



jQuery 기본

참고

- ◆ w3schools.com > > [TUTORIALS] > [JavaScript] > [Learn jQuery] > [jQuery Home] ~ [jQuery Selectors]
 - <https://www.w3schools.com/jquery/default.asp>

jQuery 소개

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

javascript

```
document.getElementById('result')  
document.getElementsByTagName('h1')
```

jQuery

```
$('#result')  
$('h1')
```

jQuery 사용방법

◆ jQuery 파일을 다운로드하는 방법

- 1) jQuery는 <http://www.jquery.com>에서 다운로드
- 2) HTML의 <head> 섹션에 <script>로 참조해서 사용

<head>

<script src="jquery-3.6.0.js"> </script>

</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="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.j  
s"> </script>
```

◆ Microsoft CDN:

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

함수 일반적인 구조

```
...<head>
<script src="../../js/jquery-3.5.1.js"></script>
</head>
<body>
<script type="text/javascript">
    //h1이 로드되지 않은 상태에서는 h1이 검색되지 않는다.
    $('h1').css('background', 'yellow');
    alert('a');
    $(document).ready(function(){
        $('h1').css('background', 'red');
    });
</script>
<h1>jQuery 테스트</h1>
</body>...
```

DOM을 의미

준비가 되면

호출할 함수 정의

jQuery 문장구조 1

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

jQuery라는 의미 선택자 선택된 요소가 수행할 작업

`$(selector).action();`

예) `$('h2').css("background", "blue");`
`$("p").show();`
`$("#id9").hide();`

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 문장구조3

◆문서가 완전히 로드 된 후 수행되도록 ready 이벤트 안에 jQuery 메소드를 작성한다.

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



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

jQuery 기본선택자

- ◆ 전체 선택자 : \$('*')
- ◆ 현재 HTML 요소 선택자 : \$(this)
- ◆ 요소 선택자 (html 태그 선택자)
 - \$('h1'), \$('p')
 - 여러 개의 태그 선택 : \$('h1, p').css('color', 'orange');
- ◆ class 선택자 : \$('.name')
- ◆ id 선택자 : \$('#name')

*선택자의 추가 예 :

https://www.w3schools.com/jquery/jquery_selectors.asp

jQuery 기본선택자 추가 사용 예시

```
<script>
```

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

```
    $("button").click(function(){
```

```
        $("p.intro").hide(); //p 요소 중 class가 'intro'인 것 들
```

```
        $("p:first").hide(); //p 요소 중 첫 번째 요소
```

```
        $("ul li:first").hide(); //ul의 자식 요소 li 중, 첫 번째 요소
```

```
        $("ul li:first-child").hide(); //ul의 자식 요소 li 중, 첫 번째 요소들
```

```
        $("[href]").hide(); //href 속성이 있는 요소들
```

```
        $("a[target='_blank']").hide(); //a 요소 중 target 속성 값이  
        '_blank'인 요소들
```

```
    });
```

```
});
```

```
</script>
```

예제 1 - 선택자(1/2)

◆css (스타일, 값)

```
<script type="text/javascript">
```

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

```
//전체 배경색을 변경
```

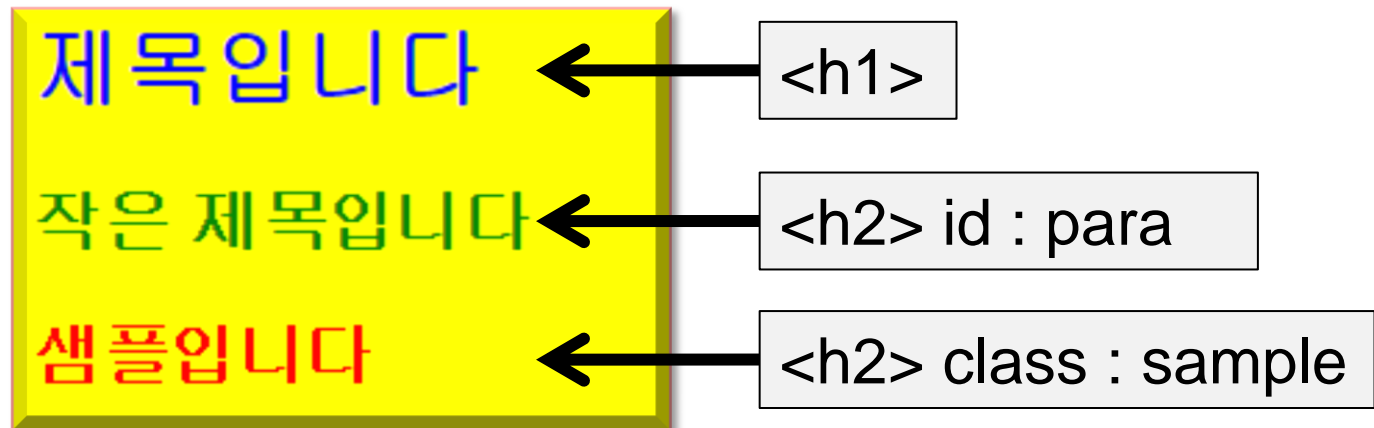
```
// h1요소의 글자색을 파랑
```

```
//id가 para인 요소의 글자색은 녹색
```

```
//class 가 sample요소의 글자색은 빨강
```

```
});
```

```
</script>
```



예제 1 - 선택자(2/2)

◆css (스타일, 값)

```
<script type="text/javascript">
$(document).ready(function(){
    $("body").css("background-color", "yellow");
    $("h1").css("color", "blue");
    $("#para").css("color", "green");
    $(".sample").css("color", "red");
});
</script>
```

```
<body>
  <h1> 제목입니다 </h1>
  <h2 id="para"> 작은 제목입니다 </h2>
  <h2 class="sample"> 샘플입니다 </h2>
</body>
```

예제 2(1/2)

```
<!DOCTYPE html>
<html>
<head>
  <script src="../../js/jquery-3.5.1.js"></script>
</head>
<script type="text/javascript">
  $(document).ready(function(){
    ...
  });
</script>
<body>
  <h2>클릭하면 사라집니다. 1</h2>
  <h2>클릭하면 사라집니다. 2</h2>
  <h2>클릭하면 사라집니다. 3</h2>
  <h2>클릭하면 사라집니다. 4</h2>
  <h2>클릭하면 사라집니다. 5</h2>
  <h2>클릭하면 사라집니다. 6</h2>
  <h2>클릭하면 사라집니다. 7</h2>
</body>
</html>
```

클릭하면 사라집니다1.

클릭하면 사라집니다2.

클릭하면 사라집니다3.

클릭하면 사라집니다4.

클릭하면 사라집니다5.

클릭하면 사라집니다6.

예제 2(2/2)

```
<!DOCTYPE html>
<html>
<head>
  <script src="../../js/jquery-3.5.1.js"></script>
</head>
<script type="text/javascript">
  $(document).ready(function(){
    $('h2').click(function(){
      $(this).hide();
    });
  });
</script>
<body>
  <h2>클릭하면 사라집니다. 1</h2>
  ...
  <h2>클릭하면 사라집니다. 7</h2>
</body>
</html>
```

jQuery core1

\$() : jQuery 함수

- \$()함수의 인수로 다른 함수를 전달하면 문서가 로드될 때 호출될 콜 백 함수를 등록한다
- window.onload 이벤트와 유사 , document.ready 이벤트에 대한 이벤트핸들러 함수 지정, 로드 직 후 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 여러 가지 조작을 가할 수 있다
- 컨텍스트는 검색의 시작점을 지정. 생략하면 문서전체에서 검색
- 컨텍스트를 지정하면 그 하위로 검색 범위가 제한

```
var myJquery1 = $("p");  
var myJquery2 = $("p", document.forms[2]);
```

- \$('p', this) → this 안의 p라는 의미(즉, this는 p의 부모)

jQuery core3

\$(element)

- DOM 엘리먼트를 인수로 전달하면 이 객체를 감싸는 jQuery 객체 리턴
- jQuery 메소드 사용가능

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

- getElementsByTagName 메서드가 리턴하는 것은 DOM 객체 배열
- DOM 객체에 대해 jQuery 메소드(css(), text(), html() 등) 호출 불가
- 이 객체를 \$()함수로 전달하여 jQuery 객체로 래핑하면 여러 가지 편리한 jQuery 메소드를 호출 할 수 있음

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

jQuery core4

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

- 인수로 전달된 html문자열로 새로운 엘리먼트를 직접 생성
- 생성된 엘리먼트를 DOM 트리의 원하는 부분에 삽입하여 실행 중에 문서를 만들 수 있음
- **propertis**는 새로 만들어진 요소의 속성, 이벤트 함수 등을 지정

```
<button type="button" onclick="test()">클릭</button>
<script>
function test(){
    var ele
    = $('<p id="p1" onclick="proc1()">Hello</p>');
    ele.appendTo("body");
}
</script>
```

jQuery core4 예제

...

```
<script>
```

```
$(document).ready(function() {  
    $("<p/>", {  
        "id" : "p1",  
        "text" : "Hello jQuery~~",  
        "click" : function(){  
            $(this).css('color', 'red')  
                .css('font-size', '+=5px');  
        }  
    }).appendTo("body");  
});
```

```
</script>
```

```
<body>
```

```
    <div>클릭하세요</div>
```

...

예제 3 - 새로운 요소 만들기(1/2)

1. 화면이 로딩되면 body안에 이미지 추가 버튼 만들어 지기
2. 버튼 클릭 시 동적으로 이미지 추가 하기
 - width, height 크기 지정
3. 이미지 클릭 시 이미지 테두리 넣기
 - border : 2px solid red

예제 3 - 새로운 요소 만들기(2/2)

```
...
$(document).ready(function() {
    $('<button onclick="addImg()">이미지추가</button><br>')
    .appendTo("body");
});

function addImg(){
    var ele = $('<img/>', {
        "src" : "../../images/사자.jpg"
        , "width" : "100px", "height" : "100px"
        , "click" : function(){
            $(this).css("border", "2px solid red");
        }
    });
    ele.appendTo("body");
}
</script>
<body>
    <h1>jQuery 예제 2 - 새로운 요소 만들기</h1>
</body>
...
```

jQuery core5

◆ html() / text()

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

`$("body").text();` → body 태그 안의 모든 문자

`$("p").text();` → 모든 p 태그의 문자

`$("p").html();` → p 태그가 여러 개인 경우, 첫 번째 p 태그만.
p 태그 안의 (태그 포함)문장

예제	<code><p>jQuery easy.</p></code> <code><p>jQuery easy.</p></code>
----	---

`$("p").text();` → "jQuery easy.jQuery easy."

`$("p").html();` → "jQuery easy."

jQuery core6(객체 조작)

◆ `$("p").length;` → p엘리먼트의 개수

◆ 반복문

1) selector 이용

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

2) object 이용 : 검색된 요소 개수만큼 지정된 fn함수를 호출

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


예제 4 - 반복문 사용 예

1) selector 이용

```
$("button").click(function(){
    $("li").each(function(index, item){
        console.log(index + "번째 item : " + item);
        alert($(this).text());
    });
});
```

2) object 이용

```
var arr = ["apple", "banana", "orange"];
$.each(arr, function(item){
    console.log(this);
});
```