

build unknown

jQuery'e İhtiyacınız Yok

Önyüz ortamları bugünlerde çok hızlı geliyor, öyle ki modern tarayıcılar DOM/DOM API'lere ait önemli gereklilikleri çoktan yerine getirdiler. DOM işleme ve olaylar için, en baştan jQuery öğrenmemize gerek kalmadı. Bu arada, üstünlükleri ile jQuery'i önemsizleştiren ve doğrudan DOM değişikliklerinin bir Anti-pattern olduğunu gösteren, React, Angular ve Vue gibi gelişmiş önyüz kütüphanelerine ayrıca teşekkür ederiz. Bu proje, IE10+ desteği ile çoğunluğu jQuery yöntemlerine alternatif olan yerleşik uygulamaları içerir.

İçerik Tablosu

- [Sorgu seçiciler](#)
- [CSS & Stil](#)
- [DOM düzenleme](#)
- [Ajax](#)
- [Olaylar](#)
- [Araçlar](#)
- [Alternatifler](#)
- [Çeviriler](#)
- [Tarayıcı desteği](#)

Sorgu seçiciler

Yaygın olan class, id ve özellik seçiciler yerine, `document.querySelector` yada

`document.querySelectorAll` kullanabiliriz. Ayrıldıkları nokta:

- `document.querySelector` ilk seçilen öğeyi döndürür
- `document.querySelectorAll` Seçilen tüm öğeleri NodeList olarak geri döndürür.
`[] .slice.call (document.querySelectorAll(selector) || [])` kullanarak bir diziye dönüştürebilirsiniz.
- Herhangi bir öğenin seçilememesi durumunda ise, jQuery `[]` döndürürken, DOM API `null` döndürecektir. Null Pointer istisnası almamak için `||` ile varsayılan değere atama yapabilirsiniz, örnek:
`document.querySelectorAll(selector) || []`

Uyarı: `document.querySelector` ve `document.querySelectorAll` biraz **YAVAŞ** olabilir, Daha hızlısını isterseniz, `getElementById`, `document.getElementsByClassName` yada `document.getElementsByTagName` kullanabilirsiniz.

- [1.0](#) Seçici ile sorgu

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

// Yerleşik
document.querySelector('selector');
```

- [1.1](#) Sınıf ile sorgu

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

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

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

- [1.2](#) Id ile sorgu

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

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

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

- [1.3](#) Özellik ile sorgu

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

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

- [1.4](#) Öğe erişimi

- Node'a erişim

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

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

- Body'e erişim

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

// Yerleşik
document.body;
```

- Özelliğe erişim

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

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

- Data özelliğine erişim

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

// Yerleşik
// getAttribute kullanarak
el.getAttribute('data-foo');
// Eğer IE 11+ kullanıyor iseniz, `dataset` ile de erişebilirsiniz
el.dataset['foo'];
```

- [1.5](#) Kardeş/Önceki/Sonraki öğeler

- Kardeş öğeler

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

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

- Önceki öğeler

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

// Yerleşik
el.previousElementSibling;
```

- Sonraki öğeler

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

// Yerleşik
el.nextElementSibling;
```

- [1.6](#) En yakın

Verilen seçici ile eşleşen ilk öğeyi döndürür, geçerli öğeden başlayarak document'a kadar geçiş yapar.

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

```
// Yerleşik - Sadece en güncellerde, IE desteklemiyor
el.closest(selector);

// Yerleşik - IE10+
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](#) Önceki atalar

Verilen seçici ile eşleşen öge veya DOM node veya jQuery nesnesi hariç, mevcut öge ile aradaki tüm önceki ataları bir set dahilinde verir.

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

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

  // eşleştirme, atadan başlar
  el = el.parentElement;
  while (el && !matchesSelector.call(el, selector)) {
    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();

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

- `e.currentTarget` ile `.radio` arasındaki dizini verir

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

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

- [1.9](#) Iframe İçeriği

Mevcut Iframe için `$('#iframe').contents()` yerine `contentDocument` döndürür.

- Iframe İçeriği

```
// jQuery
$iframe.contents();

// Yerleşik
iframe.contentDocument;
```

- Iframe seçici

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

```
// Yerleşik
iframe.contentDocument.querySelectorAll('.css');
```

[↑ üste dön](#)

CSS & Stil

- [2.1](#) CSS

- Stili verir

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

// Yerleşik
// NOT: Bilinen bir hata, eğer stil değeri 'auto' ise 'auto' döndürür
const win = el.ownerDocument.defaultView;
```

```
// null sahte tipleri döndürmemesi için
win.getComputedStyle(el, null).color;
```

- Stil değiştir

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

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

- Stil değeri al/değiştir

Eğer aynı anda birden fazla stili değiştirmek istiyor iseniz, oui-dom-utils paketi içindeki [setStyles](#) metoduna göz atınız.

- Sınıf ekle

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

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

- Sınıf çıkart

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

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

- sınıfı var mı?

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

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

- Sınıfı takas et

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

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

- [2.2](#) Genişlik ve Yükseklik

Genişlik ve Yükseklik teorik olarak aynı şekilde, örnek olarak Yükseklik veriliyor

- Window Yüksekliği

```
// window yüksekliği
$(window).height();
// kaydırma çubuğu olmaksızın, jQuery ile aynı
window.document.documentElement.clientHeight;
// kaydırma çubuğu ile birlikte
window.innerHeight;
```

- Document yüksekliği

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

// Yerleşik
document.documentElement.scrollHeight;
```

- Öge yüksekliği

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

// Yerleşik
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;
}
// Tamsayı olarak daha doğru olanı (`border-box` iken, `height` esas;
`content-box` ise, `height + padding + border` esas alınır)
el.clientHeight;
// Ondalık olarak daha doğru olanı (`border-box` iken, `height` esas;
`content-box` ise, `height + padding + border` esas alınır)
el.getBoundingClientRect().height;
```

- [2.3](#) Pozisyon ve Ara-Açıklığı

- Pozisyon

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

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

- Ara-Açıklığı

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

// Yerleşik
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](#) Üste kaydır

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

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

[↑ üste dön](#)

DOM düzenleme

- [3.1](#) Çıkartma

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

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

- [3.2](#) Metin

- Get text

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



```
// Yerleşik
el.textContent;
```

- Set text

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

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

- [3.3](#) HTML

- HTML'i alma

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

// Yerleşik
el.innerHTML;
```

- HTML atama

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

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

- [3.4](#) Sona ekleme

Ata öğenin son çocuğundan sonra öge ekleme

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

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

- [3.5](#) Öne ekleme

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

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

- [3.6](#) Öncesine Ekleme

Seçili öğeden önceki yere yeni öge ekleme

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

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

- [3.7](#) Sonrasına ekleme

Seçili öğeden sonraki yere yeni öge ekleme

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

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

- [3.8](#) eşit mi?

Sorgu seçici ile eşleşiyor ise `true` döner

```
// jQuery için not: `is` aynı zamanda `function` veya `elements` için de
geçerlidir fakat burada bir önemi bulunmuyor
$el.is(selector);

// Yerleşik
el.matches(selector);
```

- [3.9](#) Klonlama

Mevcut öğenin bir derin kopyasını oluşturur

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

// Yerleşik
el.cloneNode();

// Derin kopya için, `true` parametresi kullanınız
```

[↑ üste dön](#)

Ajax

[Fetch API](#) ajax için XMLHttpRequest yerine kullanan yeni standarttır. Chrome ve Firefox destekler, eski tarayıcılar için polyfill kullanabilirsiniz.

IE9+ ve üstü için [github/fetch](#) yada IE8+ ve üstü için [fetch-ie8](#), JSONP istekler için [fetch-jsonp](#) deneyiniz.

[↑ üste dön](#)

Olaylar

Namespace ve Delegasyon ile tam olarak değiştirmek için, <https://github.com/oneuijs/oui-dom-events> sayfasına bakınız

- [5.1](#) on ile bir öğeye bağlama

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

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

- [5.2](#) off ile bir bağlamayı sonlandırma

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

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

- [5.3](#) Tetikleyici

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

// Yerleşik
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);
```

[↑ üste dön](#)

Araçlar

- [6.1](#) isArray

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

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

- [6.2 Trim](#)

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

// Yerleşik
string.trim();
```

- [6.3 Nesne atama](#)

Türetmek için, object.assign polyfill'ini deneyiniz <https://github.com/ljharb/object.assign>

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

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

- [6.4 İçerme](#)

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

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

[↑ üste dön](#)





Alternatifler

- [jQuery'e İhtiyacınız Yok](#) - Yaygın olan olay, öge ve ajax işlemlerinin yalın Javascript'teki karşılıklarına ait örnekler
- [npm-dom](#) ve [webmodules](#) - NPM için ayrı DOM modül organizasyonları

Çeviriler

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

Tarayıcı Desteği

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

Lisans

MIT