

# jQuery 튜토리얼

jQuery 는 JavaScript 라이브러리입니다.

jQuery 는 JavaScript 프로그래밍을 크게 단순화합니다.

jQuery 는 배우기 쉽다.

## jQuery 소개

jQuery의 목적은 웹 사이트에서 JavaScript를 훨씬 쉽게 사용할 수 있도록 하는 것입니다.

## 당신이 이미 알아야 할 것

jQuery 를 공부하기 전에 다음에 대한 기본 지식이 있어야 합니다.

- HTML
- CSS
- 자바 스크립트

## jQuery 란 무엇입니까?

jQuery 는 경량의 "쓰기를 줄이고 더 많은 것을 할 수 있는" JavaScript 라이브러리입니다.

jQuery 의 목적은 웹 사이트에서 JavaScript 를 훨씬 쉽게 사용할 수 있도록 하는 것입니다.

jQuery 는 여러 줄의 JavaScript 코드가 필요한 많은 공통 작업을 수행하고 이를 한 줄의 코드로 호출 할 수 있는 메소드로 래핑합니다.

또한 jQuery 는 AJAX 호출 및 DOM 조작과 같은 JavaScript 의 복잡한 작업을 단순화합니다.

jQuery 라이브러리에는 다음 기능이 포함되어 있습니다.

- HTML / DOM 조작
- CSS 조작
- HTML 이벤트 메소드
- 효과 및 애니메이션
- AJAX
- 유틸리티

**팁 :** 또한 jQuery 에는 거의 모든 작업을 위한 플러그인이 있습니다

## 왜 jQuery인가?

다른 JavaScript 프레임 워크가 많이 있지만 jQuery 가 가장 인기 있고 확장 성이 있는 것으로 보입니다.

웹에서 가장 큰 회사는 다음과 같은 jQuery 를 사용합니다.

- 구글
- 마이크로 소프트
- IBM
- 넷플릭스

### jQuery 는 모든 브라우저에서 작동합니까?

jQuery 팀은 브라우저 간 문제에 대해 모두 알고 있으며 이 지식을 jQuery 라이브러리에 작성했습니다. jQuery 는 모든 주요 브라우저에서 동일하게 실행됩니다.

## jQuery 시작

# 웹 페이지에 jQuery 추가

웹 사이트에서 jQuery 사용을 시작하는 방법에는 여러 가지가 있습니다. 당신은 할 수 있습니다 :

- jQuery.com 에서 jQuery 라이브러리를 다운로드 하십시오.
- Google 과 같은 CDN 의 jQuery 포함

## jQuery 다운로드

다운로드 가능한 jQuery 에는 두 가지 버전이 있습니다.

- 프로덕션 버전-축소 및 압축되었으므로 라이브 웹 사이트를 위한 것입니다.
- 개발 버전-테스트 및 개발 용 (압축되지 않고 읽을 수 있는 코드)

두 버전 모두 [jQuery.com](https://jquery.com) 에서 다운로드 할 수 있습니다.

jQuery 라이브러리는 단일 JavaScript 파일이며 HTML `<script>` 태그를 사용하여 참조합니다 (`<script>` 태그가 `<head>` 섹션 내에 있어야 함에 유의 하십시오).

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

**팁 :** 다운로드 한 파일을 사용하려는 페이지와 동일한 디렉토리에 배치하십시오.

왜 `<script>` 태그 안에 **type = "text / javascript"**가 없는지 궁금하십니까?

HTML5 에서는 필요하지 않습니다. JavaScript 는 HTML5 및 모든 최신 브라우저에서 기본 스크립팅 언어입니다!

## jQuery CDN

jQuery 를 직접 다운로드하여 호스팅하지 않으려면 CDN (Content Delivery Network)에서 jQuery 를 포함시킬 수 있습니다.

Google 과 Microsoft 모두 jQuery 를 호스팅합니다.

Google 또는 Microsoft 에서 jQuery 를 사용하려면 다음 중 하나를 사용하십시오.

## Google CDN :

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

## Microsoft CDN :

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

### Google 또는 Microsoft에서 호스팅 된 jQuery를 사용하는 것의 큰 장점 :

많은 사용자가 다른 사이트를 방문 할 때 이미 Google 또는 Microsoft에서 jQuery 를 다운로드 했습니다. 결과적으로 사이트를 방문 할 때 캐시에서 로드 되므로 로드 시간이 단축됩니다. 또한 대부분의 CDN은 사용자가 파일을 요청하면 가장 가까운 서버에서 파일을 제공하므로 로딩 시간이 더 빨라집니다.

## jQuery 구문

jQuery를 사용하면 HTML 요소를 선택 (질의)하고 "조치"를 수행합니다.

## jQuery 구문

jQuery 구문은 HTML 요소 를 선택 하고 요소에 대한 작업 을 수행 하도록 맞춤화 되었습니다.

기본 구문은 **\$ ( selector ).액션 ()** 입니다.

- jQuery 를 정의 / 액세스하기 위한 \$ 부호
- HTML 요소를 "쿼리 (또는 찾기)" 하는 A ( *selector* )
- 요소에서 수행 할 jQuery 작업 ()

예 :

`$(this).hide()` -현재 요소를 숨 깁니다.

`$("p").hide()` -모든 <p> 요소를 숨 깁니다.

`$(".test").hide()` -class = "test"인 모든 요소를 숨 깁니다.

`$("#test").hide()` -id = "test"로 요소를 숨 깁니다.

### CSS 선택기에 익숙하십니까?

jQuery는 CSS 구문을 사용하여 요소를 선택합니다. 이 학습서의 다음 장에서 선택기 구문에 대해 자세히 학습합니다.

팁 : CSS를 모르는 경우 [CSS Tutorial](#)을 읽을 수 있습니다 .

## 문서 준비 이벤트

예제의 모든 jQuery 메소드가 문서 준비 이벤트 내에 있음을 알 수 있습니다.

```
$(document).ready(function(){
    // jQuery methods go here...
});
```

이는 문서로드가 완료되기 전에 모든 jQuery 코드가 실행되지 않도록하기 위한 것입니다 (준비 됨).

작업하기 전에 문서가 완전히 로드 되고 준비 될 때까지 기다리는 것이 좋습니다. 또한 헤드 섹션에서 문서 본문 앞에 JavaScript 코드를 가질 수 있습니다.

다음은 문서가 완전히 로드 되기 전에 메소드가 실행될 경우 실패 할 수 있는 조치의 예입니다.

- 아직 생성되지 않은 요소를 숨기려고 시도
- 아직 로드 되지 않은 이미지의 크기를 가져 오려고 시도

**팁 :** jQuery 팀은 문서 준비 이벤트에 대해 더 짧은 방법을 만들었습니다.

```
$(function(){  
    // jQuery methods go here...  
});
```

원하는 구문을 사용하십시오. 코드를 읽을 때 문서 준비 이벤트를 이해하기 쉽다고 생각합니다.

# jQuery 선택기

jQuery 선택기는 jQuery 라이브러리에서 가장 중요한 부분 중 하나입니다.

## jQuery 선택기

jQuery 선택기를 사용하면 HTML 요소를 선택하고 조작 할 수 있습니다.

jQuery 선택기는 이름, ID, 클래스, 유형, 속성, 속성 값 등을 기반으로 HTML 요소를 "찾기"(또는 선택)하는 데 사용됩니다. 기존 [CSS 선택기](#)를 기반으로 하며 자체 맞춤 선택기가 있습니다.

jQuery 의 모든 선택기는 달러 기호와 괄호로 시작합니다 : \$ ().

## 요소 선택기

jQuery 요소 선택기는 요소 이름을 기준으로 요소를 선택합니다.

다음 `<p>`과 같이 페이지 에서 모든 요소를 선택할 수 있습니다.

```
$("p")
```

예

사용자가 버튼을 클릭하면 모든 `<p>` 요소가 숨겨집니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me to hide paragraphs</button>

</body>
</html>
```

## #id 선택기

jQuery `#id` 선택기는 HTML 태그의 `id` 속성을 사용하여 특정 요소를 찾습니다.

ID 는 페이지 내에서 고유해야 하므로 고유 한 단일 요소를 찾으려면 `#id` 선택기를 사용해야 합니다.

특정 ID 를 가진 요소를 찾으려면 해시 문자를 쓰고 HTML 요소의 ID 를 입력하십시오.

```
$("#test")
```

예

사용자가 버튼을 클릭하면 `id = "test"`인 요소가 숨겨집니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```

<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
</script>
</head>
<body>

<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button>

</body>
</html>

```

## .class 선택기

jQuery **.class** 선택기는 특정 클래스를 가진 요소를 찾습니다.

특정 클래스를 가진 요소를 찾으려면 마침표 문자와 클래스 이름을 작성하십시오.

```
$(".test")
```

예

사용자가 버튼을 클릭하면 class = "test"인 요소가 숨겨집니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
</script>
</head>
<body>

<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>

```



```
</body>
</html>
```

## jQuery 선택기의 추가 예

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("#p.intro")</code>	Selects all <code>&lt;p&gt;</code> elements with <code>class="intro"</code>
<code>\$("#p:first")</code>	Selects the first <code>&lt;p&gt;</code> element
<code>\$("#ul li:first")</code>	Selects the first <code>&lt;li&gt;</code> element of the first <code>&lt;ul&gt;</code>
<code>\$("#ul li:first-child")</code>	Selects the first <code>&lt;li&gt;</code> element of every <code>&lt;ul&gt;</code>
<code>\$("[href]")</code>	Selects all elements with an href attribute
<code>\$("#a[target='_blank']")</code>	Selects all <code>&lt;a&gt;</code> elements with a target attribute value equal to <code>"_blank"</code>
<code>\$("#a[target!='_blank']")</code>	Selects all <code>&lt;a&gt;</code> elements with a target attribute value NOT equal to <code>"_blank"</code>
<code>\$(":button")</code>	Selects all <code>&lt;button&gt;</code> elements and <code>&lt;input&gt;</code> elements of <code>type="button"</code>
<code>\$("#tr:even")</code>	Selects all even <code>&lt;tr&gt;</code> elements
<code>\$("#tr:odd")</code>	Selects all odd <code>&lt;tr&gt;</code> elements

[jQuery 선택기 테스트](#) 를 사용 하여 다양한 선택기를 시연하십시오.

모든 jQuery 선택기에 대한 자세한 내용은 [jQuery 선택기 참조](#) 로 이동하십시오.

## 별도의 파일에 있는 함수

웹 사이트에 많은 페이지가 포함되어 있고 jQuery 함수를 쉽게 유지 관리하려면 jQuery 함수를 별도의 .js 파일에 넣을 수 있습니다.

이 튜토리얼에서 jQuery 를 시연하면 함수가 `<head>` 섹션 에 직접 추가됩니다. 그러나 때로는 다음과 같이 별도의 파일에 배치하는 것이 좋습니다 (src 속성을 사용하여 .js 파일을 참조하십시오).

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script src="my_jquery_functions.js"></script>
</head>
```

# jQuery 이벤트 메소드

jQuery는 HTML 페이지의 이벤트에 응답하도록 만들어졌습니다.

## 이벤트란 무엇입니까?

웹 페이지가 응답 할 수 있는 모든 다른 방문자의 조치를 이벤트라고 합니다.

이벤트는 무언가가 발생하는 정확한 순간을 나타냅니다.

예 :

- 요소 위로 마우스 이동
- 라디오 버튼 선택
- 요소를 클릭

**"fires / fired"** 라는 용어 는 종종 이벤트와 함께 사용됩니다. 예 : "키 누르기 이벤트가 시작됩니다. 키를 누르는 순간"

일반적인 DOM 이벤트는 다음과 같습니다.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

## 이벤트 메소드의 jQuery 구문

jQuery 에서 대부분의 DOM 이벤트에는 동등한 jQuery 메소드가 있습니다.

클릭 이벤트를 페이지의 모든 단락에 지정하려면 다음을 수행하십시오.

```
$("#p").click();
```

다음 단계는 이벤트가 발생할 때 발생할 동작을 정의하는 것입니다. 이벤트에 함수를 전달해야 합니다.

```
$("#p").click(function(){  
    // action goes here!!  
});
```

## 일반적으로 사용되는 jQuery 이벤트 메소드

### **\$ (document).ready()**

이 **\$(document).ready()** 방법을 사용하면 문서가 완전히 로드 될 때 함수를 실행할 수 있습니다. 이 이벤트는 [jQuery 구문](#) 장 에서 이미 설명했습니다.

### **click()**

이 **click()** 메소드는 이벤트 핸들러 함수를 HTML 요소에 첨부합니다.

이 기능은 사용자가 HTML 요소를 클릭하면 실행됩니다.

다음 예제는 다음과 같습니다. **click** 이벤트가 **<p>** 요소에서 발생하는 경우 ; 현재 **<p>** 요소를 숨 깁니다 :

```
<!DOCTYPE html>  
<html>  
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  
<script>  
$(document).ready(function(){  
    $("#p").click(function(){  
        $(this).hide();  
    });  
});  
</script>  
</head>  
<body>  
  
<p>If you click on me, I will disappear.</p>  
<p>Click me away!</p>  
<p>Click me too!</p>  
  
</body>  
</html>
```

### **dblclick ()**

이 `dblclick()` 메소드는 이벤트 핸들러 함수를 HTML 요소에 첨부합니다.

이 함수는 사용자가 HTML 요소를 두 번 클릭하면 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").dblclick(function(){
        $(this).hide();
    });
});
</script>
</head>
<body>

<p>If you double-click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

## mouseenter()

이 `mouseenter()` 메소드는 이벤트 핸들러 함수를 HTML 요소에 첨부합니다.

마우스 포인터가 HTML 요소에 들어갈 때 이 함수가 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#p1").mouseenter(function(){
        alert("You entered p1!");
    });
});
</script>
</head>
<body>

<p id="p1"> Enter this paragraph.</p>

</body>
</html>
```

## mouseleave()

이 `mouseleave()` 메소드는 이벤트 핸들러 함수를 HTML 요소에 첨부합니다.

마우스 포인터가 HTML 요소를 떠날 때 함수가 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("#p1").mouseleave(function(){
        alert("Bye! You now leave p1!");
    });
});
</script>
</head>
<body>

<p id="p1"> This is a paragraph.</p>

</body>
</html>
```

## mousedown()

이 `mousedown()` 메소드는 이벤트 핸들러 함수를 HTML 요소에 첨부합니다.

마우스가 HTML 요소 위에 있는 동안 왼쪽, 가운데 또는 오른쪽 마우스 버튼을 누르면 기능이 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("#p1").mousedown(function(){
        alert("Mouse down over p1!");
    });
});
</script>
</head>
<body>

<p id="p1"> This is a paragraph.</p>

</body>
</html>
```

## mouseup()

이 `mouseup()` 메소드는 이벤트 핸들러 함수를 HTML 요소에 첨부합니다.

마우스가 HTML 요소 위에 있는 동안 왼쪽, 가운데 또는 오른쪽 마우스 버튼을 놓으면 기능이 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#p1").mouseup(function(){
        alert("Mouse up over p1!");
    });
});
</script>
</head>
<body>

<p id="p1"> This is a paragraph.</p>

</body>
</html>
```

## hover()

이 `hover()` 방법은 두 가지 기능을 수행하며 `mouseenter()` 및 `mouseleave()` 방법의 조합입니다.

첫 번째 함수는 마우스가 HTML 요소에 들어갈 때 실행되고 두 번째 함수는 마우스가 HTML 요소를 떠날 때 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#p1").hover(function(){
        alert("You entered p1!");
    },
    function(){
        alert("Bye! You now leave p1!");
    });
});
});
```

```
</script>
</head>
<body>

<p id="p1"> This is a paragraph.</p>

</body>
</html>
```

## focus()

이 `focus()` 메소드는 이벤트 핸들러 함수를 HTML 양식 필드에 첨부합니다.

양식 필드에 포커스가 있으면 함수가 실행됩니다.

## blur()

이 `blur()` 메소드는 이벤트 핸들러 함수를 HTML 양식 필드에 첨부합니다.

양식 필드에 포커스가 없으면 함수가 실행됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("input").focus(function(){
        $(this).css("background-color", "yellow");
    });
    $("input").blur(function(){
        $(this).css("background-color", "green");
    });
});
</script>
</head>
<body>

Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email">

</body>
</html>
```

# on() 메소드

이 `on()` 메소드는 선택된 요소에 대해 하나 이상의 이벤트 핸들러를 첨부합니다.

클릭 이벤트를 `<p>` 요소에 첨부하십시오.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on("click", function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

`<p>` 요소에 여러 이벤트 핸들러를 연결하십시오 .

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color", "lightgray");
    },
    mouseleave: function(){
      $(this).css("background-color", "lightblue");
    },
    click: function(){
      $(this).css("background-color", "yellow");
    }
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```



# jQuery 효과- 숨기기 및 표시

숨기기, 표시, 토글, 슬라이드, 페이드 및 애니메이션. 와!

## jQuery hide() 및 show()

jQuery 를 사용하면 `hide()` and `show()` 메소드를 사용하여 HTML 요소를 숨기고 표시 할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });

  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>

<p>If you click on the "Hide" button, I will disappear.</p>
<button id="hide">Hide</button>
<button id="show">Show</button>

</body>
</html>
```

### 통사론:

```
$(selector).hide(speed, callback);

$(selector).show(speed, callback);
```

선택적인 **speed** 매개 변수는 숨기기 / 보이기 속도를 지정하며 "느린", "빠른" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 매개 변수는 `hide()` 또는 `show()` 메소드가 완료된 후 실행될 함수입니다 (나중의 장에서 콜백 함수에 대해 자세히 알아 봅니다).

다음 `hide()` 예는 다음과 같은 속도 매개 변수를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
  });
});
</script>
</head>
<body>

<button>Hide</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>

</body>
</html>
```

## jQuery toggle()

`toggle()` 메소드 를 사용하여 요소 숨기기 및 표시 간을 전환 할 수도 있습니다.

표시된 요소가 숨겨지고 숨겨진 요소가 표시됩니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").toggle();
  });
});
</script>
</head>
<body>

<button>Toggle between hiding and showing the paragraphs</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>
```

```
</body>
</html>
```

통사론:

```
$(selector).toggle(speed,callback);
```

선택적 speed 매개 변수는 "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 매개 변수는 `toggle()` 완료 후 실행되는 함수입니다.

## jQuery 효과 참조

모든 jQuery 효과에 대한 전체 개요를 보려면 [jQuery 효과 참조](#) 로 이동하십시오 .

## jQuery 효과- 페이딩

jQuery를 사용하면 가시성 내외부에서 요소를 희미하게 할 수 있습니다.

## jQuery 페이딩 방법

jQuery 를 사용하면 가시성 안과 밖에서 요소를 희미하게 할 수 있습니다.

jQuery 에는 다음과 같은 페이드 메소드가 있습니다.

- `fadeIn()`
- `fadeOut()`
- `fadeToggle()`
- `fadeTo()`

# jQuery fadeIn () 메소드

jQuery `fadeIn()` 메소드는 숨겨진 요소를 페이드하는 데 사용됩니다.

통사론:

```
$(selector).fadeIn(speed,callback);
```

선택적 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 파라미터는 페이딩이 완료된 후 실행될 함수입니다.

다음 예제는 `fadeIn()` 매개 변수가 다른 메소드 를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        $("#div1").fadeIn();
        $("#div2").fadeIn("slow");
        $("#div3").fadeIn(3000);
    });
});
</script>
</head>
<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button>Click to fade in boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;">
</div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"> </div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;">
</div>

</body>
</html>
```

# jQuery fadeOut () 메소드

jQuery `fadeOut()` 메소드는 보이는 요소를 페이드 아웃하는 데 사용됩니다.

통사론:

```
$(selector).fadeOut(speed,callback);
```

선택적 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 파라미터는 페이딩이 완료된 후 실행될 함수입니다.

다음 예제는 `fadeOut()` 매개 변수가 다른 메소드 를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeOut() with different parameters.</p>

<button>Click to fade out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;">
</div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-
color:green;"> </div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;">
</div>

</body>
</html>
```

# jQuery fadeToggle () 메소드

jQuery `fadeToggle()` 메소드는 `fadeIn()` 및 `fadeOut()` 메소드 간을 전환 합니다.

요소가 희미 해지면 희미 해 `fadeToggle()` 집니다.

요소가 희미 해지면 `fadeToggle()` 희미 해 집니다.

## 통사론:

```
$(selector).fadeToggle(speed,callback);
```

선택적 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 파라미터는 페이딩이 완료된 후 실행될 함수입니다.

다음 예제는 `fadeToggle()` 매개 변수가 다른 메소드 를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeToggle();
    $("#div2").fadeToggle("slow");
    $("#div3").fadeToggle(3000);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

<button>Click to fade in/out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;">
</div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"> </div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;">
</div>

</body>
</html>
```

# jQuery fadeTo() 메소드

jQuery `fadeTo()` 메소드를 사용하면 지정된 불투명도로 페이딩 할 수 있습니다 (0 과 1 사이의 값).

## 통사론:

```
$(selector).fadeTo(speed,opacity,callback);
```

필요한 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

`fadeTo()` 메소드의 필수 불투명도 매개 변수 는 지정된 불투명도로 페이딩을 지정합니다 (0 과 1 사이의 값).

선택적 콜백 매개 변수는 함수가 완료된 후 실행될 함수입니다.

다음 예제는 `fadeTo()` 매개 변수가 다른 메소드 를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    $("#div1").fadeTo("slow", 0.15);
    $("#div2").fadeTo("slow", 0.4);
    $("#div3").fadeTo("slow", 0.7);
  });
});
</script>
</head>
<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

<button>Click to fade in/out boxes</button><br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;">
</div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"> </div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;">
</div>

</body>
</html>
```

# jQuery 효과- 슬라이딩

jQuery 슬라이드 메소드는 요소를 위아래로 슬라이드합니다.

## jQuery 슬라이딩 방법

jQuery 를 사용하면 요소에 슬라이딩 효과를 만들 수 있습니다.

jQuery 에는 다음과 같은 슬라이드 방법이 있습니다.

- `slideDown()`
- `slideUp()`
- `slideToggle()`

## jQuery slideDown () 메소드

jQuery `slideDown()` 메소드는 요소를 아래로 슬라이드하는 데 사용됩니다.

통사론:

```
$(selector).slideDown(speed, callback);
```

선택적 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 매개 변수는 슬라이딩이 완료된 후 실행될 함수입니다.

다음 예제는 `slideDown()` 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown("slow");
  });
});
});
```



```

</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eccc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>

```

## jQuery slideUp () 메소드

jQuery `slideUp()` 메소드는 요소를 밀어 올리는 데 사용됩니다.

통사론:

```
$(selector).slideUp(speed,callback);
```

선택적 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 매개 변수는 슬라이딩이 완료된 후 실행될 함수입니다.

다음 예제는 `slideUp()` 방법을 보여줍니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideUp("slow");
  });
});
</script>

```

```

<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eccc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
}
</style>
</head>
<body>

<div id="flip">Click to slide up panel</div>
<div id="panel">Hello world!</div>

</body>
</html>

```

## jQuery slideToggle () 메소드

jQuery `slideToggle()` 메소드는 `slideDown()` 및 `slideUp()` 메소드 간을 전환 합니다.

요소가 아래로 미끄러져 있으면 `slideToggle()` 위로 미끄러집니다.

요소가 미끄러져 들어간 경우 `slideToggle()` 아래로 미칩니다.

### 통사론:

```
$(selector).slideToggle(speed, callback);
```

선택적 속도 매개 변수는 "느린", "빠른", 밀리 초 값을 가질 수 있습니다.

선택적 콜백 매개 변수는 슬라이딩이 완료된 후 실행될 함수입니다.

다음 예제는 `slideToggle()` 방법을 보여줍니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideToggle("slow");
  });
});

```

```

    });
  });
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eccc;
  border: solid 1px #c3c3c3;
}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>

<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>

</body>
</html>

```

# jQuery 효과- 애니메이션

jQuery를 사용하면 사용자 정의 애니메이션을 만들 수 있습니다.

## jQuery 애니메이션-animate () 메서드

jQuery `animate()` 메소드는 사용자 정의 애니메이션을 작성하는 데 사용됩니다.

**통사론:**

```
$(selector).animate({params}, speed, callback);
```

필수 `params` 매개 변수는 애니메이션 할 CSS 특성을 정의합니다.

선택적 속도 매개 변수는 효과의 지속 시간을 지정합니다. "slow", "fast" 또는 밀리 초 값을 사용할 수 있습니다.

선택적 콜백 매개 변수는 애니메이션이 완료된 후에 실행될 함수입니다.

다음 예제는 `animate()` 방법의 간단한 사용법을 보여줍니다. 250px의 왼쪽 속성에 도달할 때까지 `<div>` 요소를 오른쪽으로 이동합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({left: '250px'});
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>
```

기본적으로 모든 HTML 요소는 정적 위치를 가지며 이동할 수 없습니다.

위치를 조작하려면 먼저 요소의 CSS 위치 속성을 상대, 고정 또는 절대로 설정해야 합니다!

## jQuery animate ()-여러 속성 조작

여러 속성을 동시에 애니메이션 할 수 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left: '250px',
      opacity: '0.5',
      height: '150px',
    });
  });
});
</script>
</head>
<body>

<button>Start Animation</button>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>
```

```

        width: '150px'
    });
});
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>

```

## animate () 메서드로 모든 CSS 속성을 조작 할 수 있습니까?

예, 거의! 그러나 기억해야 할 중요한 사항이 하나 있습니다. 모든 속성 이름은 animate() 메서드와 함께 사용할 때 낙타로 묶어야 합니다. padding-left 대신 paddingLeft, margin-right 대신 marginRight 등을 작성해야 합니다.

또한 색상 애니메이션은 핵심 jQuery 라이브러리에 포함되지 않습니다.

색상에 애니메이션을 적용하려면 jQuery.com에서 [색상 애니메이션 플러그인](#) 을 다운로드 해야 합니다.

## jQuery animate ()-상대 값 사용

상대 값을 정의 할 수도 있습니다 (값은 요소의 현재 값을 기준으로 함). 이것은 값 앞에 + = 또는 -=를 넣어서 수행됩니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({

```

```

        left: '250px',
        height: '+=150px',
        width: '+=150px'
    });
});
});
</script>
</head>
<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>

```

## jQuery animate()-미리 정의 된 값 사용

속성의 애니메이션 값을 " **show**", " **hide**"또는 " **toggle**" 로 지정할 수도 있습니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            height: 'toggle'
        });
    });
});
</script>
</head>
<body>

<p>Click the button multiple times to toggle the animation.</p>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>

```

```
</html>
```

## jQuery animate ()-큐 기능 사용

기본적으로 jQuery 에는 애니메이션 대기열 기능이 제공됩니다.

즉 `animate()`, 서로 호출을 여러 번 작성하면 jQuery 는 이러한 메소드 호출로 "내부"큐를 생성합니다. 그런 다음 애니메이션 호출을 하나씩 실행합니다.

따라서 서로 다른 애니메이션을 수행하려는 경우 대기열 기능을 활용합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var div = $("div");
        div.animate({height: '300px', opacity: '0.4'}, "slow");
        div.animate({width: '300px', opacity: '0.8'}, "slow");
        div.animate({height: '100px', opacity: '0.4'}, "slow");
        div.animate({width: '100px', opacity: '0.8'}, "slow");
    });
});
</script>
</head>
<body>

<p>Click the button multiple times to toggle the animation.</p>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>
```

아래 예제는 먼저 `<div>`요소를 오른쪽으로 이동 한 다음 텍스트의 글꼴 크기를 증가 시킵니다.

```
<!DOCTYPE html>
<html>
```

```

<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var div = $("div");
        div.animate({left: '100px'}, "slow");
        div.animate({fontSize: '3em'}, "slow");
    });
});
</script>
</head>
<body>

<p>Click the button multiple times to toggle the animation.</p>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To
manipulate the position, remember to first set the CSS position property of the
element to relative, fixed, or absolute!</p>
<div
style="background:#98bf21;height:100px;width:100px;position:absolute;">HELLO</div>

</body>
</html>

```

# jQuery 중지 애니메이션

jQuery stop () 메서드는 애니메이션이나 효과가 끝나기 전에 중지하는 데 사용됩니다.

## jQuery stop () 메소드

jQuery stop() 메서드는 애니메이션이나 효과가 끝나기 전에 중지하는 데 사용됩니다.

이 stop() 방법은 슬라이딩, 페이딩 및 사용자 정의 애니메이션을 포함한 모든 jQuery 효과 기능에 적용됩니다.

통사론:

```
$(selector).stop(stopAll,goToEnd);
```



선택적 **stopAll** 매개 변수는 애니메이션 큐도 지워야 하는지 여부를 지정합니다. 기본값은 **false** 입니다. 즉, 활성 애니메이션 만 중지되어 대기중인 애니메이션을 나중에 수행 할 수 있습니다.

선택적 **goToEnd** 매개 변수는 현재 애니메이션을 즉시 완료할지 여부를 지정합니다. 기본값은 **false** 입니다.

따라서 기본적으로 **stop()** 메서드는 선택한 요소에서 수행중인 현재 애니메이션을 종료합니다.

다음 예제는 매개 변수 없이 **stop()** 메소드를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#stop").click(function(){
        $("#panel").stop();
    });
});
</script>
<style>
#panel, #flip {
    padding: 5px;
    font-size: 18px;
    text-align: center;
    background-color: #555;
    color: white;
    border: solid 1px #666;
    border-radius: 3px;
}

#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>

<button id="stop">Stop sliding</button>

<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

# jQuery 콜백 함수

콜백 기능은 현재 효과가 100 % 완료된 후에 실행됩니다.

## jQuery 콜백 함수

JavaScript 문은 한 줄씩 실행됩니다. 그러나 효과를 사용하면 효과가 완료되지 않은 경우에도 다음 코드 줄을 실행할 수 있습니다. 오류가 발생할 수 있습니다.

이를 방지하기 위해 콜백 함수를 만들 수 있습니다.

현재 효과가 완료된 후 콜백 함수가 실행됩니다.

전형적인 문법 : **`$ ( selector ) .hide ( speed, callback );`**

예

아래 예제에는 숨기기 효과가 완료된 후 실행될 함수 인 콜백 매개 변수가 있습니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide("slow", function(){
            alert("The paragraph is now hidden");
        });
    });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>
```

아래 예에는 콜백 매개 변수가 없으며 숨기기 효과가 완료되기 전에 경고 상자가 표시됩니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide(1000);
    alert("The paragraph is now hidden");
  });
});
</script>
</head>
<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>
</html>

```

# jQuery- 체인

jQuery 를 사용하면 액션 / 메소드를 함께 연결할 수 있습니다.

체인을 사용하면 단일 명령문 내에서 동일한 요소에서 여러 jQuery 메소드를 실행할 수 있습니다.

## jQuery 메소드 체인

지금까지 우리는 한 번에 하나씩 jQuery 문을 작성해 왔습니다.

그러나 연쇄 (chaining)라는 기술이 있습니다. 이를 통해 동일한 요소에서 여러 jQuery 명령을 하나씩 실행할 수 있습니다.

**팁 :** 이 방법으로 브라우저는 동일한 요소를 두 번 이상 찾을 필요가 없습니다.

동작을 연결하려면 단순히 이전 동작에 동작을 추가하면 됩니다.

함께 다음의 예 체인 `css()`, `slideUp()` 및 `slideDown()` 방법. "p1"요소는 먼저 빨간색으로 바뀐 다음 위로 미끄러져 내려옵니다.

```

<!DOCTYPE html>
<html>

```

```

<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#p1").css("color", "red").slideUp(2000).slideDown(2000);
    });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>
</html>

```

필요한 경우 더 많은 메소드 호출을 추가 할 수도 있습니다.

**팁 :** 체인을 연결할 때 코드 줄이 상당히 길어질 수 있습니다. 그러나 jQuery 는 구문에 대해 엄격하지 않습니다. 줄 바꿈 및 들여 쓰기를 포함하여 원하는 대로 형식을 지정할 수 있습니다.

이것은 또한 잘 작동합니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#p1").css("color", "red")
            .slideUp(2000)
            .slideDown(2000);
    });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>

<button>Click me</button>

</body>
</html>

```

jQuery는 여분의 공백을 버리고 위의 행을 하나의 긴 코드 행으로 실행합니다.

## jQuery- 컨텐츠 및 속성 가져 오기

jQuery에는 HTML 요소 및 속성을 변경하고 조작하는 강력한 방법이 포함되어 있습니다.

## jQuery DOM 조작

jQuery의 매우 중요한 부분 중 하나는 DOM을 조작할 수 있다는 것입니다.

jQuery에는 요소 및 속성에 쉽게 액세스하고 조작할 수 있는 다양한 DOM 관련 메소드가 제공됩니다.

### DOM = 문서 객체 모델

DOM은 HTML 및 XML 문서에 액세스하기 위한 표준을 정의합니다.

*"W3C DOM (문서 객체 모델)은 플랫폼과 언어 중립적 인터페이스로 프로그램과 스크립트가 컨텐츠, 구조 및 문서 스타일."*

## 컨텐츠 가져 오기-text (), html () 및 val ()

DOM 조작을 위한 간단하지만 유용한 세 가지 jQuery 메소드는 다음과 같습니다.

- **text()** -선택된 요소의 텍스트 내용을 설정하거나 반환
- **html()** -선택한 요소의 내용을 설정하거나 반환합니다 (HTML 마크 업 포함).
- **val()** -양식 필드의 값을 설정하거나 반환

다음 예제는 jQuery **text()**와 **html()**메소드로 컨텐츠를 얻는 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```

<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
});
</script>
</head>
<body>

<p id="test">This is some <b>bold</b> text in a paragraph.</p>

<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>

</body>
</html>

```

다음 예제는 jQuery `val()` 메소드를 사용하여 입력 필드의 값을 얻는 방법을 보여줍니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Value: " + $("#test").val());
    });
});
</script>
</head>
<body>

<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>

<button>Show Value</button>

</body>
</html>

```

## 속성 가져 오기-attr()

jQuery `attr()` 메소드는 속성 값을 가져 오는 데 사용됩니다.

다음 예제는 링크에서 href 속성 값을 얻는 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    alert($("#w3s").attr("href"));
  });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" id="w3c">W3Schools.com</a></p>

<button>Show href Value</button>

</body>
</html>
```

## jQuery- 컨텐츠 및 속성 설정

### 내용 설정-text (), html () 및 val ()

이전 페이지와 동일한 세 가지 방법을 사용하여 컨텐츠를 설정합니다.

- **text()** -선택된 요소의 텍스트 내용을 설정하거나 반환
- **html()** -선택한 요소의 내용을 설정하거나 반환합니다 (HTML 마크 업 포함).
- **val()** -양식 필드의 값을 설정하거나 반환

다음은 jQuery 로 콘텐츠를 설정하는 방법을 보여줍니다 **text()**, **html()** 및 **val()** 방법 :

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");
  });
});
```

```

    $("#btn2").click(function(){
        $("#test2").html("<b>Hello world!</b>");
    });
    $("#btn3").click(function(){
        $("#test3").val("Dolly Duck");
    });
});
</script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>

```

## text(), html() 및 val()에 대한 콜백 함수

위의 세 가지 jQuery 메소드 `text()`, `html()` 및 `val()` 모두 콜백 함수와 함께 제공됩니다. 콜백 함수에는 선택된 요소 목록의 현재 요소 색인과 원래 (이전) 값의 두 가지 매개 변수가 있습니다. 그런 다음 함수에서 새 값으로 사용하려는 문자열을 반환합니다.

다음의 예를 보여줍니다 `text()` 및 `html()` 콜백 기능 :

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        $("#test1").text(function(i, origText){
            return "Old text: " + origText + " New text: Hello world!
            (index: " + i + ")";
        });
    });

    $("#btn2").click(function(){
        $("#test2").html(function(i, origText){
            return "Old html: " + origText + " New html: Hello <b>world!</b>
            (index: " + i + ")";
        });
    });
});

```



```
});
</script>
</head>
<body>

<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>

<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>

<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>

</body>
</html>
```

## 속성 설정-attr ()

jQuery **attr()** 메소드는 속성 값을 설정 / 변경하는 데에도 사용됩니다.

다음 예제는 링크에서 href 속성 값을 변경 (설정)하는 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
    });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" id="w3c">W3Schools.com</a></p>

<button>Change href Value</button>

<p>Mouse over the link (or click on it) to see that the value of the href attribute has changed.</p>

</body>
</html>
```

이 **attr()** 방법을 사용하면 동시에 여러 속성을 설정할 수 있습니다.

다음 예제는 href 및 title 속성을 동시에 설정하는 방법을 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#w3s").attr({
      "href" : "https://www.w3schools.com/jquery/",
      "title" : "W3Schools jQuery Tutorial"
    });
  });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" title="some title"
id="w3c">W3Schools.com</a></p>

<button>Change href and title</button>

<p>Mouse over the link to see that the href attribute has changed and a title
attribute is set.</p>

</body>
</html>
```

## attr ()에 대한 콜백 함수

jQuery 메소드 **attr()**에는 콜백 함수도 함께 제공됩니다. 콜백 함수에는 선택된 요소 목록의 현재 요소 색인과 원래 (이전) 속성 값의 두 가지 매개 변수가 있습니다. 그런 다음 함수에서 새 속성 값으로 사용할 문자열을 반환합니다.

다음 예제 **attr()**는 콜백 함수를 보여줍니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#w3s").attr("href", function(i, origValue){
      return origValue + "/jquery/";
    });
  });
});
</script>
```

```
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" id="w3c">W3Schools.com</a></p>

<button>Change href Value</button>

<p>Mouse over the link (or click on it) to see that the value of the href
attribute has changed.</p>

</body>
</html>
```

## jQuery- 요소 추가

jQuery 를 사용하면 새로운 요소 / 콘텐츠를 쉽게 추가 할 수 있습니다.

### 새로운 HTML 콘텐츠 추가

새 콘텐츠를 추가하는 데 사용되는 네 가지 jQuery 메소드를 살펴 보겠습니다.

- **append()** -선택한 요소의 끝에 내용을 삽입합니다
- **prepend()** -선택한 요소의 시작 부분에 내용을 삽입합니다
- **after()** -선택한 요소 뒤에 내용을 삽입
- **before()** -선택한 요소 앞에 내용을 삽입

## jQuery append() 메소드

jQuery **append()** 메소드는 선택한 HTML 요소의 끝에 콘텐츠를 삽입합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
```

```

    $("p").append("Some appended text.");
  });
  $("#btn2").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>

<button id="btn1">Append text</button>
<button id="btn2">Append text items</button>

</body>
</html>

```

## jQuery prepend () 메소드

jQuery `prepend()` 메소드는 선택한 HTML 요소의 시작 부분에 콘텐츠를 삽입합니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").prepend("<b>Prepended text<b>.");
  });
  $("#btn2").click(function(){
    $("ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<ol>
  <li>List item 1</li>
  <li>List item 2</li>

```

```
<li>List item 3</li>
</ol>

<button id="btn1">Append text</button>
<button id="btn2">Append text items</button>

</body>
</html>
```

## append() 및 prepend()를 사용하여 몇 가지 새로운 요소 추가

위의 두 예제 모두에서 선택한 HTML 요소의 시작 / 끝에 일부 텍스트 / HTML 만 삽입했습니다.

그러나 `append()` 및 `prepend()` 메소드 모두 무한한 수의 새로운 요소를 매개 변수로 사용할 수 있습니다. 새로운 요소는 텍스트 / HTML (위 예제에서와 같이), jQuery 또는 JavaScript 코드 및 DOM 요소로 생성 할 수 있습니다.

다음 예에서는 몇 가지 새로운 요소를 만듭니다. 요소는 text / HTML, jQuery 및 JavaScript / DOM 으로 작성됩니다. 그런 다음 `append()` 메소드를 사용하여 텍스트에 새 요소를 추가합니다 (이것도 효과가 `prepend()` 있었습니다).

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
function appendText() {
    var txt1 = "<p>Text.</p>"; // Create element with HTML
    var txt2 = $("<p></p>").text("Text."); // Create with jQuery
    var txt3 = document.createElement("p"); // Create with DOM
    txt3.innerHTML = "Text.";
    $("body").append(txt1, txt2, txt3); // Append the new elements
}
</script>
</head>
<body>

<p>This is a paragraph.</p>
<button onclick="appendText()">Append text</button>

</body>
</html>
```

# jQuery after() 및 before() 메소드

jQuery **after()** 메소드는 선택한 HTML 요소 다음에 콘텐츠를 삽입합니다.

jQuery **before()** 메소드는 선택한 HTML 요소보다 먼저 콘텐츠를 삽입합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#img").before("<b>Before<b>");
  });

  $("#btn2").click(function(){
    $("#img").after("<i>After</i>");
  });
});
</script>
</head>
<body>

<br><br>

<button id="btn1">Insert before</button>
<button id="btn2">Insert after</button>

</body>
</html>
```

## after() 및 before()를 사용하여 몇 가지 새로운 요소 추가

또한 **after()** 및 **before()** 메소드 모두 무한한 수의 새로운 요소를 매개 변수로 사용할 수 있습니다. 새 요소는 텍스트 / HTML (위의 예에서와 같이), jQuery 또는 JavaScript 코드 및 DOM 요소를 사용하여 생성 할 수 있습니다.

다음 예에서는 몇 가지 새로운 요소를 만듭니다. 요소는 text / HTML, jQuery 및 JavaScript / DOM 으로 작성됩니다. 그런 다음 **after()** 메소드 를 사용하여 텍스트에 새 요소를 삽입합니다 (이것도 효과가 **before()** 있었습니다).

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
function afterText() {
    var txt1 = "<b>I </b>";           // Create element with HTML
    var txt2 = $("<i></i>").text("love "); // Create with jQuery
    var txt3 = document.createElement("b"); // Create with DOM
    txt3.innerHTML = "jQuery!";
    $("img").after(txt1, txt2, txt3); // Insert new elements after <img>
}
</script>
</head>
<body>



<p>Click the button to insert text after the image.</p>

<button onclick="afterText()">Insert after</button>

</body>
</html>
```

## jQuery- CSS 클래스 가져 오기 및 설정

jQuery를 사용하면 요소 스타일을 쉽게 조작 할 수 있습니다.

## CSS를 조작하는 jQuery

jQuery 에는 CSS 조작을위한 몇 가지 방법이 있습니다. 다음과 같은 방법을 살펴 보겠습니다.

- **addClass()** -선택한 요소에 하나 이상의 클래스를 추가합니다
- **removeClass()** -선택한 요소에서 하나 이상의 클래스를 제거합니다
- **toggleClass()** -선택한 요소에서 클래스 추가 / 제거 간 전환
- **css()** -스타일 속성을 설정하거나 반환

# 스타일 시트 예

이 페이지의 모든 예제에 다음 스타일 시트가 사용됩니다.

```
.important {  
  font-weight: bold;  
  font-size: xx-large;  
}  
  
.blue {  
  color: blue;  
}
```

## jQuery addClass() 메소드

다음 예제는 클래스 속성을 다른 요소에 추가하는 방법을 보여줍니다. 물론 클래스를 추가 할 때 여러 요소를 선택할 수 있습니다.

```
<!DOCTYPE html>  
<html>  
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  
<script>  
$(document).ready(function(){  
  $("button").click(function(){  
    $("h1, h2, p").addClass("blue");  
    $("div").addClass("important");  
  });  
});  
</script>  
<style>  
.important {  
  font-size: xx-large;  
  font-weight: bold;  
}  
  
.blue {  
  color: blue;  
}  
</style>  
</head>  
<body>  
  
<h1>Heading 1</h1>  
<h2>Heading 2</h2>
```



```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<div>This is some important text!</div><br>

<button>Add classes to elements</button>

</body>
</html>
```

**addClass()** 메소드 내에 여러 클래스를 지정할 수도 있습니다 .

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").addClass("important blue");
  });
});
</script>
<style>
.important {
  font-size: xx-large;
  font-weight: bold;
}

.blue {
  color: blue;
}
</style>
</head>
<body>

<div id="div1">This is some text.</div>
<div id="div1">This is some text.</div>
<br>

<button>Add classes to first element</button>

</body>
</html>
```

## jQuery removeClass() 메소드

다음 예제는 다른 요소에서 특정 클래스 속성을 제거하는 방법을 보여줍니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("h1, h2, p").removeClass("blue");
    });
});
</script>
<style>
.important {
    font-size: xx-large;
    font-weight: bold;
}

.blue {
    color: blue;
}
</style>
</head>
<body>

<h1 class="blue">Heading 1</h1>
<h2 class="blue">Heading 2</h2>

<p class="blue">This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Remove class from elements</button>

</body>
</html>

```

## jQuery toggleClass () 메소드

다음 예제는 jQuery `toggleClass()` 메소드를 사용하는 방법을 보여줍니다. 이 메소드는 선택한 요소에서 클래스 추가 / 제거간에 전환합니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("h1, h2, p").toggleClass("blue");
    });
});
</script>

```

```
<style>
.blue {
  color: blue;
}
</style>
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Toggle class</button>

</body>
</html>
```

# jQuery- 요소 제거

jQuery 를 사용하면 기존 HTML 요소를 쉽게 제거 할 수 있습니다.

## 요소 / 콘텐츠 제거

요소와 콘텐츠를 제거하기 위해 주로 두 가지 jQuery 메소드가 있습니다.

- `remove()` -선택한 요소 및 해당 하위 요소를 제거합니다
- `empty()` -선택한 요소에서 자식 요소를 제거합니다

## jQuery remove() 메소드

jQuery `remove()` 메소드는 선택한 요소와 해당 하위 요소를 제거합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
```

```

<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").remove();
    });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow">This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>
</div>
<br>

<button>Remove div element</button>

</body>
</html>

```

## jQuery empty () 메소드

jQuery `empty()` 메소드는 선택한 요소의 하위 요소를 제거합니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").empty();
    });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow">This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>
</div>
<br>

<button>Empty the div element</button>

</body>
</html>

```

# 제거 할 요소 필터링

jQuery `remove()` 메소드는 하나의 매개 변수를 허용하므로 제거 할 요소를 필터링 할 수 있습니다.

매개 변수는 jQuery 선택기 구문 중 하나 일 수 있습니다.

다음 예제는 다음을 사용 하여 `class="test"` 인 `<p>` 모든 요소를 제거합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").remove(".test");
  });
});
</script>
<style>
.test {
  color: red;
  font-size: 20px;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p class="test">This is another paragraph.</p>
<p class="test">This is another paragraph.</p>

<button>Remove all p elements with class="test"</button>

</body>
</html>
```

이 예는 모두 제거 `<p>`와 요소 `class="test"` 또는 `class="demo"`:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").remove(".test, .demo");
  });
});
```

```

});
</script>
<style>
.test {
  color: red;
  font-size: 20px;
}

.demo {
  color: green;
  font-size: 25px;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p class="test">This is p element with class="test".</p>
<p class="test">This is p element with class="test".</p>
<p class="demo">This is p element with class="demo".</p>

<button>Remove all p elements with class="test" and class="demo"</button>

</body>
</html>

```

# jQuery- css() 메소드

## jQuery CSS () 메소드

이 `css()` 메소드는 선택된 요소에 대해 하나 이상의 스타일 특성을 설정하거나 리턴합니다.

## CSS 속성 반환

지정된 CSS 속성 값을 반환하려면 다음 구문을 사용하십시오.

```
css("propertyname");
```

다음 예제는 FIRST 일치 요소의 배경색 값을 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert("Background color = " + $("p").css("background-color"));
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>

<p style="background-color:#ff0000">This is paragraph.</p>
<p style="background-color:#00ff00">This is paragraph.</p>
<p style="background-color:#0000ff">This is paragraph.</p>

<button>Return background-color of p</button>

</body>
</html>
```

## CSS 속성 설정

지정된 CSS 속성을 설정하려면 다음 구문을 사용하십시오.

```
css("propertyname","value");
```

다음 예제는 일치하는 모든 요소의 배경색 값을 설정합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css("background-color", "yellow");
  });
});
</script>
</head>
<body>
```

```
<p>This is a paragraph.</p>

<p style="background-color:#ff0000">This is paragraph.</p>
<p style="background-color:#00ff00">This is paragraph.</p>
<p style="background-color:#0000ff">This is paragraph.</p>

<p>This is paragraph.</p>

<button>Set background-color of p</button>

</body>
</html>
```

## 여러 CSS 속성 설정

여러 CSS 속성을 설정하려면 다음 구문을 사용하십시오.

```
css({"propertyname":"value","propertyname":"value",...});
```

다음 예제는 일치하는 모든 요소에 배경색과 글꼴 크기를 설정합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").css({"background-color": "yellow", "font-size": "200%"});
  });
});
</script>
</head>
<body>

<p>This is a paragraph.</p>

<p style="background-color:#ff0000">This is paragraph.</p>
<p style="background-color:#00ff00">This is paragraph.</p>
<p style="background-color:#0000ff">This is paragraph.</p>

<p>This is paragraph.</p>

<button>Set multiple styles for p</button>

</body>
</html>
```



# jQuery- 측정 기준

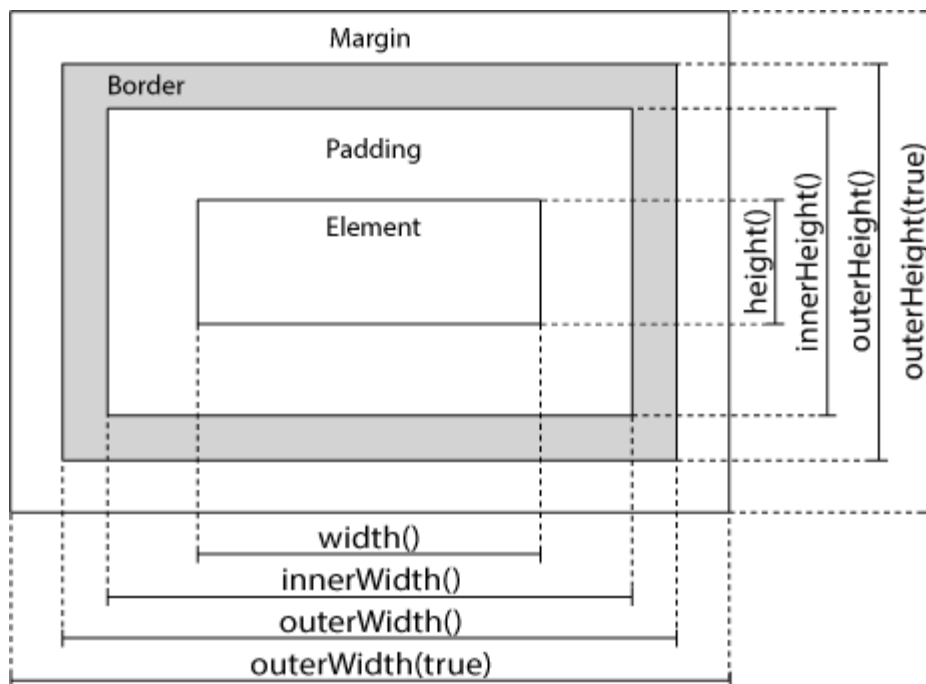
jQuery 를 사용하면 요소 및 브라우저 창 크기로 작업하기가 쉽습니다.

## jQuery 차원 메소드

jQuery 에는 차원 작업을위한 몇 가지 중요한 방법이 있습니다.

- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`

## jQuery 차원



# jQuery width () 및 height () 메서드

이 `width()` 메서드는 요소의 너비를 설정하거나 반환합니다 (패딩, 테두리 및 여백 제외).

이 `height()` 메서드는 요소의 높이를 설정하거나 반환합니다 (패딩, 테두리 및 여백 제외).

다음 예제는 지정된 `<div>` 요소의 너비와 높이를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var txt = "";
    txt += "Width: " + $("#div1").width() + "<br>";
    txt += "Height: " + $("#div1").height();
    $("#div1").html(txt);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>

<button>Display dimensions of div</button>

<p>width() - returns the width of an element.</p>
<p>height() - returns the height of an element.</p>

</body>
</html>
```

# jQuery innerWidth() 및 innerHeight() 메서드

이 `innerWidth()` 메서드는 요소의 너비를 반환합니다 (패딩 포함).

이 `innerHeight()` 메서드는 요소의 높이를 반환합니다 (패딩 포함).

다음 예제는 지정된 `<div>` 요소의 내부 너비 / 높이를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        var txt = "";
        txt += "Width of div: " + $("#div1").width() + "<br>";
        txt += "Height of div: " + $("#div1").height() + "<br>";
        txt += "Inner width of div: " + $("#div1").innerWidth() + "<br>";
        txt += "Inner height of div: " + $("#div1").innerHeight();
        $("#div1").html(txt);
    });
});
</script>
<style>
#div1 {
    height: 100px;
    width: 300px;
    padding: 10px;
    margin: 3px;
    border: 1px solid blue;
    background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>

<button>Display dimensions of div</button>

<p>innerWidth() - returns the width of an element (includes padding).</p>
<p>innerHeight() - returns the height of an element (includes padding).</p>

</body>
</html>
```

# jQuery outerWidth () 및 outerHeight () 메소드

이 `outerWidth()` 메서드는 요소의 너비를 반환합니다 (패딩 및 테두리 포함).

이 `outerHeight()` 메서드는 요소의 높이를 반환합니다 (패딩 및 테두리 포함).

다음 예제는 지정된 `<div>` 요소의 외부 너비 / 높이를 반환합니다 .

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var txt = "";
    txt += "Width of div: " + $("#div1").width() + "<br>";
    txt += "Height of div: " + $("#div1").height() + "<br>";
    txt += "Outer width of div: " + $("#div1").outerWidth() + "<br>";
    txt += "Outer height of div: " + $("#div1").outerHeight();
    $("#div1").html(txt);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>

<button>Display dimensions of div</button>

<p>outerWidth() - returns the width of an element (includes padding and border).</p>
<p>outerHeight() - returns the height of an element (includes padding and border).</p>

</body>
</html>
```

이 `outerWidth(true)` 메서드는 요소의 너비를 반환합니다 (패딩, 테두리 및 여백 포함).

이 `outerHeight(true)` 메서드는 요소의 높이 (패딩, 테두리 및 여백 포함)를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var txt = "";
    txt += "Width of div: " + $("#div1").width() + "<br>";
    txt += "Height of div: " + $("#div1").height() + "<br>";
    txt += "Outer width of div (margin included): " + $("#div1").outerWidth(true)
    + "<br>";
    txt += "Outer height of div (margin included): " +
    $("#div1").outerHeight(true);
    $("#div1").html(txt);
  });
});
</script>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>

<button>Display dimensions of div</button>

<p>outerWidth() - returns the width of an element (includes padding, border, and margin).</p>
<p>outerHeight() - returns the height of an element (includes padding, border, and margin).</p>

</body>
</html>
```

## jQuery 더 width () 및 height ()

다음 예제는 문서 (HTML 문서)와 창 (브라우저 뷰포트)의 너비와 높이를 반환합니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var txt = "";
        txt += "Document width/height: " + $(document).width();
        txt += "x" + $(document).height() + "\n";
        txt += "Window width/height: " + $(window).width();
        txt += "x" + $(window).height();
        alert(txt);
    });
});
</script>
</head>
<body>

<div id="div1"></div>

<button>Display dimensions of document and window</button>

</body>
</html>

```

다음 예제는 지정된 `<div>` 요소 의 너비와 높이를 설정합니다.

```

<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").width(500).height(500);
    });
});
</script>
<style>
#div1 {
    height: 100px;
    width: 300px;
    padding: 10px;
    margin: 3px;
    border: 1px solid blue;
    background-color: lightblue;
}
</style>
</head>
<body>

<div id="div1"></div>
<br>

```

```
<button>Resize div</button>

</body>
</html>
```

# jQuery 트래버스

## 트래버 싱이란 무엇입니까?

"이동"을 의미하는 jQuery 트래버스는 다른 요소와의 관계에 따라 HTML 요소를 "찾기"(또는 선택)하는 데 사용됩니다. 하나의 선택으로 시작하여 원하는 요소에 도달 할 때까지 해당 선택을 통해 이동하십시오.

아래 이미지는 HTML 페이지를 트리 (DOM 트리)로 보여줍니다. jQuery 트래버싱을 사용하면 선택한 (현재) 요소에서 시작하여 트리에서 쉽게 위 (조상), 아래로 (하위) 및 옆으로 (형제) 이동할 수 있습니다. 이 움직임을 DOM 트리 통과 또는 이동이라고 합니다.

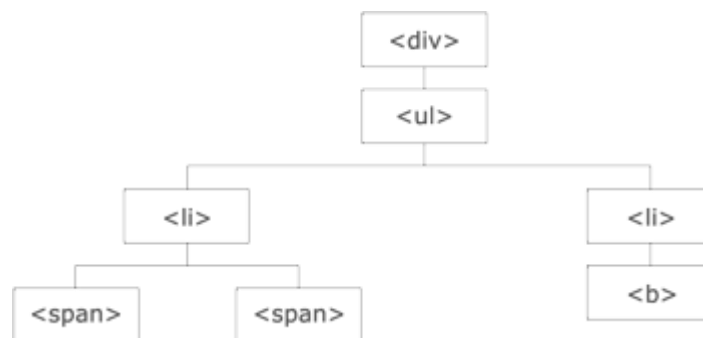


그림 설명 :

- <div> 요소는 <ul> 의 **부모(parent)** 이며 그 안의 모든 요소 의 **조상(ancestor)** 입니다.
- <UL> 요소는 <LI> 요소 및 하위 <div>의 **부모** 이다.
- 왼쪽 <LI> 요소는 <SPAN>의 **부모**이고, <UL>의 **자식(child)** 이며, <div>의 **자손(descendant)** 이다.
- <span> 요소는 왼쪽 <LI> **자식** 이며, <UL> 및 <div>의 **자손** 이다.
- 두 개의 <li> 요소는 **형제(siblings)**입니다 (동일한 부모를 공유합니다)
- 오른쪽 <LI> 요소는 <B>의 **부모**이며, <UL>의 **자식** 이고, <div>의 **자손** 이다.

- `<b>` 요소는 오른쪽 `<LI>` 자식이며 `<UL>` 및 `<div>`의 자손 이다.

조상은 부모, 조부모, 증조부모 등입니다.  
자손은 자녀, 손자, 증손자 등입니다.  
형제는 같은 부모를 공유합니다.

## DOM 탐색

jQuery 는 DOM 을 순회 할 수 있는 다양한 메소드를 제공합니다.

순회 방법의 가장 큰 범주는 트리 순회입니다.

다음 장에서는 DOM 트리에서 위, 아래 및 옆으로 이동하는 방법을 보여줍니다.

## jQuery 트래버스- 조상

jQuery 를 사용하면 DOM 트리를 탐색하여 요소의 조상을 찾을 수 있습니다.

조상은 부모, 조부모, 증조부모 등입니다.

## DOM 트리 탐색

DOM 트리를 순회하는 데 유용한 세 가지 jQuery 메소드는 다음과 같습니다.

- `parent()`
- `parents()`
- `parentsUntil()`



# jQuery parent () 메소드

이 `parent()` 메소드는 선택한 요소의 직접 상위 요소를 리턴합니다.

이 메소드는 단일 레벨의 DOM 트리 만 순회합니다.

다음 예제는 각 `<span>` 요소 의 직접 부모 요소를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parent().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="ancestors">
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>

  <div style="width:500px;">div (grandparent)
    <p>p (direct parent)
      <span>span</span>
    </p>
  </div>
</div>

</body>
</html>
```

# jQuery parents () 메소드

이 `parents()` 메서드는 선택한 요소의 모든 상위 요소를 문서의 루트 요소 (`<html>`) 까지 반환합니다 .

다음 예제는 모든 `<span>` 요소 의 모든 조상을 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parents().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>

<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>

<!-- The outer red border, before the body element, is the html element (also an ancestor) -->
</body>
</html>
```

선택적 매개 변수를 사용하여 조상 검색을 필터링 할 수도 있습니다.

다음 예제는 `<span>` 요소의 조상인 모든 `<ul>` 요소를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
```

```

display: block;
border: 2px solid lightgrey;
color: lightgrey;
padding: 5px;
margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("span").parents("ul").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>

<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>

</body>
</html>

```

## jQuery parentsUntil() 메소드

이 `parentsUntil()` 메소드는 주어진 두 인수 사이의 모든 상위 요소를 리턴합니다.

다음 예제는 a `<span>`와 `<div>` 요소 사이의 모든 조상 요소를 반환합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.ancestors * {
display: block;
border: 2px solid lightgrey;
color: lightgrey;
padding: 5px;
margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("span").parentsUntil("div").css({"color": "red", "border": "2px solid red"});
});

```

```
</script>
</head>

<body class="ancestors">body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>

</body>
</html>
```

## jQuery 트래버싱 - 자손

jQuery 를 사용하면 DOM 트리를 탐색하여 요소의 하위 항목을 찾을 수 있습니다.

자손은 자녀, 손자, 증손자 등입니다.

## DOM 트리 탐색

DOM 트리를 순회하는 데 유용한 두 가지 jQuery 메소드는 다음과 같습니다.

- `children()`
- `find()`

## jQuery children () 메소드

이 `children()` 메소드는 선택한 요소의 모든 직접 하위를 리턴합니다.

이 메소드는 단일 레벨에서 DOM 트리 아래로만 이동합니다.

다음 예제는 각 `<div>` 요소의 직계 자식 인 모든 요소를 반환합니다.

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
  $("div").children().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>

```

선택적 매개 변수를 사용하여 하위 검색을 필터링 할 수도 있습니다.

다음 예제는 `<div>` 요소의 하위 요소 인 `<p>` 클래스 이름이 "first"인 모든 요소를 반환합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
  $("div").children("p.first").css({"color": "red", "border": "2px solid red"});
});
</script>
</body>
</html>

```

```
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p class="first">p (child)
    <span>span (grandchild)</span>
  </p>
  <p class="second">p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

## jQuery find() 메소드

이 `find()` 메서드는 선택한 요소의 하위 요소를 마지막 하위까지 반환합니다.

다음 예제는 다음의 `<div>`의 자손 인 모든 `<span>` 요소를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").find("span").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
```

```
</div>

</body>
</html>
```

다음 예제는 `<div>`의 모든 자손을 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").find("*").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

## jQuery 트래버스- 형제

jQuery 를 사용하면 DOM 트리에서 옆으로 가로 질러 요소의 형제를 찾을 수 있습니다.

형제는 같은 부모를 공유합니다.

# DOM 트리에서 옆으로 순회

DOM 트리에는 옆으로 순회하는 데 유용한 jQuery 메소드가 많이 있습니다.

- `siblings()`
- `next()`
- `nextAll()`
- `nextUntil()`
- `prev()`
- `prevAll()`
- `prevUntil()`

## jQuery `siblings()` 메소드

이 `siblings()` 메서드는 선택한 요소의 모든 형제 요소를 반환합니다.

다음 예제는 `<h2>`의 모든 형제 요소를 반환합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").siblings().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
```



```
</div>

</body>
</html>
```

선택적 매개 변수를 사용하여 형제 검색을 필터링 할 수도 있습니다.

다음 예제는 `<h2>` 해당 요소의 모든 형제 `<p>` 요소를 리턴합니다.

```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").siblings("p").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>
```

## jQuery next() 메소드

이 `next()` 메서드는 선택한 요소의 다음 형제 요소를 반환합니다.

다음 예제는 `<h2>` 다음 형제를 반환합니다.

```
<!DOCTYPE html>
<html>
```

```

<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>
$(document).ready(function(){
  $("h2").next().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <p>p</p>
</div>

</body>
</html>

```

## jQuery nextAll() 메소드

이 **nextAll()** 메소드는 선택한 요소의 다음 형제 요소를 모두 반환합니다.

다음 예제는 **<h2>**의 다음 형제 요소를 모두 반환합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></s
cript>
<script>

```

```

$(document).ready(function(){
    $("h2").nextAll().css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
    <p>p</p>
    <span>span</span>
    <h2>h2</h2>
    <h3>h3</h3>
    <p>p</p>
</div>

</body>
</html>

```

## jQuery nextUntil() 메소드

이 `nextUntil()` 메소드는 주어진 두 인수 사이의 다음 형제 요소를 모두 반환합니다.

다음 예제는 `<h2>`와 `<h6>` 요소 사이의 모든 형제 요소를 반환합니다.

```

<!DOCTYPE html>
<html>
<head>
<style>
.siblings * {
    display: block;
    border: 2px solid lightgrey;
    color: lightgrey;
    padding: 5px;
    margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("h2").nextUntil("h6").css({"color": "red", "border": "2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
    <p>p</p>
    <span>span</span>
    <h2>h2</h2>
    <h3>h3</h3>
    <h4>h4</h4>

```

```
<h5>h5</h5>
<h6>h6</h6>
<p>p</p>
</div>

</body>
</html>
```

## jQuery prev(), prevAll() & prevUntil() 메소드

`prev()`, `prevAll()` 및 `prevUntil()` 방법은 위와 같지만 역 기능을 가진 방법처럼 작동 : 그들은 (대신 앞으로의 DOM 트리에 있는 형제 요소를 따라 뒤쪽으로 통과) 이전 형제 요소를 반환합니다.

## jQuery 트래버싱- 필터링

### first (), last (), eq (), filter () 및 not () 메소드

가장 기본적인 필터링 방법은 `first()`, `last()` 그리고 `eq()` 당신이 요소 그룹에서의 위치에 따라 특정 요소를 선택할 수 있는.

다른 필터링 방법, 좋아 `filter()` 하고 `not()` 당신이 특정 기준과 일치하거나 일치하지 않는 요소를 선택할 수 있습니다.

## jQuery first() 메소드

이 `first()` 메소드는 지정된 요소의 첫 번째 요소를 리턴합니다.

다음 예제는 첫 번째 `<div>` 요소를 선택합니다.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
```

```
    $("div").first().css("background-color", "yellow");
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2>
  <h3>h3</h3>
  <h4>h4</h4>
  <h5>h5</h5>
  <h6>h6</h6>
  <p>p</p>
</div>

</body>
</html>
```