



**Пишем свой jQuery
за 20 минут.**

(на самом деле нет)

Кир Белевич, Яндекс







- zeptajs.com
 - ≈5-10K
 - jQuery-совместимое API
- microjs.com
 - ≈0.1-5K
 - коллекция микробиблиотек и микрофреймворков
- 140byt.es
 - ≈0-140B
 - настоящий хардкор для реальных пацанов

04 WINS

99

SUB-ZERO

SMOKE

FIGHT!!!



```
01. ;(function(win, doc) {  
02.  
03.  
04.  
05.  
06.  
07.  
08. })(window, window.document);
```

```
01. ;(function(win, doc) {  
02.     function Yocto() {}  
03.  
04.  
05.  
06.  
07.  
08. })(window, window.document);
```



```
01. ;(function(win, doc) {  
02.     function Yocto() {}  
03.  
04.  
05.  
06.  
07.     win.Yocto = win.$$$ = Yocto;  
08. })(window, window.document);
```



```
01. ;(function(win, doc) {  
02.     function Yocto() {  
03.         if( !(this instanceof Yocto) ) {  
04.             return new Yocto();  
05.         }  
06.     }  
07.     win.Yocto = win.$$ = Yocto;  
08. })(window, window.document);
```

```
01. $$$  
02. // function Yocto() { ... }  
03.  
04. $$$ .prototype  
05. // Yocto {}  
06.  
07. $$$ () instanceof Yocto  
08. // true
```

```
01. function Yocto(selector) {  
02.     if(!(this instanceof Yocto)) { ... }  
03.  
04.     if(selector && typeof selector === 'string') {  
05.         this.result = doc.querySelectorAll(selector);  
06.     }  
07. }
```

```
01. $$$('body')
02. // Yocto {result: NodeList[1]}
03.
04. $$$('body').result
05. // [<body>...</body>]
06.
07. $$$('body').result[0]
08. // <body>...</body>
```



```
01. $$$ .prototype.makeItRed = function() {  
02.     for(var i = 0; i < this.result.length; i++) {  
03.         this.result[i].style.background = 'red';  
04.     }  
05.  
06.     return this;  
07. }
```

```
01. $$$('body')  
02.     .makeItRed()  
03.     .makeItRed()  
04.     .result[0].style.background  
05. // "red"
```

```
01. ;(function(win, doc) {  
02.     function Yocto(selector) {  
03.         if(!(this instanceof Yocto)) {  
04.             return new Yocto(selector);  
05.         }  
06.         if(selector && typeof selector === 'string') {  
07.             this.result = doc.querySelectorAll(selector);  
08.         }  
09.     }  
10.     win.Yocto = win.$$ = Yocto;  
11. })(window, window.document);
```

**А ЧТО ЕСЛИ МОЖНО
БЫЛО БЫ НАПИСАТЬ**

**\$\$\$(...)
.setAttribute(...)
.addEventListener(...)**

```
01. var div = document.createElement('div');
02. for(var k in div) {
03.     console.log(k, '-', typeof div[k]);
04. }
05. // ...
06. // innerHTML - string
07. // style - object
08. // setAttribute - function
09. // childNodes - object
```

Манипуляции с DOM

- insertAdjacentElement / insertBefore
- appendChild / replaceChild / removeChild
- cloneNode / remove
- insertAdjacentHTML
- insertAdjacentHTML / innerHTML / outerHTML
- innerText / outerText

Выборка элементов

- `getElementsByClassName` / `getElementsByTagName`
- `querySelector` / `querySelectorAll` / `contains` / `hasChildNodes`
- `matchesSelector` (`matches`)
- `parentNode` / `parentElement` / `childNodes` / `children`
- `firstChild` / `firstElementChild`
- `lastChild` / `lastElementChild`
- `nextSibling` / `nextElementSibling`
- `previousSibling` / `previousElementSibling`

Атрибуты

- className / id
- getAttribute / setAttribute
- hasAttribute / removeAttribute
- attributes
- title / lang / draggable

События

- `addEventListener` / `removeEventListener`
- `dispatchEvent`
- `click` / `focus` / `blur`

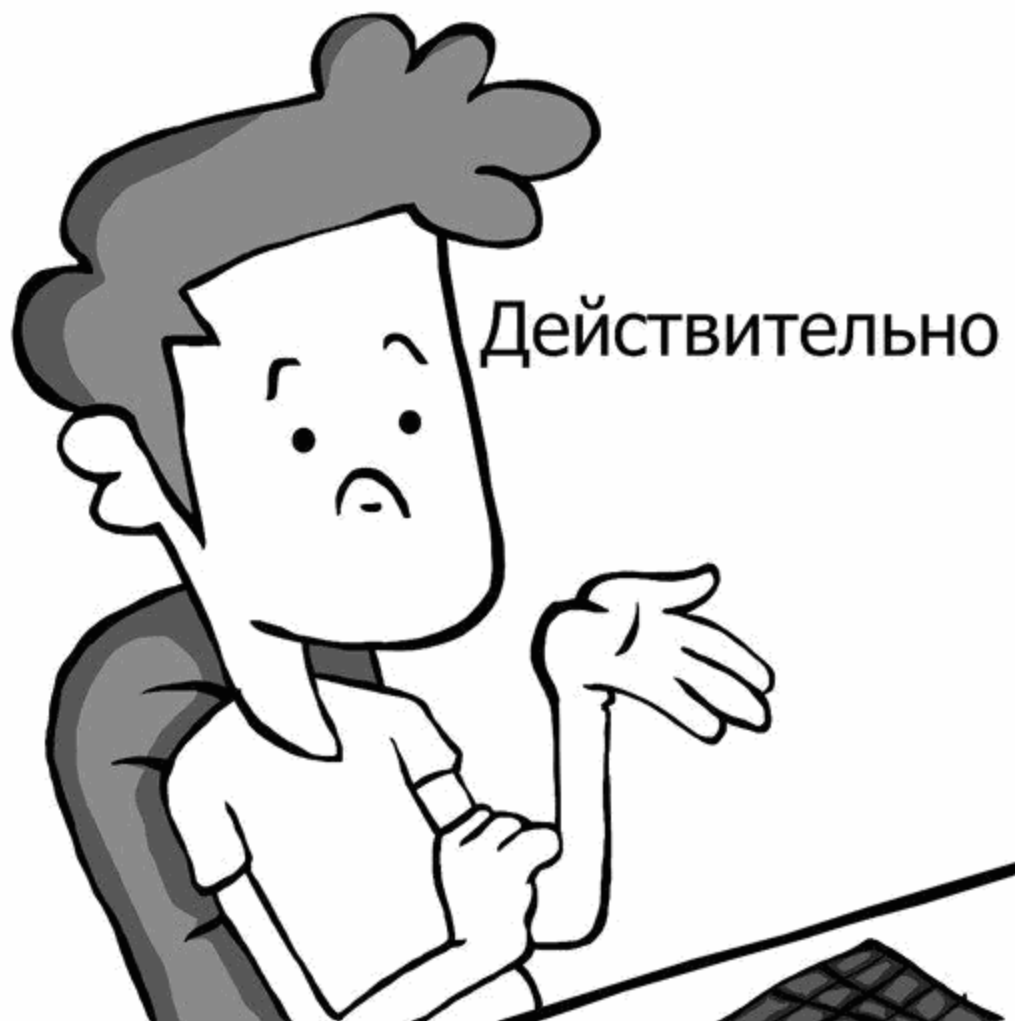
Размеры / позиция

- clientTop / clientLeft / clientHeight / clientWidth
- offsetTop / offsetLeft / offsetWidth / offsetHeight
- offsetParent
- scrollTop / scrollLeft / scrollWidth / scrollHeight
- getClientRects / getBoundingClientRect

Всякое...

- style
- textContent
- nodeType / nodeName / nodeValue / tagName
- tabIndex / accessKey
- dataset
- ...

137



```
01. for(var k in document.createElement('div')) {  
02.     (function(k) {  
03.         $$$.prototype[k] = function(arg) {  
04.             for(var i = 0; i < this.result.length; i++) {  
05.                 ...  
06.             }  
07.         }  
08.     })(k);  
09. }
```

```
01. $$$(...).getAttribute(...)
02. // return this.result[0].getAttribute(...)
03.
04. $$$(...).setAttribute(...)
05. // for(...) { elem.setAttribute(...) } return this
06.
07. $$$(...).focus()
08. // for(...) { elem.focus() } return this
```

```
01. $$$(...).innerHTML()  
02. // return this.result[0].innerHTML  
03.  
04. $$$(...).innerHTML(...)  
05. // for(...) { elem.innerHTML = ... } return this  
06.  
07. $$$(...).style({...})  
08. // for(...) { for({...}) { elem.style[k] = ... } } return this
```



```
01. $$$(...).parentNode()  
02. // result.push(elem); return this  
03.  
04. $$$(...).childNodes()  
05. // for(...) { result = result.concat(elems) } return this  
06.  
07. ...
```

```
01. $$$('div')
02.     .style({background: 'red', color: '#fff'})
03.     .setAttribute('data-test', 2)
04.     .addEventListener('click', function() { ... })
05.     .lastElementChild()
06.         .className('last-child')
07.         .querySelectorAll('span')
08.             .style({color: '#0f0'});
```

Yocto

github.com/deepsweet/yocto

- min.js ≈1K
- min.js.gz ≈600B
- кроссбраузерность?

MOAR!!!1

```
01. function Yocto(selector) {  
02.     if(selector && typeof selector === 'string') {  
03.         return [].push.apply(  
04.             this,  
05.             [].slice.call(doc.querySelectorAll(selector))  
06.         );  
07.     }  
08. }  
09. Yocto.prototype = [];  
10. Yocto.prototype.constructor = Yocto;
```

```
01. $$$('body')
02. // [<body>...</body>...]
03.
04. $$$('body') instanceof Yocto
05. $$$('body') instanceof Array
06. // true
07.
08. $$$('body').reduce
09. // function reduce() { [native code] }
```

```
01. if( !selector ) { return this; }  
02.  
03. if(selector.nodeType === 1) {  
04.     this[0] = selector;  
05.     this.length = 1;  
06.     return this;  
07. }  
08.  
09. if(typeof selector === 'string') { ... }
```

01. \$\$\$(`document.body`)

02. // [`<body>...</body>...`]

03.

04. \$\$\$()

05. // []

06.

07. \$\$\$('')

08. // []


```
01. $$$$.prototype.html = $$$$.prototype.innerHTML;
02.
03. $$$$.prototype.css = $$$$.prototype.style;
04.
05. $$$$.prototype.on = $$$$.prototype.addEventListener;
06.
07. $$$$.prototype.off = $$$$.prototype.removeEventListener;
08.
09. $$$$.prototype.find = $$$$.prototype.querySelectorAll;
```

```
01. $$$$.prototype. each = function(callback) {  
02.     [].forEach.call(this, function(el) {  
03.         callback.apply(el, arguments);  
04.     });  
05. };
```

```
01. $$$('div').each(function(el, i) {  
02.     console.log(this.tagName + ' #' + i);  
03. });  
04.  
05. // DIV #0  
06. // DIV #1  
07. // DIV #2  
08. // ...
```

```
01. ['add', 'remove', 'toggle', 'contains']
02. .forEach(function(method) {
03.     $$$.prototype[method + 'Class'] = function(cls) {
04.         for(var i = 0; i < this.length; i++) {
05.             var r = this[i].classList[method](cls);
06.             if(r !== undefined) { return r; }
07.         }
08.         return this;
09.     };
10. });
```

01. `$$$('body').addClass('test')`

02.

03. `$$$('body').removeClass('test')`

04.

05. `$$$('body').toggleClass('test')`

06. `// true|false`

07.

08. `$$$('body').hasClass('test')`

09. `// true|false`

```
01. $$$$.prototype.is = $$$$.prototype.matches ||  
02.     $$$$.prototype.webkitMatchesSelector ||  
03.     $$$$.prototype.mozMatchesSelector ||  
04.     $$$$.prototype.msMatchesSelector;
```

```
01. $$$('body').is('body')
```

```
02. // true
```

```
03.
```

```
04. $$$('div').is(':first-child')
```

```
05. // true|false
```

```
01. $$$$.ajax = function(p) {  
02.     var xhr = new XMLHttpRequest();  
03.     xhr.onload = p.onsuccess;  
04.     xhr.onerror = p.onerror;  
05.     xhr.open(p.type, p.url);  
06.     xhr.send(p.data);  
07.     return this;  
08. };
```



```
01. $$$$.ajax({  
02.     type: 'GET',  
03.     url: 'http://updates.html5rocks.com',  
04.     onSuccess: function() { console.log(this.statusText) },  
05.     onError: function() { console.log(this.status) }  
06. });  
07.  
08. // OK
```

```
01. $$$.prototype.trigger = function(name, params) {  
02.     var e = new CustomEvent(name, params || {});  
03.  
04.     this.each(function(elem) {  
05.         elem.dispatchEvent(e);  
06.     });  
07.  
08.     return this;  
09. };
```

```
01. $$$('body').on('myevent', function(e) {  
02.     console.log(e.type);  
03. });  
04.  
05. $$$('body').trigger('myevent');  
06.  
07. // 'myevent'
```

Почитать

- jQuery изнутри — введение clck.ru/4NVEy
- jQuery Annotated Source clck.ru/8sZnX
- JavaScript Library Boilerplate clck.ru/8sZnm
- JavaScript: Array vs NodeList vs HTMLCollection clck.ru/8sZnq
- Zepto.js zeptojs.com
- Microjs microjs.com
- 140byt.es www.140byt.es

Пишем свой jQuery за 20 минут

Кир Белевич, Яндекс

- tech.yandex.ru/people/398/
- deepsweet@yandex-team.ru
- twitter.com/deepsweet