> ***You Don't Need jQuery***

Você não precisa de jQuery

Ambientes Frontend evoluem rapidamente nos dias de hoje, navegadores modernos já implementaram uma grande parte das APIs DOM/BOM que são boas o suficiente. Nós não temos que aprender jQuery a partir do zero para manipulação do DOM ou eventos. Nesse meio tempo, graças a bibliotecas frontend como React, Angular e Vue, a manipulação direta do DOM torna-se um anti-padrão, jQuery é menos importante do que nunca. Este projeto resume a maioria das alternativas dos métodos jQuery em implementação nativa, com suporte ao IE 10+.

Tabela de conteúdos

Query Selector

No lugar de seletores comuns como classe, id ou atributo podemos usar `document.querySelector` ou `document.querySelectorAll` para substituição. As diferenças são:

- `document.querySelector` retorna o primeiro elemento correspondente
- `document.querySelectorAll` retorna todos os elementos correspondentes como NodeList. Pode ser convertido para Array usando `[].slice.call(document.querySelectorAll(selector) || []);`
- Se não tiver elementos correspondentes, jQuery retornaria `[]` considerando que a DOM API irá retornar `null`. Preste atenção ao Null Pointer Exception. Você também pode usar `||` para definir um valor padrão caso nenhum elemento seja encontrado, como `document.querySelectorAll(selector) || []`

> *Aviso: `document.querySelector` e `document.querySelectorAll` são bastante **LENTOS**, tente usar `getElementById`, `document.getElementsByClassName` ou `document.getElementsByTagName` se você quer ter uma maior performance.*

- 1.0 Query por seletor

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

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

- 1.1 Query por classe

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

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

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

- 1.2 Query por id

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

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

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

- 1.3 Query por atributo

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

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

- 1.4 Find sth.
  - Busca por nós

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

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

  - Buscar `body`

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

// Nativo
document.body;
```

  - Buscar atributos

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

```
// Nativo
e.getAttribute('foo');
```

- Buscar atributos `data-`

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

// Nativo
// usando getAttribute
el.getAttribute('data-foo');
// você também pode usar `dataset` se você precisar suportar apenas IE 11+
el.dataset['foo'];
```

- 1.5 Sibling/Previous/Next Elements
  - Sibling elements

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

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

  - Previous elements

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

// Nativo
el.previousElementSibling;
```

  - Next elements

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

// Nativo
el.nextElementSibling;
```

- 1.6 Closest

  Retorna o primeiro elemento que corresponda ao seletor, partindo do elemento atual para o document.

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

```
// Nativo
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

  Obtém os ancestrais de cada elemento no atual conjunto de elementos combinados, mas não inclui o elemento correspondente pelo seletor, nó do DOM, ou objeto jQuery.

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

// Nativo
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)) {
    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();
```

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

- Obter o índice do e.currentTarget entre `.radio`

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

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

- 1.9 Iframe Contents

  `$('iframe').contents()` retorna `contentDocument` para este iframe específico

  - Iframe contents

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

// Nativo
iframe.contentDocument;
```

  - Iframe Query

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

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

↑ ir para o topo

CSS & Estilo

- 2.1 CSS

  - Obter estilo

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

// Nativo
// AVISO: Bug conhecido, irá retornar 'auto' se o valor do estilo for
'auto'
const win = el.ownerDocument.defaultView;
// null significa não retornar estilos
win.getComputedStyle(el, null).color;
```

- Definir Estilo

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

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

- Get/Set Styles

  Observe que se você deseja setar vários estilos de uma vez, você pode optar por setStyles método no pacote oui-dom-utils.

- Adicionar classe

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

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

- Remover classe

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

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

- Verificar classe

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

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

- Toggle class

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

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

- 2.2 Largura e Altura

  `width` e `height` são teoricamente idênticos, vamos pegar `height` como exemplo:

- Altura da janela

```
// window height
$(window).height();
// sem scrollbar, se comporta como jQuery
window.document.documentElement.clientHeight;
// com scrollbar
window.innerHeight;
```

- Altura do Documento

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

// Nativo
document.documentElement.scrollHeight;
```

- Altura do Elemento

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

// Nativo
function getHeight(el) {
  const styles = this.getComputedStyles(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;
}
// preciso para inteiro (quando `border-box`, é `height`; quando
`content-box`, é `height + padding + border`)
el.clientHeight;
// preciso para decimal (quando `border-box`, é `height`; quando
`content-box`, é `height + padding + border`)
el.getBoundingClientRect().height;
```

- 2.3 Position & Offset

- Position

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

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

- Offset

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

// Nativo
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 Rolar para o topo

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

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

↑ ir para o topo

Manipulação do Dom

- 3.1 Remover

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

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

- 3.2 Texto
  - Obter texto

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

// Nativo
el.textContent;
```

  - Definir texto

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

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

- 3.3 HTML
  - Obter HTML

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

// Nativo
el.innerHTML;
```

  - Definir HTML

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

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

- 3.4 Append

  Incluir elemento filho após o último filho do elemento pai.

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

// Nativo
let newEl = document.createElement('div');
newEl.setAttribute('id', 'container');
newEl.innerHTML = 'hello';
el.appendChild(newEl);
```

- 3.5 Prepend

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

// Nativo
let newEl = document.createElement('div');
newEl.setAttribute('id', 'container');
newEl.innerHTML = 'hello';
el.insertBefore(newEl, el.firstChild);
```

- 3.6 insertBefore

  Insere um novo nó antes dos elementos selecionados.

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

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

- 3.7 insertAfter

  Insere um novo nó após os elementos selecionados.

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

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

↑ ir para o topo

Ajax

Substitua por fetch e fetch-jsonp

↑ ir para o topo

Eventos

Para uma substituição completa com namespace e delegation, consulte https://github.com/oneuijs/oui-dom-events

- 5.1 `Bind` num evento com `on`

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

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

- 5.2 `Unbind` num evento com `off`

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

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

- 5.3 Trigger

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

// Nativo
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);
```

↑ ir para o topo

Utilitários

- 6.1 isArray

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

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

- 6.2 Trim

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

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

- 6.3 Object Assign

  Use o polyfill `object.assign` para eetender um Object: https://github.com/ljharb/object.assign

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

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

- 6.4 Contains

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

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

[↑ ir para o topo](#)

Suporte dos Navegadores

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

Licença