

jQuery Basic



Bok, Jong Soon
javaexpert@nate.com
<https://github.com/swacademy/jQuery>

Web의 현황

- Web Page와 사용자간의 원활한 소통 필요
- Web 2.0, Ajax, RWA, RIA
- 새로운 UX, UI 의 필요
- 많은 Code, 복잡한 Code, Cross Browsing

JavaScript Library

- 하나 이상의 Subroutine이나 자주 재 사용 가능한 함수들이 모여 있는 파일.
- 기능과 UI 가 많아질수록 복잡도가 증가한다.
 - 재 사용될 수 있는 기능들을 라이브러리화 한다.
- 자주 사용되는 기능들을 모아 놓는다.
 - Prototype, jQuery, YUI, Dojo, Ext JS, etc.
- 개발해야 되는 제품의 특성에 따라 사용한다.
 - 직접 만들거나 공개 라이브러리를 사용한다.
 - 효율성(크기, 시간, 기능 등)을 고려한다.

JavaScript Library (Cont.)

■ 특징

- Cross Browser Support
- Easy Ajax working
- DOM 관련 selector, event handler, function support
- 복잡한 구조를 가진 UI를 손쉽게 구현
- UI 구현에 필요한 대부분의 component API 제공

JavaScript Library (Cont.)

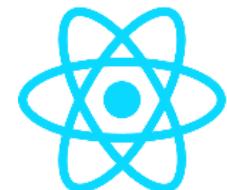
■ 장점

- 이해하기 쉽다
- 간결한 코드를 구사할 수 있다.
- 생산성이 향상된다.
- 중복작업 감소

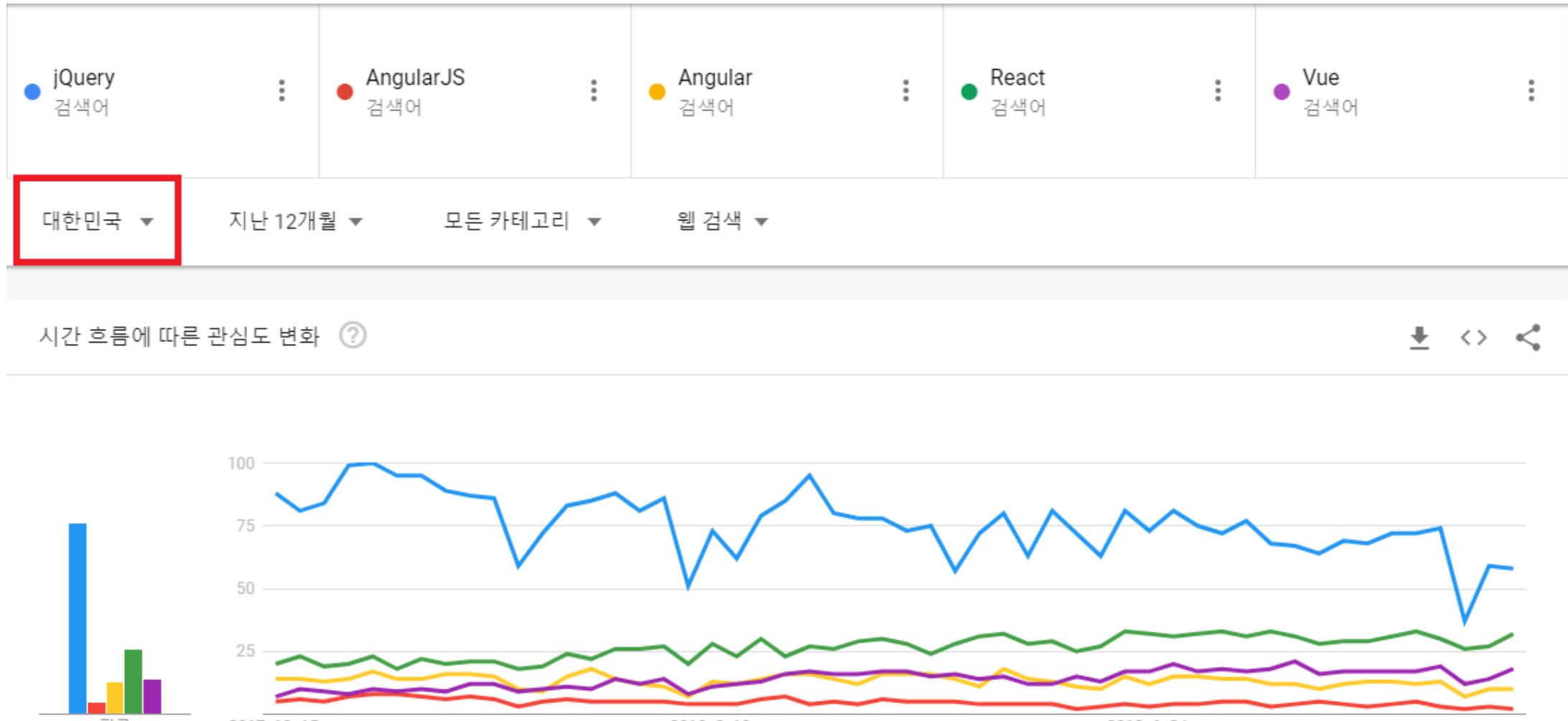
■ 단점

- 별도로 library 사용법 학습 및 이해 필요하다.
- Size가 큰 file 들을 추가로 load 해야 하는 부담이 있다.

JavaScript Library (Cont.)

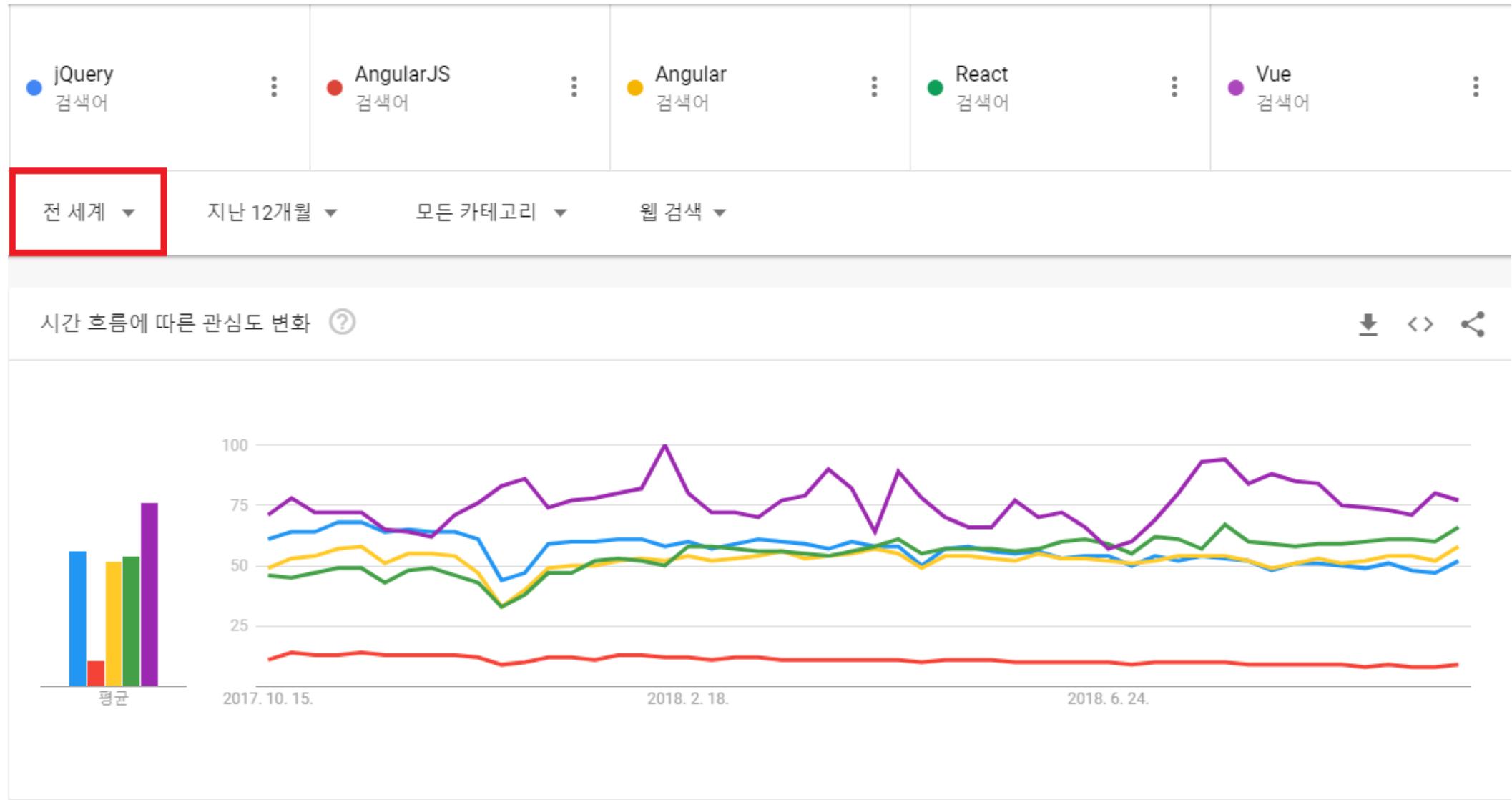


JavaScript Library (Cont.)

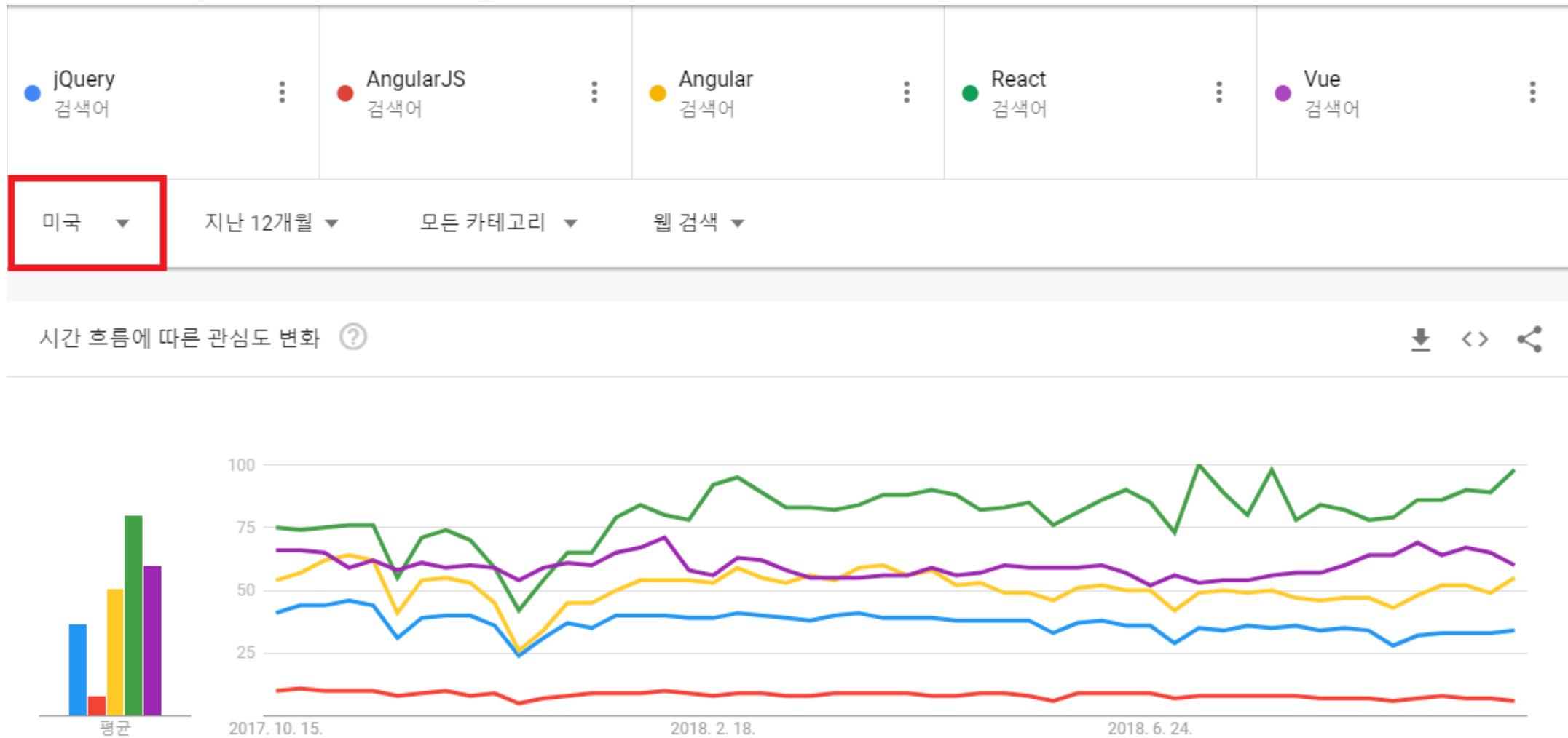


<https://trends.google.com/trends/explore?geo=KR&q=jQuery,AngularJS,Angular,React,Vue>

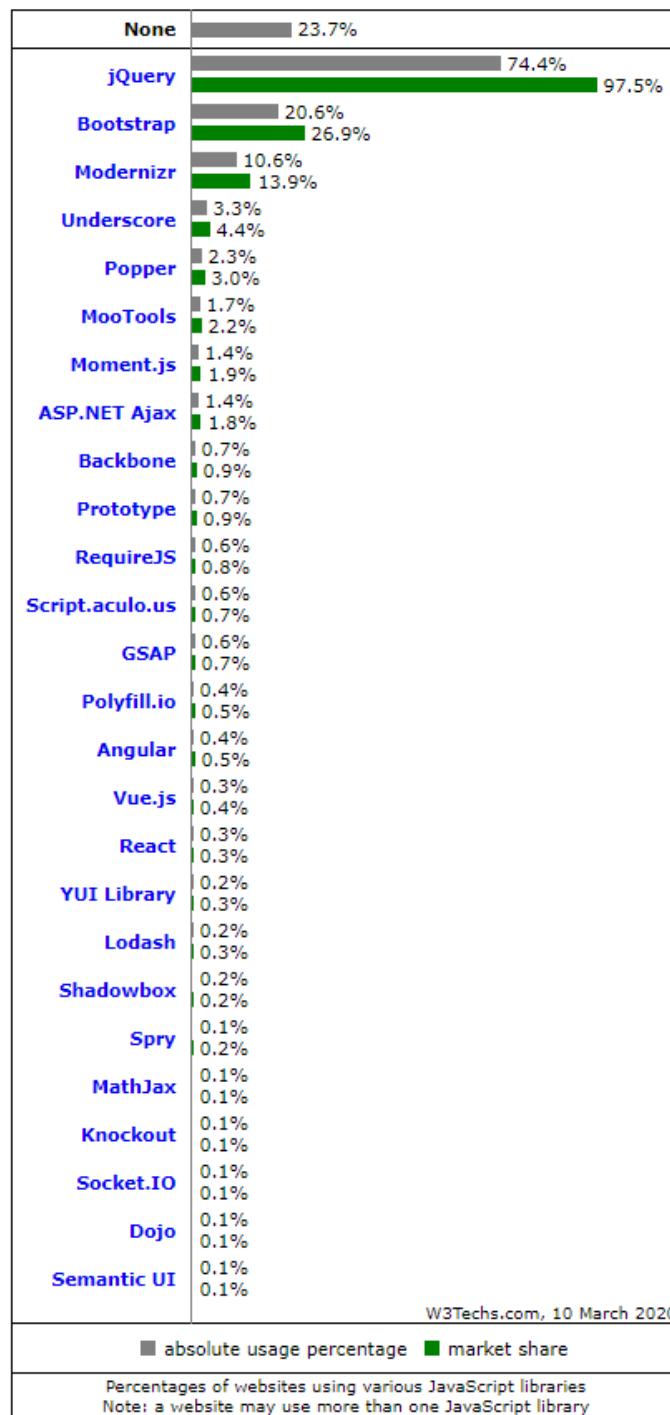
JavaScript Library (Cont.)



JavaScript Library (Cont.)



JavaScript Library (Cont.)



https://w3techs.com/technologies/overview/javascript_library

JavaScript Library (Cont.)

Market share trends for JavaScript libraries

This report shows the market share trends for JavaScript libraries since March 2019.

	2019 1 Mar	2019 1 Apr	2019 1 May	2019 1 Jun	2019 1 Jul	2019 1 Aug	2019 1 Sep	2019 1 Oct	2019 1 Nov	2019 1 Dec	2020 1 Jan	2020 1 Feb	2020 1 Mar	2020 10 Mar
jQuery	97.3%	97.3%	97.3%	97.4%	97.4%	97.4%	97.5%	97.5%	97.5%	97.3%	97.4%	97.4%	97.4%	97.5%
Bootstrap	24.5%	24.6%	24.7%	24.8%	25.0%	25.4%	25.7%	25.9%	26.0%	26.2%	26.4%	26.7%	26.9%	26.9%
Modernizr	14.9%	14.8%	14.7%	14.7%	14.7%	14.5%	14.4%	14.3%	14.2%	14.1%	14.1%	14.0%	13.9%	13.9%
Underscore	3.9%	4.0%	4.0%	4.1%	4.1%	4.0%	4.0%	4.1%	4.1%	4.2%	4.2%	4.3%	4.4%	4.4%
Popper	1.3%	1.4%	1.5%	1.7%	1.8%	1.9%	2.0%	2.2%	2.3%	2.4%	2.6%	2.8%	2.9%	3.0%
MooTools	3.0%	2.9%	2.8%	2.8%	2.8%	2.7%	2.6%	2.6%	2.6%	2.5%	2.4%	2.3%	2.3%	2.2%
Moment.js	1.7%	1.8%	1.7%	1.7%	1.8%	2.0%	2.0%	1.8%	1.8%	1.8%	1.9%	1.9%	1.9%	1.9%
ASP.NET Ajax	2.2%	2.2%	2.1%	2.1%	2.1%	2.0%	2.0%	1.9%	1.9%	1.9%	1.9%	1.8%	1.8%	1.8%
Backbone	1.3%	1.3%	1.3%	1.3%	1.3%	1.1%	1.0%	1.0%	1.0%	1.0%	1.0%	0.9%	0.9%	0.9%
Prototype	1.4%	1.4%	1.4%	1.3%	1.3%	1.2%	1.2%	1.2%	1.1%	1.1%	1.0%	1.0%	0.9%	0.9%
RequireJS										0.9%	0.9%	0.8%	0.8%	0.8%
Script.aculo.us	1.2%	1.2%	1.1%	1.1%	1.1%	1.0%	1.0%	0.9%	0.9%	0.9%	0.8%	0.8%	0.8%	0.7%
GSAP	0.6%	0.6%	0.6%	0.6%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%
Polyfill.io	0.3%	0.3%	0.3%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.5%	0.5%	0.5%	0.5%	0.5%
Angular	0.6%	0.6%	0.6%	0.6%	0.6%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%	0.5%
Vue.js	0.2%	0.2%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.4%	0.4%	0.4%	0.4%	0.4%
React	0.3%	0.2%	0.2%	0.3%	0.3%	0.5%	0.5%	0.4%	0.4%	0.4%	0.3%	0.3%	0.3%	0.3%
YUI Library	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.4%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%
Lodash	0.3%	0.3%	0.3%	0.3%	0.3%	0.5%	0.4%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%
Shadowbox	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%
Spry	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%
MathJax	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Knockout	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Socket.IO	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Dojo	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Semantic UI	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%
Ext JS	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	<0.1%	<0.1%

jQuery – <https://jquery.com>

The screenshot shows the official jQuery website. At the top, there's a dark header bar with social media icons (Twitter, Facebook, GitHub) and navigation links: Plugins, Contribute, Events, Support, and JS Foundation. Below the header is a large blue banner featuring the jQuery logo and the tagline "write less, do more.". To the right of the logo is a heart icon with a white swirl inside, and text that reads "Your donations help fund the continued development and growth of jQuery." A yellow button labeled "SUPPORT THE PROJECT" is also present. The main content area has a dark background with four key features highlighted: "Lightweight Footprint" (with an icon of a small cube), "CSS3 Compliant" (with an icon of a stylized '3'), "Cross-Browser" (with an icon of a globe with arrows), and a large orange "Download jQuery v3.3.1" button. Below these features, there's a section titled "What is jQuery?" which describes the library as fast, small, and feature-rich, making tasks like document traversal and manipulation easier. To the right, there's a "Resources" sidebar with links to the Core API Documentation, Learning Center, and Blog.

Plugins Contribute Events Support JS Foundation

jQuery
write less, do more.

Your donations help fund the continued development and growth of jQuery.

SUPPORT THE PROJECT

Download API Documentation Blog Plugins Browser Support Search

Lightweight Footprint

Only 30kB minified and zipped. Can also be included as an AMD module

CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser

Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more

Download jQuery
v3.3.1

The 1.x and 2.x branches no longer receive patches.

[View Source on GitHub →](#)
[How jQuery Works →](#)

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)

jQuery (Cont.)

- Write less, do more.
- 2005, John Resig
- 2006, First Release
- 문법이 간결하다.
- 사용하기 쉽다.
- 빠르게 배울 수 있다.
- 다른 library 들과 충돌을 일으키지 않는다.
- 다양한 plug-in 들이 존재한다.
- Browser 호환성 문제를 해결해 준다.
- MIT & GPL License.

Installation

■ Installation Issues

- jQuery 1.x vs jQuery 2.x vs jQuery 3.x
- Compressed(Product) vs uncompressed(Development)
- Downloads js file vs CDN

■ <http://jquery.com/download/>

■ <https://cdnjs.com/libraries/jquery>

```
<script src="jquery.js"></script>
<script>
    //your script goes here.
</script>
```

Installation (Cont.)

■ CDN(Content Delivery Network)

- jQuery.com CDN
 - <https://code.jquery.com/>
- Google CDN
 - <https://developers.google.com/speed/libraries#jquery>
- Microsoft CDN
 - https://docs.microsoft.com/en-us/aspnet/ajax/cdn/overview#jQuery_Releases_on_the_CDN_0
- CDNJS CDN
 - <https://cdnjs.com/libraries/jquery/>
- jsDelivr CDN
 - <https://www.jsdelivr.com/package/npm/jquery>

Recommend Sites

- <http://api.jquery.com/>
- <http://jqapi.com/>
- <http://www.w3schools.com/jquery/default.asp>

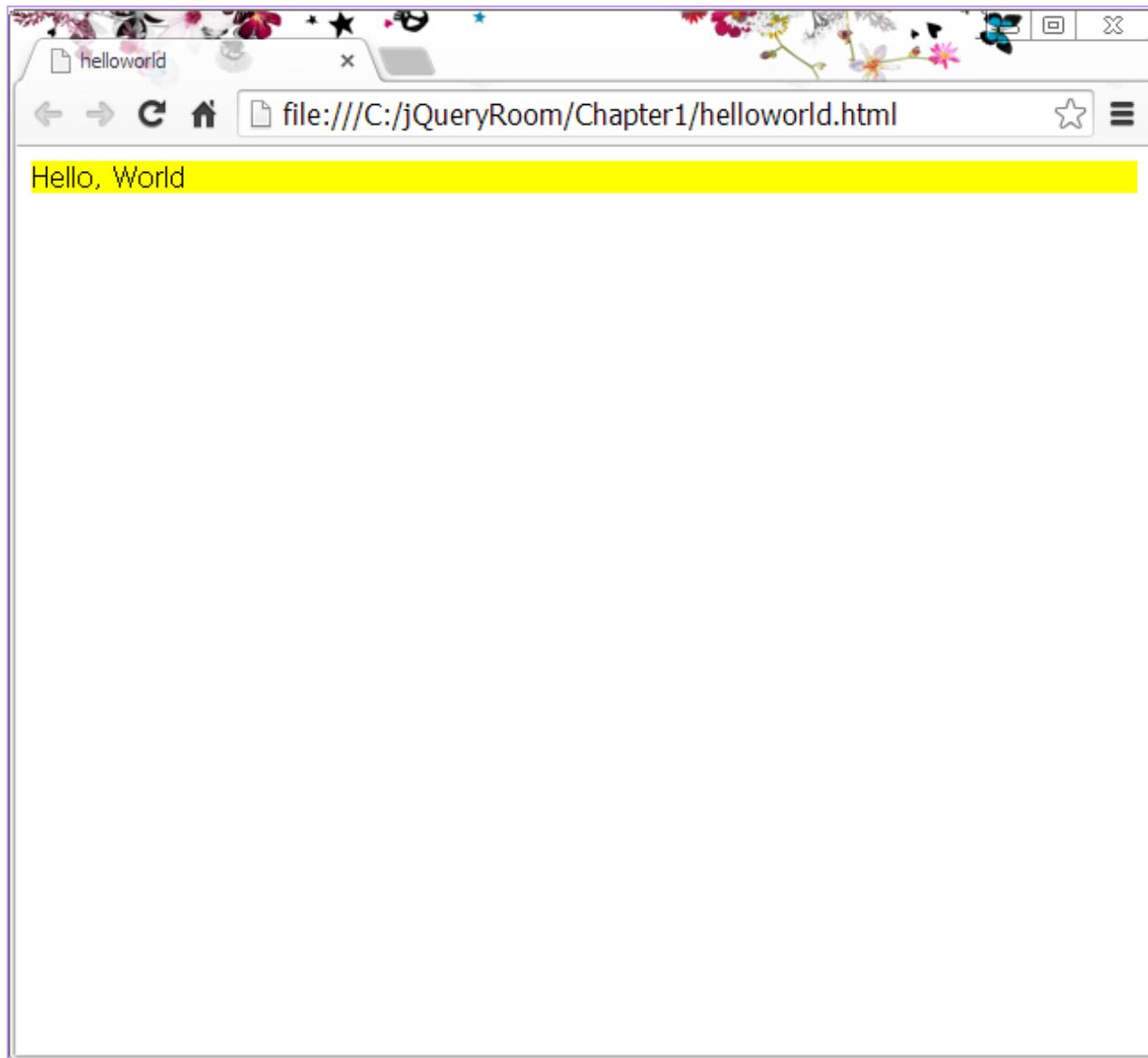
Lab : jQuery Installation

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Sublime Text, Notepad++, Visual Studio Code
- Files
 - helloworld.html

Lab : helloworld.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title> helloworld </title>
6     <script src="js/jquery-1.11.1.js"></script>
7     <script>
8       $(document).ready(
9         function(){
10           $('div').addClass('bg');
11         }
12       );
13     </script>
14     <style>
15       .bg {background-color: yellow}
16     </style>
17   </head>
18   <body>
19     <div id="helloworld">
20       Hello, World
21     </div>
22   </body>
23 </html>
```

Lab : Result



jQuery's 9 Categories

Core

Selectors

Attributes/CSS

Manipulation

Traversing

Events

Effects

UI

Ajax

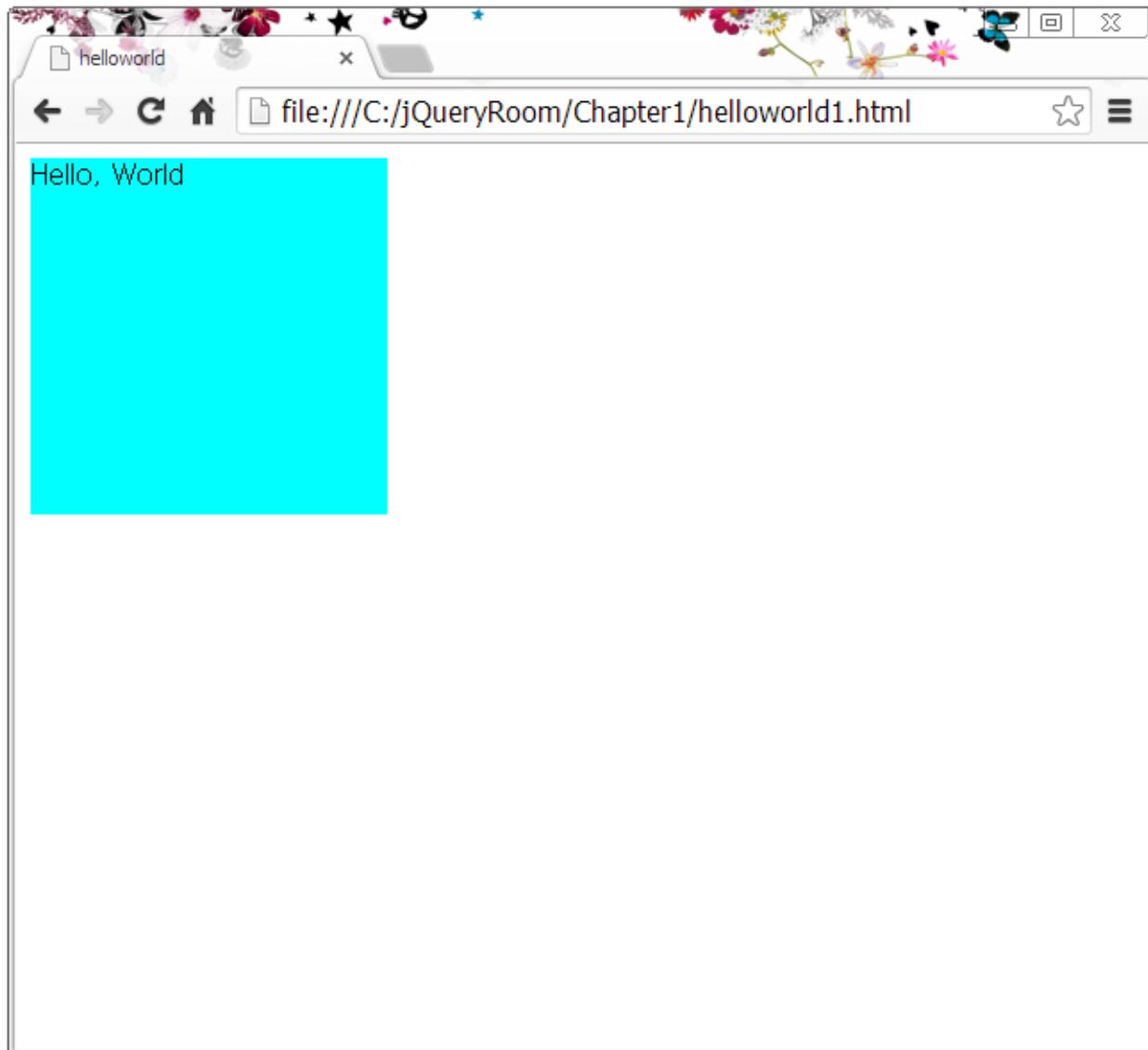
Lab : Simple Code with jQuery

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Sublime Text, Notepad++, Visual Studio Code
- Files
 - helloworld1.html

Lab : helloworld1.html

```
7 <script>
8 $(function(){
9     $('div').click(function(e){
10         $(this).text("Hello, World");
11     });
12 });
13 </script>
14 <style>
15 div {
16     width:200px; height:200px; background-color : #0FF;
17 }
18 </style>
19 </head>
20 <body>
21     <div></div>
22 </body>
```

Lab : Result



Core API



jQuery ready ()

■ window.onload = function(){}

- HTML DOM loading 후 실행할 callback 함수

■ jQuery(function())

- HTML DOM loading 후 images loading 전 실행할 callback 함수

기본 문법

```
$(document).ready(function() {  
    // DOM이 모두 로드된 후 실행할 코드 입력  
});
```

축약 문법

```
$(function() {  
    // DOM이 모두 로드된 후 실행할 코드 입력  
});
```

ready handler

- Script 는 DOM 요소가 준비될 때까지 실행을 기다린다.
 - **window.onload** 와 동일
 - 이미지를 포함하여 모두 load 될 때까지 기다린다.
 - 따라서 script 실행이 늦어진다.
- ready handler 는 DOM tree 가 만들어질 때까지만 기다린다.
 - jQuery 는 개발자가 이러한 모든 작업에 대해 신경쓰지 않도록 한다.

ready handler (Cont.)

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Hello, jQuery</title>
7      <script>
8          | alert("Hello, World at the head tag.");
9      </script>
10     </head>
11     <body>
12         <script>
13             | alert("Hello, World at body top.");
14         </script>
15         <p>Hello, World</p>
16         <p>Hello, World</p>
17         <script>
18             | alert("Hello, World at body middle.");
19         </script>
20         <p>Hello, World</p>
21         <p>Hello, World</p>
22         <script>
23             | alert("Hello, World at body bottom.");
24         </script>
25     </body>
26 </html>
```

ready handler (Cont.)

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Hello, jQuery</title>
7      <script src='js/jquery-1.12.4.js'></script>
8      <script>
9          | alert("Hello, World at the head tag.");
10     </script>
11 </head>
12 <body>
13     <script>
14         | alert("Hello, World at body top.");
15     </script>
16     <p>Hello, World</p>
17     <p>Hello, World</p>
18     <script>
19         $(document).ready(function(){
20             | alert("Hello, World at body middle.");
21         });
22     </script>
23     <p>Hello, World</p>
24     <p>Hello, World</p>
25     <script>
26         | alert("Hello, World at body bottom.");
27     </script>
28 </body>
29 </html>
```

ready handler (Cont.)

```
6 <script src="js/jquery-1.11.1.js"></script>
7 <script>
8 /* $(document).ready(
9   function(){
10     alert("Hello, World");
11   }
12 ); 첫번째 방법 */
13 /* $.ready(
14   function(){
15     alert("Hello, World");
16   }
17 ); 두번째 방법 */
18 $(
19   function(){
20     alert("Hello, World");
21   }
22 ); /* 세번째 방법 */
23 </script>
```

`$(expr, context)`

- Accepts a string containing a CSS or basic XPath selector.
- The core functionality of jQuery.
- Everything in jQuery is based upon this method.
- Finds all matching elements.
- By default, looks for DOM elements within the context of the current HTML document.

\$(expr, context) (Cont.)

■ Parameters

- **expr** : (String) : An expression to search with
- **context** : (Element | jQuery) : (Optional) A DOM element, Document or jQuery to use as context

■ Returns

- **jQuery**

\$(expr, context) (Cont.)

■ Examples

- **\$(“div > p”)**
 - Finds all **p** elements that are children of a **div** element.
- **\$(“input:radio”, document.forms[0])**
 - Searches for all inputs of type **radio** within the first **form** in the document.
- **\$(document.body).css(“background”, “black”)**
 - Set the background color of the page to black.

■ See Also

- **\$(element)**

Lab : \$(expr, context)

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - core.html

Lab : core.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(
        function(){
            $("div > p").css( "border", "1px solid red" );
        }
    );
</script>
</head>
<body>
    <p>one</p>
    <div><p>two</p></div>
    <p>three</p>
```

Lab : Result

one

two

three

`$(html)`

- Create DOM elements on-the-fly from the provided String of raw HTML.
- Parameters
 - `html` : (String) : A string of HTML to create on the fly.
- Returns
 - `jQuery`
- Example
 - `$(<div><p>Hello</p></div>) .appendTo("#body")`
 - Creates a `div` element dynamically, and appends it to the element with the `ID` of body.

Lab : \$(html)

■ Web Browsers

- IE11, Firefox, Google Chrome, Opera, Safari

■ Text Editors

- Notepad++ or Editplus

■ Files

- Core_html.html

Lab : core_html.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(
        function(){
            $( "<div/>" , {
                "class": "test",
                text: "Click me!",
                click: function() {
                    $( this ).toggleClass( "test" );
                }
            })
            .appendTo( "body" );
        }
    );
</script>
<style type="text/css">
    .test {border : 2px dotted red}
</style>
</head>
<body>

</body>
```

Lab : Result

Click me!

`$(elems)`

- Wrap jQuery functionality around a single or multiple DOM Element(s).
- Parameters
 - **elems** : (Element | Array) : DOM element(s) to be encapsulated by a jQuery object.
- Returns
 - **jQuery**

\$(elems) (Cont.)

■ Example

- **\$ (document) .find("div > p")**
 - Same as **\$ ("div > p")**.
- **\$ (document.body) .background("black")**
 - Sets the background color of the page to **black**.
- **\$ (myForm.elements) .hide ()**
 - Hides all the **input** elements within a form.

`$(fn)`

- A shorthand for `$ (document) . ready ()`
- Allows to bind a function to be executed when the DOM document has finished loading.
- Parameters
 - `fn` : (Function) : The function to execute when the DOM is ready.
- Returns
 - `jQuery`
- Example
 - `$(function() { // Document is ready }) ;`

Lab : \$(fn)

■ Web Browsers

- IE11, Firefox, Google Chrome, Opera, Safari

■ Text Editors

- Notepad++ or Editplus

■ Files

- core_fn.html

Lab : core_fn.html

```
<script>
$(function(){
    $("p:only-child").css("background-color","yellow");
})
</script>
</head>
<body>
<div style="border:1px solid;">
    <p>The first child.</p>
    <p>The last child.</p>
</div><br>

<div style="border:1px solid;">
    <p>The only child.</p>
</div><br>

<div style="border:1px solid;">
    <span>The first child.</span>
    <p>The last child.</p>
</div><br>

</body>
```

Lab : Result

The first child.

The last child.

The only child.

The first child.

The last child.

\$.holdReady()

- Holds or releases the execution of jQuery's ready event.
- Returns
 - undefined
- Example

```
$.holdReady( true );  
$.getScript( "myplugin.js", function() {  
    $.holdReady( false );  
});
```

Lab : `$.holdReady()`

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - core_holdready.html

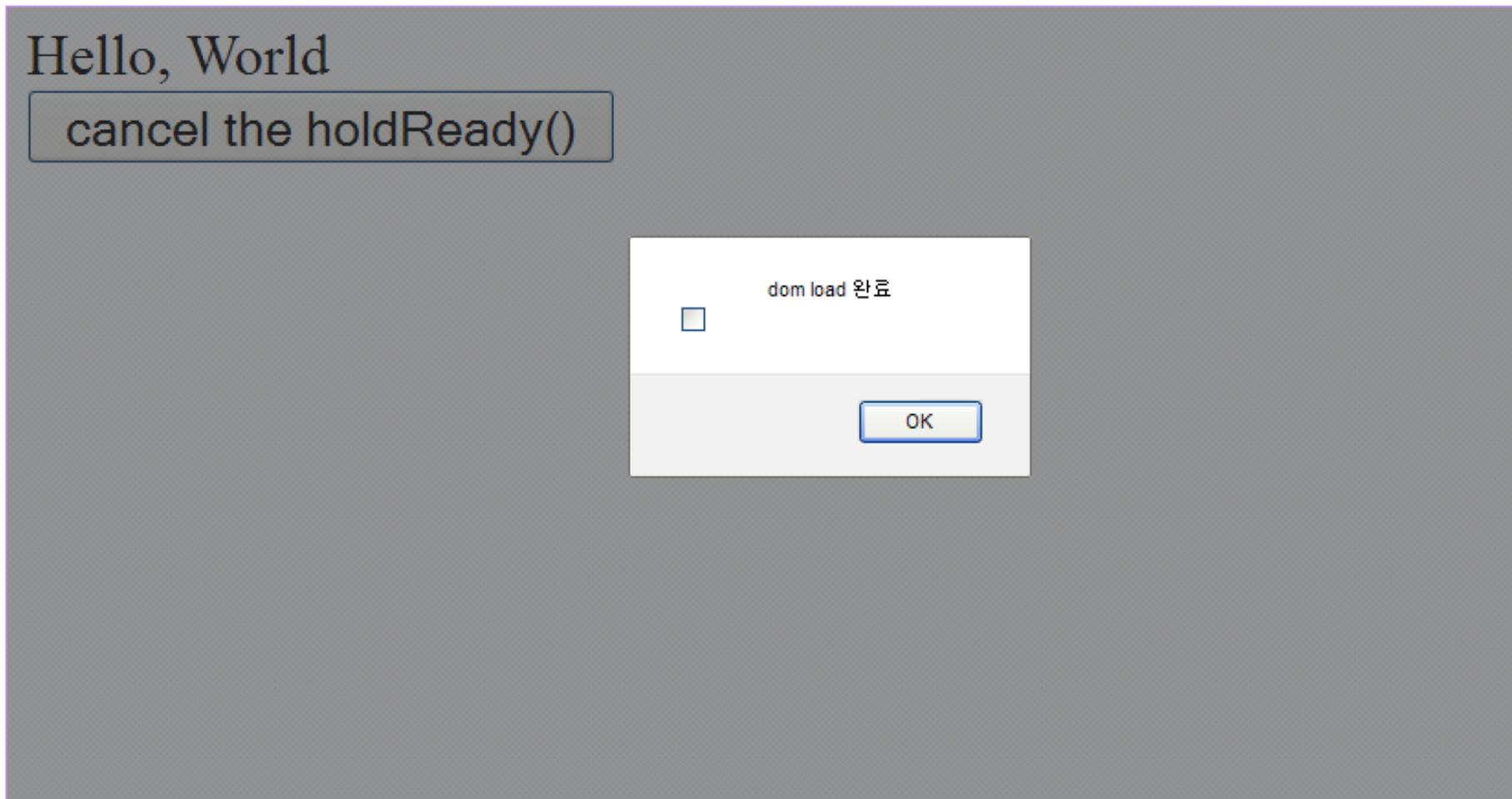
Lab : core_holdready.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $.holdReady(true);
    alert('hold ready');
    alert($('#text').html());

    $(document).ready(function(){
        alert('dom load 완료');
        alert($('#text').html());
    });

    function test(){
        $.holdReady(false);
        alert('.holdReady 끝');
    }
</script>
</head>
<body>
    <div id="text">Hello, World</div>
    <input type="button" value="cancel the holdReady()" onclick="test();"/>
</body>
```

Lab : Result



\$.noConflict()

- Relinquish jQuery's control of the \$ variable.
- Returns
 - Object
- Example
 - **\$.noConflict();**

Lab : `$.noConflict()`

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - core_noconflict.html

Lab : core_noconflict.html

```
<script src="js/jquery-1.11.1.js"></script>
</head>
<body>
  <div id="log">
    <h3>Before $.noConflict(true)</h3>
  </div>
  <script src="js/jquery-1.8.3.js"></script>
  <script>
    var $log = $( "#log" );

    $log.append( "2nd loaded jQuery version ($): " + $.fn.jquery + "<br>" );

    // Restore globally scoped jQuery variables to the first version loaded
    // (the newer version)

    jq162 = jQuery.noConflict( true );

    $log.append( "<h3>After $.noConflict(true)</h3>" );
    $log.append( "1st loaded jQuery version ($): " + $.fn.jquery + "<br>" );
    $log.append( "2nd loaded jQuery version (jq162): " + jq162.fn.jquery + "<br>" );
  </script>
</body>
```

Lab : Result

Before `$.noConflict(true)`

2nd loaded jQuery version (\$): 1.8.3

After `$.noConflict(true)`

1st loaded jQuery version (\$): 1.11.1

2nd loaded jQuery version (jq162): 1.8.3

Selectors



CSS Selectors

Pattern	Meaning	CSS Level
<code>*</code>	Any element	2
<code>E</code>	An element of type E	1
<code>E[foo]</code>	An E element with a "foo" attribute	2
<code>E[foo="bar"]</code>	An E element whose "foo" attribute value is exactly equal to "bar"	2
<code>E[foo~=“bar”]</code>	An E element whose "foo" attribute value is a list of whitespace-separated values, one of which is exactly equal to "bar"	2
<code>E[foo^="bar"]</code>	An E element whose "foo" attribute value begins exactly with the string "bar"	3
<code>E[foo\$="bar"]</code>	An E element whose "foo" attribute value ends exactly with the string "bar"	3
<code>E[foo*="bar"]</code>	An E element whose "foo" attribute value contains the substring "bar"	3
<code>E[foo =“en”]</code>	An E element whose "foo" attribute has a hyphen-separated list of values beginning (from the left) with "en"	2

CSS Selectors (Cont.)

Pattern	Meaning	CSS Level
<code>E:root</code>	An E element, root of the document	3
<code>E:nth-child(n)</code>	An E element, the n-th child of its parent	3
<code>E:nth-last-child(n)</code>	An E element, the n-th child of its parent, counting from the last one	3
<code>E:nth-of-type(n)</code>	An E element, the n-th sibling of its type	3
<code>E:nth-last-of-type(n)</code>	An E element, the n-th sibling of its type, counting from the last one	3
<code>E:first-child</code>	An E element, first child of its parent	2
<code>E:last-child</code>	An E element, last child of its parent	3
<code>E:first-of-type</code>	An E element, first sibling of its type	3
<code>E:last-of-type</code>	An E element, last sibling of its type	3
<code>E:only-child</code>	An E element, only child of its parent	3
<code>E:only-of-type</code>	An E element, only sibling of its type	3
<code>E:empty</code>	An E element that has no children (including text nodes)	3

CSS Selectors (Cont.)

Pattern	Meaning	CSS Level
E:<code>link</code>	An E element being the source anchor of a hyperlink of which the target is not yet visited (<code>:link</code>) or already visited (<code>:visited</code>)	1
E:<code>visited</code>		
E:<code>active</code>	An E element during certain user actions	1 and 2
E:<code>hover</code>		
E:<code>focus</code>		
E:<code>target</code>	An E element being the target of the referring URI	3
E:<code>lang(fr)</code>	An element of type E in language "fr" (the document language specifies how language is determined)	2
E:<code>enabled</code>	An user interface element E which is enabled or disabled	3
E:<code>disabled</code>		
E:<code>checked</code>	An user interface element E which is checked (for instance a radio-button or checkbox)	3

CSS Selectors (Cont.)

Pattern	Meaning	CSS Level
E:first-line	The first formatted line of an E element	1
E:first-letter	The first formatted letter of an E element	1
E:before	Generated content before an E element	2
E:after	Generated content after an E element	2
E:warning	An E element whose class is "warning" (the document language specifies how class is determined).	1
E#myid	An E element with ID equal to "myid".	1
E:not(s)	An E element that does not match simple selector s	3
E F	An F element descendant of an E element	1
E > F	An F element child of an E element	2
E + F	An F element immediately preceded by an E element	2
E ~ F	An F element preceded by an E element	3

Basic

■ CSS selector의 대부분을 지원한다.

- All selector ("*****")
- Class selector ("**.class**")
- Element selector ("**element**")
- ID selector ("**#id**")
- Multiple selector ("**selector1, selector2, selectorN**")

Basic (Cont.)

■ 단일 요소 선택

- **\$("p")** : 모든 <p> 요소 선택
- **\$('a')** : 모든 <a> 요소 선택
→ "", "" 중 하나 사용

■ 다중 요소 선택

- **\$('p, span, a')** : 모든 <p>, , <a> 요소 선택
→ 다중 요소 선택 시 콤마(,) 사용으로 구분

Basic (Cont.)

- ID가 지정된 요소는 # 기호로 선택
 - `$("#ID")`
→ HTML 은 하나의 페이지에 단 하나의 ID 요소만 가능
- Class 가 지정된 요소는 . 기호로 선택
 - `('.className')`
→ 한 번에 여러 개의 클래스 요소 선택 가능

Lab : Basic Selectors

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - selector.html

Lab : selector.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(
    function(){
        $('li > u').addClass('bg');
        $('#myFavorites > li').addClass('horizontal');
        jQuery('#jquery').click(function(){alert('jQuery');});
        $('.spring').click(function(){alert('Spring Framework');});
    }
);
</script>
<style type="text/css">
.bg { background-color:yellow;}
.horizontal { float:left; list-style:none; margin:10px;}
</style>
</head>
<body>
<h3>관심사항</h3>
<ul id="myFavorites">
    <li><u>Java</u>
    <li>Oracle
    <li class="spring">Spring
    <li id="jquery">j<u>Q</u>uery
</ul>
</body>
```

Lab : Result

관심사항

Java

Oracle

Spring

jQuery

Attribute

- Has Attribute selector [name]
- Equals selector [name $=$ "value"]
- Not equal selector [name $!=$ "value"]
- Starts with selector [name $^=$ "value"]
- Ends with selector [name $$=$ "value"]
- Multiple selector [name="value"][name2="value2"]
- Contains selector [name $*=$ "value"]
- Contains prefix selector [name $|=$ "value"]
- Contains word selector [name $\sim=$ "value"]

Lab : Attributes

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - selector1.html

Lab : selector1.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(
  function(){
    $("a[href*='apache']").addClass("red");
    $("a[href$='html']").addClass("green");
    $("a[href^='mailto:']").addClass("gray");
  }
);
</script>
<style type="text/css">
.red {background-color:red}
.green {background-color:green}
.gray{background-color:gray}
</style>
</head>
<body>
  <a href="tomcat.apache.com">아파치 톰캣</a>
  <a href="www.naver.com/index.html">네이버</a>
  <a href="mailto:aaa@bbb.com">이메일</a>
</body>
```

Lab : Result

[아파치 톰캣](#) [네이버](#) [이메일](#)

Basic Filter

■ **:first** selector, **:last** selector

- 선택된 요소 중 처음, 마지막 요소 선택

■ **:even** selector, **:odd** selector

- 짝수, 홀수번째 요소 선택

■ **:eq(n)** selector

- n번째 요소 선택

Lab : Even, Odd

- Web Browsers

- IE11, Firefox, Google Chrome, Opera, Safari

- Text Editors

- Notepad++ or Editplus

- Files

- EvenOdd.html

Lab : EvenOdd.html

```
<style>
    .redColor{background-color:red}
    .blueColor{background-color:blue}
</style>
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(
        function(){
            $('tr:odd').addClass('redColor');
            $('table tr:even').addClass('blueColor');
        }
    );
</script>
</head>
<body>
    <table>
        <tr>
            <td>Redhat CentOS Server</td>
        </tr>
        <tr>
            <td>Microsoft Windows Server</td>
        </tr>
        <tr>
            <td>Canonical Linux Ubuntu</td>
        </tr>
        <tr>
            <td>Linux Mint</td>
        </tr>
    </table>
</body>
```

Lab : Result

Redhat CentOS Server

Microsoft Windows Server

Canonical Linux Ubuntu

Linux Mint

Basic Filter (Cont.)

- **:lt(n)** selector, **:gt(n)** selector
 - n 보다 작은, n 보다 큰 요소 선택
- **:header** selector
 - 제목 요소 선택
- **:not()** selector
 - 선택자에 해당되지 않는 요소 선택
- **:animated** selector
 - 현재 움직이고 있는 요소 선택

Child Filter

- :nth-child() selector
 - 부모 요소의 n 번째 자식 요소 선택

Content Filter

■ **:contains(str)** selector

- 특정 텍스트를 포함하는 요소 선택

■ **:empty** selector

- 빈 요소 선택

■ **:parent** selector

- 자식이 있는 부모 요소 선택

■ **:has()** selector

- Selector에 해당하는 요소를 포함하고 있는 요소 선택

Lab : Contains

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - contains.html

Lab : contains.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(
        function(){
            $('td:contains("Boston")').addClass('redColor');
        }
    );
</script>
<style type="text/css">
    .redColor { color:red; }
</style>
</head>
<body>
    <table>
        <tr>
            <td>Boston</td>
        </tr>
        <tr>
            <td>London</td>
        </tr>
    </table>
</body>
```

Lab : Result

Boston

London

Form

- **:input, :checkbox, :radio**
- **:text, :password, :file**
- **:button, :submit, :image, :reset**
- **:focus**
- **:checked**
- **:selected**
- **:enabled, :disabled**

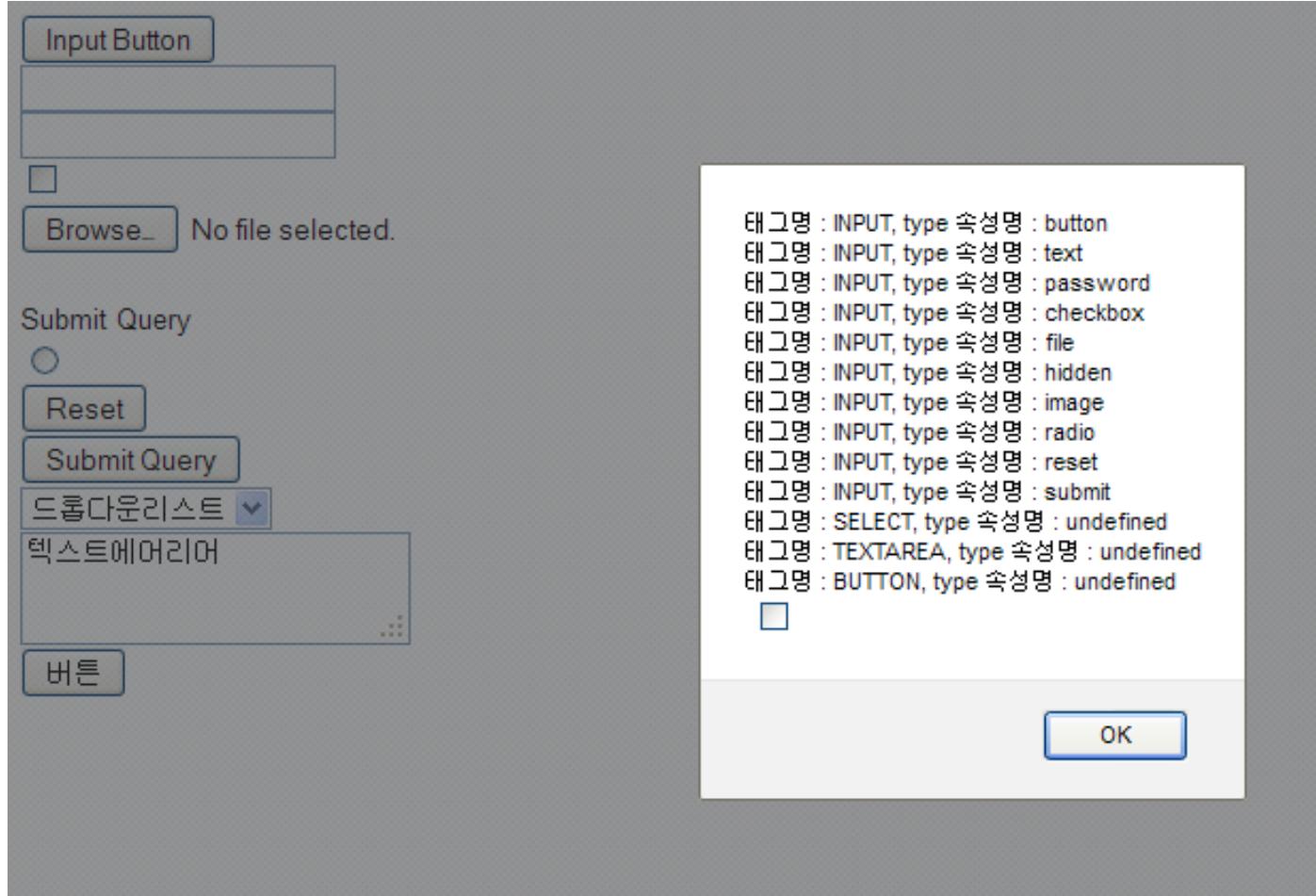
Lab : Input form

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - Input_form.html

Lab : input_form.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(
    function(){
        var result = "";
        alert($('input').size());
        $('input').each(function(index){
            result += "태그명 : " + this.tagName +
                      ", type 속성명 : " + $(this).attr('type') +
                      "\n";
        });
        alert(result);
    }
);
</script>
</head>
<body>
<input type="button" value="Input Button"/><br />
<input type="text"/><br />
<input type="password"/><br />
<input type="checkbox"/><br />
<input type="file"/><br />
<input type="hidden"/><br />
<input type="image"/><br />
<input type="radio"/><br />
<input type="reset"/><br />
<input type="submit"/><br />
<select><option>드롭다운리스트</option></select><br />
<textarea>텍스트에어리어</textarea><br />
<button>버튼</button><br />
</body>
```

Lab : Result



Hierarchy

- Child selector ("parent > child")
 - 부모의 자식 요소 선택
- Descendant selector ("ancestor descendant")
 - 하위 요소를 모두 선택
- Next Adjacent selector ("prev + next")
 - prev 다음에 인접한 next 요소 선택
- Next siblings selector ("prev ~ siblings")
 - prev 이후에 나오는 형제 요소들을 선택

Visibility Filter

■ **:visible** selector

- 보이는 요소 선택

■ **:hidden** selector

- 보이지 않는 요소 선택

Attributes / CSS



.addClass()

- Adds the specified class(es) to each of the set of matched elements.
- Parameters
 - **class** : (String) : A CSS class to add to the elements.
- Returns
 - **jQuery**
- Example
 - `$(“p”).addClass(“selected”)`

Lab : .addClass()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - addClass.html

Lab : addClass.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(
  function(){
    $("body p:nth-child(2)").addClass( "selected" );
  }
);
</script>
<style>
p {
  margin: 8px;
  font-size: 16px;
}
.selected {
  color: red;
}
</style>
</head>
<body>
<p>Hello</p>
<p>and</p>
<p>Goodbye</p>
</body>
```

Lab : Result

Hello
and
Goodbye

Lab : .addClass()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - addClass1.html

Lab : addClass1.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(
    function(){
        $( "p:last" ).addClass( "selected highlight" );
    }
);
</script>
<style>
p {
    margin: 8px;
    font-size: 16px;
}
.selected {
    color: red;
}
.highlight {
    background: yellow;
}
</style>
</head>
<body>
    <p>Hello</p>
    <p>and</p>
    <p>Goodbye</p>
</body>
```

Lab : Result

Hello
and

Goodbye

Lab : .addClass()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - addClass2.html

Lab : addClass2.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    $("div").addClass(function( index, currentClass ) {
        var addedClass;
        if( currentClass === "red" ) {
            addedClass = "green";
            $("p").text("There is one green div");
        }
        return addedClass;
    });
});
</script>
<style>
div { background: white; }
.red { background: red; }
.red.green { background: green; }
</style>
</head>
<body>
<p>Hello</p>
<p>and</p>
<p class="red">Goodbye</p>
<p>Hello, World</p>
</body>
```

Lab : Result

Hello

and

Goodbye

Hello, World

.removeClass()

- Removes all or the specified class from the set of matched elements.
- Parameters
 - **class** : (String) : (optional) A CSS to remove from the elements.
- Returns
 - **jQuery**
- Examples
 - `$(“p”).removeClass()`
 - `$(“p”).removeClass(“selected”)`

Lab : .removeClass()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - removeClass.html

Lab : removeClass.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(function(){
        $("p:even").removeClass("blue");
    });
</script>
<style>
    p {
        margin: 4px; font-size: 16px; font-weight: bolder;
    }
    .blue { color: blue; }
    .under { text-decoration: underline; }
    .highlight { background: yellow; }
</style>
</head>
<body>
    <p class="blue under">Hello</p>
    <p class="blue under highlight">and</p>
    <p class="blue under">then</p>
    <p class="blue under">Goodbye</p>
</body>
```

Lab : Result

Hello

and

then

Goodbye

.toggleClass()

- Adds the specified class if it is not present, removes it if it is present.
- Parameters
 - **class** : (String) : A CSS class with which to toggle the elements.
- Returns
 - **jQuery**
- Examples
 - **\$("p").toggleClass("selected")**

Lab : .toggleClass()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - toggleClass.html

Lab : toggleClass.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    $("p").click(function() {
        $(this).toggleClass( "highlight" );
    });
});
</script>
<style>
p {
    margin: 4px;          font-size: 16px;
    font-weight: bolder;   cursor: pointer;
}
.blue { color: blue; }
.highlight { background: yellow; }
</style>
</head>
<body>
    <p class="blue">Click to toggle</p>
    <p class="blue highlight">highlight</p>
    <p class="blue">on these</p>
    <p class="blue">paragraphs</p>
</body>
```

Lab : Result

Click to toggle
highlight
on these
paragraphs

.attr()

- Access a property on the first matched element.
- Makes it easy to retrieve a property value from the first matched element.
- Parameters
 - **name** : (String) : The name of the property to access.
- Returns
 - **Object**
- Examples
 - `$("img").attr("src");`

.attr()

- Set a key/value object as properties to all matched elements.
- Serves as the best way to set a large number of properties on all matched elements.
- Parameters
 - **properties** : (Map) : Key/value pairs to set as object properties.
- Returns
 - **jQuery**
- Examples
 - `$("img").attr({src:"test.jpg", alt:"Test Image"});`

.attr()

- Set a single property to a value, on all matched elements.

- Parameters

- **key** : (String) : The name of the property to set.
 - **value** : (Object) : The value to set the property to.

- Returns

- **jQuery**

- Examples

- **\$("img").attr("src", "test.jpg");**

Lab : .attr()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - attr.html

Lab : attr.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    var title = $("em").attr("title");
    $("div").text( title );
});
</script>
<style>
em {
    color: blue;
    font-weight: bold;
}
div {
    color: red;
}
</style>
</head>
<body>
    <p>Once there was a <em title="huge, gigantic">large</em> dinosaur...</p>
```

The title of the emphasis is:<div></div>

Lab : Result

Once there was a *large* dinosaur...

The title of the emphasis is:
huge, gigantic

Lab : .attr()

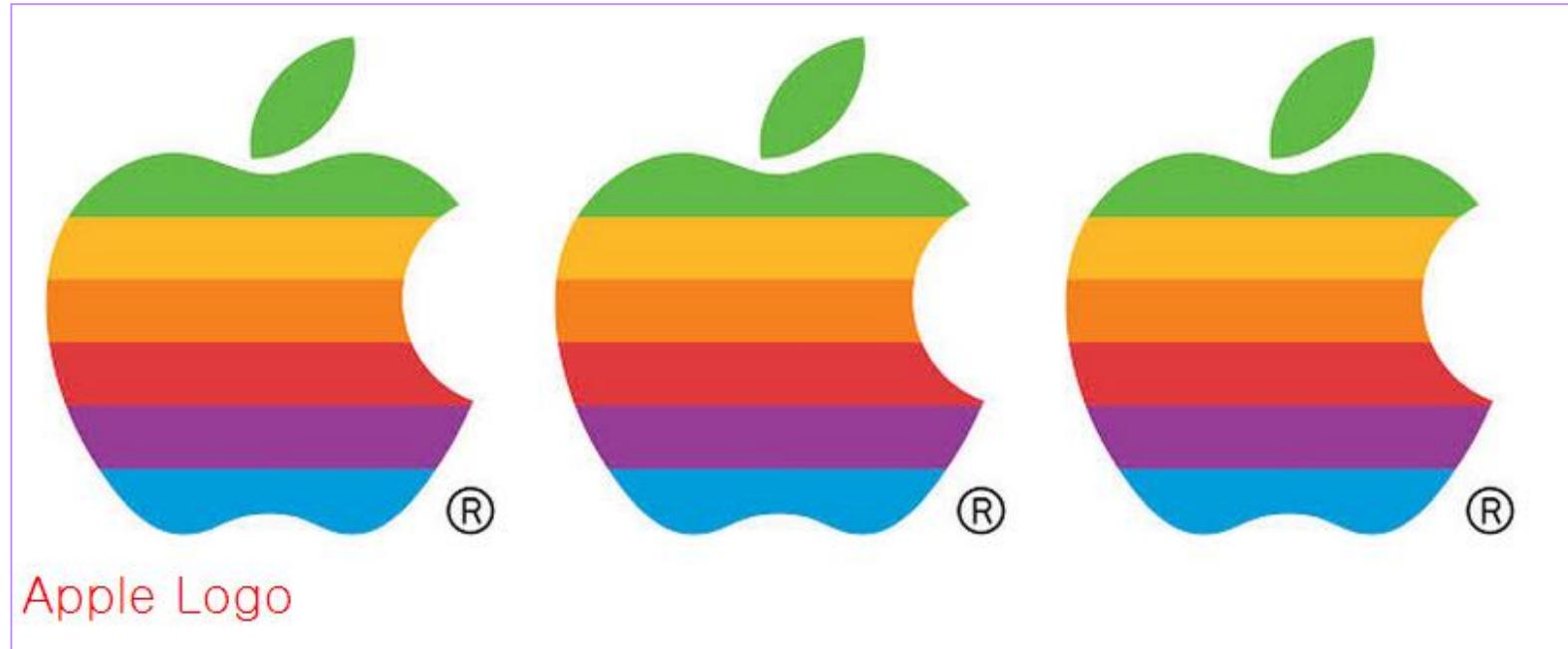
- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - attr1.html

Lab : attr1.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    $("img").attr({
        src: "images/applelogo.png",
        title: "Apple Company",
        alt: "Apple Logo"
    });
    $("div").text( $("img").attr("alt") );
});
</script>
<style>
img {
    padding: 10px;
}
div {
    color: red;
    font-size: 24px;
}
</style>
</head>
<body>
<img>
<img>
<img>

<div><b>Attribute of Ajax</b></div>
</body>
```

Lab : Result



.removeAttr()

- Remove an attribute from each of the matched elements.
- Parameters
 - **name** : (String) : The name of the attribute to remove.
- Returns
 - **jQuery**
- Examples
 - `$("input").removeAttr("disabled");`

Lab : .removeAttr()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - removeAttr.html

Lab : removeAttr.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(function(){
        $("button").click(function(){
            $("p").removeAttr("style");
        });
    });
</script>
</head>
<body>
    <h1>This is a heading</h1>

    <p style="font-size:120%;color:red">This is a paragraph.</p>
    <p style="font-weight:bold;color:blue">This is another paragraph.</p>

    <button>Remove the style attribute from all p elements</button>
</body>
```

Lab : Result

This is a heading

This is a paragraph.

This is another paragraph.

Remove the style attribute from all p elements

.hasClass()

- Checks if any of the selected elements have a specified class name.
- If ANY of the selected elements has the specified class name, this method will return "true".
- Parameters
 - **classname** : (String) : The name of the class to test.
- Returns
 - **Boolean**
- Examples
 - `$(“p”).hasClass(“title”);`

Lab : .hasClass()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - hasClass.html

Lab : hasClass.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    $("#result1").append( $("p:first").hasClass( "selected" ).toString() );
    $("#result2").append( $("p:last").hasClass( "selected" ).toString() );
    $("#result3").append( $("p").hasClass( "selected" ).toString() );
});
</script>
<style>
p {
    margin: 8px;
    font-size: 16px;
}
.selected {
    color: red;
}
</style>
</head>
<body>
<p>black and is the first paragraph.</p>
<p class="selected">red and is the second paragraph.</p>
<div id="result1">First paragraph has selected class: </div>
<div id="result2">Second paragraph has selected class: </div>
<div id="result3">At least one paragraph has selected class: </div>
</body>
```

Lab : Result

black and is the first paragraph.

red and is the second paragraph.

First paragraph has selected class: false

Second paragraph has selected class: true

At least one paragraph has selected class: true

.html()

- Get the html contents of the first matched element.
- Returns
 - **String**
- Examples
 - `$(“div”).html();`

.html()

- Set the html contents of every matched element.
- Parameters
 - **val** : (String) : Set the html contents to the specified value.
- Returns
 - **jQuery**
- Examples
 - `$("div").html("new stuff");`

Lab : .html()

■ Web Browsers

- IE11, Firefox, Google Chrome, Opera, Safari

■ Text Editors

- Notepad++ or Editplus

■ Files

- css_html.html

Lab : css_html.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
  $(document).ready(function(){
    $( "div" ).html( "<span class='red'>Hello <b>Again</b></span>" );
  });
</script>
<style>
  .red {
    color: red;
  }
</style>
</head>
<body>
  <span>Hello</span>
  <div></div>
  <div></div>
  <div></div>
</body>
```

Lab : Result

Hello

Hello Again

Hello Again

Hello Again

.prop()

- Get the value of a property for the first element in the set of matched elements or set one or more properties for every matched element.
- Parameters
 - **propertyName** : (String) : The name of the property to get.
- Returns
 - **Anything**
- Examples
 - `$(“em”).prop(“checked”);`

Lab : .prop()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - prop.html

Lab : prop.html

```
<script src="js/jquery-1.11.1.js"></script>
<style>
  p { margin: 20px 0 0; }
  b { color: blue; }
</style>
<script>
$(document).ready(function(){
  $("input").change(function() {
    var $input = $( this );
    $( "p" ).html(
      ".attr( \"checked\" ): <b>" + $input.attr( "checked" ) + "</b><br>" +
      ".prop( \"checked\" ): <b>" + $input.prop( "checked" ) + "</b><br>" +
      ".is( \":checked\" ): <b>" + $input.is( ":checked" ) + "</b>";
    }).change();
  });
</script>
</head>
<body>
  <input id="check1" type="checkbox" checked="checked">
  <label for="check1">Check me</label>
  <p></p>
</body>
```

Lab : Result

Check me

```
.attr( "checked" ): checked  
.prop( "checked" ): true  
.is( ":checked" ): true
```

.prop() (Cont.)

- Set one or more properties for the set of matched elements.
- Parameters
 - **propertyName** : (String) : The name of the property to get.
 - value : (Anything) : A value to set for the property.
- Returns
 - **jQuery**
- Examples
 - `$(“input”).prop(“checked”, true);`

Lab : .prop()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - prop1.html

Lab : prop1.html

```
<script src="js/jquery-1.11.1.js"></script>
<style>
  img {
    padding: 10px;
  }
  div {
    color: red;
    font-size: 24px;
  }
</style>
<script>
$(document).ready(function(){
  $("input[type='checkbox']").prop({
    disabled: true
  });
})
</script>
</head>
<body>
  <input type="checkbox" checked="checked">낚시
  <input type="checkbox">등산
  <input type="checkbox">독서
  <input type="checkbox" checked="checked">영화감상
</body>
```

Lab : Result



낚시



등산



독서



영화감상

.removeProp()

- Remove a property for the set of matched elements.
- Parameters
 - **propertyName** : (String) : The name of the property to remove.
- Returns
 - **jQuery**
- Examples
 - `$(“p”).removeProp(“checked”);`

Lab : .removeProp()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - removeProp.html

Lab : removeProp.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    jQuery("button").click(function () {
        jQuery("#check1").removeProp("checked");
    });
});
</script>
<style>
.highlight {
    border : 2px solid #1E8CBE;
}
</style>
</head>
<body>
    <input id="check1" type="checkbox" checked="checked">
    <label for="check1">Checkbox</label>

    <button>Click Here</button>
</body>
```

Lab : Result



Checkbox

[Click Here](#)

.val()

- Get the current value of the first matched element.
- Returns
 - **String**
- Examples
 - `$(“input”).val();`

.val() (Cont.)

- Set the value of every matched element.
- Parameters
 - **value** : (String) : Set the property to the specified value.
- Returns
 - **jQuery**
- Examples
 - `$("input").val("test");`

Lab : .val()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - val.html

Lab : val.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(function(){
        $("button").click(function(){
            $("input:text").val("Michael Jackson");
        });
    });
</script>
</head>
<body>
    <p>Name: <input type="text" name="user"></p>
    <button>Set the value of the input field</button>
</body>
```

Lab : Result

Name: Michael Jackson

Set the value of the input field

.text()

- Get the text contents of all matched elements.
- Returns
 - **String**
- Examples
 - `$(“p”).text();`

.text() (Cont.)

- Set the text contents of all matched elements.
- This has the same effect as html().
- Parameters
 - **value** : (String) : The text value to set the contents of the element to.
- Returns
 - **String**
- Examples
 - `$(“input”).text(“Some new text.”);`

Lab : .text()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - text.html

Lab : text.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
    $(document).ready(function(){
        $("button").click(function(){
            $("p").text("Hello world!");
        });
    });
</script>
</head>
<body>
    <button>Set text content for all p elements</button>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
</body>
```

Lab : Result

Set text content for all p elements

Hello world!

Hello world!

.css()

- Access a style property on the first matched element.
- Makes it easy to retrieve a style property value from the first matched element. .
- Returns
 - **String**
- Examples
 - `$(“p”).css(“color”);`

.css() (Cont.)

- Set a key/value object as style properties to all matched elements.
- Parameters
 - **properties** : (Map) : Key/value pairs to set as style properties.
- Returns
 - **jQuery**
- Examples
 - `$("p").css({color:"red", background:"blue"});`

.css() (Cont.)

- Set a single style property to a value, on all matched elements.
- Parameters
 - **key** : (String) : The name of the property to set.
 - **value** : (Object) : The value to set the property to.
- Returns
 - **jQuery**
- Examples
 - `$("p").css("color", "red");`

Lab : .css()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - css.html

Lab : css.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
  $("div").click(function() {
    var color = $( this ).css( "background-color" );
    $("#result").html( "That div is <span style='color:" +
      color + ";'>" + color + "</span>." );
  });
});
</script>
<style>
div {
  width: 60px;    height: 60px;
  margin: 5px;    float: left;
}
</style>
</head>
<body>
<span id="result">&ampnbsp</span>
<div style="background-color:blue;"></div>
<div style="background-color:rgb(15,99,30);"></div>
<div style="background-color:#123456;"></div>
<div style="background-color:#f11;"></div>
</body>
```

Lab : Result



That div is `rgb(0, 0, 255)`.

Lab : .css()

- Web Browsers
 - IE11, Firefox, Google Chrome, Opera, Safari
- Text Editors
 - Notepad++ or Editplus
- Files
 - css1.html

Lab : css1.html

```
<script src="js/jquery-1.11.1.js"></script>
<script>
$(document).ready(function(){
    $("p").mouseover(function() {
        $(this).css("color", "red");
    });
});
</script>
<style>
p {
    color: blue;
    width: 200px;
    font-size: 14px;
}
</style>
</head>
<body>
<p>Just roll the mouse over me.</p>
<p>Or me to see a color change.</p>
</body>
```

Lab : Result

Just roll the mouse over me.

Or me to see a color change.

.height()

- Returns the css height value for the first matched element.
- If used on document, returns the document's height (innerHeight).
- If used on window, returns the viewport's (window) height.
- Returns
 - Number
- Examples
 - `$ (document) .height () ;`

.width()

- Returns the css width value for the first matched element.
- If used on document, returns the document's width (innerWidth).
- If used on window, returns the viewport's (window) width.
- Returns
 - Number
- Examples
 - `$ (document) .width () ;`

.innerHeight()

- Returns the inner height value (without border) for the first matched element.
- If used on document, returns the document's height (innerHeight).
- If used on window, returns the viewport's (window) height.
- Returns
 - Number
- Examples
 - `$("#testdiv").innerHeight();`

.innerWidth()

- Returns the inner width value (without border) for the first matched element.
- If used on document, returns the document's Width (innerWidth).
- If used on window, returns the viewport's (window) width.
- Returns
 - Number
- Examples
 - `$("#testdiv").innerWidth();`

.outerHeight()

- Returns the outer height value (including border) for the first matched element.
- Cannot be used on document or window.
- Returns
 - Number
- Examples
 - `$("#testdiv").outerHeight();`

.outerWidth()

- Returns the outer width value (including border) for the first matched element.
- Cannot be used on document or window.
- Returns
 - **Number**
- Examples
 - `$("#testdiv").outerWidth();`

.scrollLeft()

- Returns how many pixels the user has scrolled to the right (scrollLeft).
- Works on containers with overflow: auto and window/document.
- Returns
 - Number
- Examples
 - `$("#testdiv").scrollLeft();`

.scrollTop()

- Returns how many pixels the user has scrolled to the bottom (scrollTop).
- Works on containers with overflow: auto and window/document.
- Returns
 - Number
- Examples
 - `$("#testdiv").scrollTop();`

.offset()

- Access a style property on the first matched element.
- Makes it easy to retrieve a style property value from the first matched element. .
- Returns
 - **String**
- Examples
 - `$(“p”).css(“color”);`