

우리가 HTML태그의 속성이라고 부르는 것을 Javascript에서는 값을 갖는 속성의 경우 **attribute**, 값을 갖지 않고 속성 이름만 지정하는 경우 **property**라고 합니다. **attribute**는 ``태그의 `src`속성과 같은 것이 있고 **property**는 `disabled`, `readonly`, `checked` 등의 속성이 있습니다.

#01. 요소의 판별

`index()` 함수

특정 요소가 부모 태그 안에서 갖는 인덱스번호를 리턴하는 함수 (0부터 시작함)

```
<div>
  <a href="#">...</a>
  <div id='hello'></div>
  <p class="memo"></p>
</div>

<script>
  // 부모 <div>를 기준으로 2번째 요소이므로 인덱스 1이 리턴된다.
  var idx = $("#hello").index();
</script>
```

`$(e.currentTarget)`

복수 요소에 대한 이벤트에 전달된 콜백함수 안에서 이벤트가 발생한 주체를 의미하는 객체

```
<button class='btn'>button1</button>
<button class='btn'>button1</button>
<button class='btn'>button1</button>

<script>
  // `btn`이라는 클래스를 갖는 모든 요소에 대해서 일괄적으로 적용되는 이벤트
  $('btn').on("click", (e) => {
    // 버튼은 한번에 하나씩만 누를 수 있다.
    // 이 안에서 $(e.currentTarget)는 클릭된 주체를 의미한다.
  });
</script>
```

#02. HTML 요소의 속성 제어

특정 요소에 적용되어 있는 속성값 조회하기

```
var foo = $("#bar").attr('속성이름');
```

특정 요소의 적용하기

개별적용

```
$("#bar").attr('속성이름', '값');
```

일괄적용 (json형태 사용)

```
$("#bar").attr({
  속성이름: '값',
  속성이름: '값',
  속성이름: '값'
  ...
});
```

- 속성이름에 공백이나 - 기호가 포함된 경우는 반드시 따옴표 적용.
- 그 외에는 따옴표 처리가 필수는 아님

data-* 속성

data-000 형태로 개발자가 필요에 따라 직접 정의하는 속성.

JAVASCRIPT에서 사용할 변수값을 숨겨놓기 위한 목적으로 사용한다.

```
<div id="foo" data-hello="안녕하세요" data-world="jQuery">
  ...
</div>
```

data속성값 조회하기

```
var foo = $("#bar").data('hello');
```

data속성값 적용/변경하기

```
$("#bar").data('hello', '반갑습니다');
```

#03. HTML 요소의 CSS 속성 제어

특정 요소에 적용되어 있는 속성값 조회하기

```
var foo = $("#bar").css('속성이름');
```

특정 요소의 적용하기

개별적용

```
$("#bar").css('속성이름', '값');
```

일괄적용 (json형태 사용)

```
$("#bar").css({  
  속성이름: '값',  
  속성이름: '값',  
  속성이름: '값'  
  ...  
});
```

#04. HTML요소의 CSS 클래스 제어

CSS 클래스 적용여부 검사

```
var foo = $("#bar").hasClass('클래스이름'); // true,false 리턴
```

CSS 클래스 적용

```
$("#bar").addClass('클래스이름 클래스이름 ...');
```

- 두 개 이상의 클래스를 적용하고자 하는 경우 공백으로 구분.

CSS 클래스 삭제

```
$("#bar").removeClass('클래스이름 클래스이름 ...');
```

두 개 이상의 클래스를 삭제하고자 하는 경우 공백으로 구분.

CSS 클래스 적용/삭제 자동 반복

```
$("#bar").toggleClass('클래스이름 클래스이름 ...');
```

두 개 이상의 클래스를 처리하고자 하는 경우 공백으로 구분.

요소의 숨김,표시 처리

함수	설명
show([time, [function]])	요소를 표시한다.
hide([time, [function]])	요소를 숨긴다.
toggle([time, [function]])	요소의 숨김과 표시를 자동 반복한다.
fadeIn([time, [function]])	페이드 효과를 적용하여 요소를 표시한다.
fadeOut([time, [function]])	페이드 효과를 적용하여 요소를 숨긴다.
fadeToggle([time, [function]])	페이드 효과를 적용하여 요소의 숨김과 표시를 자동 반복한다.
slideDown([time, [function]])	요소를 아래로 펼쳐서 표시한다.
slideUp([time, [function]])	요소를 위로 접어서 요소를 숨긴다.
slideToggle([time, [function]])	요소를 위,아래로 접고 펼치는 효과를 사용하여 숨김과 표시를 자동 반복한다.

- 시간값은 1/1000초 단위로 지정한다.
- 두 번째 파라미터인 콜백함수는 처리가 완료된 후 호출된다.

#05. not()

- 복수 요소를 지정하고 있는 jQuery객체 중에서 특정 요소를 제외한 항목들을 지정하는 기능.

this와 함께 사용하는 경우

```
$(".btn").on("click", (e) => {
  // ".btn"중에서 클릭된 요소.
  $(e.currentTarget)
  // '.btn'중에서 클릭되지 않은 나머지 요소(들)
  $(".btn").not(e.currentTarget)
});
```

jQuery 객체를 사용하여 not() 함수 사용

".btn"중에서 id값이 "#hello"인 요소 제외

```
$(".btn").not( $("#hello" ) )
```

".btn"중에서 3번째 요소만 제외

```
$(".btn").not( $(".btn:nth-child(3)" ) )    // 1부터 카운트  
$(".btn").not( $(".btn:eq(2)" ) )          // 0부터 카운트
```

css 셀렉터를 사용하여 not() 함수 사용

".btn"중에서 id값이 "#hello"인 요소 제외

```
$(".btn").not( "#hello" )
```

".btn"중에서 3번째 요소만 제외

```
$(".btn").not( ".btn:nth-child(3)" )    // 1부터 카운트  
$(".btn").not( ".btn:eq(2)" )          // 0부터 카운트
```