에서
만나요!

삼성 청년 SW 아카데미