

You Don't Need jQuery

오늘날 프론트엔드 개발 환경은 급격히 진화하고 있고, 모던 브라우저들은 이미 충분히 많은 DOM/BOM API들을 구현했습니다. 우리는 jQuery를 DOM 처리나 이벤트를 위해 처음부터 배울 필요가 없습니다. React, Angular, Vue같은 프론트엔드 라이브러리들이 주도권을 차지하는 동안 DOM을 바로 처리하는 것은 안티패턴이 되었고, jQuery의 중요성은 줄어들었습니다. 이 프로젝트는 대부분의 jQuery 메소드의 대안을 IE 10+ 이상을 지원하는 네이티브 구현으로 소개합니다.

목차

1. [Query Selector](#)
2. [CSS & Style](#)
3. [DOM 조작](#)
4. [Ajax](#)
5. [이벤트](#)
6. [유틸리티](#)
7. [대안방법](#)
8. [번역](#)
9. [브라우저 지원](#)

Query Selector

평범한 class, id, attribute같은 selector는 `document.querySelector` 나 `document.querySelectorAll` 으로 대체할 수 있습니다.

- `document.querySelector` 는 처음 매칭된 엘리먼트를 반환합니다.
- `document.querySelectorAll` 는 모든 매칭된 엘리먼트를 NodeList로 반환합니다. `[].slice.call` 을 사용해서 Array로 변환할 수 있습니다.
- 만약 매칭된 엘리먼트가 없으면 jQuery는 `[]` 를 반환하지만 DOM API는 `null` 을 반환합니다. Null Pointer Exception에 주의하세요.

안내: `document.querySelector` 와 `document.querySelectorAll` 는 꽤 느립니다, `getElementById` 나 `document.getElementsByClassName`, `document.getElementsByTagName` 를 사용하면 퍼포먼스가 향상을 기대할 수 있습니다.

- [1.0](#) selector로 찾기

```
// jQuery
$('selector');
```

```
// Native
document.querySelectorAll('selector');
```

- [1.1](#) class로 찾기

```
// jQuery
$('.class');
```

```
// Native
document.querySelectorAll('.class');
```

```
// or
document.getElementsByClassName('class');
```

- [1.2](#) id로 찾기

```
// jQuery
$('#id');

// Native
document.querySelector('#id');

// or
document.getElementById('id');
```

- [1.3](#) 속성(attribute)으로 찾기

```
// jQuery
$('a[target=_blank]');

// Native
document.querySelectorAll('a[target=_blank]');
```

- [1.4](#) 자식에서 찾기

```
// jQuery
$el.find('li');

// Native
el.querySelectorAll('li');
```

- [1.5](#) 형제/이전/다음 엘리먼트 찾기

- 형제 엘리먼트

```
// jQuery
$el.siblings();

// Native
[].filter.call(el.parentNode.children, function(child) {
  return child !== el;
});
```

- 이전 엘리먼트

```
// jQuery
$el.prev();

// Native
el.previousElementSibling;
```

- 다음 엘리먼트

```
// jQuery
$el.next();

// Native
el.nextElementSibling;
```

- [1.6](#) Closest

현재 엘리먼트부터 document로 이동하면서 주어진 선택터와 일치하는 가장 가까운 엘리먼트를 반환합니다.

```
// jQuery
$el.closest(selector);

// Native - 최신 브라우저만, IE는 미지원
el.closest(selector);

// Native - IE10 이상
function closest(el, selector) {
    const matchesSelector = el.matches || el.webkitMatchesSelector ||
    el.mozMatchesSelector || el.msMatchesSelector;

    while (el) {
        if (matchesSelector.call(el, selector)) {
            return el;
        } else {
            el = el.parentElement;
        }
    }
    return null;
}
```

- [1.7](#) Parents Until

주어진 선택터에 매칭되는 엘리먼트를 찾기까지 부모 태그들을 위로 올라가며 탐색하여 저장해두었다가 DOM 노드 또는 jQuery object로 반환합니다.

```
// jQuery
$el.parentsUntil(selector, filter);

// Native
function parentsUntil(el, selector, filter) {
    const result = [];
    const matchesSelector = el.matches || el.webkitMatchesSelector ||
    el.mozMatchesSelector || el.msMatchesSelector;

    // match start from parent
    el = el.parentElement;
    while (el && !matchesSelector.call(el, selector)) {
        result.push(el);
        el = el.parentElement;
    }
    return result;
}
```

```

    if (!filter) {
        result.push(el);
    } else {
        if (matchesSelector.call(el, filter)) {
            result.push(el);
        }
    }
    el = el.parentElement;
}
return result;
}

```

- [1.8](#) Form

- Input/Textarea

```

// jQuery
$('#my-input').val();

// Native
document.querySelector('#my-input').value;

```

- e.currentTarget이 .radio 의 몇번째인지 구하기

```

// jQuery
$(e.currentTarget).index('.radio');

// Native
[].indexOf.call(document.querySelectorAll('.radio'), e.currentTarget);

```

- [1.9](#) Iframe Contents

`$('#iframe').contents()` 는 iframe에 한정해서 `contentDocument` 를 반환합니다.

- Iframe contents

```

// jQuery
$iframe.contents();

// Native
iframe.contentDocument;

```

- Iframe에서 찾기

```

// jQuery
$iframe.contents().find('.css');

// Native
iframe.contentDocument.querySelectorAll('.css');

```

- [1.10](#) body 얻기

```
// jQuery
$('body');

// Native
document.body;
```

- [1.11](#) 속성 얻기 및 설정

- 속성 얻기

```
// jQuery
$el.attr('foo');

// Native
el.getAttribute('foo');
```

- 속성 설정하기

```
// jQuery, DOM 변형 없이 메모리에서 작동됩니다.
$el.attr('foo', 'bar');

// Native
el.setAttribute('foo', 'bar');
```

- data- 속성 얻기

```
// jQuery
$el.data('foo');

// Native (`getAttribute` 사용)
el.getAttribute('data-foo');
// Native (IE 11 이상의 지원만 필요하다면 `dataset`을 사용)
el.dataset['foo'];
```

[↑ 목차로 돌아가기](#)

CSS & Style

- [2.1](#) CSS

- style값 얻기

```
// jQuery
$el.css("color");

// Native
// NOTE: 알려진 버그로, style값이 'auto'이면 'auto'를 반환합니다.
```

```
const win = el.ownerDocument.defaultView;
// null은 가상 스타일은 반환하지 않음을 의미합니다.
win.getComputedStyle(el, null).color;
```

- style값 설정하기

```
// jQuery
$el.css({ color: "#ff0011" });

// Native
el.style.color = '#ff0011';
```

- Style값들을 동시에 얻거나 설정하기

만약 한번에 여러 style값을 바꾸고 싶다면 oui-dom-utils 패키지의 [setStyles](#)를 사용해보세요.

- class 추가하기

```
// jQuery
$el.addClass(className);

// Native
el.classList.add(className);
```

- class 제거하기

```
// jQuery
$el.removeClass(className);

// Native
el.classList.remove(className);
```

- class를 포함하고 있는지 검사하기

```
// jQuery
$el.hasClass(className);

// Native
el.classList.contains(className);
```

- class 토글하기

```
// jQuery
$el.toggleClass(className);

// Native
el.classList.toggle(className);
```

- [2.2](#) 폭과 높이

폭과 높이는 이론상 동일합니다. 높이로 예를 들겠습니다.

- Window의 높이

```
// window 높이
$(window).height();
// jQuery처럼 스크롤바를 제외하기
window.document.documentElement.clientHeight;
// 스크롤바 포함
window.innerHeight;
```

- 문서 높이

```
// jQuery
$(document).height();

// Native
document.documentElement.scrollHeight;
```

- Element 높이

```
// jQuery
$el.height();

// Native
function getHeight(el) {
  const styles = this.getComputedStyle(el);
  const height = el.offsetHeight;
  const borderTopWidth = parseFloat(styles.borderTopWidth);
  const borderBottomWidth = parseFloat(styles.borderBottomWidth);
  const paddingTop = parseFloat(styles.paddingTop);
  const paddingBottom = parseFloat(styles.paddingBottom);
  return height - borderBottomWidth - borderTopWidth - paddingTop -
paddingBottom;
}
// 정수로 정확하게 (`border-box`일 때 이 값은 `height`이고, `content-box`일
때, 이 값은 `height + padding + border`)
el.clientHeight;
// 실수로 정확하게 (`border-box`일 때 이 값은 `height`이고, `content-box`일
때, 이 값은 `height + padding + border`)
el.getBoundingClientRect().height;
```

- [2.3](#) Position & Offset

- Position

```
// jQuery
$el.position();
```

```
// Native
{ left: el.offsetLeft, top: el.offsetTop }
```

- Offset

```
// jQuery
$el.offset();

// Native
function getOffset (el) {
  const box = el.getBoundingClientRect();

  return {
    top: box.top + window.pageYOffset -
document.documentElement.clientTop,
    left: box.left + window.pageXOffset -
document.documentElement.clientLeft
  }
}
```

- [2.4 Scroll Top](#)

```
// jQuery
$(window).scrollTop();

// Native
(document.documentElement && document.documentElement.scrollTop) ||
document.body.scrollTop;
```

[↑ 목차로 돌아가기](#)

DOM 조작

- [3.1 제거](#)

```
// jQuery
$el.remove();

// Native
el.parentNode.removeChild(el);
```

- [3.2 Text](#)

- text 가져오기

```
// jQuery
$el.text();
```



```
// Native
el.textContent;
```

- text 설정하기

```
// jQuery
$el.text(string);

// Native
el.textContent = string;
```

- [3.3](#) HTML

- HTML 가져오기

```
// jQuery
$el.html();

// Native
el.innerHTML;
```

- HTML 설정하기

```
// jQuery
$el.html(htmlString);

// Native
el.innerHTML = htmlString;
```

- [3.4](#) 해당 엘리먼트의 자식들 뒤에 넣기(Append)

부모 엘리먼트의 마지막 자식으로 엘리먼트를 추가합니다.

```
// jQuery
$el.append("<div id='container'>hello</div>");

// Native
el.insertAdjacentHTML("beforeend", "<div id='container'>hello</div>");
```

- [3.5](#) 해당 엘리먼트의 자식들 앞에 넣기(Prepend)

```
```js // jQuery $el.prepend("
```

```
hello
");
```

```
// Native
```

```
el.insertAdjacentHTML("afterbegin", "
```

hello  
");

- [3.6] (#3.6) <a name='3.6'></a> 해당 엘리먼트 앞에 넣기(insertBefore)

새 노드를 선택한 엘리먼트 앞에 넣습니다.

```
```js
// jQuery
$newEl.insertBefore(queryString);

// Native
const target = document.querySelector(queryString);
target.parentNode.insertBefore(newEl, target);
```

- [3.7](#) 해당 엘리먼트 뒤에 넣기(insertAfter)

새 노드를 선택한 엘리먼트 뒤에 넣습니다.

```
// jQuery
$newEl.insertAfter(queryString);

// Native
const target = document.querySelector(queryString);
target.parentNode.insertBefore(newEl, target.nextSibling);
```

- [3.8](#) is

query selector와 일치하면 true 를 반환합니다.

```
// jQuery
$el.is(selector);

// Native
el.matches(selector);
```

- [3.9](#) clone

엘리먼트의 복제본을 만듭니다.

```
// jQuery
$el.clone();

// Native
el.cloneNode();

// Deep clone은 파라미터를 `true` 로 설정하세요.
```

- [3.10](#) empty

모든 자식 노드를 제거합니다.

```
// jQuery
$el.empty();

// Native
el.innerHTML = '';
```

[↑ 목차로 돌아가기](#)

Ajax

[Fetch API](#) 는 XMLHttpRequest를 ajax로 대체하는 새로운 표준 입니다. Chrome과 Firefox에서 작동하며, polyfill을 이용해서 구형 브라우저에서 작동되도록 만들 수도 있습니다.

IE9 이상에서 지원하는 [github/fetch](#) 혹은 IE8 이상에서 지원하는 [fetch-ie8](#), JSONP 요청을 만드는 [fetch-jsonp](#)를 이용해보세요.

[↑ 목차로 돌아가기](#)

이벤트

namespace와 delegation을 포함해서 완전히 알아 옆길 원하시면 <https://github.com/oneuijs/oui-dom-events> 를 고려해보세요.

- [5.1](#) 이벤트 Bind 걸기

```
// jQuery
$el.on(eventName, eventHandler);

// Native
el.addEventListener(eventName, eventHandler);
```

- [5.2](#) 이벤트 Bind 풀기

```
// jQuery
$el.off(eventName, eventHandler);

// Native
el.removeEventListener(eventName, eventHandler);
```

- [5.3](#) 이벤트 발생시키기(Trigger)

```
// jQuery
$(el).trigger('custom-event', {key1: 'data'});

// Native
if (window.CustomEvent) {
  const event = new CustomEvent('custom-event', {detail: {key1: 'data'}});
} else {
```

```
const event = document.createEvent('CustomEvent');
event.initCustomEvent('custom-event', true, true, {key1: 'data'});
}

el.dispatchEvent(event);
```

[↑ 목차로 돌아가기](#)

유틸리티

- [6.1](#) 배열인지 검사(isArray)

```
// jQuery
$.isArray(range);

// Native
Array.isArray(range);
```

- [6.2](#) 앞뒤 공백 지우기(Trim)

```
// jQuery
$.trim(string);

// Native
string.trim();
```

- [6.3](#) Object Assign

사용하려면 object.assign polyfill을 사용하세요. <https://github.com/ljharb/object.assign>

```
// jQuery
$.extend({}, defaultOpts, opts);

// Native
Object.assign({}, defaultOpts, opts);
```

- [6.4](#) Contains

```
// jQuery
$.contains(el, child);

// Native
el !== child && el.contains(child);
```

- [6.5](#) inArray

```
// jQuery
$.inArray(item, array);
```

```
// Native
array.indexOf(item);
```

- [6.6](#) map

```
// jQuery
$.map(array, function(value, index) {
});

// Native
Array.map(function(value, index) {
});
```

[↑ 목차로 돌아가기](#)

대안방법

- [You Might Not Need jQuery](#) - 일반 자바스크립트로 공통이벤트, 엘리먼트, ajax 등을 다루는 방법 예제.
- [npm-dom](#) 과 [webmodules](#) - 개별 DOM모듈을 NPM에서 찾을 수 있습니다.

번역

- [한국어](#)
- [简体中文](#)
- [Bahasa Melayu](#)
- [Bahasa Indonesia](#)
- [Português\(PT-BR\)](#)
- [Tiếng Việt Nam](#)
- [Español](#)
- [Русский](#)
- [Türkçe](#)
- [Italian](#)

브라우저 지원

 Chrome	 Firefox	 IE	 Opera	 Safari
Latest ✓	Latest ✓	10+ ✓	Latest ✓	6.1+ ✓

License

MIT