

- Prototype 1.6.0

API

Prototype API

API Prototype Prototype

!

CHM *Remigijus Jodelis.*
2008/04/29

Prototype

<http://www.prototypejs.org/>

<http://www.prototypejs.org/api>



Prototype

JavaScript

Prototype

jQuery

Prototype

ORain orainyday@gmail.com <http://blog.csdn.net/orain>
2009-3-17

`$`, `$$`, `$A`, `$F`, `$H`, `$R`, `$w`, `Try.these`, `document.getElementsByClassName`

Prototype "" \$ Prototype \$

Javascript

`$()` Prototype \$ ID DOM

`document.getElementById`

Prototype Javascript

`$`

`$(id | element) -> HTMLElement` Or `$((id | element)...) -> [HTMLElement]`

ID DOM DOM Prototype

`$$`

`$(cssRule...) -> [HTMLElement...]`

CSS (CSS Selectors) CSS DOM document

`$A`

`$A(iterable) -> actualArray`

Array `Array.from` Array

`$F`

`$F(element) -> value`

`Form.Element.getValue` `Form.Element.getValue`

\$H

```
$H([obj]) -> Hash
```

Hash Hash JavaScript

\$R

```
$R(start, end[, exclusive = false]) -> ObjectRange
```

ObjectRange ObjectRange , \$R

\$w

```
$w(String) -> Array
```

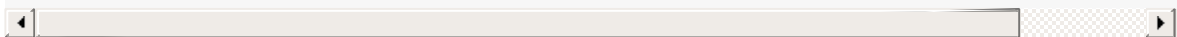
Ruby %w{foo bar} Perl qw(foo bar)

Try.these

```
Try.these(Function...) -> firstOKResult
```

document.getElementsByClassName

```
document.getElementsByClassName(className[, element]) -> [HTMLElement]
```



className CSS DOM

element CSS

\$

```
$(id | element) -> HTMLElement $((id | element)...) -> [HTMLElement  
HTML ID HTML ,
```

\$ Prototype

document.getElementById

ID DOM

```
function foo(element) { element = $(element); /* ... */ }
```

— ID

```
      $      $  
ID      null DOM      ID  
      Element.extend      Prototype DOM
```

```
Element.hide('itemId');  
// ...  
$('#itemId').hide();  
// DOM
```

\$

```
['item1', 'item2', 'item3'].each(Element.hide);  
$('#item1', 'item2', 'item3').invoke('hide');  
//
```

How Prototype extends the DOM

\$\$

```
$(cssRule...) -> [HTMLElement...]
```

CSS DOM HTML

DOM CSS

DOM

\$()

document.getElementById()getElementsByTagName() Prototype

getElementsByClassName()

[

```
$( 'div' );  
// -> DIV document.getElementsByTagName('div')  
$( '#contents' );  
// -> $( 'contents' )  
$( 'li.faux' );  
// -> Class='faux' LI
```

\$\$

Element#getEle

CSS

\$\$ CSS

CSS

v1.5.0

- div
- #a li
- CSS 2.1 [attr][attr=value][attr~=value]
[attr|=value] [attr!=value]
- CSS .highlighted .example.wrong
- ID #item1

v1.5.1

CSS3

pseudo-elements

::first-letter 1.5.1

- `#a > li`
`#a li ID 'a' li`
`'a' li`
- `~=`
- `:not` pseudo-class `#a *:not(li) #a LI`
- `:nth:first:last` `tr:nth-child(even) even`
`:nth-child() an+b an 2n, -3n, 5na b`
`1, 2, -3b an+b` `tr:nth-cl`
`color:red;} tr:nth-child(3)` `tr:nth`
`CSS3` `li:first-child (LI` `p:nth-]`
`type(3)`
- `:empty`
- `:enabled:disabled:checked`

```

$$('#contents a[rel]');
// -> ID "contents" rel
$$('a[href="#"]');
// -> href "#"
$$('#navbar a', '#sidebar a');
// -> ID "navbar" "sidebar"

```

v1.5.1

```

$$('a:not([rel~=nofollow])');
// -> rel "nofollow"
$$('table tbody > tr:nth-child(even)');
// -> tbody even
$$('div:empty');
// -> DIV

```

\$A

```
$A(iterable) -> actualArray
```

Array

Array.from

\$A NodeList DOM HTMLCollection arguments

Array

Array Prototype Array mix in

Prototype Array

Array nullundefined false Array toArray

Prototype toArray ""

DOM document.getElementsByTagName()

NodeList IE NodeList.prototype Enumerable

NodeList Array

```
var paras = $A(document.getElementsByTagName('p'));
paras.each(Element.hide);
$(paras.last()).show();
```

each Element.hide

DO

hide DOM

```
$A(document.getElementsByTagName('p')).map(Element.extend).invoke('
```

Array join

```
// ...  
function showArgs() { alert(Array.prototype.join.call(arguments, ' ',  
// ...  
function showArgs() { alert($A(arguments).join(' ', ' '));}
```



\$F

```
$F(element) -> value
```

Form.Element.getValue

Form.Element.getValue

\$H

`$H([obj]) -> Hash`

Hash	map	“”	Hash
Prototype	Hash	Hash	Hash

`$H`

```
var tmp = {a:'aa', b:'bb'};
var h = $H(tmp);
h.set('c', 'cc');

h.toQueryString(); //a=aa&b=bb&c=cc
$H(tmp).toQueryString(); //a=aa&b=bb
```

`$H` `Hash` **v1.5** `Hash`

\$R

```
$R(start, end[, exclusive = false]) -> ObjectRange
```

ObjectRange

ObjectRange

\$R

ObjectRange

\$R start end

exclusive

```
$R(0, 10).include(10)
// -> true
$A($R(0, 5)).join(', ')
// -> '0, 1, 2, 3, 4, 5'
$A($R('aa', 'ah')).join(', ')
// -> 'aa, ab, ac, ad, ae, af, ag, ah'
$R(0, 10, true).include(10)
// -> false
$R(0, 10, true).each(function(value){
    // 10 value 0 9
});
```

ObjectRange mix in

Enumerable

Array

toArray \$A

Enumerable min() max()

\$w

```
$w(String) -> Array
```

Array Ruby

%w{foo bar} Perl

qw(foo bar)

:-)

```
$w('apples bananas kiwis')  
// -> ['apples', 'bananas', 'kiwis']
```

```
$w('apples bananas kiwis').each(function(fruit){ var message = 'I l  
//  
})
```

Element

```
$w('ads navbar funkyLinks').each(Element.hide);
```

document.getElementsByClassName

```
document.getElementsByClassName(className[, element]) -> [HTMLElement]
```

className CSS element

Prototype 1.6

document.getElementsByClassName

NodeList Array v1.6

\$\$ Element#select

HTML

```
<body>
  <div id="one" class="foo">

  </div>
  <div id="two" class="foo bar thud">

  </div>
  <ul id="list">
    <li id="item_one" class="thud">List item 1</li>
    <li>List item 2</li>
    <li id="item_two" class="thud">List item 3</li>
  </ul>
</body>
```

JavaScript

```
document.getElementsByClassName('foo');
// -> [HTMLElement, HTMLElement] (div#one, div#two)
document.getElementsByClassName('thud');
// -> [HTMLElement, HTMLElement, HTMLElement] (div#two, li#item_one)
document.getElementsByClassName('thud', $('list'));
// -> [HTMLElement, HTMLElement] (li#item_one, li#item_two)
```

Try.these

```
Try.these(Function...) -> firstOKResult
```

Prototype **Ajax**

XMLHttpRequest IE6

XMLHttpRequest Javascript IE

Try.these

Try.these undefined

get

```
getTransport: function(){
    return Try.these(
        function() { return new XMLHttpRequest() },
        function() { return new ActiveXObject('Msxml2.XMLHT
        function() { return new ActiveXObject('Microsoft.XM
    ) || false;
}
```

Ajax

[Ajax Options](#), [Ajax.PeriodicalUpdater](#), [Ajax.Request](#), [Ajax.Responders](#),
[Ajax.Response](#), [Ajax.Updater](#)

Prototype AJAX Prototype Web AJAX
Ajax

Methods

Ajax Options

AJAX

Ajax.PeriodicalUpdater

```
new Ajax.PeriodicalUpdater(container, url[, options])
```

AJAX “decay”

Ajax.Request

```
new Ajax.Request(url[, options])
```

AJAX

Ajax.Responders

```
Ajax.Responders.register(responder)  
Ajax.Responders.unregister(responder)
```

Prototype AJAX

Ajax.Response 1.6

Ajax

Ajax.Updater

```
new Ajax.Updater(container, url[, options])
```

AJAX

Ajax

AJAX

Ajax

asynchronous	true	XMLHttpRequest
contentType	'application/x-www-form-urlencoded'	Content-Typeform enctype URL
encoding	'UTF-8'	
method	'post'	HTTP 'put' 'delete' 'get' Ruby On Rails Protoc 'post' '_method'
parameters	''	'get' URL URL URL Hash Prototype
postBody	None	'post' method

requestHeaders		<ul style="list-style-type: none">objectarray (0, 2...) (1, 3...) Prototype <ul style="list-style-type: none">X-Requested-With 'XMLHttpRequest'X-Prototype-Version Prototype 1.5.0Accept 'text/javascript, text/html, application/xml, text/xml, */*'Content-type contentType encoding
----------------	--	---

evalJS	true	content-type application/ecmas application/javascript, application/x-ecmascript, application/x-javascript, text/ecmascript, text/javascript, text/x-ecmascript text/x- javascript <u>SOP</u> Prototype
--------	------	--

evalJSON	true	content-type Ajax.Response#responseText Ajax.Response#responseJSON 'force' 'false'	application/json <u>SOP</u>
sanitizeJSON	false true	Ajax.Response#responseText	

Ajax.Request

onCreate (v1.5.1)	Ajax.Request URL	XHR
onComplete	on	X
onException	XHR	Ajax.Rec
onFailure	2xy onFailure	
onInteractive		
onLoaded	XHR	
onLoading	XHR	
onSuccess	2xy	
onUninitialized	XHR	
onXYZ	XYZ XMLHttpRequest onComplete	XYZ

responders

Ajax.Responders

Ajax.Request

XMLHttpRequest

X

null this

onCreate	Ajax XHR
onComplete	on
onException	XHR
onInteractive	
onLoaded	XHR
onLoading	XHR
onUninitialized	XHR

XHR

X

Ajax.Rec

Ajax.PeriodicalUpdater

```
new Ajax.PeriodicalUpdater(container, url[, options])
```

AJAX “decay”

“”

Ajax.Update decay

Ajax.PeriodicalUpdater Ajax.Updater

frequency 2 0.5Hz2 Ajax

decay 1 1

decay

```
new Ajax.PeriodicalUpdater('items', '/items', { method: 'get', freq
```

		Decay		Decay		
1	00:00	2	n/a	1	3	
2	00:03	1	yes	1	3	decay“ ” 1 decay 1
3	00:06	1	no	2	6	decay decay

4	00:12	2	no	4	12	
5	00:24	4	no	8	24	
6	00:48	8	yes	1	3	decay 1

PeriodicalUpdater

PeriodicalUpdater stop start

Ajax.PeriodicalUpdater **Ajax.Updater**

Ajax.Updater Ajax.Updater Ajax.Request evalJS
 getHeader onComplete

Ajax.Request

```
new Ajax.Request(url[, options])
```

AJAX .

AJAX “boilerplate

options hash eval Javascript

onSuccess

Ajax

new

```
var url = '/proxy?url=' + encodeURIComponent('http://www.google.com')
// proxy SOP
new Ajax.Request(url, {
  method: 'get',
  onSuccess: function(transport) {
    var notice = $('notice');
    if (transport.responseText.match(/href="http:\/\/pr
      notice.update('Yeah! You are in the Top 10!')
    else
      notice.update('Damn! You are beyond #10...')
    }
  }
});
```

XMLHttpRequest

1. Created
2. Initialized
3. Request sent
4. Response being received

5. Response received

Ajax Prototype AJAX

1. `onCreate` `AJAX`
2. `onUninitialized`""[Created]
3. `onLoading`""[Initialized]
4. `onLoaded`""[Request sent]
5. `onInteractive`""[Response being received]
6. `onXYZXYZ` `onSuccess` `onFailure`
7. `onComplete`

Response received

`onFailure` `onFailureonComplete`

`XMLHttpRequest`

`onLoaded` `onInteractive`

`onCreateonUninitialized`

`onSuccess` `onFailure,`

`Ajax.Request` `XHR`""

```
//  
new Ajax.Request('/your/url', {  
  onComplete: function(transport) {  
    if (200 == transport.status)  
      // yada yada yada  
  }  
});
```

"" "2xy"

Prototype

```
new Ajax.Request('/your/url',{
    onSuccess: function(transport){
        // yada yada yada
    }
});
```

Javascript

AJAX

SOP

content-type Javascript

responseText

eval

AJAX Javascript Prototype

MIME

- application/ecmascript
- application/javascript
- application/x-ecmascript
- application/x-javascript
- text/ecmascript
- text/javascript
- text/x-ecmascript
- text/x-javascript

MIME

1

success() XHR status

2xy

transport.status

2 HTTP

XHR `getResponseHeader` XHR

`Ajax.Response#getHeader`

```
new Ajax.Request('/your/url', {
  onSuccess: function(response){
    // null
    if ((response.getHeader('Server') || '').match(/Apache/))
      ++gApacheCount;
    //
  }
});
```

3 JSON

JSON `X-JSON` JSON

`Ajax.Response` `headerJSON`

```
new Ajax.Request('/your/url', {
  onSuccess: function(transport) {
    transport.headerJSON
  }
});
```

Ajax.Responders

```
Ajax.Responders.register(responder) Ajax.Responders.unregister(re
```

Prototype AJAX

AJAX

Ajax.Request Ajax.Updater

Ajax.PeriodicalUpdater

AJAX

Prototype

Ajax.Responders

Prototype

```
//  
var responders = new Object();  
responders.onCreate = function() { Ajax.activeRequestCount++; }; //  
responders.onComplete = function() { Ajax.activeRequestCount--; };  
Ajax.Responders.register(responders);
```

Prototype

Ajax.activeRe

Prototype

onCreate onComplete

```
Ajax.Responders.register({  
  onCreate: function() { Ajax.activeRequestCount++; },  
  onComplete: function() { Ajax.activeRequestCount--; }  
});
```

onCreate

onUninitialized

...

Ajax.Response 1.6

Ajax

XmlHttpRequest ActiveX
headerJSON JSON
Ajax.Request XmlHttpRequest Ajax.Request
Ajax.Response

Ajax.Response

status	Number		
statusText	String	HTTP	
readyState	Number	0 "Uninitialized"1 "Loading"2 "Loaded"3 "Interactive"4 "Complete"	
responseText	String		
responseXML	document null	content-type	application/xml XML
responseJSON	Object, Array null	content-type	application/json
headerJSON	Object, Array null	X-JSON	null
request	Object	Ajax.Request Ajax.Updater	
transport	Object	xmlHttpRequest	

Ajax.Response

getHeader(name)	String null	null	XHR
getAllHeaders()	String	\r\n	Http

	null		
getResponseHeader(name)	String	getHeader	XmlHttp
getAllResponseHeaders()	String	getAllResponseHeaders	\r\n getAllH

Ajax.Updater

```
new Ajax.Updater(container, url[, options])
```

AJAX

Ajax.Updater **Ajax.Request**

Aja

```
new Ajax.Updater('items', '/items', {  
  parameters: { text:$F('text') }  
});
```

onComplete

Ajax.Updater AJAX DOM

evalScripts false <script>

Element.update 1.6.0
insertion None **Insertion** **Insertion.Bottom 1.6.0**
'top' 'bottom' 'before' 'after'

AJAX XHTML

```
new Ajax.Updater('items', '/items',  
  { parameters: { text: $F('text') },  
    insertion: Insertion.Bottom  
  });
```

evalScripts

evalScripts: true <script>

eval()

- Prototype var

var myVariant = 'this is a example'

-

```
// Ajax.Updater
function coolFunc() {
    // Amazing stuff!
}
```

coolFunc var

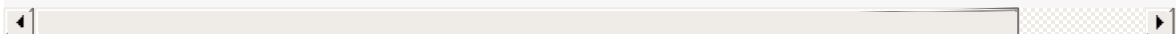
```
// Ajax.Updater
coolFunc = function() {
    // Amazing stuff!
}
```

success/failure

container DOM success failure

```
new Ajax.Updater({ success: 'items' }, '/items', {
    parameters: { text: $F('text') },
    insertion: Insertion.Bottom
});
```

```
new Ajax.Updater({ success: 'items', failure: 'notice' },
    '/items',
    { parameters: { text: $F('text') }, insertion: Insertion.Bottom
});
```



Array

clear, clone, compact, each, first, flatten, from, indexOf, inspect, last, reduce,
reverse, size, toArray, toJSON, uniq, without

Prototype Javascript

- **Enumerable** `Enumerable`
-

Prototype

for...in **bye-bye**

Javascript

`for...in`

ECMA 262

ECMAScript

`concat``join``pop``push`

`for...in` §12.6.4 `in`

`for...in`

`for...in`

`for...in`

Prototype

`Array.prototype`

Enumerable Prototype `Array`

```
for (var index = 0; index < myArray.length; ++index) {  
    var item = myArray[index];  
    // ...  
}
```

iterators each

```
myArray.each(function(item) {  
    // ...  
});
```

Prototype for...in

```
//  
for (var index = 0, len = myArray.length; index < len; ++index) {  
    var item = myArray[index];  
    // ...  
}
```

for index < myArray.length Javascript

clear

```
clear() -> Array
```

clone

```
clone() -> newArray
```

compact

```
compact() -> newArray
```

null/undefined

each


```
each(iterator) -> Array
```

first

```
first() -> value
```

```
undefined
```

flatten

```
flatten() -> newArray
```

from

```
Array.from(iterable) -> actualArray
```

```
$A()    $A
```

indexOf

```
indexOf(value) -> position
```

```
-1
```

inspect

```
inspect() -> String
```

```
"['a', ['b', [Object Object]], 'c']"
```

last

```
last() -> value
```

```
undefined
```

reduce

```
reduce() -> Array | singleValue
```

reverse

```
reverse([inline = true]) -> Array
```

`inline` `false`

size

```
size() -> Number
```

toArray

```
toArray() -> newArray
```

`Enumerable` `toArray`

toJSON 1.5.1

```
toJSON() -> String
```

JSON

uniq

```
uniq() -> newArray
```

without

```
without(value...) -> newArray
```

clear

`clear()` -> Array

```
var guys = ['Sam', 'Justin', 'Andrew', 'Dan'];  
guys.clear();  
// -> []  
guys  
// -> []
```

clone

```
clone() -> newArray
```

```
var fruits = ['Apples', 'Oranges'];  
var myFavs = fruits.clone();  
myFavs.pop();  
// fruits -> ['Apples', 'Oranges']  
// myFavs -> ['Apples']
```

compact

```
compact() -> newArray
```

```
null/undefined
```

```
['frank', , 'sue', , 'sally', null].compact()  
// -> ['frank', 'sue', 'sally']
```

each

```
each(iterator) -> Array
```

each

Enumerable

first

```
first() -> value
```

```
undefined
```

```
['Ruby', 'Php', 'Python'].first()  
// -> 'Ruby'  
[].first()  
// -> undefined
```

flatten

```
flatten() -> newArray
```

```
['frank', ['bob', 'lisa'], ['jill', ['tom', 'sally']]].flatten()  
// -> ['frank', 'bob', 'lisa', 'jill', 'tom', 'sally']
```


from

```
Array.from(iterable) -> actualArray
```

\$A()

\$A

indexOf

```
indexOf(value) -> position
```

```
-1
```

```
==
```

```
===
```

Javascript Array

```
[3, 5, 6, 1, 20].indexOf(1)
// -> 3
[3, 5, 6, 1, 20].indexOf(90)
// -> -1
[0, false, 15].indexOf(false)
// -> 0 0 1 0 == false!
```

inspect

```
inspect() -> String
```

```
"['a', ['b', [Object Object]], 'c']"
```

`inspect` `Object.inspect`

```
['Apples', {good: 'yes', bad: 'no'}, 3, 34].inspect()  
// -> "['Apples', [object Object], 3, 34]"
```

`join`

```
['apples', 'bananas', 'kiwis'].join(', ')  
// -> 'apples, bananas, kiwis'
```

last

```
last() -> value
```

```
undefined
```

```
['Ruby', 'Php', 'Python'].last()  
// -> 'Python'  
[].last()  
// -> undefined
```

reduce

```
reduce() -> Array | singleValue
```

```
[3].reduce();  
// -> 3  
[3, 5].reduce();  
// -> [3, 5]
```

reverse

```
reverse([inline = true]) -> Array
```

`inline`

`false`

```
var nums = [3, 5, 6, 1, 20];  
nums.reverse(false)  
// -> [20, 1, 6, 5, 3]  
nums  
// -> [3, 5, 6, 1, 20]  
nums.reverse()  
// -> [20, 1, 6, 5, 3]  
nums  
// -> [20, 1, 6, 5, 3]
```

size

```
size() -> Number
```

Enumerable

size

length

Array.size `return this.length;`

Enumerable.size `return this.toArray().length;`

toArray

```
toArray() -> newArray
```

```
    Enumerable toArray
```

```
clone    Enumerable    toArray
```


toJSON

1.5.1

toJSON() -> String

JSON

```
['a', {b: null}].toJSON();  
//-> '["a", {"b": null}]'
```

uniq

```
uniq() -> newArray
```

```
['Sam', 'Justin', 'Andrew', 'Dan', 'Sam'].uniq();  
// -> ['Sam', 'Justin', 'Andrew', 'Dan']  
['Prototype', 'prototype'].uniq();  
// -> ['Prototype', 'prototype']
```

- $O(n^2)$
- JavaScript [Array.concat](#)

without

```
without(value...) -> newArray
```

```
[3, 5, 6, 1, 20].without(3)  
// -> [5, 6, 1, 20]  
[3, 5, 6, 1, 20].without(20, 6)  
// -> [3, 5, 1]
```

Class

`addMethods`, `create`

Prototype OOP

`Object.extend`

`addMethods` 1.6.0

```
addMethods(methods)
```

`create`

```
create([superclass][, methods...]) -> Class
```

addMethods 1.6.0

addMethods(methods)

Class#addMethods Class.create

\$super

```
var Animal = Class.create({
  initialize: function(name, sound) {
    this.name = name;
    this.sound = sound;
  },
  speak: function() {
    alert(this.name + " says: " + this.sound + "!");
  }
});
// Animal
var Snake = Class.create(Animal, {
  initialize: function($super, name) {
    $super(name, 'hissssssssss');
  }
});
var ringneck = new Snake("Ringneck", "hissssssss");
ringneck.speak(); //-> "Ringneck says: hissssssss!"
// Snake#speak ( $super )
Snake.addMethods({
  speak: function($super) {
    $super();
    alert("You should probably run. He looks really mad.");
  }
});
ringneck.speak();
//-> "Ringneck says: hissssssss!"
//-> "You should probably run. He looks really mad."
// Animal#speak
Animal.addMethods({
  speak: function() {
    alert(this.name + 'snarls: ' + this.sound + '!');
  }
});
ringneck.speak();
//-> "Ringneck snarls: hissssssss!"
//-> "You should probably run. He looks really mad."
```



create

```
create([superclass][, methods...]) -> Class
```

Class.create initialize

1.6 Class.create Class

\$super

Class.addMethods

- superclass null
- subclasses

JavaScript constructor

```
var Animal = Class.create({
  initialize: function(name, sound) {
    this.name = name;
    this.sound = sound;
  },
  speak: function() {
    alert(this.name + " says: " + this.sound + "!");
  }
});
// Animal
var Snake = Class.create(Animal, {
  initialize: function($super, name) {
    $super(name, 'hisssssssss');
  }
});
var ringneck = new Snake("Ringneck");
```

```

ringneck.speak();
//-> "Ringneck says: hisssssssss!"

var rattlesnake = new Snake("Rattler");
rattlesnake.speak();
//-> "Rattler says: hisssssssss!"

// Enumerable
var AnimalPen = Class.create(Enumerable, {
  initialize: function() {
    var args = $A(arguments);
    if (!args.all( function(arg) { return arg instanceof Snake; } ))
      throw "Only animals in here!"
    this.animals = args;
  },
  // Enumerable _each
  _each: function(iterator) {
    return this.animals._each(iterator);
  }
});

var snakePen = new AnimalPen(ringneck, rattlesnake);
snakePen.invoke('speak');
//-> "Ringneck says: hisssssssss!"
//-> "Rattler says: hisssssssss!"

```

1.6

Prototype 1.6

`Class.create`

`initialize`

Ruby

```

var Animal = Class.create();
Animal.prototype = {
  initialize: function(name, sound) {
    this.name = name;
    this.sound = sound;
  },
  speak: function() {
    alert(name + " says: " + sound + "!");
  }
};

var snake = new Animal("Ringneck", "hisssssssss");
snake.speak();
// -> "Ringneck says: hisssssssss!"
var Dog = Class.create();
Dog.prototype = Object.extend(new Animal(), {
  initialize: function(name) {
    this.name = name;
    this.sound = "woof";
  }
});

```



```
});  
var fido = new Dog("Fido");  
fido.speak();  
// -> "Fido says: woof!"
```

Date

toJson

toJson 1.5.1

```
toJson() -> String
```

JSON ISO

toJSON

1.5.1

toJSON() -> String

JSON ISO

```
new Date(1969, 11, 31, 19).toJSON();  
// -> '"1969-12-31T19:00:00"'
```

Element

[absolutize](#), [addClassName](#), [addMethods](#), [adjacent](#), [ancestors](#), [childElements](#), [classNames](#), [cleanWhitespace](#), [clonePosition](#), [cumulativeOffset](#), [cumulativeScrollOffset](#), [descendantOf](#), [descendants](#), [down](#), [empty](#), [extend](#), [fire](#), [firstDescendant](#), [getDimensions](#), [getElementsByClassName](#), [getElementsBySelector](#), [getHeight](#), [getOffsetParent](#), [getStyle](#), [getWidth](#), [hasClassName](#), [hide](#), [identify](#), [immediateDescendants](#), [insert](#), [inspect](#), [makeClipping](#), [makePositioned](#), [match](#), [next](#), [nextSiblings](#), [observe](#), [positionedOffset](#), [previous](#), [previousSiblings](#), [readAttribute](#), [recursivelyCollect](#), [relativize](#), [remove](#), [removeClassName](#), [replace](#), [scrollTo](#), [select](#), [setOpacity](#), [setStyle](#), [show](#), [siblings](#), [stopObserving](#), [toggle](#), [toggleClassName](#), [undoClipping](#), [undoPositioned](#), [up](#), [update](#), [viewportOffset](#), [visible](#), [wrap](#), [writeAttribute](#)

Element

Element

DOM

DOM

“Prototype

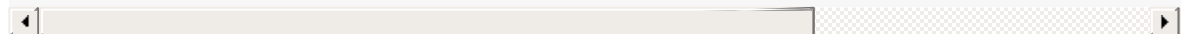
DOM” Prototype DOM

```
<div id="message" class=""></div>
```

```
// div#message CSS
$('message').addClassName('read');
// -> div#message
//
Element.toggleClassName('message', 'read');
// -> div#message
```

Element

```
$('message').addClassName('read').update('I read this message!').se
```



Element 1.6

```
new Element(tagName[, attributes])
```

Element DOM

Element#update

“/”

Element#writeAttribute

```
var a = document.createElement('a');
a.setAttribute('class', 'foo');
a.setAttribute('href', '/foo.html');
a.appendChild(document.createTextNode("Next page"));
```

```
var a = new Element('a', { 'class': 'foo', href: '/foo.html' }).up()
```

absolutize 1.6

```
absolutize(element) -> HTMLElement
```

```
element.style.position = 'absolute'
```

addClassName

```
addClassName(element, className) -> HTMLElement
```

CSS element

addMethods

```
addMethods([methods])
addMethods(tagName, methods)
```

[illegible]

adjacent 1.6

```
Element.adjacent(element[, selectors...]) -> [HTMLElement...]
```

```
someElement.adjacent([selectors...]) -> [HTMLElement...]
```

selectors

ancestors

```
ancestors(element) -> [HTMLElement...]
```

element ...

childElements 1.5.1

```
childElements(element) -> [HTMLElement...]
```

classNames

```
classNames(element) -> Enumerable
```

classNames Enumerable CSS

cleanWhitespace

```
cleanWhitespace(element) -> HTMLElement
```

XML HTML

```
<div>  
  <p></p>  
</div>
```

<div> element.childNodes <p> ,
Prototype cleanWhitespace

clonePosition 1.6

```
clonePosition(element, source[, options]) -> HTMLElement
```

source options / element

cumulativeOffset 1.6

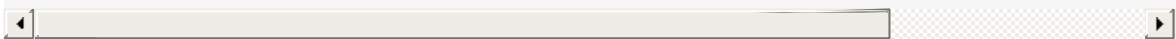
```
cumulativeOffset(element) -> [Number, Number] { left: Number, top:
```



`element`
`offset X` `offset[0]` `offset.left Y` `offset[1]` `offset.top`

cumulativeScrollOffset 1.6

```
cumulativeScrollOffset(element) -> [Number, Number] { left: Number
```



`offset X` `offset[0]` `offset.left Y` `offset[1]` `offset.top`

descendantOf

```
descendantOf(element, ancestor) -> Boolean
```

`element` `ancestor`

descendants

```
descendants(element) -> [HTMLElement...]
```

`element`

down

```
down(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

`element` `cssRule` `index` *index* `cssRule`

empty

```
empty(element) -> Boolean
```

`element`

extend

```
extend(element)
```

`element` `Element.Methods` `Element.Methods.Simulated` `element`
`inputtextarea` `select` `Form.Element.Methods` `form` `Form.M`

fire 1.6.0

```
fire(eventName[, memo]) -> Event
```

firstDescendant 1.5.1

```
firstDescendant(element) -> HTMLElement
```

DOM `firstChild` `firstChild` [

getDimensions

```
getDimensions(element) -> {height: Number, width: Number}
```

`element` `width` `height`

getElementsByClassName []

```
getElementsByClassName(element, className) -> [HTMLElement...]
```

`element` `className` CSS

getElementsBySelector []

```
getElementsBySelector(element, selector...) -> [HTMLElement...]
```

CSS `element`

getHeight

```
getHeight(element) -> Number
```

`element`

getOffsetParent 1.6

```
getOffsetParent(element) -> HTMLElement
```


`element` Containing Block CSS `position` `relative` `absolute`
`body`

getStyle

```
getStyle(element, property) -> String | null
```

`element` CSS `property` CSS camelized CSS
`border-width` `borderWidth`

DOM `element.style.xxx`

getWidth

```
getWidth(element) -> Number
```

`element`

hasClassName

```
hasClassName(element, className) -> Boolean
```

`element` `className` CSS

hide

```
hide(element) -> HTMLElement
```

`element`

identify 1.6

```
identify(element) -> id
```

`element` `id` `id` `id` `id`

immediateDescendants []

```
immediateDescendants(element) -> [HTMLElement...]
```

insert 1.6

```
insert(element, { position: content }) -> HTMLElement
```

```
insert(element, content) -> HTMLElement
```

position content content content
position beforeaftertop bottom

inspect

```
inspect(element) -> String
```

element

makeClipping

```
makeClipping(element) -> HTMLElement
```

'hidden' CSS clip element

makePositioned

```
makePositioned(element) -> HTMLElement
```

CSS block element CSS position 'static' undefined
'relative'

match

```
match(element, selector) -> Boolean
```

element selector CSS

next

```
next(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

cssRule element index index cssRule

nextSiblings

```
nextSiblings(element) -> [HTMLElement...]
```

element

observe

```
observe(element, eventName, handler[, useCapture = false]) -> HTML
```

element

positionedOffset 1.6

```
positionedOffset(element) -> [Number, Number] { left: Number, top:
```

element Containing Block CSS position relative absolute
element Element#getOffsetParent
offset X offset[0] offset.left Y offset[1] offset.top

previous

```
previous(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

cssRule element index index cssRule

previousSiblings

```
previousSiblings(element) -> [HTMLElement...]
```

element

readAttribute

```
readAttribute(element, attribute) -> String | null
```

elemnet attribute null

recursivelyCollect

```
recursivelyCollect(element, property) -> [HTMLElement...]
```

element property property element DOM

relativize 1.6

```
relativize(element) -> HTMLElement
```

```
element.style.position = 'relative'
```

1.6.0.3 2066

```
element._originalWidth = element.style.width;  
element._originalHeight = element.style.height;
```

remove

```
remove(element) -> HTMLElement
```

element

removeClassName

```
removeClassName(element, className) -> HTMLElement
```

element className CSS

replace

```
replace(element[, html]) -> HTMLElement
```

```
html    element    element  
IE      element.outerHTML = html
```

scrollTo

```
scrollTo(element) -> HTMLElement
```

elemnet

select 1.6

```
select(element, selector...) -> [HTMLElement...]
```

CSS

element

setOpacity 1.5.1

```
Element.setOpacity(element, opacity) -> [HTMLElement...]  
someElement.setOpacity(opacity) -> [HTMLElement...]
```

`element` `opacity 0 1` `0 1`

setStyle

```
setStyle(element, styles) -> HTMLElement
```

`element` `CSS` `styles / Hash` `camelized`
`width` `borderWidth` `border-width`

show

```
show(element) -> HTMLElement
```

`element`

siblings

```
siblings(element) -> [HTMLElement...]
```

`element`

stopObserving

```
stopObserving(element, eventName, handler) -> HTMLElement
```

`element`

toggle

```
toggle(element) -> HTMLElement
```

`element`

toggleClassName

```
toggleClassName(element, className) -> HTMLElement
```

`element` `CSS` `className`

undoClipping

```
undoClipping(element) -> HTMLElement
```

`element` `CSS` `overflow` `Element.makeClipping()`

undoPositioned

```
undoPositioned(element) -> HTMLElement
```

`element` `Element.makePositioned`

up

```
up(element, [cssRule][, index = 0]) -> HTMLElement | undefined
```

`element` `cssRule` `index` *index* `cssRule`

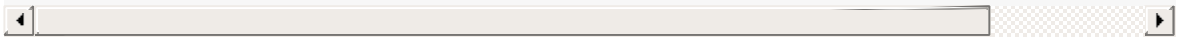
update

```
update(element[, newContent]) -> HTMLElement
```

`element` `newContent`
`element.innerHTML = newContent`

viewportOffset 1.6

```
viewportOffset(element) -> [Number, Number] { left: Number, top: N
```



`element`
`offset X` `offset[0]` `offset.left Y` `offset[1]` `offset.top`

visible

```
visible(element) -> Boolean
```

`Boolean` `element` `style` `"display:none;"`

wrap 1.6

```
Element.wrap(element, wrapper[, attributes]) -> HTMLElement  
someElement.wrap(wrapper[, attributes]) -> HTMLElement
```

`element` `wrapper` `wrapper`

writeAttribute 1.6

```
writeAttribute(element, attribute[, value = true]) -> HTMLElement  
writeAttribute(element, attributes) -> HTMLElement
```

element hash /

absolutize 1.6

```
absolutize(element) -> HTMLElement
```

```
element.style.position = 'absolute'
```


addClassName

```
addClassName(element, className) -> HTMLElement
```

CSS `element`

```
<div id="mutsu" class="apple fruit"></div>
```

```
$('#mutsu').addClassName('food')  
$('#mutsu').className  
// -> 'apple fruit food'  
$('#mutsu').classNames()  
// -> ['apple', 'fruit', 'food']
```

addMethods

```
addMethods([methods])  
addMethods(tagName, methods)
```

hash	Element hash
HTML tagName	HTML

Element.addMethods Element \$() Element

```
$(element).myOwnMethod([args...]);
```

```
Element.myOwnMethod(element|id[, args...]);
```

Element.addMethods hash hash

element \$()

[methods]

```
var myVeryOwnElementMethods = {  
  myFirstMethod: function(element[, args...]) {  
    element = $(element);  
    //  
    return element;  
  },  
  mySecondMethod: function(element[, args...]) {  
    element = $(element);  
    //  
    return element;  
  }  
};
```

v1.5.1

Element.addMethods HTML

```
Element.addMethods('DIV', my_div_methods);
```

```
// DIV
```

```
Element.addMethods(['DIV', 'SPAN'], my_div_methods);  
// DIV SPAN
```

Element.addMethods

getAttribute

innerHTML

Prototype Prototype

<div>

E1

```
Element.addMethods({  
  wrap: function(element, tagName) {  
    element = $(element);  
    var wrapper = document.createElement('tagName');  
    element.parentNode.replaceChild(wrapper, element);  
    wrapper.appendChild(element);  
    return Element.extend(wrapper);  
  }  
});
```

```
//  
<p id="first">Some content...</p>
```

```
$(element).wrap('div'); // -> HTMLElement (div)
```

```
//  
<div><p id="first">Some content...</p></div>
```

Element.wrap <div>

Element.extend

```
$(element).wrap('div').setStyle({backgroundImage: 'url(images/round
```

Web

Ajax.Updater DOM

```
Element.addMethods({
  ajaxUpdate: function(element, url, options) {
    element = $(element);
    element.update(' HTMLElement
```

element Ajax spinner.gif

Ajax.Up

element Ajax

Element.addMethods

Element.addMethods

Element.MethodsElement.Methods.SimulatedForm.Methods

Form.Element.Methods DOM Form.Methods form

Form.Element.Methods inputselect textarea

Element.addMethods

“Please wait...”

```
Form.Element.Methods.processing = function(element, text) {
  element = $(element);
  if (element.tagName.toLowerCase() == 'input' && ['button',
  {
    element.value = (text === undefined ? 'Please wait.'
    element.disable()
  }
  return element;
};

Element.addMethods();
```

```
addMethods(tagName, methods)
```

adjacent 1.6

```
Element.adjacent(element[, selectors...]) -> [HTMLElement...]  
someElement.adjacent([selectors...]) -> [HTMLElement...]
```

selectors

```
<ul id="cities">  
  <li class="us" id="nyc">New York</li>  
  <li class="uk" id="lon">London</li>  
  <li class="us" id="chi">Chicago</li>  
  <li class="jp" id="tok">Tokyo</li>  
  <li class="us" id="la">LosAngeles</li>  
  <li class="us" id="aus">Austin</li>  
</ul>
```

```
$('nyc').adjacent('li.us');  
//-> [<li#chi, li#la, li#aus>]
```

ancestors

```
ancestors(element) -> [HTMLElement...]
```

```
element ...
```

element parentNode

Prototype DOM

```
<html>
  [...]
  <body>
    <div id="father">
      <div id="kid"> </div>
    </div>
  </body>
</html>
```

```
$('#kid').ancestors();
// -> [div#father, body, html]
// "body" "html"
document.getElementsByTagName('html')[0].ancestors();
// -> []
```

childElements 1.5.1

```
childElements(element) -> [HTMLElement...]
```

0

Prototype DOM

```
<div id="australopithecus">
  <div id="homo-erectus">
    <div id="homo-neanderthalensis"></div>
    <div id="homo-sapiens"></div>
  </div>
</div>
```

```
$('australopithecus').childElements();
// -> [div#homo-erectus]
$('homo-erectus').childElements();
// -> [div#homo-neanderthalensis, div#homo-sapiens]
$('homo-sapiens').childElements();
// -> []
```


classNames

```
classNames(element) -> Enumerable
```

```
classNames      Enumerable CSS
```

```
Element#classNames      Element#addClassName
```

```
Element#removeClassName
```

```
Element#hasClassName
```

```
CSS
```

```
$w(element.classNames)
```

```
Enumerable CSS
```

```
className
```

```
Enumerable CSS
```

```
set(className) add(className)
```

```
remove(className) Element.addClassName
```

```
Element.toggleClassName Element.removeClassName
```

```
<div id="mutsu" class="apple fruit food"></div>
```

```
$('mutsu').classNames().inspect()  
// -> "#<Enumerable:['apple', 'fruit', 'food']>"  
// CSS  
$('mutsu').className = 'fruit round'  
$('mutsu').classNames().inspect()  
// -> "#<Enumerable:['fruit', 'food']>"
```

cleanWhitespace

```
cleanWhitespace(element) -> HTMLElement
```

XML HTML

```
<div>  
<p></p>  
</div>
```

```
<div> element.childNodes <p> ,  
<p> Prototype
```

`cleanWhitespace`

`Element.cleanWhitespace` `nextSibling`
`previousSibling` `firstChild` `lastChild` `DOM`

`Element.up` `Element.down`

`Element.previous`

HTML

```
<ul id="apples">  
  <li>Mutsu</li>  
  <li>McIntosh</li>  
  <li>Ida Red</li>  
</ul>
```

```
var element = $('apples');  
element.firstChild.innerHTML;  
// -> undefined
```

`ul#apples`

```
element.cleanWhitespace();
```

DOM

```
<UL id="apples"><LI>Mutsu</LI><LI>McIntosh</LI><LI>Ida Red</LI></UL>
```



`firstChild`

```
element.firstChild.innerHTML;  
// -> 'Mutsu'
```

clonePosition 1.6

```
clonePosition(element, source[, options]) -> HTMLElement
```

```
source options / element
```

CSS containing block

```
setLeft    true    source CSS    left    element
setTop     true    source CSS    top     element
setWidth   true    source CSS    width   element
setHeight  true    source CSS    height  element

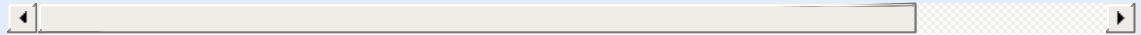
                                element CSS    left
offsetLeft 0            elemnet CSS    left    source    left
                                offsetLeft

                                element CSS    top
offsetTop  0            elemnet CSS    top     source    top
                                offsetTop
```

```
options hash            element width height options
{setWidth:false, setWidth:false}
```

cumulativeOffset 1.6

```
cumulativeOffset(element) -> [Number, Number] { left: Number, to
```



element
offset X
offset.top

offset[0]

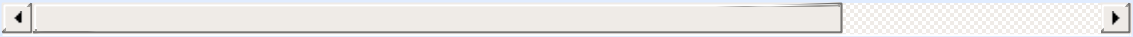
offset.left Y

offset[1]

offsetLeft offsetTop

cumulativeScrollOffset 1.6

```
cumulativeScrollOffset(element) -> [Number, Number] { left: Number
```



offset X	offset[0]	offset.left Y	offset[1]
<code>offset.top</code>			

`scrollLeft` `scrollTop`

descendantOf

```
descendantOf(element, ancestor) -> Boolean
```

```
element  ancestor
```

```
Element.descendantOf  ancestor  $()  ancestor ID
```

```
<div id="australopithecus">  
  <div id="homo-herectus">  
    <div id="homo-sapiens"></div>  
  </div>  
</div>
```

```
$('homo-sapiens').descendantOf('australopithecus');  
// -> true  
$('homo-herectus').descendantOf('homo-sapiens');  
// -> false
```

descendants

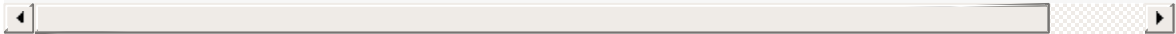
```
descendants(element) -> [HTMLElement...]
```

`element`

Prototype DOM

```
<div id="australopithecus">
  <div id="homo-herectus">
    <div id="homo-neanderthalensis"></div>
    <div id="homo-sapiens"></div>
  </div>
</div>
```

```
$('australopithecus').descendants();
// -> [div#homo-herectus, div#homo-neanderthalensis, div#homo-sapie
$('homo-sapiens').descendants();
// -> []
```



down

```
down(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

element	cssRule	index	<i>index</i>	cssRule
---------	---------	-------	--------------	---------

Element.down Prototype DOM

Element.up

Element.next Element.previous

Prototype / CSS

```
$(element).down(1).next('li', 2).hide();
```

DOM

element	firstChild	Element.down
---------	------------	--------------

index	element	Element.des
-------	---------	-------------

0

cssRule	Element.down
---------	--------------

cssRule	index	Element.down	CSS
---------	-------	--------------	-----

undefined

```
<ul id="fruits">
  <li id="apples">
    <ul>
      <li id="golden-delicious">Golden Delicious</li>
      <li id="mutsu" class="yummy">Mutsu</li>
      <li id="mcintosh" class="yummy">McIntosh</li>
      <li id="ida-red">Ida Red</li>
    </ul>
  </li>
</ul>
```

```
        </ul>
    </li>
</ul>
```

```
$('#fruits').down();
// $('#fruits').down(0);
// -> li#apple
$('#fruits').down(3);
// -> li#golden-delicious
$('#apples').down('li');
// -> li#golden-delicious
$('#apples').down('li.yummy');
// -> li#mutsu
$('#fruits').down('.yummy', 1);
// -> li#mcintosh
$('#fruits').down(99);
// -> undefined
```

empty

```
empty(element) -> Boolean
```

```
element
```

```
<div id="wallet"> </div>  
<div id="cart">full!</div>
```

```
$('wallet').empty();  
// -> true  
$('cart').empty();  
// -> false
```

extend

```
extend(element)
```

element	Element.Methods	Element.Methods.Simulated	element
inputtextarea	select	Form.Element.Methods	form
Form.Methods			

Prototype

```
element.update('hello world');
```

Element

```
element.update('hello world').addClassName('greeting');
```

Element

Element.siblings Prototype

\$\$()

Prototype DOM .

fire 1.6.0

```
fire(eventName[, memo]) -> Event
```

Element#fire

memo

Element#fire

Element#fire

MyEvent:disable DOM

mousewheel DOMMouseScroll

```
document.observe("widget:frobbed", function(event) {
    console.log("Element with ID (" + event.target.id + ") frobbed");
});

var someNode = $('foo');
someNode.fire("widget:frobbed", { widgetNumber: 19 });
//-> "Element with ID (foo) frobbed widget #19."
```

Event#stop

stopped

true

Element#fire

stopped

firstDescendant 1.5.1

`firstDescendant(element) -> HTMLElement`

DOM	<code>firstChild</code>	<code>firstChild</code>	[
------------	-------------------------	-------------------------	----------

```
<div id="australopithecus">
  <div id="homo-erectus"><!--Latin is super -->
    <div id="homo-neanderthalensis"></div>
    <div id="homo-sapiens"></div>
  </div>
</div>
```

```
$('australopithecus').firstDescendant();
// -> div#homo-herectus
// DOM firstChild
$('homo-herectus').firstChild;
// -> "Latin is super"
//
$('homo-herectus').firstDescendant();
// -> div#homo-neanderthalensis
```

getDimensions

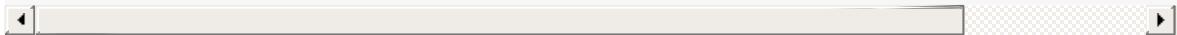
```
getDimensions(element) -> {height: Number, width: Number}
```

```
element  width  height
```

CSS

none 0

```
<div id="rectangle" style="font-size: 10px; width: 20em; height: 100px">
```



```
var dimensions = $('rectangle').getDimensions();  
// -> {width: 200, height: 100}  
dimensions.width;  
// -> 200  
dimensions.height;  
// -> 100
```

getElementsByClassName

```
getElementsByClassName(element, className) -> [HTMLElement...]
```

element className CSS

Prototype 1.6

document.getElementsByClassName

NodeList Array v1.6

\$\$ Element#select

0

element CSS

```
<ul id="fruits">
  <li id="apples">apples
    <ul>
      <li id="golden-delicious">Golden Delicious</li>
      <li id="mutsu" class="yummy">Mutsu</li>
      <li id="mcintosh" class="yummy">McIntosh</li>
      <li id="ida-red">Ida Red</li>
    </ul>
  </li>
  <li id="exotic" class="yummy">exotic fruits
    <ul>
      <li id="kiwi">kiwi</li>
      <li id="granadilla">granadilla</li>
    </ul>
  </li>
</ul>
```

```
$('#fruits').getElementsByClassName('yummy');
// -> [li#mutsu, li#mcintosh, li#exotic]
$('#exotic').getElementsByClassName('yummy');
// -> []
```


getElementsBySelector

```
getElementsBySelector(element, selector...) -> [HTMLElement...]
```

CSS

element

Prototype 1.6

Element#getElementsBySelector

Element#select

\$\$()

CSS

```
<ul id="fruits">
  <li id="apples">
    <h3 title="yummy!">Apples</h3>
    <ul id="list-of-apples">
      <li id="golden-delicious" title="yummy!">Golden Delicious</li>
      <li id="mutsu" title="yummy!">Mutsu</li>
      <li id="mcintosh">McIntosh</li>
      <li id="ida-red">Ida Red</li>
    </ul>
    <p id="saying">An apple a day keeps the doctor away</p>
  </li>
</ul>
```

```
$('#apples').getElementsBySelector('[title="yummy!"]');
// -> [h3, li#golden-delicious, li#mutsu]
$('#apples').getElementsBySelector('p#saying', 'li[title="yummy!"]')
// -> [li#golden-delicious, li#mutsu, p#saying]
$('#apples').getElementsBySelector('[title="disgusting!"]');
// -> []
```

Element#getElementsBySelector getElementsByTagName

```
var nodes = $(someUL).getElementsByTagName('li').map(Element.prototype);
var nodes2 = someUL.getElementsByTagName('li');
```

Array Prototype

Enumerable

DOM

Element#getElementsByTagName

1.5.1

getHeight

```
getHeight(element) -> Number
```

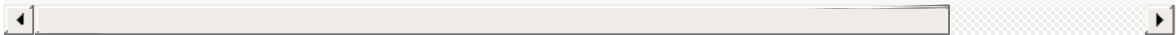
`element`

CSS

none 0

[Element.getHeight\(\)](#)

```
<div id="rectangle" style="font-size: 10px; width: 20em; height: 100px">
```



```
$('#rectangle').getHeight();  
// -> 100
```

getOffsetParent

1.6

getOffsetParent(element) -> HTMLElement

element	Containing Block	CSS	position	relative	absolute
body					

element CSS containing block

getStyle

```
getStyle(element, property) -> String | null
```

element	CSS	property	CSS	camelized
width	border-width	borderWidth		
DOM	element.style.xxx			

C

CSS CSS

1 left top right bottom width height

```
$(element).getStyle('font-size');  
//  
$(element).getStyle('fontSize');  
// -> '12px'
```

Internet Explorer CSS HTML

```
<style>  
    #test { font-size: 12px; margin-left: 1em; }  
</style>  
<div id="test"></div>
```

```
$('#test').getStyle('margin-left');  
// -> IE '1em'  
// -> '12px'
```

display 'none' Safari null

CSS

DOM 2

getWidth

```
getWidth(element) -> Number
```

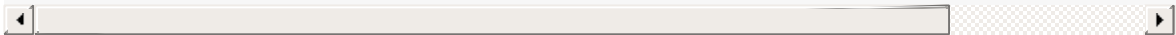
`element`

CSS

none 0

[Element.getWidth\(\)](#)

```
<div id="rectangle" style="font-size: 10px; width: 20em; height: 100px">
```



```
$('#rectangle').getWidth();  
// -> 200
```

hasClassName

```
hasClassName(element, className) -> Boolean
```

```
element  className  CSS
```

```
<div id="mutsu" class="apple fruit food"></div>
```

```
$('#mutsu').hasClassName('fruit');  
// -> true  
$('#mutsu').hasClassName('vegetable');  
// -> false
```

hide

```
hide(element) -> HTMLElement
```

`element`

```
<div id="error-message"></div>
```

```
$('error-message').hide();  
// -> HTMLElement ( div#error-message)
```

Prototype

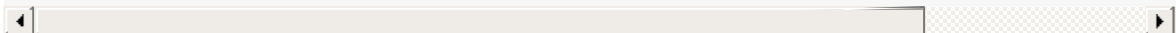
`Element.toggleEl`

`Element.hide` 1.5

Enumerables

```
['content', 'navigation', 'footer'].each(Element.hide);  
// -> ['content', 'navigation', 'footer']  
// #content, #navigation #footer
```

```
$('content', 'navigation', 'footer').invoke('hide');  
// -> [HTMLElement, HTMLElement, HTMLElement] (#content, #navigation, #footer)  
// #content, #navigation #footer
```



identify 1.6

```
identify(element) -> id
```

element	id	id	id	id
---------	----	----	----	----

HTML

```
<ul>  
  <li id="apple">apple</li>  
  <li>orange</li>  
</ul>
```

JavaScript

```
$('apple').identify();  
// -> 'apple'  
$('apple').next().identify();  
// -> 'anonymous_element_1'
```

HTML

```
<ul>  
  <li id="apple">apple</li>  
  <li id="anonymous_element_1">orange</li>  
</ul>
```

immediateDescendants deprecated

```
immediateDescendants(element) -> [HTMLElement...]
```

Prototype 1.6

[Element#immediateDescendants](#)

[Element#childElements](#)

0

Prototype DOM

```
<div id="australopithecus">
  <div id="homo-erectus">
    <div id="homo-neanderthalensis"></div>
    <div id="homo-sapiens"></div>
  </div>
</div>
```

```
$('australopithecus').immediateDescendants();
// -> [div#homo-erectus]
$('homo-erectus').immediateDescendants();
// -> [div#homo-neanderthalensis, div#homo-sapiens]
$('homo-sapiens').immediateDescendants();
// -> []
```

insert 1.6

```
insert(element, { position: content }) -> HTMLElement
insert(element, content) -> HTMLElement
```

	position	content		content	cont
	position	beforeaftertop	bottom		

HTMLDOM toHTML toElemen

HTML <script> Javascript HTML
String#evalScripts

inspect

```
inspect(element) -> String
```

element

inspect

Object.inspect

```
<ul>
  <li id="golden-delicious">Golden Delicious</li>
  <li id="mutsu" class="yummy apple">Mutsu</li>
  <li id="mcintosh" class="yummy">McIntosh</li>
</ul>
```

```
$('golden-delicious').inspect();
// -> '<li id="golden-delicious">'
$('mutsu').inspect();
// -> '<li id="mutsu" class="yummy apple">'
$('mutsu').next().inspect();
// -> '<li>'
```

makeClipping

```
makeClipping(element) -> HTMLElement
```

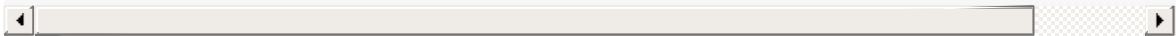
```
'hidden' CSS clip element
```

Element.undoClipping

element

```
<div id="framer">  
    
</div>
```

```
$('framer').makeClipping().setStyle({width: '100px', height: '100px'})  
// -> HTMLElement
```



makePositioned

```
makePositioned(element) -> HTMLElement
```

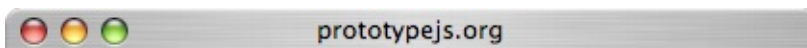
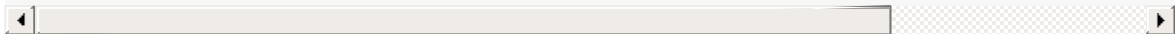
CSS block
'relative'

element CSS position 'static' undefin

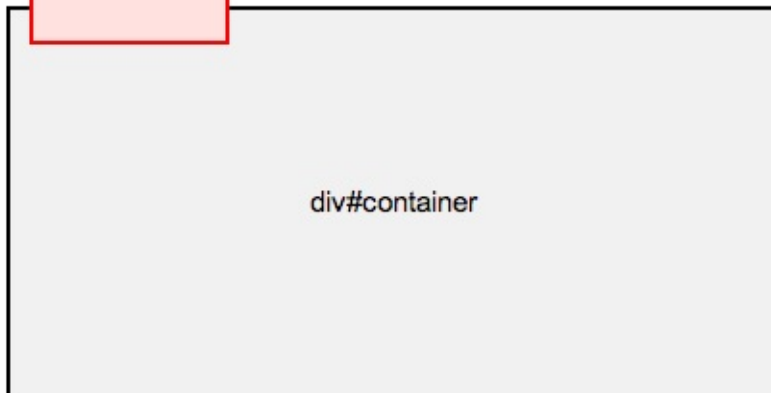
element CSS

Element.undoPositioned

```
<p>lorem [...]</p>  
<div id="container">  
  <div id="element" style="position: absolute; top: 20px; left: 20px;">  
</div>  
</div>
```

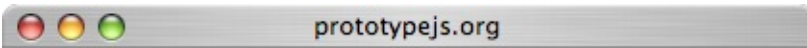


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
ei
er
ni
div#element
niam, quis nostrud exercitation ullamco laboris
a commodo consequat.

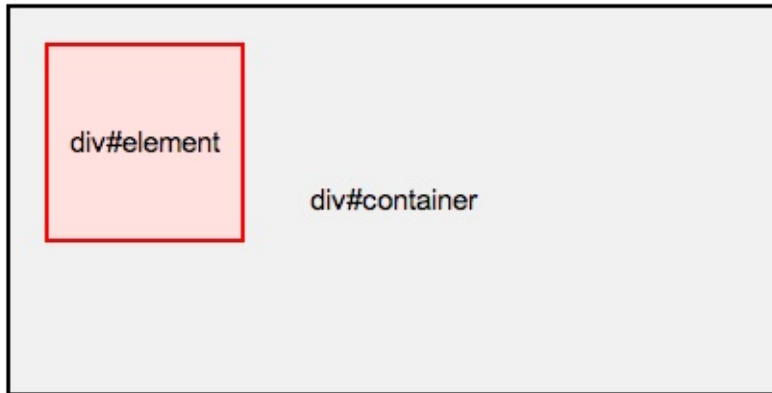


div#element

```
$('container').makePositioned();  
// -> HTMLElement
```



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



match

```
match(element, selector) -> Boolean
```

element **selector** **CSS**

```
<ul id="fruits">
  <li id="apples">
    <ul>
      <li id="golden-delicious">Golden Delicious</li>
      <li id="mutsu" class="yummy">Mutsu</li>
      <li id="mcintosh" class="yummy">McIntosh</li>
      <li id="ida-red">Ida Red</li>
    </ul>
  </li>
</ul>
```

```
$('#fruits').match('ul');
// -> true
$('#mcintosh').match('li#mcintosh.yummy');
// -> true
$('#fruits').match('p');
// -> false
```


next

```
next(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

cssRule	element	index	<i>index</i>	cssRule
undefined				

Element.next Prototype DOM

Element.up

Element.down Element.previous

Prototype / CSS

head boc

```
$(element).down(1).next('li', 2).hide();
```

DOM

	element	nextSibling	Element.next
index	element		Element.i
0			
	cssRule	Element.next	
	cssRule	index	Element.next CSS
	undefined		

```
<ul id="fruits">
  <li id="apples">
    <h3 id="title">Apples</h3>
```

```
        <ul id="list-of-apples">
            <li id="golden-delicious">Golden Delicious</li>
            <li id="mutsu">Mutsu</li>
            <li id="mcintosh" class="yummy">McIntosh</li>
            <li id="ida-red" class="yummy">Ida Red</li>
        </ul>
        <p id="saying">An apple a day keeps the doctor away</p>
    </li>
</ul>
```

```
$( 'list-of-apples' ).next();
//
$( 'list-of-apples' ).next(0);
// -> p#sayings
$( 'title' ).next(1);
// -> ul#list-of-apples
$( 'title' ).next('p');
// -> p#sayings
$( 'golden-delicious' ).next('.yummy');
// -> li#mcintosh
$( 'golden-delicious' ).next('.yummy', 1);
// -> li#ida-red
$( 'ida-red' ).next();
// -> undefined
```

nextSiblings

```
nextSiblings(element) -> [HTMLElement...]
```

`element`

`head` `body`

0 `element`

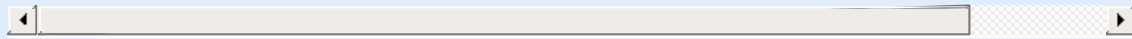
Prototype DOM

```
<ul>
  <li id="golden-delicious">Golden Delicious</li>
  <li id="mutsu">Mutsu</li>
  <li id="mcintosh">McIntosh</li>
  <li id="ida-red">Ida Red</li>
</ul>
```

```
$('mutsu').nextSiblings();
// -> [li#mcintosh, li#ida-red]
$('ida-red').nextSiblings();
// -> []
```

observe

```
observe(element, eventName, handler[, useCapture = false]) -> HTML
```



element

Event.observe

Event.observe

```
$(element).observe('click', function(event){  
    alert(Event.element(event).innerHTML);  
});  
// -> HTMLElement HTML
```

positionedOffset 1.6

positionedOffset(element) -> [Number, Number] { left: Number, top: Number }

element	Containing Block	CSS position	relative	absolute
element		Element#getOffsetParent		
offset X	offset[0]	offset.left	Y	offset[1]
offset.top				

offsetLeft offsetTop CSS

position

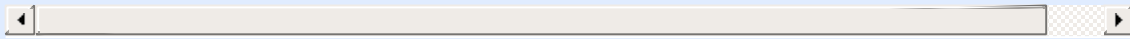
" Calculates the cumulative offsetLeft and offsetTop of an element and all its parents until it reaches an element with a position of static.

"

```
positionedOffset: function(element) {
    var valueT = 0, valueL = 0;
    do {
        valueT += element.offsetTop || 0;
        valueL += element.offsetLeft || 0;
        element = element.offsetParent;
        if (element) {
            if (element.tagName.toUpperCase() == 'BODY')
                var p = Element.getStyle(element, 'position');
            if (p != 'static') break;
        }
    } while (element);
    return Element._returnOffset(valueL, valueT);
}
```

previous

```
previous(element[, cssRule][, index = 0]) -> HTMLElement | undefi
```



cssRule

element

index

index

cssRule

Element.previous Prototype DOM

Element.up

Element.down Element.next

Prototype / CSS

head body

```
$(element).down(1).next('li', 2).hide();
```

DOM

element

previousSibling

Element.previous

index

element

Element.previous

element 0

cssRule

Element.previous

cssRule

index

Element.previous CSS

undefined

```
<ul id="fruits">
  <li id="apples">
    <h3>Apples</h3>
```

```
        <ul id="list-of-apples">
            <li id="golden-delicious" class="yummy">Gol
            <li id="mutsu" class="yummy">Mutsu</li>
            <li id="mcintosh">McIntosh</li>
            <li id="ida-red">Ida Red</li>
        </ul>
        <p id="saying">An apple a day keeps the doctor away
    </li>
</ul>
```

```
$('#saying').previous();
//
$('#saying').previous(0);
// -> ul#list-of-apples
$('#saying').previous(1);
// -> h3
$('#saying').previous('h3');
// -> h3
$('#ida-red').previous('.yummy');
// -> li#mutsu
$('#ida-red').previous('.yummy', 1);
// -> li#golden-delicious
$('#ida-red').previous(5);
// -> undefined
```

previousSiblings

```
previousSiblings(element) -> [HTMLElement...]
```

element

head body

0 element

Prototype DOM

```
<ul>
  <li id="golden-delicious">Golden Delicious</li>
  <li id="mutsu">Mutsu</li>
  <li id="mcintosh">McIntosh</li>
  <li id="ida-red">Ida Red</li>
</ul>
```

```
$('mcintosh').previousSiblings();
// -> [li#mutsu, li#golden-delicious]
$('golden-delicious').previousSiblings();
// -> []
```


readAttribute

```
readAttribute(element, attribute) -> String | null
```

elemnet	attribute	null
---------	-----------	------

getAttribute Safari Internet E

getAttribute "" .apply .call

Internet Explorer

```
<a id="tag" href="/tags/prototype" rel="tag" title="viewrelated boc
```

```
$('#tag').readAttribute('href');  
// -> '/tags/prototype'  
$('#tag').readAttribute('title');  
// -> 'view related bookmarks.'  
$('#tag').readAttribute('my_widget');  
// -> 'some info.'
```

recursivelyCollect

```
recursivelyCollect(element, property) -> [HTMLElement...]
```

element

property

property

element

DOM

Element.ancestors Element.descendants Element.nextSibling
Element.previousSiblings Element.siblings

Prototype DOM

```
<ul id="fruits">
  <li id="apples">
    <ul id="list-of-apples">
      <li id="golden-delicious"><p>Golden Delicious</p>
      <li id="mutsu">Mutsu</li>
      <li id="mcintosh">McIntosh</li>
      <li id="ida-red">Ida Red</li>
    </ul>
  </li>
</ul>
```

```
$('#fruits').recursivelyCollect('firstChild');
// -> [li#apples, ul#list-of-apples, li#golden-delicious, p]
```

relativize

1.6

```
relativize(element) -> HTMLElement
```

```
    element.style.position = 'relative'
```

1.6.0.3 2066

```
element._originalWidth  = element.style.width;
```

```
element._originalHeight = element.style.height;
```

remove

```
remove(element) -> HTMLElement
```

```
element
```

Element.hide

```
//  
<ul>  
  <li id="golden-delicious">Golden Delicious</li>  
  <li id="mutsu">Mutsu</li>  
  <li id="mcintosh">McIntosh</li>  
  <li id="ida-red">Ida Red</li>  
</ul>
```

```
$('mutsu').remove();  
// -> HTMLElement ( li#mutsu)
```

```
//  
<ul>  
  <li id="golden-delicious">Golden Delicious</li>  
  <li id="mcintosh">McIntosh</li>  
  <li id="ida-red">Ida Red</li>  
</ul>
```

removeClassName

```
removeClassName(element, className) -> HTMLElement
```

```
element  className  CSS
```

```
<div id="mutsu" class="apple fruit food"></div>
```

```
$('#mutsu').removeClassName('food');  
// -> HTMLElement  
$('#mutsu').classNames();  
// -> ['apple', 'fruit']
```

replace

```
replace(element[, html]) -> HTMLElement
```

```
html  element  element  
IE    element.outerHTML = html
```

Opera 9 input replace

```
$('foo').replace('<p>Bar</p>')
```

Web Forms 2

replace input

Ele

```
'<p>Bar</p>')
```

html HTML toString JavaScript

```
html    <script>      element    <script>
```

Element.replace() String#evalScripts

html Element.replace element

Element.r

```
<div id="food">  
  <div id="fruits">  
    <p id="first">Kiwi, banana <em>and</em> apple.</p>  
  </div>  
</div>
```

HTML

```
$('first').replace('<ul id="favorite"><li>kiwi</li><li>banana</li><li>apple</li></ul>','<ul id="favorite"><li>kiwi</li><li>banana</li><li>apple</li></ul>')  
// -> HTMLElement (p#first)  
$('fruits').innerHTML; // -> '<ul id="favorite"><li>kiwi</li><li>banana</li><li>apple</li></ul>'
```

<script>

```
$('#favorite').replace('<p id="still-first">Melon, oranges <em>and</em>  
// -> HTMLElement (ul#favorite) "removed!"  
$('#fruits').innerHTML  
// -> '<p id="still-first">Melon, oranges <em>and</em> grapes.</p>'
```



```
$('#still-first').replace('Melon, oranges and grapes.');
```

```
// -> HTMLElement (p#still-first)  
$('#fruits').innerHTML  
// -> 'Melon, oranges and grapes.'
```

toString()

```
$('#fruits').replace(123);  
// -> HTMLElement  
$('#food').innerHTML;  
// -> '123'
```

scrollTo

```
scrollTo(element) -> HTMLElement
```

```
elemnet
```

HTML anchors

```
$(element).scrollTo();  
// -> HTMLElement
```


select 1.6

```
select(element, selector...) -> [HTMLElement...]
```

CSS

element

\$\$()

CS

```
<ul id="fruits">
  <li id="apples">
    <h3 title="yummy!">Apples</h3>
    <ul id="list-of-apples">
      <li id="golden-delicious" title="yummy!" >Golden Delicious</li>
      <li id="mutsu" title="yummy!">Mutsu</li>
      <li id="mcintosh">McIntosh</li>
      <li id="ida-red">Ida Red</li>
    </ul>
    <p id="saying">An apple a day keeps the doctor away</p>
  </li>
</ul>
```

```
$('apples').select('[title="yummy!"]');
// -> [h3, li#golden-delicious, li#mutsu]
$('apples').select('p#saying', 'li[title="yummy!"]');
// -> [li#golden-delicious, li#mutsu, p#saying]
$('apples').select('[title="disgusting!"]');
// -> []
```

Element#select getElementsByTagName

```
var nodes = $(someUL.getElementsByTagName('li')).map(Element.prototype.select);
var nodes2 = someUL.select('li');
```

Array Prototype

Enumerable

DOM Element#select

1.6

setOpacity 1.5.1

```
Element.setOpacity(element, opacity) -> [HTMLElement...]  
someElement.setOpacity(opacity) -> [HTMLElement...]
```

| | | | |
|----------------------|----------------------|------------------|------------------|
| <code>element</code> | <code>opacity</code> | <code>0 1</code> | <code>0 1</code> |
|----------------------|----------------------|------------------|------------------|

`Element.setStyle` `setOpacity`

```
var element = $('myelement');  
// 50%  
element.setOpacity(0.5);  
// CSS  
element.setStyle({ opacity: 0.5 });  
element.setStyle("opacity: 0.5");
```

setStyle

```
setStyle(element, styles) -> HTMLElement
```

| element | CSS | styles / Hash | camelized |
|---------|--------------|---------------|--------------|
| | border-width | borderWidth | border-width |

```
$(element).setStyle({ backgroundColor: '#900', fontSize: '12px' });  
// -> HTMLElement
```

| | float | opacity | float |
|---------|-------|---------|-------|
| opacity | 0 | | 1 |

```
$(element).setStyle({ cssFloat: 'left', opacity: 0.5 });  
// -> HTMLElement  
$(element).setStyle({  
  'float': 'left', // float  
  opacity: 0.5  
});  
// -> HTMLElement
```

CSS

DOM 2

```
CSS  
null $(element).setStyle({ backgroundColor:  
null }) IE  
Firefox  
backgroundColor: '' }) IE Firefox
```

show

```
show(element) -> HTMLElement
```

```
element
```

```
<div id="error-message" style="display:none;"></div>
```

```
$('#error-message').show();  
// -> HTMLElement ( div#error-message )
```

Element.show CSS Prototype CSS

<style> CSS

show

CSS

display

```
<style>  
    #hidden-by-css { display: none; }  
</style>
```

```
[...]
```

```
<div id="hidden-by-css"></div>
```

```
$('#hidden-by-css').show(); //  
// -> HTMLElementdiv#error-message )
```

Prototype

Element.toggleEl

Element.hide 1.5

Enumerables

```
['content', 'navigation', 'footer'].each(Element.show);  
// -> ['content', 'navigation', 'footer']  
// #content, #navigation #footer
```

```
$('content', 'navigation', 'footer').invoke('show');  
// -> [HTMLElement, HTMLElement, HTMLElement] (#content, #navigation, #footer)  
// #content, #navigation #footer
```



siblings

```
siblings(element) -> [HTMLElement...]
```

element

head body

0

element

Prototype DOM

Examples

```
<ul>
  <li id="golden-delicious">Golden Delicious</li>
  <li id="mutsu">Mutsu</li>
  <li id="mcintosh">McIntosh</li>
  <li id="ida-red">Ida Red</li>
</ul>
```

```
$('mutsu').siblings();
// -> [li#golden-delicious, li#mcintosh, li#ida-red]
```

stopObserving

```
stopObserving(element, eventName, handler) -> HTMLElement
```

`element`

Event.stopObserving

Event.stopObservin

```
$(element).stopObserving('click', coolAction);  
// -> HTMLElement 'coolAction'
```

toggle

```
toggle(element) -> HTMLElement
```

element

```
<div id="welcome-message"></div>
<div id="error-message" style="display:none;"></div>
```

```
$('welcome-message').toggle();
// -> HTMLElement div#welcome-message
$('error-message').toggle();
// -> HTMLElement div#error-message
```

Element.toggle CSS Prototype CSS

<style> CSS toggle CSS display

Javascript

```
<style>
  #hidden-by-css { display: none; }
</style>

[...]
```

```
<div id="hidden-by-css"></div>
```

```
$('hidden-by-css').toggle(); //
// -> HTMLElementdiv#hidden-by-css
```

Prototype

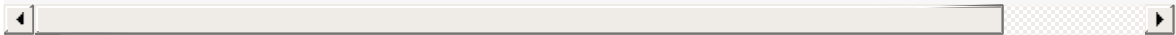
Element.hide 1.5

Element.toggleEl

Enumerables

```
['error-message', 'welcome-message'].each(Element.toggle);  
// -> ['error-message', 'welcome-message']  
//   div#error-message  div#confirmation-message
```

```
$('error-message', 'welcome-message').invoke('toggle');  
// -> [HTMLElement, HTMLElement]div#error-message  div#welcome-mess  
//   div#error-message  div#confirmation-message
```



toggleClassName

```
toggleClassName(element, className) -> HTMLElement
```

```
element CSS className
```

```
<div id="mutsu" class="apple"></div>
```

```
$('#mutsu').hasClassName('fruit');  
// -> false  
$('#mutsu').toggleClassName('fruit');  
// -> element  
$('#mutsu').hasClassName('fruit');  
// -> true
```

undoClipping

```
undoClipping(element) -> HTMLElement
```

```
element CSS overflow Element.makeClipping()
```

```
<div id="framer">  
    
</div>
```

```
$('framer').undoClipping();  
// -> HTMLElement CSS overflow
```



undoPositioned

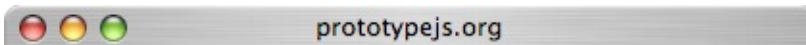
```
undoPositioned(element) -> HTMLElement
```

```
element    Element.makePositioned
```

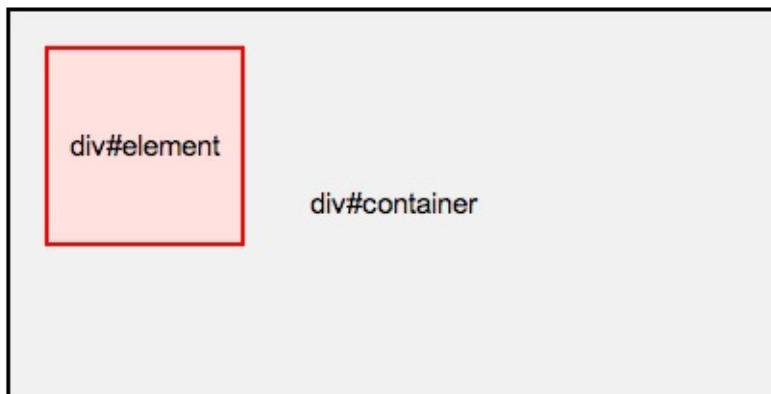
```
element            element CSS            position  
'absolute' 'relative'    'fixed'
```

```
<p>lorem [...]</p>  
<div id="container">  
  <div id="element" style="position: absolute; top: 20px; left: 20px;">  
</div>  
</div>
```

```
$('container').makePositioned();  
// -> HTMLElement
```

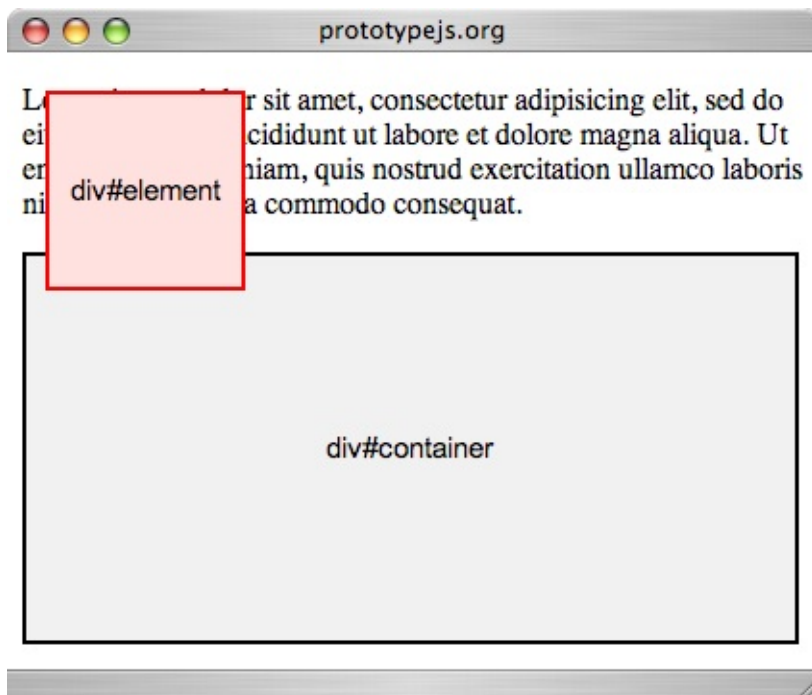


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



```
Element.undoPositioned
```

```
$('#container').undoPositioned();  
// -> HTMLElement
```



up

```
up(element, [cssRule][, index = 0]) -> HTMLElement | undefined
```

element

cssRule

index

index

cssRule

Element.up Prototype DOM

Element.down

Element.next Element.previous

Prototype / CSS

```
$(element).up(1).next('li', 2).hide();
```

DOM

element

parentNode

Element.up

index

element

Element.anc

cssRule

Element.up

cssRule

index

Element.up CSS

undefined

```
<html>
  [...]
  <body>
    <ul id="fruits">
      <li id="apples" class="keeps-the-doctor-awa
        <ul>
          <li id="golden-delicious">G
          <li id="mutsu" class="yummy
```

```

<li id="mcintosh" class="yu
<li id="ida-red">Ida Red</l
</ul>
</li>
</ul>
</body>
</html>
```

```

$('fruits').up();
//
$('fruits').up(0);
// -> body
$('mutsu').up(2);
// -> ul#fruits
$('mutsu').up('li');
// -> li#apples
$('mutsu').up('.keeps-the-doctor-away');
// -> li#apples
$('mutsu').up('ul', 1);
// -> ul#fruits
$('mutsu').up('div');
// -> undefined
```

update

```
update(element[, newContent]) -> HTMLElement
```

```
element  newContent  
element.innerHTML = newContent
```

newContent HTML toString() JavaScript

newContent <script> Javascript

String#evalScripts

Element.update element

IE6

```
<div id="fruits">carrot, eggplant and cucumber</div>
```

```
$('fruits').update('kiwi, banana and apple');  
// -> HTMLElement  
$('fruits').innerHTML  
// -> 'kiwi, banana and apple'
```

```
$('fruits').update();  
// -> HTMLElement  
$('fruits').innerHTML;  
// -> '' ()
```

HTML

```
$('fruits').update('<p>Kiwi, banana <em>and</em> apple.</p>');  
// -> HTMLElement  
$('fruits').innerHTML;
```



```
// -> '<p>Kiwi, banana <em>and</em> apple.</p>'
```

<script> HTML

```
$('#fruits').update('<p>Kiwi, banana <em>and</em> apple.</p><script>  
// -> HTMLElement"updated"  
$('#fruits').innerHTML; // -> '<p>Kiwi, banana <em>and</em> apple.</p>'
```

toString()

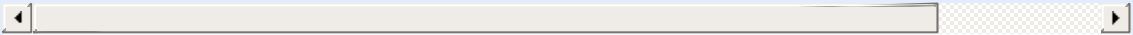
```
$('#fruits').update(123);  
// -> HTMLElement  
$('#fruits').innerHTML;  
// -> '123'
```

toString

```
var Fruit = Class.create();  
Fruit.prototype = {  
  initialize: function(fruit){ this.fruit = fruit; },  
  toString: function(){ return 'I am a fruit and my name is "'  
}  
var apple = new Fruit('apple');  
$('#fruits').update(apple);  
$('#fruits').innerHTML;  
// -> 'I am a fruit and my name is "apple".'
```

viewportOffset 1.6

viewportOffset(element) -> [Number, Number] { left: Number, top:



| | | | |
|-------------------------|------------------------|----------------------------|------------------------|
| <code>element</code> | | | |
| <code>offset X</code> | <code>offset[0]</code> | <code>offset.left Y</code> | <code>offset[1]</code> |
| <code>offset.top</code> | | | |

visible

```
visible(element) -> Boolean
```

| Boolean | element | style | "display:none;" |
|---------|---------|-------|-----------------|
|---------|---------|-------|-----------------|

```
<div id="visible"></div>  
<div id="hidden" style="display:none;"></div>
```

```
$('visible').visible();  
// -> true  
$('hidden').visible();  
// -> false
```

CSS Prototype CSS

<style> CSS **visible** Javascript CSS

```
<style>  
    #hidden-by-css { display: none; }  
</style>  
  
[...]  
  
<div id="hidden-by-css"></div>
```

```
$('hidden-by-css').visible();  
// -> true
```

wrap 1.6

```
Element.wrap(element, wrapper[, attributes]) -> HTMLElement
someElement.wrap(wrapper[, attributes]) -> HTMLElement
```

element wrapper wrapper

Internet Explorer textarea wrap

\$('foo').wrap('p') IE HTML wrap

textarea Element.wrap('foo', 'p')

Element#wrap —

wrapper HTMLElement HTML

Prototype Element#writeAttribute / wrapper

HTML

```
<table id="data">
  <tr>
    <th>Foo</th>
    <th>Bar</th>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
  </tr>
</table>
```

JavaScript

```
// 1
var div = new Element('div', { 'class': 'table-wrapper' });
$( 'data' ).wrap( div );
// 2
$( 'data' ).wrap( 'div', { 'class': 'table-wrapper' } );
// DIV
```

HTML

```
<div class="table-wrapper">
  <table id="data">
    <tr>
      <th>Foo</th>
      <th>Bar</th>
    </tr>
    <tr>
      <td>1</td>
      <td>2</td>
    </tr>
  </table>
</div>
```

writeAttribute 1.6

```
writeAttribute(element, attribute[, value = true]) -> HTMLElement  
writeAttribute(element, attributes) -> HTMLElement
```



`element` `hash /`

Element.Methods

[absolutize](#), [addClassName](#), [adjacent](#), [ancestors](#), [childElements](#), [classNames](#), [cleanWhitespace](#), [clonePosition](#), [cumulativeOffset](#), [cumulativeScrollOffset](#), [descendantOf](#), [descendants](#), [down](#), [empty](#), [fire](#), [firstDescendant](#), [getDimensions](#), [getElementsByClassName](#), [getElementsBySelector](#), [getHeight](#), [getOffsetParent](#), [getStyle](#), [getWidth](#), [hasClassName](#), [hide](#), [identify](#), [immediateDescendants](#), [insert](#), [inspect](#), [makeClipping](#), [makePositioned](#), [match](#), [next](#), [nextSiblings](#), [observe](#), [positionedOffset](#), [previous](#), [previousSiblings](#), [readAttribute](#), [recursivelyCollect](#), [relativize](#), [remove](#), [removeClassName](#), [replace](#), [scrollTo](#), [select](#), [setOpacity](#), [setStyle](#), [show](#), [siblings](#), [stopObserving](#), [toggle](#), [toggleClassName](#), [undoClipping](#), [undoPositioned](#), [up](#), [update](#), [viewportOffset](#), [visible](#), [wrap](#), [writeAttribute](#)

Element.Methods DOM

```
//  
$(element).hide();  
//      Enumerable ID "article"  
$('articles').descendants();
```

absolutize 1.6

```
absolutize(element) -> HTMLElement
```

```
element.style.position = 'absolute'
```

addClassName

```
addClassName(element, className) -> HTMLElement
```

CSS `element`

adjacent 1.6

```
Element.adjacent(element[, selectors...]) -> [HTMLElement...]
someElement.adjacent([selectors...]) -> [HTMLElement...]
```

selectors

ancestors

```
ancestors(element) -> [HTMLElement...]
```

element ...

childElements 1.5.1

```
childElements(element) -> [HTMLElement...]
```

classNames []

```
classNames(element) -> Enumerable
```

classNames Enumerable CSS

cleanWhitespace

```
cleanWhitespace(element) -> HTMLElement
```

XML HTML

```
<div>
  <p></p>
</div>
```

<div> element.childNodes <p> ,
Prototype cleanWhitespace

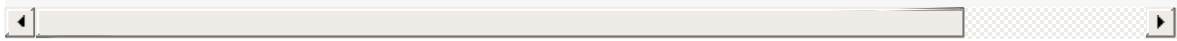
clonePosition 1.6

```
clonePosition(element, source[, options]) -> HTMLElement
```

source options / element

cumulativeOffset 1.6

```
cumulativeOffset(element) -> [Number, Number] { left: Number, top:
```



element
offset X offset[0] offset.left Y offset[1] offset.top

cumulativeScrollOffset 1.6

```
cumulativeScrollOffset(element) -> [Number, Number] { left: Number
```



offset X offset[0] offset.left Y offset[1] offset.top

descendantOf

```
descendantOf(element, ancestor) -> Boolean
```

element ancestor

descendants

```
descendants(element) -> [HTMLElement...]
```

element

down

```
down(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

element cssRule index *index* cssRule

empty

```
empty(element) -> Boolean
```

element

fire 1.6.0

```
fire(eventName[, memo]) -> Event
```

firstDescendant 1.5.1

```
firstDescendant(element) -> HTMLElement
```

DOM firstChild firstChild [

getDimensions

```
getDimensions(element) -> {height: Number, width: Number}
```

element width height

getElementsByClassName []

```
getElementsByClassName(element, className) -> [HTMLElement...]
```

element className CSS

getElementsBySelector []

```
getElementsBySelector(element, selector...) -> [HTMLElement...]
```

CSS element

getHeight

```
getHeight(element) -> Number
```

element

getOffsetParent 1.6

```
getOffsetParent(element) -> HTMLElement
```

element Containing Block CSS position relative absolute
body

getStyle

```
getStyle(element, property) -> String | null
```

element CSS property CSS camelized CSS

border-width borderWidth

getWidth

```
getWidth(element) -> Number
```

element

hasClassName

```
hasClassName(element, className) -> Boolean
```

element className CSS

hide

```
hide(element) -> HTMLElement
```

element

identify 1.6

```
identify(element) -> id
```

element id id id id

immediateDescendants deprecated

```
immediateDescendants(element) -> [HTMLElement...]
```

insert 1.6

```
insert(element, { position: content }) -> HTMLElement  
insert(element, content) -> HTMLElement
```

position content content content
position beforeaftertop bottom

inspect

```
inspect(element) -> String
```

element

makeClipping

```
makeClipping(element) -> HTMLElement
```

'hidden' CSS clip element

makePositioned

```
makePositioned(element) -> HTMLElement
```

'relative' CSS block element CSS position 'static' undefined posi

match

```
match(element, selector) -> Boolean
```

element selector CSS

next

```
next(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```

cssRule element index index cssRule

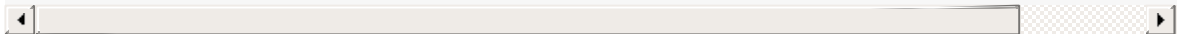
nextSiblings

```
nextSiblings(element) -> [HTMLElement...]
```

element

observe

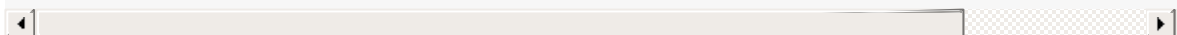
```
observe(element, eventName, handler[, useCapture = false]) -> HTML
```



element

positionedOffset 1.6

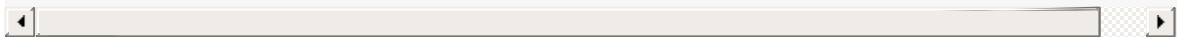
```
positionedOffset(element) -> [Number, Number] { left: Number, top:
```



element Containing Block CSS position relative absolute
 element Element#getOffsetParent
 offset X offset[0] offset.left Y offset[1] offset.top

previous

```
previous(element[, cssRule][, index = 0]) -> HTMLElement | undefined
```



cssRule element index index cssRule

previousSiblings

```
previousSiblings(element) -> [HTMLElement...]
```

element

readAttribute

```
readAttribute(element, attribute) -> String | null
```

element attribute null

recursivelyCollect

```
recursivelyCollect(element, property) -> [HTMLElement...]
```

element property property element DOM

relativize 1.6

```
relativize(element) -> HTMLElement
```

```
element.style.position = 'relative'
```

1.6.0.3 2066

```
element._originalWidth = element.style.width;
element._originalHeight = element.style.height;
```

remove

```
remove(element) -> HTMLElement
```

element

removeClassName

```
removeClassName(element, className) -> HTMLElement
```

element className CSS

replace

```
replace(element[, html]) -> HTMLElement
```

html element element
IE element.outerHTML = html

scrollTo

```
scrollTo(element) -> HTMLElement
```

elemnet

select 1.6

```
select(element, selector...) -> [HTMLElement...]
```

CSS

element

setOpacity 1.5.1

```
Element.setOpacity(element, opacity) -> [HTMLElement...]  
someElement.setOpacity(opacity) -> [HTMLElement...]
```

element opacity 0 1 0 1

setStyle

```
setStyle(element, styles) -> HTMLElement
```

element CSS styles / Hash camelized
width borderWidth border-width

show

```
show(element) -> HTMLElement
```

`element`

siblings

```
siblings(element) -> [HTMLElement...]
```

`element`

stopObserving

```
stopObserving(element, eventName, handler) -> HTMLElement
```

`element`

toggle

```
toggle(element) -> HTMLElement
```

`element`

toggleClassName

```
toggleClassName(element, className) -> HTMLElement
```

`element` `CSS` `className`

undoClipping

```
undoClipping(element) -> HTMLElement
```

`element` `CSS` `overflow` `Element.makeClipping()`

undoPositioned

```
undoPositioned(element) -> HTMLElement
```

`element` `Element.makePositioned`

up

```
up(element, [cssRule][, index = 0]) -> HTMLElement | undefined
```

element cssRule index *index* cssRule

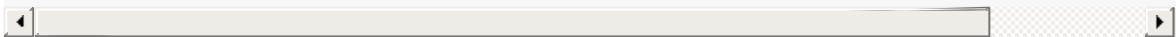
update

```
update(element[, newContent]) -> HTMLElement
```

```
element    newContent  
element.innerHTML = newContent
```

viewportOffset 1.6

```
viewportOffset(element) -> [Number, Number] { left: Number, top: N
```



```
element  
offset X          offset[0]          offset.left Y          offset[1]          offset.top
```

visible

```
visible(element) -> Boolean
```

```
Boolean    element    style    "display:none;"
```

wrap 1.6

```
Element.wrap(element, wrapper[, attributes]) -> HTMLElement  
someElement.wrap(wrapper[, attributes]) -> HTMLElement
```

```
element    wrapper    wrapper
```

writeAttribute 1.6

```
writeAttribute(element, attribute[, value = true]) -> HTMLElement  
writeAttribute(element, attributes) -> HTMLElement
```

```
element hash /
```


Element.Methods.Simulated

hasAttribute

Element.Methods.Simulated HTMLElement

hasAttribute simulated

hasAttribute(element, attribute) -> Boolean

Internet Explorer 6 7 DOM

hasAttribute

hasAttribute simulated

```
hasAttribute(element, attribute) -> Boolean
```

Internet Explorer 6 7 DOM

hasAttribute

```
<a id="link" href="http://prototypejs.org">Prototype</a>
```

```
$('link').hasAttribute('href');  
// -> true
```

Enumerable

all, any, collect, detect, each, eachSlice, entries, find, findAll, grep, inGroupsOf,
include, inject, invoke, map, max, member, min, partition, pluck, reject, select,
size, sortBy, toArray, zip

Enumerable Prototype

Enumerable module

Enumerable Ruby Enumerable

Prototype

EnumerableArray Hash

DOM Ajax

context

Enumerable iterator iterator context
iterator iterator this context

```
var myObject = {};  
['foo', 'bar', 'baz'].each(function(name, index) {  
    this[name] = index;  
}, myObject); // contextthis myObject  
myObject  
//-> { foo: 0, bar: 1, baz: 2}
```

context

Ruby

Enumerable

Enumerable

- map collect
- find detect

- `findAll` `select`
- `include` `member`
- `entries` `toArray`

Enumerable API

`collect` `invoke` `pluck` `each`

`Enumerable` `each` `Enumerable` `collect`

- `invoke`
- `pluck`

`reject` `findAll` VS. `partition`

`findAll`/`select` `reject`

Enumerable

`Enumerable` `_each` `“”`

Prototype

```
_each: function(iterator) {
  for (var i = 0, length = this.length; i < length; i++)
    iterator(this[i]); //
}
```

`each` `Enumerable`

Prototype

`ArrayHash` `ObjectRange`

`Enumerable`

```
var YourObject = Class.create();
```

```
Object.extend(YourObject.prototype, Enumerable);
Object.extend(YourObject.prototype, {
  initialize: function() {
    //
  },
  _each: function(iterator) {
    // iterator
  },
  // Enumerable
});
```

```
var obj = new YourObject();
[...]  
obj.pluck('somePropName');  
obj.invoke('someMethodName');  
obj.size();
```

all

```
all([iterator = Prototype.K[, context]]) -> Boolean
```

```
Enumerable  true  true          false  iterator                                bool
```

any

```
any([iterator = Prototype.K[, context]]) -> Boolean
```

```
Enumerable  true  true          false  iterator                                bool
```

collect

```
collect(iterator[, context]) -> Array
```

```
iterator  Enumerable                                map iterator
```

detect

```
detect(iterator[, context]) -> firstElement | undefined
```

```
iterator  true                                find  iterator
```

each

```
each(iterator[, context]) -> Enumerable
```

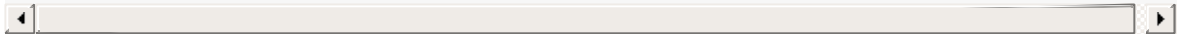
Enumerable

Enumerable

iterator

eachSlice

```
eachSlice(size[, iterator = Prototype.K[, context]]) -> [slice...]
```



Enumerable

entries

```
entries() -> Array
```

toArray

find

```
find(iterator) -> firstElement | undefined
```

iterator

true

detect

find find

iterator

findAll

```
findAll(iterator[, context]) -> Array
```

iterator

true

select

grep

```
grep(regex[, iterator = Prototype.K[, context]]) -> Array
```

iterator

inGroupsOf

```
inGroupsOf(size[, filler = null]) -> [group...]
```

filler

include

```
include(object) -> Boolean
```

Enumerable == member

inject

```
inject(accumulator, iterator[, context]) -> accumulatedValue
```

iterator accumulator iterator accumulator
accumulator
iterator Prototype accumulator

invoke

```
invoke(methodName[, arg...]) -> Array
```

each collect Enumerable

map

```
map(iterator) -> Array
```

iterator Enumerable collect

max

```
max([iterator = Prototype.K[, context]]) -> maxValue
```

Enumerable iterator iterator Enumerable

member

```
member(object) -> Boolean
```

Enumerable == include

min

```
min([iterator = Prototype.K[, context]]) -> minValue
```

Enumerable iterator iterator Enumerable

partition

```
partition([iterator = Prototype.K[, context]]) -> [TrueArray, FalseArray]
```





true false Javascript

bool

ite

pluck

```
pluck(propertyName) -> Array
```

collect

reject

```
reject(iterator[, context]) -> Array
```

iterator false

select

```
select(iterator) -> Array
```

iterator true findAll

size

```
size() -> Number
```

Enumerable

sortBy

```
sortBy(iterator[, context]) -> Array
```

iterator Enumerable

toArray

```
toArray() -> Array
```

Enumerable Array entries

zip

```
zip(Sequence...[, iterator = Prototype.K]) -> Array
```


all

```
all([iterator = Prototype.K[, context]]) -> Boolean
```

```
Enumerable  true  true          false  iterator          bool
```

Prototype.K Prototype

```
bool                                false                                iterator bool
```

bool

```
context  iterator      iterator  this  context
```

```

[[]].all()
// -> true (           false)

$R(1, 5).all()
// -> true ( [1..5]      true)

[0, 1, 2].all()
// -> false (0           false)

[9, 10, 15].all(function(n) { return n >= 10; })
// -> false ( 9iterator   false)

$( { name: 'John', age: 29, oops: false } ).all(function(pair) { return pair.value; })
// -> false ( oops/false  false)

```

Enumerable any

```
any([iterator = Prototype.K[, context]]) -> Boolean
```

Enumerable	true	true	false	iterator	boolean

```
bool          true          iterator bool
bool
```

```

[ ].any()
// -> false (true)

$(0, 2).any()
// -> true (2 true)

[2, 4, 6, 8, 10].any(function(n) { return 0 == n % 3; })
// -> true ( 6 iterator true 3 )

$H({ opt1: null, opt2: false, opt3: '', opt4: 'pfew!' }).any(function(pair) {
    return pair.value;
})
// -> true ( opt4/'pfew!' true)

```

Enumerable all

collect

```
collect(iterator[, context]) -> Array
```

iterator Enumerable

map iterator

context iterator iterator this context

```
['Hitch', "Hiker's", 'Guide', 'To', 'The', 'Galaxy'].collect(function(s) {
  return s.charAt(0).toUpperCase();
}).join('')
// -> 'HHGTTG'
$R(1,5).collect(function(n) { return n * n; })
// -> [1, 4, 9, 16, 25]
```

pluck

collect

detect

```
detect(iterator[, context]) -> firstElement | undefined
```

iterator

true

find

iterator

context

iterator

iterator

this

context

each

```
each(iterator[, context]) -> Enumerable
```

Enumerable

Enumerable

iter

each **Enumerable** iterator

1.

2. 0

context iterator iterator this context

\$break **\$continue**

\$continue Prototype 1.5

iterator

return

JavaScript

break continue

ite

iterator iterator

Prototype

\$break

each

```
['one', 'two', 'three'].each(function(s) { alert(s); });
['hello', 'world'].each(function(s, index) { alert(index + ': ' +
// '0: hello' '1: world'
// inject
// ...
var result = [];
$(1,10).each(function(n) {
    if(0 == n % 2)
        throw $continue;
    if (n > 6)
        throw $break;
    result.push(n);
});
```

```
});  
// result -> [1, 3, 5]  
// inject  
$R(1,10).inject([], function(result, n){  
  if(n < 6 && 0 != n % 2)  
    result.push(n);  
  return result;  
})
```

each VS. _each

Enumerable

Enumerable

Enumerable.each _each

1. break/continue
2. value/index

invoke each

eachSlice

```
eachSlice(size[, iterator = Prototype.K[, context]]) -> [slice...
```

Enumerable

context iterator iterator this context

```
var students = [
  { name: 'Sunny', age: 20 }, { name: 'Audrey', age: 21 },
  { name: 'Matt', age: 20 }, { name: 'Élodie', age: 26 },
  { name: 'Will', age: 21 }, { name: 'David', age: 23 },
  { name: 'Julien', age: 22 }, { name: 'Thomas', age: 21 },
  { name: 'Serpil', age: 22 }
];
students.eachSlice(4, function(toon) {
  return toon.pluck('name');
})
// -> [ ['Sunny', 'Audrey', 'Matt', 'Élodie'],
//      ['Will', 'David', 'Julien', 'Thomas'],
//      ['Serpil'] ]

students.eachSlice(2).first()
// -> [{ name: 'Sunny', age: 20 }, { name: 'Audrey', age: 21 }]
```

eachSlice

entries

```
entries() -> Array
```

```
  toArray
```

find

```
find(iterator) -> firstElement | undefined
```

iterator

true

detect

find

find

iterator

findAll

```
//  
function isPrime(n) {  
  if (2 > n)  
    return false;  
  if (0 == n % 2)  
    return (2 == n);  
  for (var index = 3; n / index > index; index += 2)  
    if (0 == n % index)  
      return false;  
  return true;  
}  
// isPrime  
$R(10,15).find(isPrime)  
// -> 11  
[ 'hello', 'world', 'this', 'is', 'nice'].find(function(s) { return  
// -> 'is'
```

findAll

```
findAll(iterator[, context]) -> Array
```

iterator true select

grepgrep findAll

context iterator iterator this context

```
$R(1, 10).findAll(function(n) { return 0 == n % 2; })  
// -> [2, 4, 6, 8, 10]  
[ 'hello', 'world', 'this', 'is', 'nice'].findAll(function(s) { ret  
// -> ['hello', 'world']
```

findAll reject

partition

grep

```
grep(filter[, iterator = Prototype.K[, context]]) -> Array
```

iterator

match Enumerable

filter match

filter.match true

iterator **map**

context iterator iterator this context

Prototype 1.6 grep filter

```
//  
['hello', 'world', 'this', 'is', 'cool'].grep(/(.)\1/)  
// -> ['hello', 'cool']  
// 0 5  
$R(1,30).grep(/[05]$/)  
// -> [5, 10, 15, 20, 25, 30]  
// 0 5 1  
$R(1,30).grep(/[05]$/, function(n) { return n - 1; })  
// -> [4, 9, 14, 19, 24, 29]  
// CSS  
// (Selector "match" )  
$('foo').childElements().grep(new Selector("li.active"));
```

inGroupsOf

```
inGroupsOf(size[, filler = null]) -> [group...]
```

filler

```
var students = [  
  { name: 'Sunny', age: 20 }, { name: 'Audrey', age: 21 },  
  { name: 'Matt', age: 20 }, { name: 'Élodie', age: 26 },  
  { name: 'Will', age: 21 }, { name: 'David', age: 23 },  
  { name: 'Julien', age: 22 }, { name: 'Thomas', age: 21 },  
  { name: 'Serpil', age: 22 }  
];  
  
students.pluck('name').inGroupsOf(4) {  
  // -> [ ['Sunny', 'Audrey', 'Matt', 'Élodie'],  
  // ['Will', 'David', 'Julien', 'Thomas'],  
  // ['Serpil', null, null, null] ]  
}
```

eachSlice

include

```
include(object) -> Boolean
```

```
Enumerable == member
```

```
===
```

```
Enumerable any
```

```
$R(1,15).include(10)
// -> true
['hello', 'world'].include('HELLO')
// -> false
[1, 2, '3', '4', '5'].include(3)
// -> true (== )
```

inject

```
inject(accumulator, iterator[, context]) -> accumulatedValue
```

accumulator	iterator	accumulator	iterator	accumulator
	Prototype			

context iterator iterator this context

```
$R(1,10).inject(0, function(acc, n) {
    return acc + n;
})
// -> 55 (1 10 )

$R(2,5).inject(1, function(acc, n) {
    return acc * n;
})
// -> 120 (5 )

['hello', 'world', 'this', 'is', 'nice'].inject(
    [],
    function(array, value, index) {
        if (0 == index % 2)
            array.push(value);
        return array;
    }
)
// -> ['hello', 'this', 'nice']

//
var array1 = [];
var array2 = [1, 2, 3].inject(array1, function(array, value) {
    array.push(value * value);
    return array;
});
array2 // -> [1, 4, 9]
array1 // -> [1, 4, 9]
array2.push(16);
array1 // -> [1, 4, 9, 16]
```

[inject](#) JavaScript

invoke

`invoke(methodName[, arg...]) -> Array`

`each` `collect`

Enumerable

`each` `collect`

```
['hello', 'world', 'cool!'].invoke('toUpperCase')
// ['HELLO', 'WORLD', 'COOL!']
['hello', 'world', 'cool!'].invoke('substring', 0, 3)
// ['hel', 'wor', 'coo']
// Prototype
$('navBar', 'adsBar', 'footer').invoke('hide')
// Prototype
// "invoke"
$$('#windows div.close').invoke('addClassName', 'active').invoke('s
```

`pluck`

map

```
map(iterator) -> Array
```

```
  iterator  Enumerable
```

```
collect
```

max

```
max([iterator = Prototype.K[, context]]) -> max_value
```

Enumerable

iterator iterator

Enumerable

context iterator iterator this context

```
$R(1,10).max()  
// -> 10  
['hello', 'world', 'gizmo'].max()  
// -> 'world'  
  
function Person(name, age) {  
    this.name = name;  
    this.age = age;  
}  
var john = new Person('John', 20);  
var mark = new Person('Mark', 35);  
var daisy = new Person('Daisy', 22);  
[john, mark, daisy].max(function(person) {  
    return person.age;  
})  
// -> 35
```

member

```
member(object) -> Boolean
```

`Enumerable`

`==`

`include`

min

```
min([iterator = Prototype.K[, context]]) -> minValue
```

Enumerable

iterator iterator

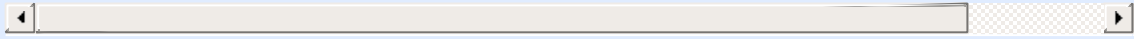
Enumerable

context iterator iterator this context

```
$R(1,10).min()  
// -> 1  
['hello', 'world', 'gizmo'].min()  
// -> 'gizmo'  
  
function Person(name, age) {  
    this.name = name;  
    this.age = age;  
}  
var john = new Person('John', 20);  
var mark = new Person('Mark', 35);  
var daisy = new Person('Daisy', 22);  
[john, mark, daisy].min(function(person) {  
    return person.age;  
})  
// -> 20
```

partition

```
partition([iterator = Prototype.K[, context]]) -> [TrueArray, FalseArray]
```



true false Javascript

bool

findAll/select

reject

Enumerable

partiti

context

iterator

iterator

this

context

```
['hello', null, 42, false, true, , 17].partition()  
// -> [['hello', 42, true, 17], [null, false, undefined]]  
$(1, 10).partition(function(n) {  
    return 0 == n % 2;  
})  
// -> [[2, 4, 6, 8, 10], [1, 3, 5, 7, 9]]
```

pluck

```
pluck(propertyName) -> Array
```

collect

collect

```
['hello', 'world', 'this', 'is', 'nice'].pluck('length')  
// -> [5, 5, 4, 3, 4]  
document.getElementsByClassName('superfluous').pluck('tagName').sort()  
// -> sorted list of unique canonical tag names for elements with the  
// specific CSS class...
```

invoke Enumerable

reject

```
reject(iterator[, context]) -> Array
```

iterator	false
----------	-------

context	iterator	iterator	this	context
---------	----------	----------	------	---------

```
$R(1, 10).reject(function(n) {  
  return 0 == n % 2;  
})  
// -> [1, 3, 5, 7, 9]  
[ 'hello', 'world', 'this', 'is', 'nice'].reject(function(s) {  
  return s.length >= 5;  
})  
// -> ['this', 'is', 'nice']
```

[findAll](#) [select](#)

select

```
select(iterator) -> Array
```

findAll

size

```
size() -> Number
```

```
Enumerable
```

```
$R(1, 10).size()  
// -> 10  
['hello', 42, true].size()  
// -> 3  
$H().size()  
// -> 0
```

```
Enumerable
```

```
length
```

sortBy

```
sortBy(iterator[, context]) -> Array
```

iterator

Enumerable

iterator

sort

<

sortBy

Array sort

context

iterator

iterator

this

context

```
['hello', 'world', 'this', 'is', 'nice'].sortBy(function(s) {
  return s.length;
})
// -> ['is', 'this', 'nice', 'hello', 'world']
['hello', 'world', 'this', 'is', 'cool'].sortBy(function(s) {
  var md = s.match(/[aeiouy]/g);
  return null == md ? 0 : md.length;
})
// -> [ 'world', 'this', 'is', 'hello', 'cool']
// ()
```

toArray

```
toArray() -> Array
```

Enumerable Array **entries**

\$A Enumerable

```
$R(1, 5).toArray()  
// -> [1, 2, 3, 4, 5]
```

Enumerable **Array.toArray**

zip

```
zip(Sequence...[, iterator = Prototype.K]) -> Array
```

zip Haskell Ruby

```
var firstNames = ['Justin', 'Mislav', 'Tobie', 'Christophe'];
var lastNames = ['Palmer', 'Marohnić', 'Langel', 'Porteneuve'];

firstNames.zip(lastNames)
// -> [['Justin', 'Palmer'], ['Mislav', 'Marohnić'],
// ['Tobie', 'Langel'], ['Christophe', 'Porteneuve']]

firstNames.zip(lastNames, function(a) {
    return a.join(' ');
})
// -> ['Justin Palmer', 'Mislav Marohnić', 'Tobie Langel', 'Christophe Porteneuve']

var cities = ['Memphis', 'Zagreb', 'Montreal', 'Paris'];
firstNames.zip(lastNames, cities, function(p) {
    return p[0] + ' ' + p[1] + ', ' + p[2];
})
// -> ['Justin Palmer, Memphis', 'Mislav Marohnić, Zagreb',
// 'Tobie Langel, Montreal', 'Christophe Porteneuve, Paris']

firstNames.zip($R(1, 100), function(a) {
    return a.reverse().join('.');
})
// -> ['1. Justin', '2. Mislav', '3. Tobie', '4. Christophe']
```

Event

[element](#), [extend](#), [findElement](#), [isLeftClick](#), [observe](#), [pointerX](#), [pointerY](#), [stop](#), [stopObserving](#), [unloadCache](#)



W3C MSIE

Safari [keypress/keydown](#) MSIE

Prototype

Prototype

Event

Event

[KEY_LEFT](#), [KEY_UP](#), [KEY_RIGHT](#), [KEY_DOWN](#), [KEY_DELETE](#), [KEY_HOME](#), [KEY_END](#),
[KEY_PAGEUP](#), [KEY_PAGEDOWN](#)

[observeelement](#) [stop](#)

1.6

Prototype 1.6

Event

```
$('#foo').observe('click', respondToClick);
function respondToClick(event) {
    var element = Event.element(event);
    element.addClassName('active');
}
```

```
$('#foo').observe('click', respondToClick);
function respondToClick(event) {
    var element = event.element(); //
    element.addClassName('active');
```

```
}
```

Event **Event.extend**

Element **Element.e>**

Event.observe Prototype

Event.extend

element

```
Event.element(event) -> Element
```

DOM

extend 1.6

```
Event.extend(event)
```

Event.Methods event

findElement

```
Event.findElement(event, tagName) -> Element
```

DOM DOM

isLeftClick

```
Event.isLeftClick(event) -> Boolean
```

“”

observe

```
Event.observe(element, eventName, handler[, useCapture = false])
```

DOM

pointerX

```
Event.pointerX(event) -> Number
```

pointerY

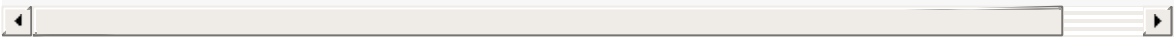
```
Event.pointerY(event) -> Number
```

stop

```
Event.stop(event)
```

stopObserving

```
Event.stopObserving(element, eventName, handler[, useCapture = false])
```



unloadCache 1.6

```
Event.unloadCache()
```

observe **1.6**

element

```
Event.element(event) -> Element
```

DOM

v1.5.1

Event.element

```
Event.observe(document.body, 'click', function(event) {  
  var element = Event.element(event);  
  if ('P' == element.tagName)  
    element.hide();  
});
```

findElement element

Prototype 1.5.0

1.5.1

DOM

Event

Element.Methods **\$()**

```
Event.observe(document.body, 'click', function(event) {  
  var element = $(Event.element(event));  
  /* ... */  
});
```


extend 1.6

```
Event.extend(event)
```

```
Event.Methods  event
```

Event.observe Element#observe

onclick

findElement

```
Event.findElement(event, tagName) -> Element
```

DOM DOM

tagName

HTMLDocument

```
Event.observe(document.body, 'click', function(event) {  
    var elt = Event.findElement(event, 'P');  
    if (elt != document)  
        $(elt).hide();  
});
```

up CSS

```
Event.observe(document.body, 'click', function(event) {  
    // CSS 'container'  
    var elt = $(Event.element(event)).up('.container');  
    // DIV  
    // elt = $(Event.element(event)).up('div', 1);  
    // CSS 'holder' DIV  
    // elt = $(Event.element(event)).up('div.holder', 1);  
    elt.hide();  
});
```

element

isLeftClick

```
Event.isLeftClick(event) -> Boolean
```

```
"""
```

```
"""
```

observe

```
Event.observe(element, eventName, handler[, useCapture = false])
```

DOM

HTML `<body onload="return myFunction(`
Level-0 `window.onload = myFunction; observe`
element+event

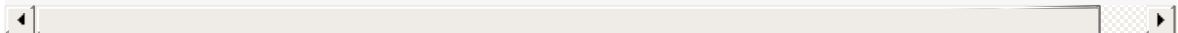
1. DOM Prototype DOM
2. DOM **DOM Level 2** 1.6 'click'
- 3.
4. *capturing* *bubbling* DOM

...

DOM DOM HTML

(X)HTML

```
<form id="signinForm" method="post" action="/auth/signin">... </form>
```



`checkForm`

```
Event.observe('signinForm', 'submit', checkForm);
```

DOM `<head> <script>`

```
Event.observe(window, 'load', function() {  
    Event.observe('signinForm', 'submit', checkForm);  
});
```

...

DOM

this

Prototype

bindAsEventListener

```
var Checks = {  
    // 'generic'  
    generic: function(event) {  
        //  
    }  
};  
  
Event.observe('signinForm', 'submit', Checks.generic.bindAsEventLis
```

stopObserving **unloadCache**

pointerX

```
Event.pointerX(event) -> Number
```

pointerY

```
Event.pointerY(event) -> Number
```

stop

```
Event.stop(event)
```

-
-

99.9% Prototype

```
Event.observe('signInForm', 'submit', function(event) {  
    var login = $F('login').strip();  
    if ('' == login) {  
        Event.stop(event);  
        //  
    }  
});
```


stopObserving

```
Event.stopObserving(element, eventName, handler[, useCapture = fa
```

observe

stopObserving

observe

```
var obj = {
  ...
  fx: function(event) {
    ...
  }
};
Event.observe(elt, 'click', obj.fx.bindAsEventListener(obj));
...
//
Event.stopObserving(elt, 'click', obj.fx.bindAsEventListener(obj));
```

bindAsEventListener

“”

script.aculo.us

```
var obj = {
  ...
  fx: function(event) {
    ...
  }
};
obj.bfx = obj.fx.bindAsEventListener(obj);
Event.observe(elt, 'click', obj.bfx);
...
Event.stopObserving(elt, 'click', obj.bfx);
```

unloadCache

unloadCache 1.6

Event.unloadCache()

observe 1.6

1.6 v1.6

MSIE

MSIE

observe

Prototype

unl

Prototype MSIE

unloadCache

Form

disable, enable, findFirstElement, focusFirstElement, getElements, getInputs,
request, reset, serialize, serializeElements

Form

“Prototype DOM”

Form Element formElementForm Control

<form>

<input><select>

disable

```
disable(formElement) -> HTMLFormElement
```

enable

```
enable(formElement) -> HTMLFormElement
```

findFirstElement

```
findFirstElement(formElement) -> HTMLElement
```

focusFirstElement

```
focusFirstElement(formElement) -> HTMLFormElement
```

getElements

```
getElements(formElement) -> array
```

getInputs

```
getInputs(formElement [, type [, name]]) -> array
```

INPUT INPUT type name

request 1.5.1

```
request([options]) -> new Ajax.Request
```

HTTP getpost Ajax.Request action URL options Ajax.Reques

reset

```
reset(formElement) -> HTMLFormElement
```

serialize

```
serialize(formElement[, getHash = false]) -> String | object
```

Ajax getHash true Hash name ""

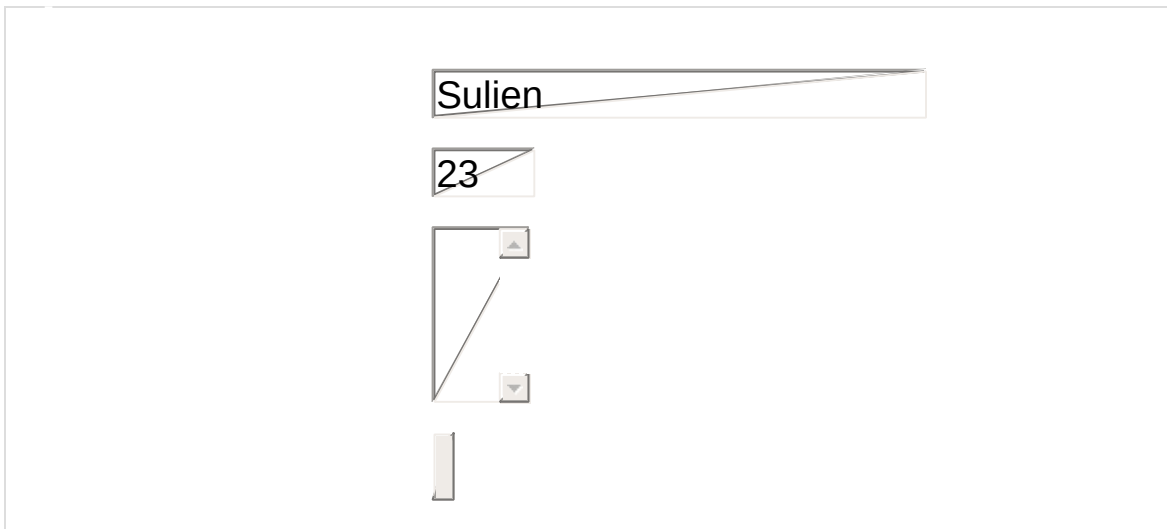
serializeElements

```
serializeElements(elements[, getHash = false]) -> string | object
```

Ajax getHash true Hash name ""

disable

```
disable(formElement) -> HTMLFormElement
```



```
var form = $('disable-example');  
// form.disable() form.enable()  
form[form.disabled ? 'enable' : 'disable']();  
form.disabled = !form.disabled;
```

```
form[form.disabled ? 'enable' : 'disable']();
```

```
form.disabled = true
```

```
form[form.disabled ? 'enable' : 'disable']();
```

JavaScript

`form['disable']() ==> form.disable()`

enable

```
enable(formElement) -> HTMLFormElement
```

`disable()`

HTML

findFirstElement

```
findFirstElement(formElement) -> HTMLElement
```

INPUTSELECT TEXTAREA

document

Tab

focusFirstElement

```
focusFirstElement(formElement) -> HTMLFormElement
```

Form.findFirstElement()

activate()

getElements

```
getElements(formElement) -> array
```

OPTION SELECT

getInputs

```
getInputs(formElement [, type [, name]]) -> array
```

```
INPUT INPUT type name
```

```
var form = $('myform') form.getInputs()  
// -> INPUT  
form.getInputs('text')  
// -> type = 'text' INPUT  
var buttons = form.getInputs('radio', 'education')  
// -> 'education'  
//  
buttons.invoke('disable')
```

INPUT

document

Tab

request 1.5.1

```
request([options]) -> new Ajax.Request
```

HTTP	Ajax.Request getpost	action URL	options Ajax.Req
------	-------------------------	------------	------------------

request() options Ajax.Request

- method Ajax.Request method options m
method "POST"
- options parameters "" hash

HTML

```
<form id="person-example" method="POST" action="/user/info">
  <fieldset>
    <legend>User info</legend>
    <div>
      <label for="username">Username:</label>
      <input type="text" name="username" id="user
    </div>
    <div>
      <label for="age">Age:</label>
      <input type="text" name="age" id="age" valu
    </div>
    <div>
      <label for="hobbies">Your hobbies are:</lat
      <select name="hobbies[]" id="hobbies" multi
        <option>coding</option>
        <option>swimming</option>
        <option>hiking</option>
        <option>drawing</option>
      </select>
    </div>
    <div class="buttonrow">
      <input type="submit" value="serialize!" />
    </div>
  </fieldset>
</form>
```

Ajax

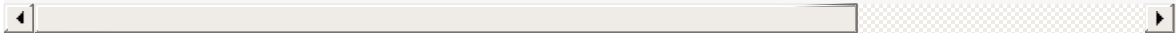
```
$('#person-example').request();  
//  
//  
$('#person-example').request({  
    onComplete: function(){  
        alert('Form data saved!')  
    }  
})
```

HTTP

options method par

interests hobbies

```
$('#person-example').request({  
    method: 'get',  
    parameters: { interests: 'JavaScript', 'hobbies[]': ['program  
    onComplete: function(){ alert('Form data saved!') }  
})
```



reset

```
reset(formElement) -> HTMLFormElement
```

```
Form.reset('contact')  
//  
$('contact').reset()  
// reset
```

HTML `reset()`

serialize

```
serialize(formElement[, getHash = false]) -> String | object
```

Ajax

getHash true Hash name ""

```
getHash true {name: "johnny", color: "blue"}
```

"name=johnny&color=blue"

Ajax

Prototype 1.5 Ajax

Hash getHash

```
$('#person-example').serialize()  
// -> 'username=sulien&age;=22&hobbies;=coding&hobbies;=hiking'  
$('#person-example').serialize(true)  
// -> {username: 'sulien', age: '22', hobbies: ['coding', 'hiking']}
```

User info

Username:

Age:

Your hobbies are:

coding

swimming

hiking

drawing

serialize!

W3C HTML JavaScript

"hobbies"

SELECT PHP RORRuby on

"hobbies[]"

array JavaScript

—

serializeElements

```
serializeElements(elements[, getHash = false]) -> string | object
```

1

Ajax

getHash true Hash name

Form.serialize

getHash

Form

serializeElement

"text" INPUT

```
Form.serializeElements($('myform').getInputs('text'))  
// ->
```

Form.Element

activate, clear, disable, enable, focus, getValue, present, select, serialize,
setValue

/

Prototype `Form.Element` `Field`

“Prototype `Field`”

`INPUTSELECT` `TEXTAREA` `Form.Element`

```
Form.Element.activate('myfield')  
Field.activate('myfield')  
$('myfield').activate()
```

`Form.Element`

activate

```
activate(element) -> HTMLElement
```

clear

```
clear(element) -> HTMLElement
```

disable

```
disable(element) -> HTMLElement
```

enable

```
enable(element) -> HTMLElement
```

focus

```
focus(element) -> HTMLElement
```

getValue

```
getValue(element) -> string | array
```

\$F()

present

```
present(element) -> boolean
```

true false

select

```
select(element) -> HTMLElement
```

serialize

```
serialize(element) -> string
```

name=value URL

setValue

```
setValue(element, value) -> HTMLElement
```

value

activate

activate(element) -> HTMLElement

```
Form.Element.focus('myelement').select() // select
$('myelement').activate()
```

;)

Form.Element.focus(element) element el
select

Prototype 3562 3567 Element.rela
3564 3569

```
$(element).focus();
```

```
element = $(element);  
element.focus();
```

Prototype 1.6.0.3

clear

```
clear(element) -> HTMLElement
```

```
$('#some_field').onfocus = function() {  
    //  
    if (this._cleared)  
        return  
    // "this"  
    this.clear()  
    this._cleared = true  
}
```

disable

```
disable(element) -> HTMLElement
```

```
disable      true
```

Form.disable()

JavaScript

enable

```
enable(element) -> HTMLElement
```

`Form.disable()`

focus

focus(element) -> HTMLElement

\$('#inputElement').focus() JavaScript

Form.Element.select()

```
Form.Element.focus('searchbox')
//
//  focus() :
$('#searchbox').focus()
```

Prototype 1.6.0.3	Form.Element	focus	select
element		element	
Prototype		3562 3567	Element.rela
3564 3569			

\$(element).focus();

```
element = $(element);
element.focus();
```

Prototype 1.6.0.3

getValue

```
getValue(element) -> string | array
```

\$F()

name

DOM

ID

ID "contact" "company"

```
var form = $('contact');  
var input = form['company'];  
Form.Element.getValue(input);  
//  
$(input).getValue();  
//  $()  
//  
$F(input);
```

element ID"element has no properties"

present

```
present(element) -> boolean
```

```
true false
```

User Details

Please fill out the following fields:

Username

Email Address

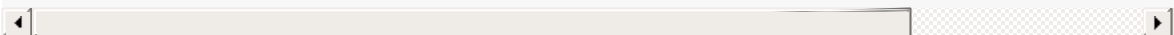
submit

JavaScript

```
$('#example').onsubmit = function(){
    var valid, msg = $('#msg')

    //
    valid = $(this.username).present() && $(this.email).present()

    if (valid) {
        // true
        // return true
        msg.update('Passed validation!').style.color = 'green'
    }
    else {
        msg.update('Please fill out all the fields.').style.color = 'red'
    }
    return false
}
```



select

```
select(element) -> HTMLElement
```

`$('#inputElement').select()` JavaScript

`Form.Element.focus()`

```
$('#searchbox').onfocus = function() {  
    Form.Element.select(this)  
    //  
    this.select()  
}
```

+ **activate!**

activate — JavaScript

Prototype 1.6.0.3	Form.Element	focus	select
element		element	
Prototype		3562	3567
3564	3569		Element.rela

```
$(element).focus();
```

```
element = $(element);  
element.focus();
```

Prototype 1.6.0.3

serialize

```
serialize(element) -> string
```

```
name=value URL
```

Ajax —

Ajax

getValue()

Function

argumentNames, bind, bindAsEventListener, curry, defer, delay, methodize, wrap

Prototype **binding**

“” **this**

JavaScript **this** **bind()**

Prototype

Prototype **this** Prototype

Prototype **Function** **String**

charAt

argumentNames 1.6

`someFunction.argumentNames() -> Array`

bind

`bind(thisObj[, arg...]) -> Function`

this **thisObj** **thisObj**

arg **args**

bindAsEventListener

`bindAsEventListener(thisObj[, arg...]) -> Function`

bind

curry 1.6.0

```
curry(arg...) -> Function
```

arg

defer 1.6.0

```
defer(arg...) -> Number
```

JavaScript

delay 1.6.0

```
delay(seconds[, arg...]) -> Number
```

seconds arg

methodize

```
someFunction.methodize() -> Function
```

this

wrap 1.6.0

```
wrap(wrapperFunction[, arg...]) -> Function
```

wrapperFunction

wrapperFunction

argumentNames 1.6

```
someFunction.argumentNames() -> Array
```

```
var fn = function(foo, bar) {  
    return foo + bar;  
};  
fn.argumentNames();  
//-> ['foo', 'bar']  
Prototype.emptyFunction.argumentNames();  
//-> []
```


bind

```
bind(thisObj[, arg...]) -> Function
```

this thisObj

thisObj
arg args

Function

Ja

JavaScript “[scope]”

```
window.name = "the window object";
function scopeTest() {
    return this.name
}
//
scopeTest()
// -> "the window object"
var foo = {
    name: "the foo object!",
    otherScopeTest: function() {
        return this.name
    }
}
foo.otherScopeTest()
// -> "the foo object!"
```

otherScoptTest

```
// ...
//
window.test = foo.otherScopeTest
//
test()
// -> "the window object"
```

Prototype

```
var obj = {
  name: 'A nice demo',
  fx: function() {
    alert(this.name);
  }
};

window.name = 'I am such a beautiful window!';
function runFx(f) {
  f();
}
var fx2 = obj.fx.bind(obj);
runFx(obj.fx);
runFx(fx2);
```

Try it out!

[bind](#)

```
var obj = {
  name: 'A nice demo',
  fx: function() {
    alert(this.name + '\n' + $A(arguments).join(', '));
  }
};

var fx2 = obj.fx.bind(obj, 1, 2, 3);
fx2(4, 5);
// name "1, 2, 3, 4, 5"
```

Try it out!

OK

JavaScript

bindAsEventListener

```
bindAsEventListener(thisObj[, arg...]) -> Function
```

bind

“”

Function

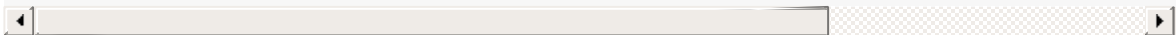
bind()

DOM Level 0 MSIE

Event.observe

```
var obj = { name: 'A nice demo' };
function handler(e) {
    var tag = Event.element(e).tagName.toLowerCase();
    var data = $A(arguments);
    data.shift();
    alert(this.name + '\nClick on a ' + tag + '\nOther args: '
}

Event.observe(window, 'load', function(){
    Event.observe(document.body, 'click', handler.bindAsEventLi
    // obj.name "1, 2, 3"
})
```



curry

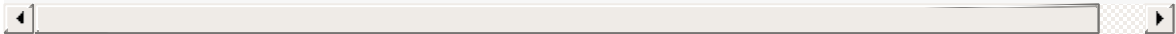
1.6.0

```
curry(arg...) -> Function
```

arg

Function#curry Function#bind

```
String.prototype.splitOnSpaces = String.prototype.split.curry(" ");  
"foo bar baz thud".splitOnSpaces();  
//-> ["foo", "bar", "baz", "thud"]
```



defer 1.6.0

```
defer(arg...) -> Number
```

JavaScript

[delay](#) [seconds](#) [0.01](#) [JavaScript](#) [Google](#)

Chrome V8

[delay](#)

[window.setTimeout](#)

[setTimeout](#)

[setTimeout](#) ""

"" JavaScript

[window.setTimeout](#) ID

[window.cle](#)

```
function hideNewElement() {
    $('inserted').hide();
};
function insertThenHide(markup) {
    $('container').insert(markup);
    // IE DOM
    hideNewElement.defer();
}
insertThenHide("<div id='inserted'>Lorem ipsum</div>");
```

delay 1.6.0

```
delay(seconds[, arg...]) -> Number
```

seconds **arg**

window.setTimeout ID

JavaScript

Function#defer

```
//  
window.setTimeout(function() {  
    Element.addClassName('foo', 'bar');  
}, 1000);  
  
//  
Element.addClassName.delay(1, 'foo', 'bar');  
//  
var id = Element.hide.delay(5, 'foo');  
window.clearTimeout(id);
```

methodize

```
someFunction.methodize() -> Function
```

this

methodize

thispro

```
//  
var fn = function(target, foo) {  
    target.value = foo;  
};  
var object = {};  
  
//  
fn(object, 'bar');  
object.value  
//-> 'bar'  
  
// fn  
object.fnMethodized = fn.methodize();  
object.fnMethodized('boom!');  
object.value  
//-> 'boom!'
```

prototype

Element.Methods Event.Methods prototype DOM

```
Event.observe('inputId', 'click', handler)
```

```
$('inputId').observe('click', handler)
```

```
var input = document.getElementById('inputId');  
input.observe = Event.observe.methodize();  
input.observe('click', handler);
```

wrap 1.6.0

wrap(wrapperFunction[, arg...]) -> Function

wrapperFunction

wrapperFunction

Function#wrap

```
String.prototype.capitalize = String.prototype.capitalize.wrap(
  function(proceed, eachWord) {
    if (eachWord && this.include(" ")) {
      //
      return this.split(" ").invoke("capitalize")
    }
    else {
      //
      return proceed();
    }
  }
);

"hello world".capitalize()
// "Hello world"
"hello world".capitalize(true)
// "Hello World"
```


Hash

clone, each, get, inspect, keys, merge, remove, set, toJSON, toObject,
toQueryString, unset, update, values

Hash JavaScript
"/" hash hash

hash

Hash new JavaScript new Hash(obj)
JavaScript Hash \$H\$H
hash

Prototype 1.6

- Hash Hash Ha

Hash "/" Hash
Hash#set(key, value) Hash#unset(key) "/"

```
var myhash = new Hash();  
// API --> API  
myhash.name = "Bob";    --> myhash.set('name', 'Bob');  
myhash.name;           --> myhash.get('name');  
delete myhash.name;     --> myhash.unset('name');
```

Hash API

- `$H(object)` `new Hash(object)` Hash
- `Hash#merge` `Hash`
- `Hash#update` `Hash#merge`
- `Hash#clone` `Hash`

- `Hash#toObject` `Hash`
 Prototype `“/”`
- `Hash.toQueryString` `Object.toQueryString`
 `Hash.toQueryString` Prototype
- `Hash#remove` `Hash#unset`
- `Hash.toJSON` `Object.toJSON` `Hash#toJSON`

Prototype < 1.6

`hash` `$H`

`Enumerable` `Hash “”`

```
var h = new Hash({ ... });
h['each'] = 'my own stuff';
h.map();
// -> 'each'
```

`Enumerable` `each` — `hash`

`clone` 1.6

```
clone() -> newHash
```

`hash`

`each`

```
each(iterator) -> Hash
```

`hash “/”`

`get` 1.6

```
get(key) -> value
```

`hash`

inspect

```
inspect() -> String
```

hash

keys

```
keys() -> [String...]
```

hash

merge 1.6

```
merge(object) -> newHash
```

`object` hash `v1.6.0` hash `v1.6.0`

remove

```
remove(key) -> value  
remove(key1, key2...) -> Array
```

hash `v1.6.0`

set 1.6

```
set(key, value) -> value
```

hash `key` `valuevalue`

toJSON 1.5.1

```
toJSON() -> String
```

JSON

toObject 1.6

```
toObject() -> Object
```

hash `Object` `Hash` `Object "/"`

toQueryString 1.6

```
toQueryString() -> String
```

hash URL

unset 1.6

```
unset(key) -> value
```

hash

update 1.6

```
update(object) -> Hash
```

`object` "/" hash hash

values

```
values() -> Array
```

hash

clone 1.6

clone() -> newHash

hash

```
var h = new Hash({ a: 'apple'});
var clone = h.clone();
h.unset('a');
h.inspect();
// -> #<Hash:{}>
clone.inspect();
// -> #<Hash:{'a': 'apple'}>
```

each

```
each(iterator) -> Hash
```

```
hash "/"
```

Enumerable each iterator

"/" iterator

1. key"" String
2. value"" undefined)

```
for ... in
```

Hash Enumerable hash

```
var h = $H({ version: 1.5, author: 'Sam Stephenson' });
h.each(function(pair) {
  alert(pair.key + ' = ' + pair.value + '');
});
//
// --> 'version = "1.5"' 'author = "Sam Stephenson"'
```

get 1.6

`get(key) -> value`

hash

```
var h = new Hash({ a: 'apple', b: 'banana', c: 'coconut' });  
h.get('a');  
// -> 'apple'  
h.get('d');  
// -> undefined
```

inspect

```
inspect() -> String
```

```
hash
```

`inspect`

`Object.inspect`

```
$H({ name: 'Prototype', version: 1.5 }).inspect()  
// -> "<#Hash:{name: 'Prototype', version: 1.5}>"
```

Hash “/”

JavaScript “/”

—

keys

```
keys() -> [String...]
```

```
hash
```

```
for ... in
```

```
$H({ name: 'Prototype', version: 1.5 }).keys().sort()  
// -> ['name', 'version']  
$H().keys()  
// -> []
```

merge

1.6 modified

merge(object) -> newHash

object hash

v1.6.0

hash

v1.

"" hash

object

""

object Hash Object

1.6.0 Hash#merge hash

```
var h = $H({ name: 'Prototype', version: 1.5 });  
h.merge({version: 1.6, author: 'Sam' }).inspect();  
// -> #<Hash:{'name': 'Prototype', 'version': 1.6, 'author': 'Sam'}  
h.inspect();  
// -> #<Hash:{'name': 'Prototype', 'version': 1.5}>
```

1.6.0 Hash#merge hash

```
var h = $H({ name: 'Prototype', version: 1.5 });  
h.merge({version: 1.6, author: 'Sam' }).inspect();  
// -> #<Hash:{'name': 'Prototype', 'version': 1.6, 'author': 'Sam'}  
h.inspect();  
// -> #<Hash:{'name': 'Prototype', 'version': 1.6, 'author': 'Sam'}>
```

1.6.0


Hash#update

remove

```
remove(key) -> value  
remove(key1, key2...) -> Array
```

hash **v1.6.0**

Prototype 1.6.0

[Hash#unset\(\)](#) 

```
['foo', 'bar'].each(function(key) {  
  hash.unset(key);  
});
```

```
var h = new Hash({ a:'apple', b:'banana', c:'coconut' })  
h.remove('a', 'c')  
// -> ['apple', 'coconut']  
h.values()  
// -> ['banana']
```

set 1.6

```
set(key, value) -> value
```

hash	key	value	value
------	-----	-------	-------

```
var h = new Hash({ a: 'apple', b: 'banana', c: 'coconut' });
h.set('d', 'orange');
// -> 'orange'
h.inspect();
// -> #<Hash:{'a': 'apple', 'b': 'banana', 'c': 'coconut', 'd': 'orange'}>
h.set('a', 'kiwi');
// -> 'kiwi'
h.inspect();
// -> #<Hash:{'a': 'kiwi', 'b': 'banana', 'c': 'coconut', 'd': 'orange'}>
```



toJSON

1.5.1

toJSON() -> String

JSON

```
$H({name: 'Violet', occupation: 'character', age: 25 }).toJSON();  
//-> '{"name": "Violet", "occupation": "character", "age": 25}'
```

toObject 1.6

toObject() -> Object

hash	Object	Hash	Object "/"
------	--------	------	------------

```
var h = new Hash({ a: 'apple', b: 'banana', c: 'coconut' });
var obj = h.toObject();

Object.inspect(h);
// -> #<Hash:{'a': 'kiwi', 'b': 'banana', 'c': 'coconut'}>
Object.inspect(obj);
// -> [object Object]
obj.a = 'orange';
h.get('a');
// -> 'apple'
```

toQueryString 1.6

```
toQueryString() -> String
```

hash URL

Ajax

undefined `"/"/`

encodeURIComponent URI

```
$H({ action: 'ship', order_id: 123, fees: ['f1', 'f2'], 'label': 'a'  
// -> 'action=ship_o_id=123&fees;=f1&fees;=f2&label;=a%20demo'  
// hash  
$H().toQueryString()  
// -> ''
```

```
var hash = new Hash({a: 'aa', b: null, c: undefined, d: 'dd'});  
hash.toQueryString()  
// -> 'a=aa&b;=&c;&d;=dd'
```

Hash.toQueryString(obj)

Object.toQueryString

unset 1.6

unset(key) -> value

hash

```
var h = new Hash({ a: 'apple', b: 'banana', c: 'coconut' });
h.unset('a');
// -> 'apple'
h.inspect();
// -> #<Hash:{'b': 'banana', 'c': 'coconut'}>
h.unset('d');
// -> undefined
h.inspect();
// -> #<Hash:{'b': 'banana', 'c': 'coconut'}>
```


update 1.6

update(object) -> Hash

object **"/" hash hash**

"/" hash

object

"/"

object Hash Object

```
var h = $H({ name: 'Prototype', version: 1.5 });  
h.update({ version: 1.6, author: 'Sam' }).inspect();  
// -> #<Hash:{'name': 'Prototype', 'version': 1.6, 'author': 'Sam'}  
h.inspect();  
// -> #<Hash:{'name': 'Prototype', 'version': 1.6, 'author': 'Sam'}
```

Hash#merge

values

```
values() -> Array
```

```
hash
```

```
“”
```

```
for ... in —
```

```
undefined
```

```
$H({ name: 'Prototype', version: 1.5 }).values().sort()  
// -> [1.5, 'Prototype']  
$H().values()  
// -> []
```

Insertion

After, Before, Bottom, Top

Prototype 1.6

Insertion

Element#insert

Insertion HTML

HTML

<script>

Inse

After []

```
new Insertion.After(element, html)
```

html element

Before []

```
new Insertion.Before(element, html)
```

html element

Bottom []

```
new Insertion.Bottom(element, html)
```

html element

Top []

```
new Insertion.Top(element, html)
```

html element

After

```
new Insertion.After(element, html)
```

html element

Prototype 1.6

Insertion

Element#insert

HTML

<script>

Inse

String#evalScripts

HTML

```
<div>
  <p id="animal_vegetable_mineral">
    In short, in all things vegetable, animal, and mine
  </p>
</div>
```

JavaScript

```
new Insertion.After(
  'animal_vegetable_mineral',
  "<p>I am the very model of a modern major general.</p>"
);
```

HTML

```
<div>
  <p id="animal_vegetable_mineral">
    In short, in all things vegetable, animal, and mine
  </p>
  <p>
    I am the very model of a modern major general.
  </p>
</div>
```

Before

```
new Insertion.Before(element, html)
```

html element

Prototype 1.6

Insertion

Element#insert

HTML

<script>

Inse

String#evalScripts

HTML

```
<div>
  <p id="modern_major_general">
    I am the very model of a modern major general.
  </p>
</div>
```

JavaScript

```
new Insertion.Before(
  'modern_major_general',
  "<p>In short, in all things vegetable, animal, and mineral.
");
```

HTML

```
<div>
  <p>
    In short, in all things vegetable, animal, and mine
  </p>
  <p id="modern_major_general">
    I am the very model of a modern major general.
  </p>
</div>
```

Bottom

```
new Insertion.Bottom(element, html)
```

html element

Prototype 1.6

Insertion Element#insert

HTML

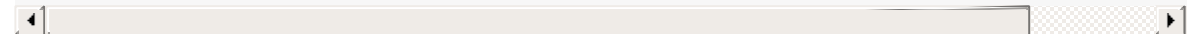
<script>

Inse

String#evalScripts

HTML

```
<div id="modern_major_general">
  <p>In short, in all things vegetable, animal, and mineral..
</div>
```

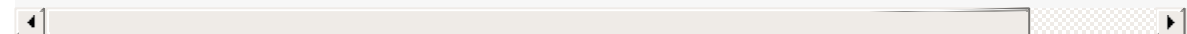


JavaScript

```
new Insertion.Bottom(
  'modern_major_general',
  "<p>I am the very model of a modern major general.</p>"
);
```

HTML

```
<div id="modern_major_general">
  <p>In short, in all things vegetable, animal, and mineral..
  <p>I am the very model of a modern major general.</p>
</div>
```



Top

```
new Insertion.Top(element, html)
```

```
html element
```

Prototype 1.6

Insertion

Element#insert

HTML

<script>

Inse

String#evalScripts

HTML

```
<div id="modern_major_general">
  <p>I am the very model of a modern major general.</p>
</div>
```

JavaScript

```
new Insertion.Top(
  'modern_major_general',
  "<p>In short, in all things vegetable, animal, and mineral.
");
```

HTML

```
<div id="modern_major_general">
  <p>In short, in all things vegetable, animal, and mineral..
  <p>I am the very model of a modern major general.</p>
</div>
```

Number

[abs](#), [ceil](#), [floor](#), [round](#), [succ](#), [times](#), [toColorPart](#), [toJSON](#), [toPaddedString](#)

Prototype JavaScript [Number](#)

- [Number#succ](#) [ObjectRange](#)
- [Number#times](#) [Ruby](#)
- [Number#toColorPart](#) [Number#toPaddedString](#)

```
$R(1, 10).each(function(index) {  
    // 1 10 10  
});  
  
(5).times(function(index) {  
    // 0 5 5  
    // 5 JavaScript  
    // Number  
    // 5.toString() (5).toString()  
});  
  
(128).toColorPart()  
// -> '80'  
(10).toColorPart()  
// -> '0a'  
'#' + [128, 10, 16].invoke('toColorPart').join('')  
// -> '#800a10'
```

abs [1.6.0](#)

`abs()` -> Number

[Number](#)

ceil [1.6.0](#)

`ceil()` -> Number

[Number](#)

floor 1.6.0

```
floor() -> Number
```

Number

round 1.6.0

```
round() -> Number
```

Number

succ

```
succ() -> Number
```

Number

ObjectRange

times

```
times(iterator) -> Number
```

Ruby [0...n]

toColorPart

```
toColorPart() -> String
```

Number

Number [0...255] CSS

toJSON 1.5.1

```
toJSON() -> String
```

JSON

toPaddedString 1.5.1

```
toPaddedString(length[, radix]) -> String
```

Number length 0 radix
radix 3 7 10

abs

1.6.0

`abs()` -> Number

Number

Math.abs

```
Math.abs(-5) //-> 5  
(-5).abs() //-> 5  
(5).abs() //-> 5
```

ceil

1.6.0

```
ceil() -> Number
```

Number

Math.ceil

```
Math.ceil(4.1) //-> 5  
(4.1).ceil() //-> 5  
(-4.1).ceil() //-> -4
```

floor

1.6.0

floor() -> Number

Number

Math.floor

```
Math.floor(4.6) //-> 4
(4.6).floor() //-> 4
(-4.1).floor() //-> -5
```

round

1.6.0

round() -> Number

Number

Math.round

```
Math.round(4.5) //-> 5
(4.5).round() //-> 5
(4.49).round() //-> 4
(-4.5).round() //-> -4
```

SUCC

`succ()` -> Number

Number

ObjectRange

```
(5).succ() // -> 6  
$A($R(1, 5)).join('') // -> '12345'
```

times

```
times(iterator) -> Number
```

Ruby [0...n]

iterator 0

Number

```
var s = '';  
(5).times(function(n) { s += n; });  
s // -> '01234'
```

toColorPart

toColorPart() -> String

Number

Number [0...255] CSS

```
128.toColorPart() // -> '70'  
10.toColorPart() // -> '0a'  
'#' + [128, 10, 16].invoke('toColorPart').join('')  
// -> '#800a10'
```


toJSON

1.5.1

```
toJSON() -> String
```

JSON

```
(45).toJSON(); //-> '45'
```

toPaddedString 1.5.1

```
toPaddedString(length[, radix]) -> String
```

| Number | length | 0 | radix |
|--------|--------|---|-------|
| radix | 3 7 10 | | |

```
(13).toPaddedString(4); // -> '0013'  
(13).toPaddedString(2); // -> '13'  
(13).toPaddedString(1); // -> '13'  
(13).toPaddedString(4, 16); // -> '000d'  
(13).toPaddedString(4, 2); // -> '1101'
```

ObjectRange

include

Range

Range""

“

Prototype Number String succ

ObjectRange Enumerable

ObjectRange \$R

```
$A($R(1, 5)).join(', ')  
// -> '1, 2, 3, 4, 5'  
$R(1, 5).zip(  
  ['one', 'two', 'three', 'four', 'five'],  
  function(tuple) {  
    return tuple.join(' = ');  
  }  
)  
// -> ['1 = one', '2 = two', '3 = three', '4 = four', '5 = five']
```

String String SUCC

```
$A($R('a', 'e'))  
// -> ['a', 'b', 'c', 'd', 'e']  
$A($R('ax', 'ba'))  
// -> ['ax', 'ay', 'az', 'a{', 'a|', 'a}', 'a~'...]
```

include

include(value) -> Boolean

include

```
include(value) -> Boolean
```

<

value ObjectRange

value ObjectRange

ObjectRange include

```
var Sequence = Class.create({
  initialize: function(original){
    this.original = original;
  },
  // succ
  succ: function(){
    //
    return new Sequence(this.original + 2);
  },
  // toString include
  // $R(new Sequence(1), new Sequence(10))
  toString: function(){
    return this.original;
  }
});

var result = [];
var range = $R(new Sequence(1), new Sequence(10));
range.each(function(s){
  result.push(s.toString());
});
result.join(' ');
// -> '1 3 5 7 9'

range.include(new Sequence(4));
// -> true
```

include

```
$R(1, 10).include(5) // -> true  
$R('a', 'h').include('x') // -> false  
$R(1, 10).include(10) // -> true  
$R(1, 10, true).include(10) // -> false
```

Object

clone, extend, inspect, isArray, isElement, isFunction, isHash, isNumber,
isString, isUndefined, keys, toHTML, toJSON, toQueryString, values

Prototype `Object`

Prototype

Prototype hash

clone

```
Object.clone(obj) -> Object
```

extend

```
Object.extend(dest, src) -> alteredDest
```

`src` `dest` prototype

Prototype

inspect

```
Object.inspect(obj) -> String
```

`obj`

isArray 1.6

```
isArray(obj) -> Boolean
```

`obj` `true` `false`

isElement 1.6

```
isElement(obj) -> Boolean
```

```
obj DOM 1      true false
```

isFunction 1.6

```
isFunction(obj) -> Boolean
```

```
obj Function    true false
```

isHash 1.6

```
isHash(obj) -> Boolean
```

```
obj Hash      true  false
```

isNumber 1.6

```
isNumber(obj) -> Boolean
```

```
obj Number     true false
```

isString 1.6

```
isString(obj) -> Boolean
```

```
obj String     true false
```

isUndefined 1.6

```
isUndefined(obj) -> Boolean
```

```
obj undefined   true  false
```

keys

```
Object.keys(obj) -> [String...]
```

```
obj Hash
```

toHTML 1.6

```
toHTML(obj) -> String
```

```
obj toHTML      obj toHTML      obj String.interpret
```

toJson 1.5.1

```
toJson(obj) -> String
```

JSON

toQueryString 1.6

```
toQueryString(obj) -> String
```

URL

values

```
Object.values(obj) -> Array
```

`obj` `Hash`

clone

```
Object.clone(obj) -> Object
```

```
var o = { name: 'Prototype', version: 1.5, authors: ['sam', 'contri']
var o2 = Object.clone(o);

o2.version = '1.5 weird';
o2.authors.pop();

o.version
// -> 1.5
o2.version
// -> '1.5 weird'
o.authors
// -> ['sam']
//
```

extend

```
Object.extend(dest, src) -> alteredDest
```

| | | | |
|-----|------|-----------|-----------|
| src | dest | prototype | Prototype |
|-----|------|-----------|-----------|

Prototype OOP Prototype

Class.create

Element.extend

Prototype DOM

inspect

```
Object.inspect(obj) -> String
```

```
obj
```

- undefined null

- `inspect` `tc`

Prototype `inspect` —

```
Object.inspect()  
// -> 'undefined'  
Object.inspect(null)  
// -> 'null'  
Object.inspect(false)  
// -> 'false'  
Object.inspect([1, 2, 3])  
// -> '[1, 2, 3]'  
Object.inspect('hello')  
// -> '"hello'"
```

isArray 1.6

isArray(obj) -> Boolean

| obj | true | false |
|-----|------|-------|
|-----|------|-------|

```
Object.isArray([]); //-> true  
Object.isArray($w()); //-> true  
Object.isArray({ }); //-> false
```

isElement

1.6

isElement(obj) -> Boolean

obj DOM 1 true false

```
Object.isElement(new Element('div'));  
//-> true  
Object.isElement(document.createElement('div'));  
//-> true  
Object.isElement($('id_of_an_exiting_element'));  
//-> true  
Object.isElement(document.createTextNode('foo'));  
//-> false
```

isFunction 1.6

```
isFunction(obj) -> Boolean
```

| obj | Function | true | false |
|-----|----------|------|-------|
|-----|----------|------|-------|

```
Object.isFunction($); //-> true  
Object.isFunction(123); //-> false
```

isHash

1.6

isHash(obj) -> Boolean

| | | | |
|-----|------|------|-------|
| obj | Hash | true | false |
|-----|------|------|-------|

```
Object.isHash(new Hash({ })); //-> true  
Object.isHash($H({ })); //-> true  
Object.isHash({ }); //-> false
```

isNumber 1.6

```
isNumber(obj) -> Boolean
```

| obj | Number | true | false |
|-----|--------|------|-------|
|-----|--------|------|-------|

```
Object.isNumber(0); //-> true  
Object.isNumber(1.2); //-> true  
Object.isNumber("foo"); //-> false
```


isString

1.6

```
isString(obj) -> Boolean
```

| | | | |
|-----|--------|------|-------|
| obj | String | true | false |
|-----|--------|------|-------|

```
Object.isString("foo"); //-> true  
Object.isString(""); //-> true  
Object.isString(123); //-> false
```

isUndefined

1.6

isUndefined(obj) -> Boolean

| | | | |
|-----|-----------|------|-------|
| obj | undefined | true | false |
|-----|-----------|------|-------|

```
Object.isUndefined(); //-> true
Object.isUndefined(undefined); //-> true
Object.isUndefined(null); //-> false
Object.isUndefined(0); //-> false
Object.isUndefined(""); //-> false
```

keys

```
Object.keys(obj) -> [String...]
```

`obj` **Hash**

`for...in`

```
Object.keys()  
// -> []  
Object.keys({ name: 'Prototype', version: 1.5 }).sort()  
// -> ['name', 'version']
```

toHTML 1.6

toHTML(obj) -> String

obj toHTML obj toHTML obj String.interpret

```
var Bookmark = Class.create({
  initialize: function(name, url) {
    this.name = name;
    this.url = url;
  },
  toHTML: function() {
    return '<a href="#{url}">#{name}</a>'.interpolate(t
  }
});

var api = new Bookmark('Prototype API', 'http://prototypejs.org/api
Object.toHTML(api);
//-> '<a href="http://prototypejs.org/api">Prototype API</a>'

Object.toHTML("Hello world!");
//-> "Hello world!"

Object.toHTML();
//-> ""

Object.toHTML(null);
//-> ""

Object.toHTML(undefined);
//-> ""

Object.toHTML(true);
//-> "true"

Object.toHTML(false);
//-> "false"

Object.toHTML(123);
//-> "123"
```

toJson

1.5.1

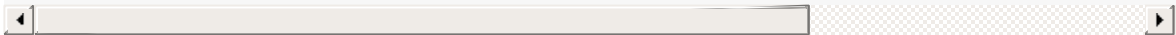
```
toJson(obj) -> String
```

JSON

Prototype JSON

Example

```
var data = {name: 'Violet', occupation: 'character', age: 25, pets:  
Object.toJson(data);  
//-> '{"name": "Violet", "occupation": "character", "age": 25, "pet
```



toQueryString 1.6

```
toQueryString(obj) -> String
```

URL

Ajax

undefined `"/"/`

encodeURIComponent URI

```
Object.toQueryString({
  action: 'ship',
  order_id: 123,
  fees: ['f1', 'f2'],
  'label': 'a demo'
})
// -> 'action=ship_o_id=123&fees;=f1&fees;=f2&label;=a%20demo'
```

values

```
Object.values(obj) -> Array
```

```
obj  Hash
```

```
for...in
```

```
Object.values()  
// -> []  
Object.values({ name: 'Prototype', version: 1.5 }).sort()  
// -> [1.5, 'Prototype']
```

PeriodicalExecuter

stop

Wind

PeriodicalExecuter

PeriodicalExecuter

PeriodicalExecuter

```
new PeriodicalExecuter(pollChatRoom, 3);
new PeriodicalExecuter(function(pe) {
    if (!confirm('Want me to annoy you again later?'))
        pe.stop();
}, 5);
//
```

setInterval IE
setInterval

FireFox

```
var interval = setInterval(function(){
    if (!confirm('Want me to annoy you again later?'))
        clearInterval(interval);
}, 5000)
```

stop

stop()

PeriodicalExecuter

stop

stop()

PeriodicalExecuter

PeriodicalExecuter

PeriodicalExecuter registerCallback

registerCallback

```
PeriodicalExecuter.prototype.resume = function(){
    if(!this.timer)
        this.registerCallback();
};
```

```
var pe = new PeriodicalExecuter(function(){
    alert('');
}, 5);

//
pe.stop();
//
pe.resume();
```

```
var gCallCount = 0;
new PeriodicalExecuter(function(pe) {
    if (++gCallCount > 3)
        pe.stop();
    else
        alert(gCallCount);
}, 1);
// 1, 2, 3
```

Position

absolutize, clone, cumulativeOffset, offsetParent, overlap, page,
positionedOffset, prepare, realOffset, relativize, within,
withinIncludingScrolloffsets

Prototype 1.6

Position

Element

Position DOM UI

absolutize []

```
absolutize(element)
```

```
element.style.position = 'absolute'
```

clone []

```
clone(source, target[, options]) -> [Number, Number]
```

```
source options / target
```

cumulativeOffset []

```
cumulativeOffset(element) -> [Number, Number]
```

```
element
```

offsetParent []

```
offsetParent(element) -> HTMLElement
```

```
element Containing Block  
body
```

CSS

```
position relative absolute
```

overlap []

```
overlap(mode, element) -> Number
```

0 1

Position.within element

element mode

page []

```
page(element) -> [Number, Number]
```

element

positionedOffset []

```
positionedOffset(element) -> [Number, Number]
```

element Containing Block
element

CSS position relative absolute
Position.offsetParent(element)

prepare []

```
prepare()
```

Position.withinIncludingScrolloffsets

realOffset []

```
realOffset(element) -> [Number, Number]
```

relativize []

```
relativize(element)
```

```
element.style.position = 'relative'
```

within []

```
within(element, x, y) -> Boolean
```

x, y

element

Position.overlap

withinIncludingScrolloffsets []

```
withinIncludingScrolloffsets(element, x, y) -> Boolean
```

`x, y` `element` `element` `Position.within` `Position.pre`
`withinIncludingScrolloffsets --> Position.overlap`

absolutize

```
absolutize(element)
```

```
element.style.position = 'absolute'
```

Element#absolutize

clone

```
clone(source, target[, options]) -> [Number, Number]
```

```
source options / target
```

Element#clonePosition

CSS containing block

```
setLeft    true    source CSS    left    target
setTop     true    source CSS    top     target
setWidth   true    source CSS    width   target
setHeight  true    source CSS    height  target
           target CSS    left
offsetLeft 0       target CSS    left    source left
           offsetLeft
           target CSS    top
offsetTop  0       target CSS    top     source top
           offsetTop
```

```
options hash          target width height options
{setWidth:false, setWidth:false}
```

cumulativeOffset

```
cumulativeOffset(element) -> [Number, Number]
```

```
element
```

Element#cumulativeOffset

offsetLeft offsetTop

offsetParent

offsetParent(element) -> HTMLElement

| | | | | | |
|---------|------------------|-----|----------|----------|----------|
| element | Containing Block | CSS | position | relative | absolute |
| body | | | | | |

Element#getOffsetParent

element CSS containing block

overlap

```
overlap(mode, element) -> Number
```

```
    0  1
horizontal      Position.within element element m
```

```
element      x, y      Position.within element
```

```
element      x, yPosition.overlap
```

Position.within

```
var element = $('some_positioned_element');
Position.cumulativeOffset(element);
// -> [100, 100] 100px
element.getDimensions();
// -> { width: 150, height: 150 }

Position.within(element, 175, 145);
// -> true
Position.overlap('horizontal', element);
// -> 0.5
Position.overlap('vertical', element);
// -> 0.3 3/10
```

page

```
page(element) -> [Number, Number]
```

```
  element
```

Element#viewportOffset

positionedOffset

positionedOffset(element) -> [Number, Number]

| | | | | | |
|---------|------------------|-----|----------|----------|--------------------------------|
| element | Containing Block | CSS | position | relative | absolute |
| element | | | | | Position.offsetParent(element) |

Element#positionedOffset

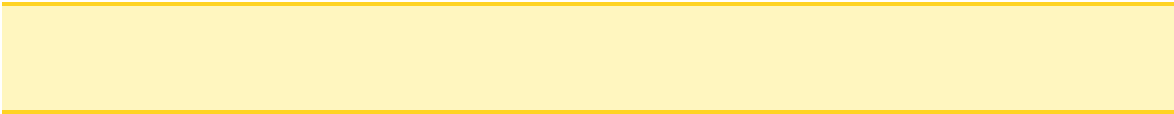
offsetLeft offsetTop CSS

position

prepare

prepare()

Position.withinIncludingScrolloffsets



realOffset

```
realOffset(element) -> [Number, Number]
```

Element#cumulativeScrollOffset

`scrollLeft` `scrollTop`

relativize

```
relativize(element)
```

```
element.style.position = 'relative'
```

Element#relativize

within

```
within(element, x, y) -> Boolean
```

x, y

element

Position.overlap

Position.cumulativeOffset element

e

(x, y)

element

true

```
var element = $('some_positioned_element');
Position.cumulativeOffset(element);
// -> [100, 100] 100px
Element.getDimensions(element);
// -> { width: 150, height: 150 }

Position.within(element, 200, 200);
// -> true
Position.within(element, 260, 260);
// -> false
```

withinIncludingScrolloffsets

```
withinIncludingScrolloffsets(element, x, y) -> Boolean
  x, y      element      element  Position.within  Position
--> withinIncludingScrolloffsets --> Position.overlap
```



Position.within element Script.aculo.us

Position.prepare Position.prepare

Prototype

K, emptyFunction

Prototype Prototype

“”

Prototype

Prototype

Prototype.Version Prototype

script.aculo.us

Prototype

Prototype Prototype

DOM Level 3 XPath Prototype.Brows

boolean

Prototype

E

Prototype.K

K

```
K(argument) -> argument
```

K Prototype

emptyFunction

```
emptyFunction([argument...])
```

`emptyFunction`

emptyFunction

```
emptyFunction([argument...])
```

```
emptyFunction
```

K

```
K(argument) -> argument
```

K **Prototype**

Enumerable

```
Prototype.K('hello world!');  
// -> 'hello world!'  
Prototype.K(1.5);  
// -> 1.5  
Prototype.K(Prototype.K);  
// -> Prototype.K
```

String

blank, camelize, capitalize, dasherize, empty, endsWith, escapeHTML, evalJSON, evalScripts, extractScripts, gsub, include, inspect, interpolate, isJSON, parseQuery, scan, startsWith, strip, stripScripts, stripTags, sub, succ, times, toArray, toJSON, toQueryParams, truncate, underscore, unescapeHTML, unfilterJSON

Prototype `String.prototype` `String`
`String#strip` `replace` `String#sub` `String#gsub`
 `String#toQueryParams`

blank 1.5.1

```
blank() -> Boolean
```

camelize

```
camelize() -> string
```

Camel `'foo-bar'` `'fooBar'`

capitalize

```
capitalize() -> string
```

dasherize

```
dasherize() -> string
```

`"_"` `"-"`

empty 1.5.1

```
empty() -> Boolean
```

endsWith 1.5.1

```
endsWith(substring) -> Boolean
```

substring

escapeHTML

```
escapeHTML() -> string
```

HTML

evalJSON 1.5.1

```
evalJSON([sanitize = false]) -> object
```

JSON

sanitize

true

evalScripts

```
evalScripts() -> [returnedValue...]
```

script

script

extractScripts

```
extractScripts() -> [script...]
```

script

gsub

```
gsub(pattern, replacement) -> string
```

pattern

replacement

replacement

Template

pattern

include 1.5.1

```
include(substring) -> Boolean
```

substring

inspect

```
inspect([useDoubleQuotes = false]) -> String
```

'\'

interpolate 1.6

```
interpolate(object[, pattern]) -> string
```

object

isJSON 1.5.1.1

```
isJSON() -> boolean
```

JSON

parseQuery

toQueryParams

scan

```
scan(pattern, iterator) -> string
```

pattern

startsWith 1.5.1

```
startsWith(substring) -> Boolean
```

substring

strip

```
strip() -> string
```

stripScripts

```
stripScripts() -> string
```

HTML `script`

stripTags

```
stripTags() -> string
```

HTML

sub

```
sub(pattern, replacement[, count = 1]) -> string
```

`count` `pattern` `replacement` `replacement` **Template** `pattern`

succ

```
succ() -> string
```

ObjectRange Unicode

times 1.5.1

```
times(count) -> string
```

`count`

toArray

```
toArray() -> [character...]
```

toJSON 1.5.1

```
toJSON() -> String
```

JSON

toQueryParams

```
toQueryParams([separator = '&']) -> Object
```

URI `"/"`

truncate

```
truncate([length = 30[, suffix = '...']]) -> string
```

underscore

```
underscore() -> string
```

Camel `"_"`

unescapeHTML

```
unescapeHTML() -> string
```

HTML

unfilterJSON 1..5.1

```
unfilterJSON([filter = Prototype.JSONFilter]) -> String
```

Ajax JSON JavaScript

blank

1.5.1

`blank() -> Boolean`

```
'' .blank(); //-> true
' ' .blank(); //-> true
' a ' .blank(); //-> false
```

camelize

```
camelize() -> string
```

```
Camel      'foo-bar'  'fooBar'
```

Prototype CSS DOM style

```
'background-color'.camelize();  
// -> 'backgroundColor'  
'-moz-binding'.camelize();  
// -> 'MozBinding'
```

capitalize

```
capitalize() -> string
```

```
'hello'.capitalize();  
// -> 'Hello'  
'HELLO WORLD!'.capitalize();  
// -> 'Hello world!'
```

dasherize

```
dasherize() -> string
```

```
"_"  "-"
```

```
'border_bottom_width'.dasherize();  
// -> 'border-bottom-width'
```

DOM

style CSS

String#underscore

```
'borderBottomWidth'.underscore().dasherize();  
// -> 'border-bottom-width'
```

empty

1.5.1

```
empty() -> Boolean
```

```
''.empty(); //-> true  
' '.empty(); //-> false
```

endsWith 1.5.1

endsWith(substring) -> Boolean

substring

```
'slaughter'.endsWith('laughter')  
// -> true
```

escapeHTML

```
escapeHTML() -> string
```

```
HTML
```

```
'<div class="article">This is an article</div>'.escapeHTML();  
// -> "&lt;div class="article"&gt;This is an article&lt;/div&gt;"
```


evalJSON 1.5.1

```
evalJSON([sanitize = false]) -> object
```

JSON

sanitize

true

JSON

SyntaxError

```
var person = '{ "name": "Violet", "occupation": "character" }'.eval
person.name;
//-> "Violet"

person = 'grabUserPassword()'.evalJSON(true);
//-> SyntaxError: Badly formed JSON string: 'grabUserPassword()'
```

```
person = '/*-secure-\n{"name": "Violet", "occupation": "character"}
person.name;
//-> "Violet"
```

sanitize true XSS

String#evalJSON String#unfilterJSON

Prototype.JSONFilter

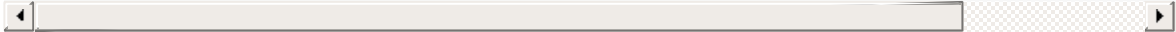
evalScripts

```
evalScripts() -> [returnedValue...]
```

script

script

```
'lorem... <script>2 + 2</script>'.evalScripts();  
// -> [4]  
'<script>2 + 2</script><script>alert("hello world!")</script>'.eval  
// -> [4, undefined] 'hello world!'
```



extractScripts

```
extractScripts() -> [script...]
```

```
script
```

String#evalScripts

```
'lorem... <script>2 + 2</script>'.extractScripts();  
// -> ['2 + 2']  
'<script>2 + 2</script><script>alert("hello world!")</script>'.extr  
// -> ['2 + 2', 'alert("hello world!")']
```

```
var myScripts = '<script>2 + 2</script><script>alert("hello world!"  
// -> ['2 + 2', 'alert("hello world!")']  
  
var myReturnedValues = myScripts.map(function(script) {  
    return eval(script);  
});  
// -> [4, undefined] 'hello world!'
```

gsub

```
gsub(pattern, replacement) -> string
```

pattern replacement replacement **Template** pattern

replacement String#gsub JavaScript replace()

```
var mouseEvents = 'click dblclick mousedown mouseup mouseover mousemove';
mouseEvents.gsub(' ', ', ');
// -> 'click, dblclick, mousedown, mouseup, mouseover, mousemove, mousemove';
mouseEvents.gsub(/\s+/, ', ');
// -> 'click, dblclick, mousedown, mouseup, mouseover, mousemove, mousemove';
```

replacement

```
mouseEvents.gsub(/\w+/, function(match){
    return 'on' + match[0].capitalize();
});
// -> 'onClick onDbldclick onMouseDown onMouseup onMouseover onMousemove';

var markdown = '![a pear](/img/pear.jpg) ![an orange](/img/orange.jpg)';
markdown.gsub(/!\[([.*?])\]\([.*?]\)/, function(match){
    return '';
});
// -> ' ';
```

Template Ruby

match()

```
markdown.gsub(/!\[([.*?])\]\([.*?]\)/, '');
// -> ' ';
```

String#gsub

String#sub

"g"

include 1.5.1

`include(substring) -> Boolean`

`substring`

```
'Prototype framework'.include('frame');  
//-> true  
'Prototype framework'.include('frameset');  
//-> false
```

inspect

```
inspect([useDoubleQuotes = false]) -> String
```

```
`
```

`inspect`

`Object.inspect`

```
'I\'m so happy.'.inspect();  
// -> '\`I\\\'m so happy.\`' 'I\'m so happy.'  
'I\'m so happy.'.inspect(true);  
// -> '"I'm so happy."' "I'm so happy."
```

interpolate 1.6

```
interpolate(object[, pattern]) -> string
```

object

Template evaluate

```
"#{animals} on a #{transport}".interpolate({ animals:"Pigs", transport:"Surfboard"})  
// -> "Pigs on a Surfboard"
```

```
var syntax = /(^\|.\|r|\n)(\<%=s*(\w+)\s*%\>)/;  
// '<%= field %>'  
var html = '<div>Name: <b><%= name %></b>, Age: <b><%=age%></b></div>'  
html.interpolate({ name: 'John Smith', age: 26 }, syntax);  
// -> <div>Name: <b>John Smith</b>, Age: <b>26</b></div>
```


isJSON

1.5.1.1

isJSON() -> boolean

JSON

```
"something".isJSON()  
// -> false  
"\something\".isJSON()  
// -> true  
"{ foo: 42 }".isJSON()  
// -> false  
"{ \"foo\": 42 }".isJSON()  
// -> true
```

parseQuery

toQueryParams

scan

```
scan(pattern, iterator) -> string
```

pattern

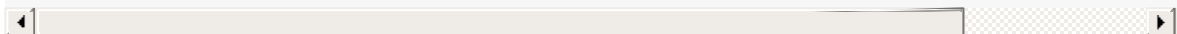
pattern iterator String#gsub

```
'apple, pear & orange'.scan(/\w+/, alert);  
// -> 'apple pear orange' 'apple' 'pear' 'orange'
```

```
var fruits = [];  
'apple, pear & orange'.scan(/\w+/, function(match){  
  fruits.push(match[0])  
});  
fruits.inspect()  
// -> ['apple', 'pear', 'orange']
```

DOM

```
'failure-message, success-message & spinner'.scan(/(\w|-)+/, Element)  
// -> 'failure-message, success-message & spinner'
```



"g"

startsWith 1.5.1

startsWith(substring) -> Boolean

substring

```
'Prototype JavaScript'.startsWith('Pro');  
//-> true
```

strip

```
strip() -> string
```

```
' hello world! '.strip();  
// -> 'hello world!'
```

stripScripts

```
stripScripts() -> string
```

HTML **script**

```
'a <a href="#">link</a><script>alert("hello world!")</script>'.stri  
// -> 'a <a href="#">link</a>'
```



stripTags

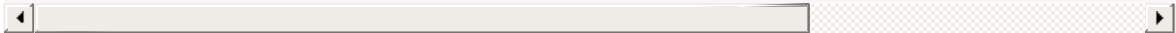
```
stripTags() -> string
```

HTML

`<script>` `String#stripTags`

`String#stripScripts`

```
'a <a href="#">link</a>'.stripTags();  
// -> 'a link'  
  
'a <a href="#">link</a><script>alert("hello world!")</script>'.stri  
// -> 'a linkalert("hello world!")'  
  
'a <a href="#">link</a><script>alert("hello world!")</script>'.stri  
// -> 'a link'
```



sub

```
sub(pattern, replacement[, count = 1]) -> string
```

| count | pattern | replacement | replacement | Template | pattern |
|-------|---------|-------------|-------------|----------|---------|
|-------|---------|-------------|-------------|----------|---------|

String#gsub

String#sub

1

String#sub

String#gsub

String#gsub

```
var fruits = 'apple pear orange';
fruits.sub(' ', ' ', ' ');
// -> 'apple, pear orange'
fruits.sub(' ', ' ', ' ', 1);
// -> 'apple, pear orange'
fruits.sub(' ', ' ', ' ', 2);
// -> 'apple, pear, orange'

fruits.sub(/\w+/, function(match){
  return match[0].capitalize() + ', '
}, 2);
// -> 'Apple, Pear, orange'

var markdown = '![a pear](/img/pear.jpg) ![an orange](/img/orange.jpg)';
markdown.sub(/!\[([.*?])\]\([.*?]\)/, function(match){
  return '';
});
// -> ' ![an orange](/img/orange.jpg)'

markdown.sub(/!\[([.*?])\]\([.*?]\)/, '');
// -> ' ![an orange](/img/orange.jpg)'
```

"g"

SUCC

```
succ() -> string
```

ObjectRange Unicode

```
'a'.succ(); // -> 'b'  
'aaaa'.succ(); // -> 'aaab'
```

times

1.5.1

```
times(count) -> string
```

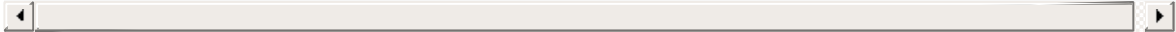
count

```
"echo ".times(3); //-> "echo echo echo "
```

toArray

toArray() -> [character...]

```
'a'.toArray();  
// -> ['a']  
'hello world!'.toArray();  
// -> ['h', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd', '!']
```



toJson

1.5.1

toJson() -> String

JSON

```
'The "Quoted" chronicles'.toJson();  
//-> '"The \"Quoted\" chronicles"'
```

toQueryParams

```
toQueryParams([separator = '&']) -> Object
```

URI "/"

separator

"&"

"#" "/"

URL "#"

http://xxx/News/Content.as

id=100#first JavaScript

location.hash URL "#"

String#toQueryParams

undefined

```
'section=blog&id=45'.toQueryParams();  
// -> {section: 'blog', id: '45'}  
  
'section=blog;id=45'.toQueryParams();  
// -> {section: 'blog', id: '45'}  
  
'http://www.example.com?section=blog&id=45#comments'.toQueryParams  
// -> {section: 'blog', id: '45'}  
  
'section=blog&tag=javascript&tag=prototype&tag=doc'.toQueryParam  
// -> {section: 'blog', tag: ['javascript', 'prototype', 'doc']}  
  
'tag=ruby%20on%20rails'.toQueryParams();  
// -> {tag: 'ruby on rails'}  
  
'id=45&raw;'.toQueryParams();  
// -> {id: '45', raw: undefined}
```

truncate

```
truncate([length = 30[, suffix = '...']]) -> string
```

String#truncate

30 "..."

String#truncate

```
'A random sentence whose length exceeds 30 characters.'.truncate();  
// -> 'A random sentence whose len...'  
'Some random text'.truncate();  
// -> 'Some random text.'  
'Some random text'.truncate(10);  
// -> 'Some ra...'  
'Some random text'.truncate(10, ' [...]');  
// -> 'Some [...]'
```

```
'String'.truncate(6);  
// -> 'String'
```

underscore

```
underscore() -> string
```

Camel **"_"**

```
'borderBottomWidth'.underscore();  
// -> 'border_bottom_width'
```

DOM

style CSS

String#dasherize

```
'borderBottomWidth'.underscore().dasherize();  
// -> 'border-bottom-width'
```

unescapeHTML

```
unescapeHTML() -> string
```

```
HTML
```

```
'x &gt; 10'.unescapeHTML()  
// -> 'x > 10'  
'<h1>Pride &amp; Prejudice</h1>'.unescapeHTML()  
// -> 'Pride & Prejudice'
```


unfilterJSON 1..5.1

```
unfilterJSON([filter = Prototype.JSONFilter]) -> String
```

Ajax JSON JavaScript

```
'/*-secure-\n{"name": "Violet", "occupation": "character", "age": 25}\n// -> '{"name": "Violet", "occupation": "character", "age": 25}'
```



Template

evaluate

Template Ruby

`#{fieldName}`

`{fieldName}`

```
//  
var myTemplate = new Template('The TV show #{title} was created by  
//  
var show = {title: 'The Simpsons', author: 'Matt Groening', network  
//  
myTemplate.evaluate(show);  
// -> The TV show The Simpsons was created by Matt Groening.
```

```
//  
var conversion1 = {from: 'meters', to: 'feet', factor: 3.28};  
var conversion2 = {from: 'kilojoules', to: 'BTUs', factor: 0.9478};  
var conversion3 = {from: 'megabytes', to: 'gigabytes', factor: 1024  
  
//  
var templ = new Template('Multiply by #{factor} to convert from #{f  
  
//  
[conversion1, conversion2, conversion3].each( function(conv){  
    templ.evaluate(conv);  
});  
// -> Multiply by 3.28 to convert from meters to feet.  
// -> Multiply by 0.9478 to convert from kilojoules to BTUs.  
// -> Multiply by 1024 to convert from megabytes to gigabytes.
```

```
// JavaScript
var t = new Template('in #{lang} we also use the \#{variable} synt
var data = {lang: 'Ruby', variable: '(not used)'};
t.evaluate(data);
// -> in Ruby we also use the #{variable} syntax for templates.

//
//  "#"  "#anchor"
var temp = new Template('http://#{site}/#{page}?#{queryString}#anch
temp.evaluate({site: 'www.xxx.com', page: 'test.htm', queryString:
// -> 'http://www.xxx.com/test.htm?sn=110#anchor'
```

Ruby

Template

```
var syntax = /(^\|.|\r|\n)(\<%=s*(\w+)\s*%\>)/;
// '<%= field %>'
var t = new Template('<div>Name: <b><%= name %></b>, Age: <b><%=age
t.evaluate( {name: 'John Smith', age: 26} );
// -> <div>Name: <b>John Smith</b>, Age: <b>26</b></div>
```

evaluate

```
evaluate(object) -> String
```

object object

evaluate

```
evaluate(object) -> String
```

object

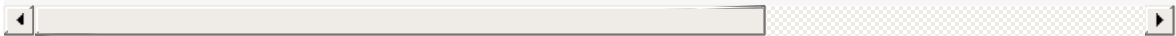
object

```
var hrefTemplate = new Template('/dir/showAll?lang=#{language}&cate
var selection = {category: 'books' , language: 'en-US'};
hrefTemplate.evaluate(selection);
// -> '/dir/showAll?lang=en-US&categ=books&lv='

hrefTemplate.evaluate({language: 'jp', levels: 3, created: '10/12/2
// -> '/dir/showAll?lang=jp&categ=&lv=3'

hrefTemplate.evaluate({});
// -> '/dir/showAll?lang=&categ=&lv='

hrefTemplate.evaluate(null);
// ->
```



TimedObserver

[Form.Element.Observer](#), [Form.Observer](#)

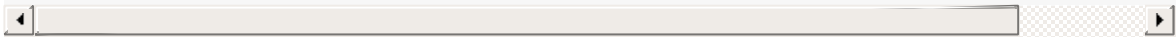
TimedObserver

TimedObserver `getValue()`

`Form.Element.Observer`

TimedObserver

```
new Form.Element.Observer( 'myelement', 0.2, // 200
    function(el, value){
        alert('The form control has changed value to: ' + v
    }
)
```



TimedObserver 0.2

Form.Element.Observer

```
new Form.Element.Observer(element, frequency, callback)
```

Form.Observer

```
new Form.Observer(element, frequency, callback)
```

name

Form.Element.Observer

```
new Form.Element.Observer(element, frequency, callback)
```

Form.Element.Observer

Form.Element.getValue()

getValue()

Abstract.TimedObserver

Form.Observer

```
new Form.Observer(element, frequency, callback)
```

name

Form.Observer Form.serialize() getValue

Abstract.TimedObserver

Login Preferences

Current settings:

Greeting message

Hello world!

Login options

- ☒ allow others to see my last login date
- ☐ land on recent changes overview instead of the Dashboard

save

JavaScript

```
new Form.Observer('example', 0.3, function(form, value){
    $('msg').update('Your preferences have changed. Resubmit to
    form.down().setStyle({ background:'lemonchiffon', borderCol
})
$('example').onsubmit = function() {
    $('msg').update('Preferences saved!').style.color = 'green'
    this.down().setStyle({ background:'', borderColor:'' })
}
```

```
    return false  
}
```

```
this.down().setStyle({ background:'', borderColor:'' });
```

```
this.down().setStyle({ background:null, borderColor:null });
```

IE IE

style null Firefox

fire

1.6.0

```
fire(eventName[, memo]) -> Event
```

```
document
```

document.fire **Element#fire**

observe 1.6.0

```
observe(eventName, handler) -> document
      "dom:loaded"
```

document.observe **Element#observe**

Event.observe(document, eventName, handler)

"dom:loaded"

Prototype DOM

DOMContentLoaded

Prototype

DOMContentLoaded

"dom.loaded"

DOMContentLoaded Prototype DOM

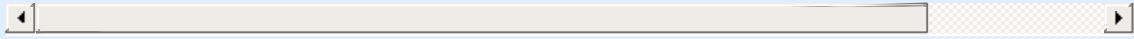
window.onload

dom:loaded HTML

```
document.observe("dom:loaded", function() {
  // initially hide all containers for tab content
  $$('div.tabcontent').invoke('hide');
});
```

stopObserving 1.6.0

```
document.stopObserving(eventName, handler[, useCapture = false])
```



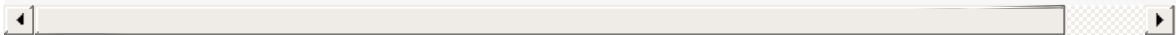
`document.stopObserving` `Element#stopObserving`

document.viewport

getDimensions, getHeight, getScrollOffsets, getWidth

getDimensions 1.6.0

```
document.viewport.getDimensions() -> { width: Number, height: Number }
```

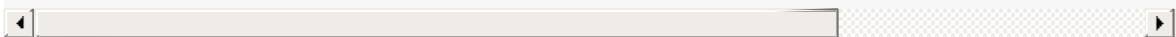


getHeight 1.6.0

```
document.viewport.getHeight() -> Number
```

getScrollOffsets 1.6.0

```
document.viewport.getScrollOffsets() -> [Number, Number] { top: Number, left: Number }
```

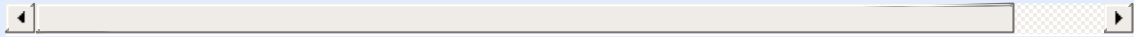


getWidth 1.6.0

```
document.viewport.getWidth() -> Number
```

getDimensions 1.6.0

```
document.viewport.getDimensions() -> { width: Number, height: Num
```



viewport —

```
document.viewport.getDimensions();  
//-> { width: 776, height: 580 }
```

getHeight

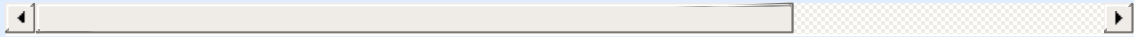
1.6.0

```
document.viewport.getHeight() -> Number
```

```
document.viewport.getDimensions().height
```

getScrollOffsets 1.6.0

```
document.viewport.getScrollOffsets() -> [Number, Number] { top: |
```



```
document.viewport.getScrollOffsets();  
//-> { left: 0, top: 0 }  
  
window.scrollTo(0, 120);  
document.viewport.getScrollOffsets();  
//-> { left: 0, top: 120 }
```

getWidth

1.6.0

```
document.viewport.getWidth() -> Number
```

```
document.viewport.getDimensions().width
```


Same Origin Policy

JavaScript“origin”

Netscape Navigator2.0 Navigator2.01 Navigator2.02
“origin”

URL

<http://www.example.com/dir2/other.html>

<http://www.example.com/dir/inner/other.html>

<https://www.example.com/dir2/other.html>

<http://en.example.com/dir2/other.html>

<http://example.com/dir2/other.html>

<http://www.example.com:81/dir2/other.html>

XMLHttp

XMLHttp

XMLHttpRequest.status

| | |
|-----|-------------------------------|
| 100 | Continue |
| 101 | Switching protocols |
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 203 | Non-Authoritative Information |
| 204 | No Content |
| 205 | Reset Content |
| 206 | Partial Content |
| 300 | Multiple Choices |
| 301 | Moved Permanently |
| 302 | Found |
| 303 | See Other |
| 304 | Not Modified |
| 305 | Use Proxy |
| 307 | Temporary Redirect |
| 400 | Bad Request |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |

| | |
|-----|-------------------------------|
| 407 | Proxy Authentication Required |
| 408 | Request Timeout |
| 409 | Conflict |
| 410 | Gone |
| 411 | Length Required |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 414 | Request-URI Too Long |
| 415 | Unsupported Media Type |
| 416 | Requested Range Not Suitable |
| 417 | Expectation Failed |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 502 | Bad Gateway |
| 503 | Service Unavailable |
| 504 | Gateway Timeout |
| 505 | HTTP Version Not Supported |

XMLHttpRequestStatus

| | |
|---|--------------------------------------|
| 0 | () open |
| 1 | () send |
| 2 | () send http |
| 3 | () http responseBody
responseText |
| 4 | () , responseBody responseBodyText |

\$super

Prototype

Prototype \$super

Prototype

\$super

```
var baseClass = Class.create({
  initialize: function(name){
    this.name = name;
  },
  show: function(msg){
    alert(msg + ' ' + this.name);
  }
});

var subClass = Class.create(baseClass, {
  show: function($super, msg){
    $super(msg + '\n');
  }
});

var subObj = new subClass('ORain');
subObj.show('Hello, World');
```

show \$super Prototype

setValue

```
setValue(element, value) -> HTMLElement
```

`value`

Prototype

```
$('#inputid').setValue('.....')  
//  
Form.Element.setValue('inputid', '.....')  
  
// SELECT  
$('#multiSelect').setValue(['', ''])
```

`element` ID"element has no properties"