Anda tidak memerlukan jQuery

Mutakhir ini perkembangan dalam persekitaran frontend berlaku begitu pesat sekali. Justeru itu kebanyakan pelayar moden telahpun menyediakan API yang memadai untuk pengaksesan DOM/BOM. Kita tak payah lagi belajar jQuery dari asas untuk memanipulasi DOM dan acara-acara. Projek ini menawarkan perlaksanaan alternatif kepada kebanyakan kaedah-kaedah jQuery yang menyokong IE 10+.

Isi Kandungan

Pemilihan Elemen

Pemilihan elemen yang umum seperti class, id atau atribut, biasanya kita boleh pakai `document.querySelector` atau `document.querySelectorAll` sebagai ganti. Bezanya terletak pada

- `document.querySelector` akan mengembalikan elemen pertama sekali yang sepadan dijumpai
- `document.querySelectorAll` akan mengembalikan kesemua elemen yang sepadan dijumpai kedalam sebuah NodeList. Ia boleh ditukar kedalam bentuk array menggunakan `[].slice.call`
- Sekiranya tiada elemen yang sepadan dijumpai, jQuery akan mengembalikan `[]` dimana API DOM pula akan mengembalikan `null`. Sila ambil perhatian pada Null Pointer Exception

> *AWAS: `document.querySelector` dan `document.querySelectorAll` agak **LEMBAB** berbanding `getElementById`, `document.getElementsByClassName` atau `document.getElementsByTagName` jika anda menginginkan bonus dari segi prestasi.*

- 1.1 Pemilihan menggunakan class

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

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

- 1.2 Pemilihan menggunakan id

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

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

- 1.3 Pemilihan menggunakan atribut

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

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

- 1.4 Cari sth.
  - Find nodes

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

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

  - Cari body

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

    // Native
    document.body;
    ```

  - Cari Attribute

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

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

  - Cari atribut data

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

    // Native
    // menggunakan getAttribute
    el.getAttribute('data-foo');
    // anda boleh juga gunakan `dataset` jika ingin pakai IE 11+
    el.dataset['foo'];
    ```

- 1.5 Sibling/Previous/Next Elements
  - Sibling elements

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

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

  - Previous elements

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

// Native
el.previousElementSibling;
```

  - Next elements

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

- 1.6 Closest

  Return the first matched element by provided selector, traversing from current element to document.

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

// Native
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

  Get the ancestors of each element in the current set of matched elements, up to but not including the element matched by the selector, DOM node, or jQuery object.

```
// 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)) {
    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;
    ```

  - Get index of e.currentTarget between `.radio`

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

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

- 1.9 Iframe Contents

  `$('iframe').contents()` returns `contentDocument` for this specific iframe

  - Iframe contents

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

```
// Native
iframe.contentDocument;
```

- Iframe Query

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

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

↑ back to top

CSS & Style

- 2.1 CSS
  - Get style

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

    // Native
    // NOTE: Known bug, will return 'auto' if style value is 'auto'
    const win = el.ownerDocument.defaultView;
    // null means not return presudo styles
    win.getComputedStyle(el, null).color;
    ```

  - Set style

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

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

  - Get/Set Styles

    Note that if you want to set multiple styles once, you could refer to setStyles method in oui-dom-utils package.

  - Add class

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

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

  - Remove class

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

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

  - has class

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

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

  - Toggle class

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

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

- 2.2 Width & Height

  Width and Height are theoretically identical, take Height as example:

  - Window height

    ```
    // window height
    $(window).height();
    // without scrollbar, behaves like jQuery
    window.document.documentElement.clientHeight;
    // with scrollbar
    window.innerHeight;
    ```

  - Document height

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

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

  - Element height

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

```
// Native
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;
}
// accurate to integer (when `border-box`, it's `height`; when
`content-box`, it's `height + padding + border`)
el.clientHeight;
// accurate to decimal (when `border-box`, it's `height`; when
`content-box`, it's `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;
```

⬆ back to top

DOM Manipulation

- 3.1 Remove

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

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

- 3.2 Text

  - Get text

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

    // Native
    el.textContent;
    ```

  - Set text

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

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

- 3.3 HTML

  - Get HTML

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

    // Native
    el.innerHTML;
    ```

  - Set HTML

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

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

- 3.4 Append

  append child element after the last child of parent element

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

  // Native
  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>");

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

- 3.6 insertBefore

  Insert a new node before the selected elements

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

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

- 3.7 insertAfter

  Insert a new node after the selected elements

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

  // Native
  ```

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

⬆ back to top

Ajax

Replace with fetch and fetch-jsonp

⬆ back to top

Events

For a complete replacement with namespace and delegation, refer to https://github.com/oneuijs/oui-dom-events

- 5.1 Bind an event with on

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

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

- 5.2 Unbind an event with off

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

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

- 5.3 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);
  ```

⬆ back to top

Utility

- 6.1 isArray

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

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

- 6.2 Trim

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

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

- 6.3 Object Assign

  Extend, use object.assign polyfill 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);
  ```

⬆ back to top

Terjemahan

- 한국어
- 简体中文
- English
- Русский
- Türkçe

Sokongan Pelayar

Chrome	Firefox	IE	Opera	Safari

Latest ✓	Latest ✓	10+ ✓	Latest ✓	6.1+ ✓

Lesen

MIT