Вам не нужен jQuery

В наше время среда фронт энд разработки быстро развивается, современные браузеры уже реализовали значимую часть DOM/BOM APIs и это хорошо. Вам не нужно изучать jQuery с нуля для манипуляцией DOM'ом или обектами событий. В то же время, благодаря лидирующим фронт энд библиотекам, таким как React, Angular и Vue, манипуляция DOM'ом напрямую становится противо шаблонной, jQuery никогда не был менее важен. Этот проект суммирует большинство альтернатив методов jQuery в нативном исполнении с поддержкой IE 10+.

Содержание

1. Query Selector
2. CSS & Style
3. Манипуляция DOM
4. Ajax
5. События
6. Утилиты
7. Альтернативы
8. Переводы
9. Поддержка браузеров

Query Selector

Для часто используемых селекторов, таких как class, id или attribute мы можем использовать `document.querySelector` или `document.querySelectorAll` для замены. Разница такова:

- `document.querySelector` возвращает первый совпавший элемент
- `document.querySelectorAll` возвращает все совспавшие элементы как коллекцию узлов(NodeList). Его можно конвертировать в массив используя `[].slice.call(document.querySelectorAll(selector) || []);`
- Если никакие элементы не совпадут, jQuery вернет `[]` где DOM API вернет `null`. Обратите внимание на указатель исключения Null (Null Pointer Exception). Вы так же можете использовать `||` для установки значения по умолчанию если не было найдемо совпадений `document.querySelectorAll(selector) || []`

> *Заметка:* `document.querySelector` *и* `document.querySelectorAll` *достаточно* ***МЕДЛЕННЫ****, старайтесь использовать* `getElementById`, `document.getElementsByClassName` *или* `document.getElementsByTagName` *если хотите улучшить производительность.*

- 1.0 Query by selector

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

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

- 1.1 Запрос по классу

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

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

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

- 1.2 Запрос по ID

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

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

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

- 1.3 Запрос по атрибуту

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

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

- 1.4 Найти среди потомков
  - Найти nodes

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

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

  - Найти body

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

// Нативно
document.body;
```

  - Найти атрибуты

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

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

- Найти data attribute

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

// Нативно
// используя getAttribute
el.getAttribute('data-foo');
// также можно использовать `dataset`, если не требуется поддержка
ниже IE 11.
el.dataset['foo'];
```

- 1.5 Родственные/Предыдущие/Следующие Элементы
  - Родственные элементы

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

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

  - Предыдущие элементы

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

// Нативно
el.previousElementSibling;
```

  - Следующие элементы

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

// Нативно
el.nextElementSibling;
```

- 1.6 Closest

  Возвращает первый совпавший элемент по предоставленному селектору, обходя от текущего элементы до документа.

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

// Нативно - Only latest, NO IE
el.closest(selector);

// Нативно - 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 Родители до

  Получить родителей кажого элемента в текущем сете совпавших элементов, но не включая элемент совпавший с селектором, узел DOM'a, или объект jQuery.

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

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

  // Совпадать начиная от родителя
  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 От

- Input/Textarea

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

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

- получить индекс e.currentTarget между `.radio`

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

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

- 1.9 Контент Iframe

  `$('iframe').contents()` возвращает `contentDocument` для именно этого iframe

  - Контент Iframe

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

// Нативно
iframe.contentDocument;
```

  - Iframe Query

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

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

↑ Наверх

CSS & Style

- 2.1 CSS

  - Получить стиль

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

// Нативно
// ЗАМЕТКА: Известная ошика, возвращает 'auto' если значение стиля
```

```
'auto'
const win = el.ownerDocument.defaultView;
// null означает не возвращать псевдостили
win.getComputedStyle(el, null).color;
```

- Присвоение style

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

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

- Получение/Присвоение стилей

  Заметьте что если вы хотите присвоить несколько стилей за раз, вы можете сослаться на setStyles метод в oui-dom-utils package.

- Добавить класс

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

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

- Удалить class

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

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

- Имеет класс

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

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

- Переключать класс

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

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

- 2.2 Ширина и Высота

  Ширина и высота теоритечески идентичны, например возьмем высоту:

  - высота окна

    ```
    // Высота окна
    $(window).height();
    // без скроллбара, ведет себя как jQuery
    window.document.documentElement.clientHeight;
    // вместе с скроллбаром
    window.innerHeight;
    ```

  - высота документа

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

    // Нативно
    document.documentElement.scrollHeight;
    ```

  - Высота элемента

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

    // Нативно
    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;
    }
    // С точностью до целого числа (когда `border-box`, это `height`; когда
    `content-box`, это `height + padding + border`)
    el.clientHeight;
    // с точностью до десятых (когда `border-box`, это `height`; когда
    `content-box`, это `height + padding + border`)
    el.getBoundingClientRect().height;
    ```

- 2.3 Позиция и смещение

  - Позиция

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

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

- Смещение

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

// Нативно
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 Прокрутка вверх

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

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

↑ Наверх

Манипуляции DOM

- 3.1 Remove

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

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

- 3.2 Текст

- Получить текст

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

```
// Нативно
el.textContent;
```

- Присвоить текст

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

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

- 3.3 HTML
  - Получить HTML

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

// Нативно
el.innerHTML;
```

  - Присвоить HTML

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

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

- 3.4 Append

  Добавление элемента ребенка после последнего ребенка элемента родителя

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

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

- 3.5 Prepend

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

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

- 3.6 insertBefore

Вставка нового элемента перед выбранным элементом

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

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

- 3.7 insertAfter

  Вставка новго элемента после выбранного элемента

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

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

- 3.8 is

  Возвращает `true` если совпадает с селектором запроса

```
// jQuery - заметьте что `is` так же работает с `function` или `elements`
которые не имют к этому отношения
$el.is(selector);

// Нативно
el.matches(selector);
```

↑ Наверх

Ajax

Заменить с fetch и fetch-jsonp

↑ Наверх

События

Для полной замены с пространством имен и делегация, сослаться на oui-dom-events

- 5.1 Связать событие используя on

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

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

- 5.2 Отвязать событие используя off

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

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

- 5.3 Trigger

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

// Нативно
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);
```

↑ Наверх

Утилиты

- 6.1 isArray

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

// Нативно
Array.isArray(range);
```

- 6.2 Trim

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

// Нативно
string.trim();
```

- 6.3 Назначение объекта

  Дополнительно, используйте полифил object.assign https://github.com/ljharb/object.assign

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

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

- 6.4 Contains

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

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

↑ Наверх

Альтернативы

- You Might Not Need jQuery - Примеры как исполняются частые события, элементы, ajax и тд с ванильным javascript.
- npm-dom и webmodules - Отдельные DOM модули можно найти на NPM

Переводы

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

Поддержка браузеров

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

License