



jQuery 기본

jQuery

- ◆ jQuery는 존 레식(John Resig)이 2006년에 BarCamp NYC에서 발표
- ◆ jQuery는 자바 스크립트 라이브러리
- ◆ jQuery를 사용하면 자바 스크립트 프로그래밍의 양을 상당히 줄일 수 있다. (짧고 간결하다)
- ◆ jQuery는 배우기 쉽다.
- ◆ 무료이다.
- ◆ jQuery는 자바스크립트를 쉽게 쓸 수 있도록 만든 라이브러리
- ◆ 자바스크립트의 대체용 언어가 아니다
- ◆



javascript	<code>document.getElementById('result')</code> <code>document.getElementsByTagName('h1')</code>
jQuery	<code>\$('#result')</code> <code>\$('h1')</code>

jQuery 사용방법

◆ jQuery 파일을 다운로드하는 방법

- jQuery는 <http://www.jquery.com>에서 다운로드
- 다운로드 후 사용

```
<head>
```

```
    <script src="jquery-3.4.1min.js">
```

```
</head>
```

◆ 구글이나 마이크로소프트에서 제공하는 CDN을 사용하여 jQuery 파일을 포함하는 방법

- 공개 서버로부터 네트워크를 통하여 웹페이지를 실행할 때마다 다운로드 받을 수도 있다 : (CDN:Content Delivery Network)방식

jQuery 사용방법 CDN

◆ jQuery CDN

```
<script src="http://code.jquery.com/jquery-3.4.1.min.js"> </script>
```

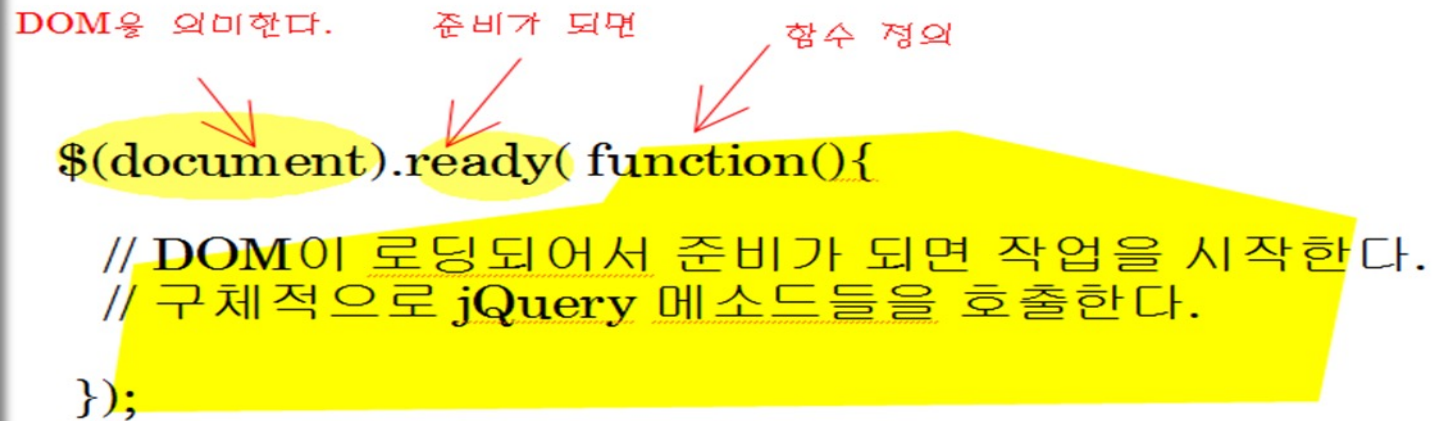
◆ Google CDN:

```
< script  
src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">  
</script>
```

◆ Microsoft CDN:

```
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.1.min.js">  
</script>
```

함수 일반적인 구조



The diagram shows a jQuery `ready` function call with three red arrows pointing to its components: `$(document)` is labeled "DOM을 의미한다." (It means DOM), `.ready()` is labeled "준비가 되면" (When ready), and `function()` is labeled "함수 정의" (Function definition). The entire function body is highlighted in yellow.

```
$(document).ready(function(){  
    // DOM이 로딩되어서 준비가 되면 작업을 시작한다.  
    // 구체적으로 jQuery 메소드들을 호출한다.  
});
```

<script> h1이 로드되지 않은 상태에서는 검색되지 않는다

```
    $('h1').css('background', 'yellow');
```

</script>

<script>

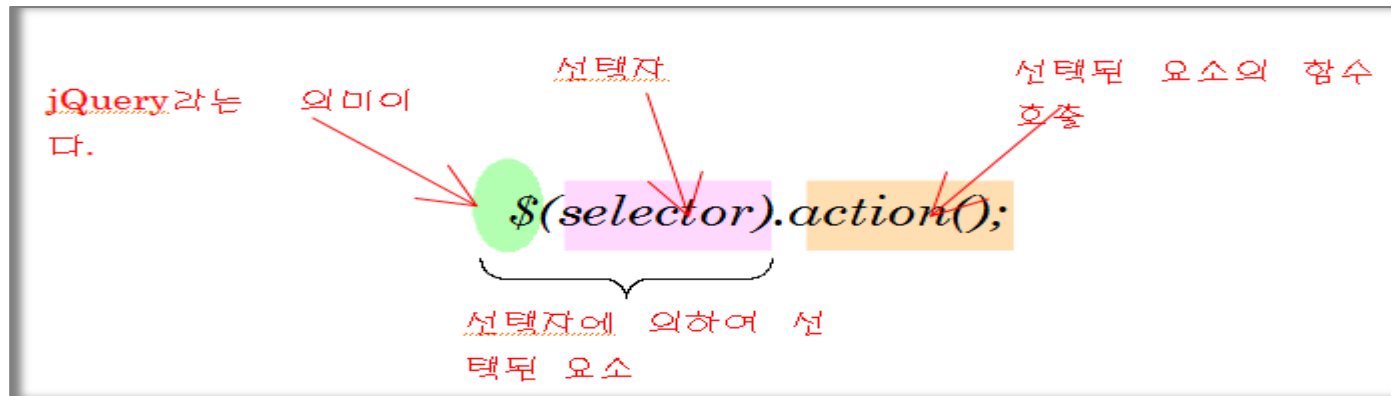
```
    $(document).ready(function(){  
        $('h1').css('background', 'yellow');
```

```
    });
```

</script>

jQuery 문장구조 1

- ◆ \$(...) 안에 선택자를 넣어서 원하는 요소를 선택하고, 선택된 요소에 대하여 여러 가지 조작을 한다.



- `$('h2').css("background", "blue")`
- `$("p").show()` : 모든 `<p>` 요소들을 찾아서 화면에 표시한다.
- `$(".group1").slideUp()` : `class=group1`인 요소를 슬라이드업 방식으로 표시한다.
- `$("#id9").hide()` : `id=id9`인 요소를 화면에서 감춘다.

jQuery 문장구조2

◆ Javascript

```
Var hlist = document.getElementsByTagName('h1')  
hlist[0].style.backgroundColor = "yellow";  
Hlist[0].style.color = "red";
```

◆ jQuery

```
var jq = $("h1");  
jq.css("background-color", "yellow");  
jq.css('color', 'red');
```

메소드 체이닝(Method Chaining)

```
$("h1").css("background-color","yellow").css("color", "red")
```

jQuery 기본선택자

- ◆ 타입선택자 (요소선택자) html 태그

`$('h1') , $('p')`

여러 개의 태그 선택 : `$('h1, p').css('color', 'orange');`

- ◆ .class 선택자 : `$('.name')`

- ◆ #id선택자 : `$('#name')`

- ◆ *전체선택자 : `$('*')`

예제 1 - 선택자

◆ css(스타일, 값)

```
<script type="text/javascript">
  $(document).ready(function(){
    //전체 배경색을 변경
    // h1요소의 글자색을 파랑
    //id가 para인 요소의 글자색은 녹색
    //class 가 sample요소의 글자색은 빨강
  });
</script>
<body>
  <h1> 제목입니다 </h1>
  <h2 id="para"> 작은 제목입니다 </h2>
  <h2 class="sample"> 샘플입니다 </h2>
</body>
```

제목입니다

작은 제목입니다

샘플입니다

첫 번째 jQuery 프로그램

```
<!DOCTYPE html>
<html>
<head>
<script src="http://code.jquery.com/jquery-3.4.1.min.js"></script>
</script>
<script>
    $(document).ready(function () {
        });
</script>
</head>
<body>
<h2>클릭하면 사라집니다1.</h2>
<h2>클릭하면 사라집니다2.</h2>
<h2>클릭하면 사라집니다3.</h2>
<h2>클릭하면 사라집니다4.</h2>
<h2>클릭하면 사라집니다5.</h2>
<h2>클릭하면 사라집니다6.</h2>
<h2>클릭하면 사라집니다7.</h2>

</body>
</html>
```

[웹브라우저로 보기](#)

클릭하면 사라집니다1.

클릭하면 사라집니다2.

클릭하면 사라집니다3.

클릭하면 사라집니다4.

클릭하면 사라집니다5.

클릭하면 사라집니다6.

jQuery core1

\$ () : jQuery 함수

- `$()`함수의 인수로 다른 함수를 전달하면 문서가 로드될 때 호출될 콜백 함수를 등록한다
- `window.onload` 이벤트와 유사 , `document.ready` 이벤트에 대한 이벤트핸들러 함수 지정 , 로드 직후 곧바로 실행

<head>

<script>

```
$( function() {  
    $('h1').css('background', 'yellow');  
});
```

</script>

</head>

<body>

<h1>제목입니다 </h1>

</body>

jQuery core2

\$(선택자,[컨텍스트])

- 선택자로 작업대상 엘리먼트를 검색한다
- 선택된 엘리먼트는 jQuery객체에 저장되며 이후 메서드를 호출하여 검색된 엘리먼트에 여러가지 조작을 가할 수 있다
- 컨텍스트는 검색의 시작점을 지정, 생략하면 문서전체에서 검색
- 컨텍스트를 지정하면 그 하위로 검색 범위가 제한

```
var vp1 = $("p");
```

```
var vp2 = $("p", document.forms[2]);
```

```
$( 'p', this ) -> this안에 p this는 p의 부모가 됨
```

현재 이벤트 대상이 되는 객체를 this로 받음

jQuery core3

\$(element)

DOM 엘리먼트 를 인수로 전달하면 이 객체를 감싸는 jQuery 객체를 리턴
jQuery 메소드 사용가능

```
var elem = document.getElementsByTagName('h1')[0];  
elem.style.color = 'red';
```

getElementsByTagName 메서드가 리턴하는 것은 DOM객체 배열이며
DOM 객체에 대해 css()나 text() html()같은 jQuery 메소드를 직접 호출할
수 없다

그래서 이 객체를 \$()함수로 전달하여 jQuery 객체로 래핑하면
여러 가지 편리한 jQuery 메소드를 호출 할 수 있다

```
$(elem).css('color','red');
```

jQuery core4

`$("html"), $("html", {propertis})`

인수로 전달된 html문자열로 새로운 엘리먼트를 직접 생성한다
생성된 엘리먼트를 DOM트리의 원하는 부분에 삽입하여 실행 중에 문서를 만들 수 있다

propertis 는 새로 만들어진 요소의 속성, 이벤트 함수 등을 지정

```
<body>
```

```
<button type="button">새로만들기</button>
```

```
<script>
```

```
    var elem = $('<p id='p1' onclick="proc1()">Hello jQuery.</p>')  
    elem.appendTo('body');
```

```
</script>
```

```
<body>
```

jQuery core5

```
<head>
```

```
<script src="jquery-3.4.1.min.js"> </script>
```

```
<script>
```

```
$(document).ready(function() {
```

```
    $("<p/>", {
```

```
        "id" : "p1",
```

```
        "text" : "Hello jQuery~~",
```

```
        "click" : function(){
```

```
            $(this).css('color', 'red').css('font-size', +=5px);
```

```
        }
```

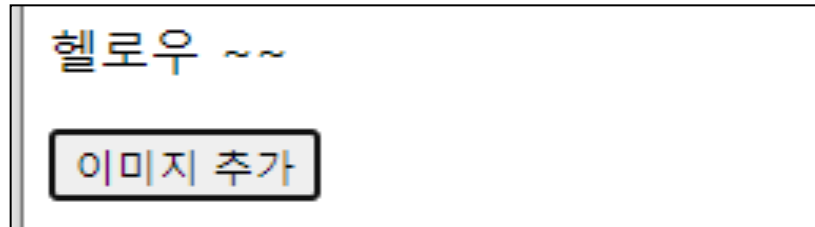
```
    }).appendTo("body");
```

```
})
```

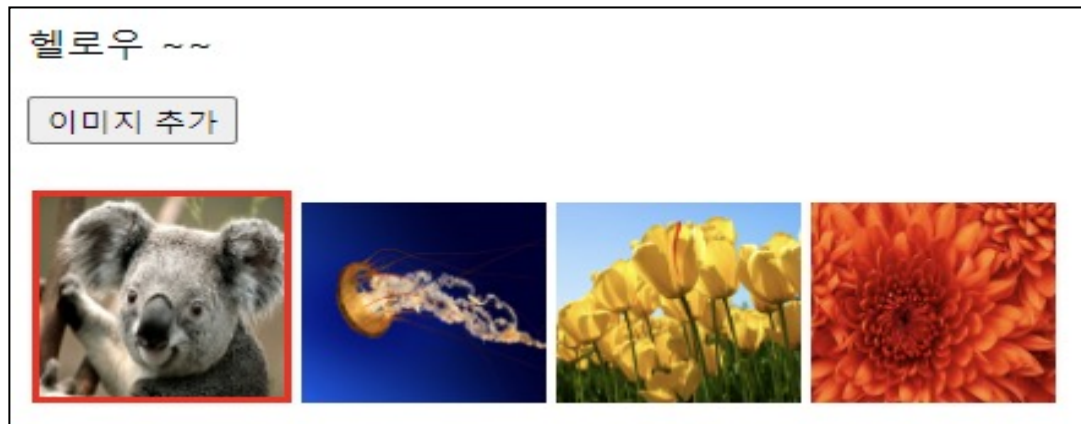
```
<div> 클릭하세요 </div>
```

jQuery core6 새로운 요소 만들기

//body안에 이미지 추가 버튼 만들어 놓기



/버튼 클릭시 동적으로 이미지 추가 하기- width, height 크기지정
//이미지클릭 - 이미지테두리 -border : 2px solid red



jQuery core7

◆ html() / text()

`$("body").html();` // body 태그 안의 모든 태그를 포함한 문장

`$("body").text();` // body 태그안의 모든 문자만

`$("p").text();` //여러개인경우 모든 p태그의 문자만

`$("p").html();` //p가 여러개인경우 첫번째 인거만 태그를 포함한 문장

//모든것을 대상 으로 실행시 - 반복루프를 사용

`<body>`

`<p>1홍길동</p>`

`<p>2개나리</p>`

`<p>3진달래</p>`

`</body>`

jQuery core8(객체 조작)

◆ `$("p").length;` -- p엘리먼트의 개수

◆ 반복문

`$(요소선택).each(function(매개변수1, 매개변수2){...});`

`$.each("요소선택, function(매개변수1, 매개변수2){...});`

검색된 DOM요소의 개수만큼 지정된 fn함수를 호출한다

