zepto.js

the aerogel-weight
mobile javascript framework

hello.

hello.

Why not use Prototype, jQuery, etc.

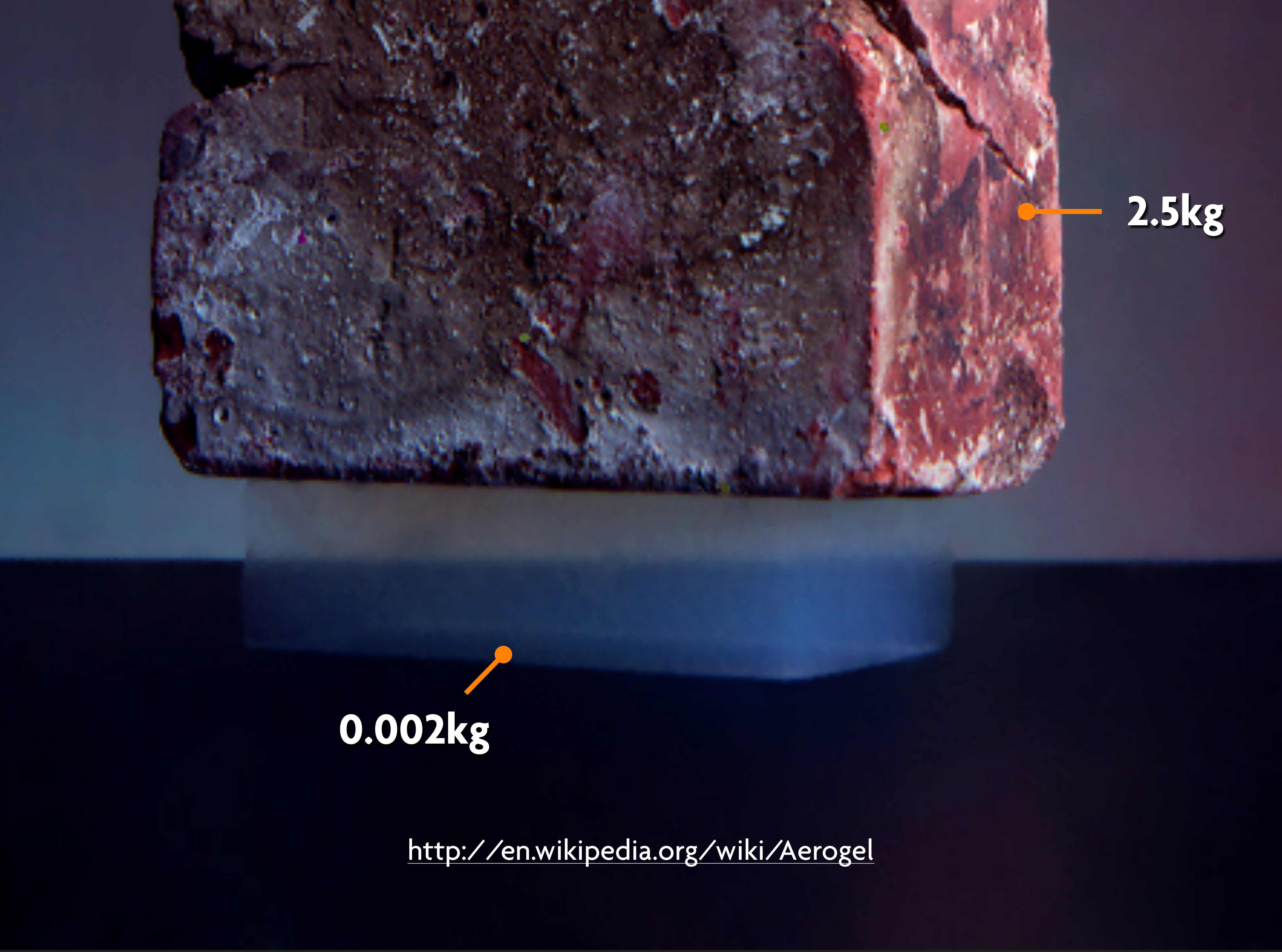
- Lots of code = long time to download
- Caching on mobile devices is not so great
- Most code is for browsers that you don’t need to support (IE6 doesn’t run on an iPhone!)
- Natively supported features are duplicated, for example JSON support and animations

Goals for a mobile JavaScript framework

- ✓ Very small codebase
- ✓ Easy to use API for common DOM tasks
- ✓ Easy to extend & customize
- ✓ No need to deal with browser cruft (IE6, etc)
- ✓ Inlineable

zepto.JS

the aerogel-weight
mobile javascript framework



http://en.wikipedia.org/wiki/Aerogel

zepto.JS

the aerogel-weight
mobile javascript framework

✓ Size: ~2K

- ✓ jQuery-compatible API
- ✓ Uses mobile WebKit features as much as possible
- ✓ Easily replaceable with larger libs if your app grows
- ✓ Open source (MIT license)

```
$('p').html('test').css('color:red');
```

```
get(): return array of all elements found
get(0): return first element found
each(callback): iterate over array of all elements found
index('selector'): return an integer indicating the position of 'selector' in array of all elements found
first(): remove all but the first element from the list of found elements

find('selector'): find all children/grandchildren that match the given selector
closest('selector'): traverses the DOM upwards to find the first matching element
next(): next siblings
prev(): previous siblings
is('selector'): returns true/false if first element matches the selector

remove(): remove element

html('new html'): set the contents of the element(s)
append, prepend, before, after: like html, but add html to element contents (or before/after)
html(): get first elements .innerHTML
show(): forces elements to be displayed (only works correctly for block elements right now)
hide(): removes a elements from layout

offset(): get object with top: left: width: height: properties (in px)
height(): get first elements height in px
width(): get first elements width in px

attr('attribute'): get element attribute
attr('attribute', 'value'): set element attribute

css('css property', 'value'): set a CSS property
css({ property1: value1, property2: value2 }): set multiple CSS properties
css('css property'): get this CSS property of the first element

addClass('classname'): adds a CSS class name
removeClass('classname'): removes a CSS class name
hasClass('classname'): returns true of first element has a classname set

bind(type, function): add an event listener (see below)
delegate(selector, type, function): add an event listener w/ event delegation (see below)
live(type, function): add an event listener that listens to the selector for current and future elements
trigger(type): triggers an event
```

```
get(): return array of all elements found
get(0): return first element found
each(callback): iterate over array of all elements found
index('selector'): return an integer indicating the position of 'selector' in array of all elements found
first(): remove all but the first element from the list of found elements

find('selector'): find all children/grandchildren that match the given selector
closest('selector'): traverses the DOM upwards to find the first matching element
next(): next siblings
prev(): previous siblings
is('selector'): returns true/false if first element matches the selector

remove(): remove element

html('new html'): set the contents of the element(s)
append, prepend, before, after: like html, but add html to element contents (or before/after)
html(): get first elements .innerHTML
show(): forces elements to be displayed (only works correctly for block elements right now)
hide(): [illegible] a elements from layout

offset(): [illegible] object with top: left: width: height: properties (in px)
height(): get first elements height in px
width(): get first elements width in px

attr('attribute'): get element attribute
attr('attribute', 'value'): set element attribute

css('css property', 'value'): set a CSS property
css({ property1: value1, property2: value2 }): set multiple CSS properties
css('css property'): get this CSS property of the first element

addClass('classname'): adds a CSS class name
removeClass('classname'): removes a CSS class name
hasClass('classname'): returns true of first element has a classname set

bind(type, function): add an event listener (see below)
delegate(selector, type, function): add an event listener w/ event delegation (see below)
live(type, function): add an event listener that listens to the selector for current and future elements
trigger(type): triggers an event
```

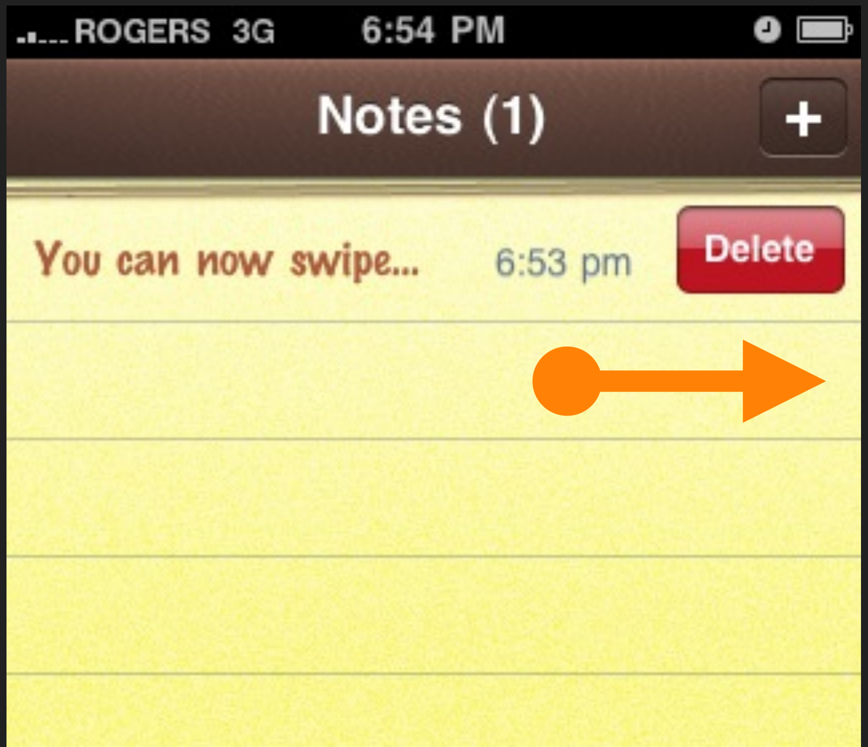
✓ **Basically, everything**

✓ Ajax

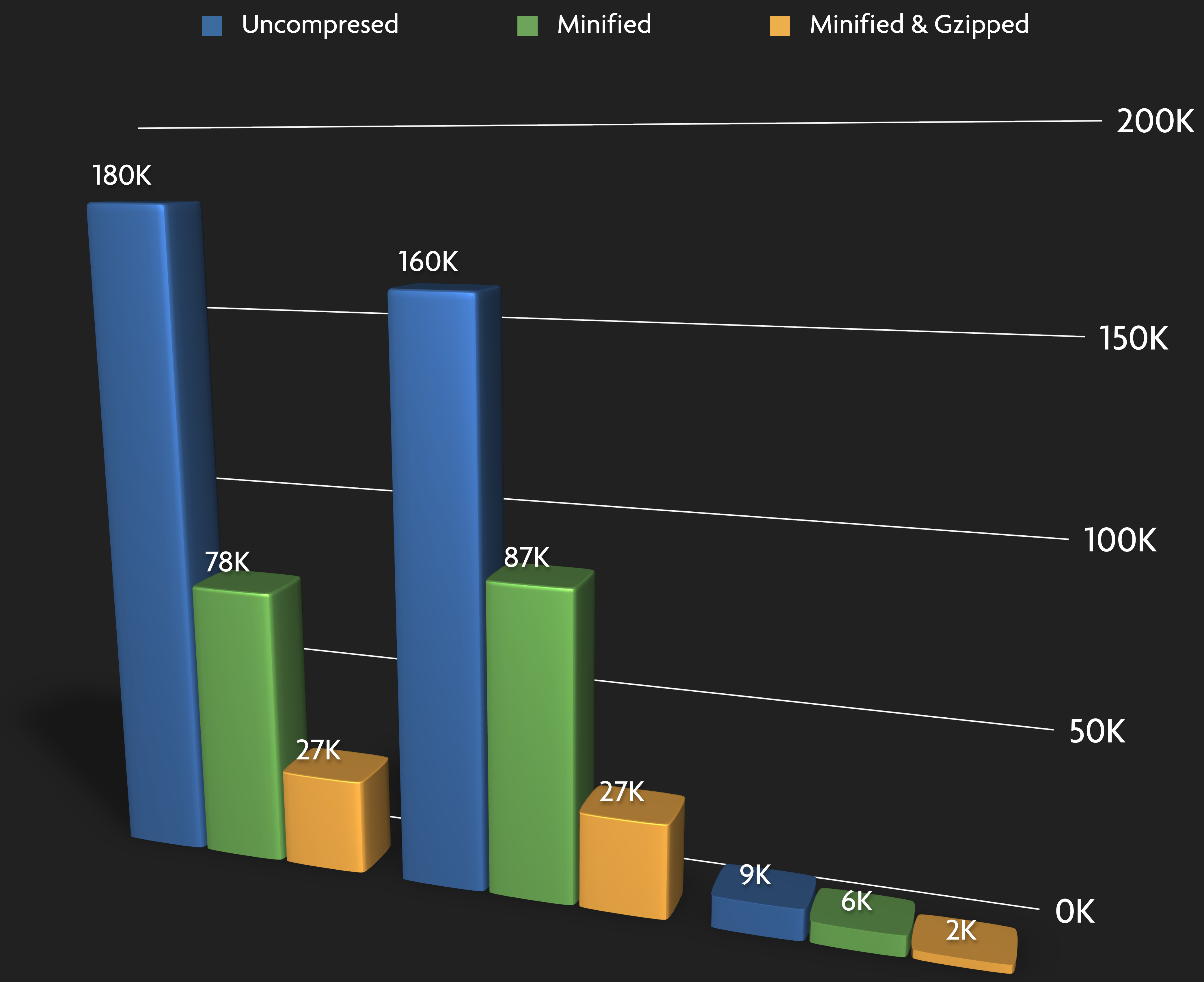
```
$.get(url, callback)
$.post(url, callback)
$.getJSON(url, callback)

$('selector').load('url'[, callback]);
$('selector').load('url #fragment-selector'[, callback]);
```

✓ Tap & Swipe

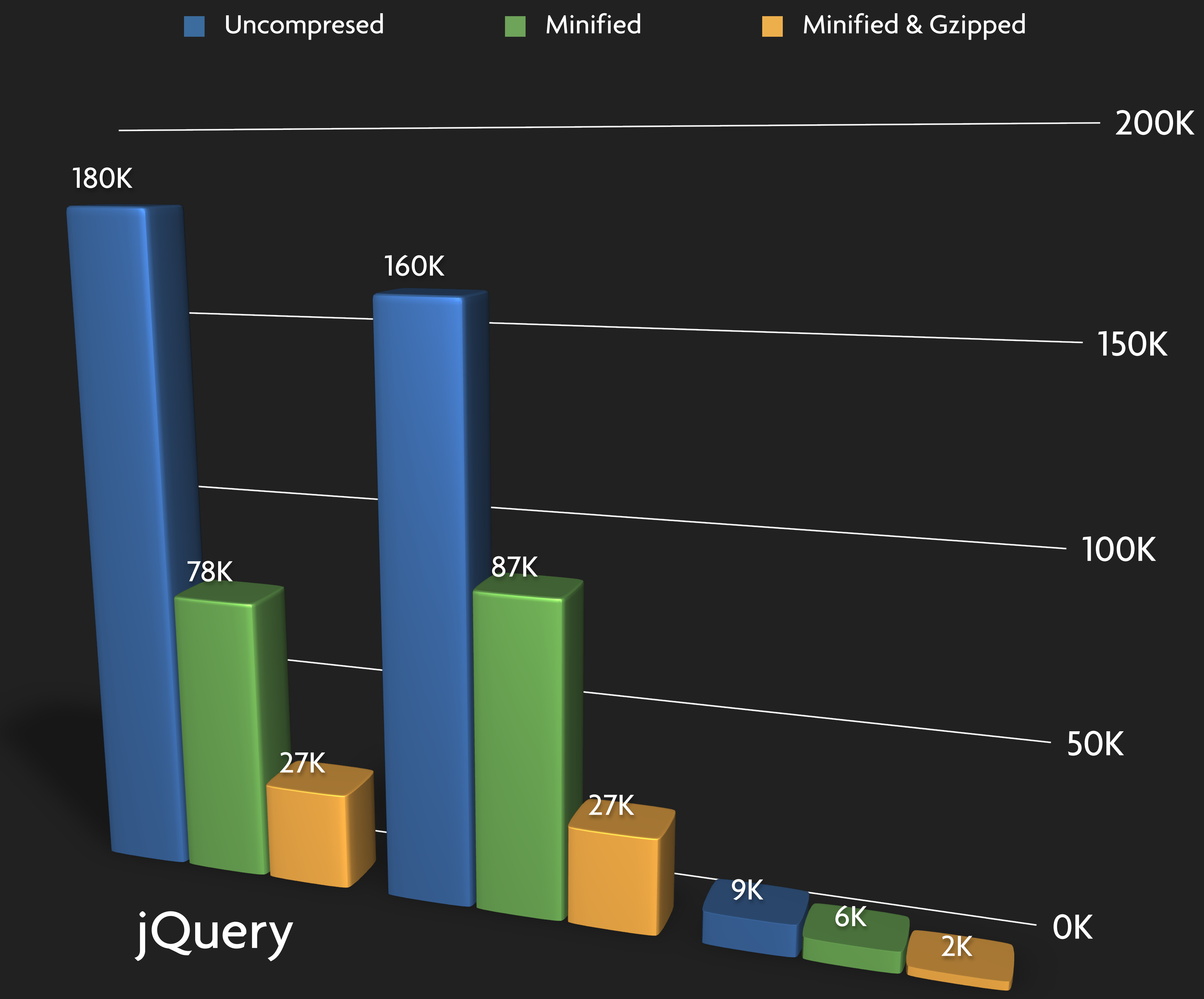
```
$('some selector').tap(function(){ ... });
$('some selector').doubleTap(function(){ ... });

$('some selector').swipe(function(){ ... });
```



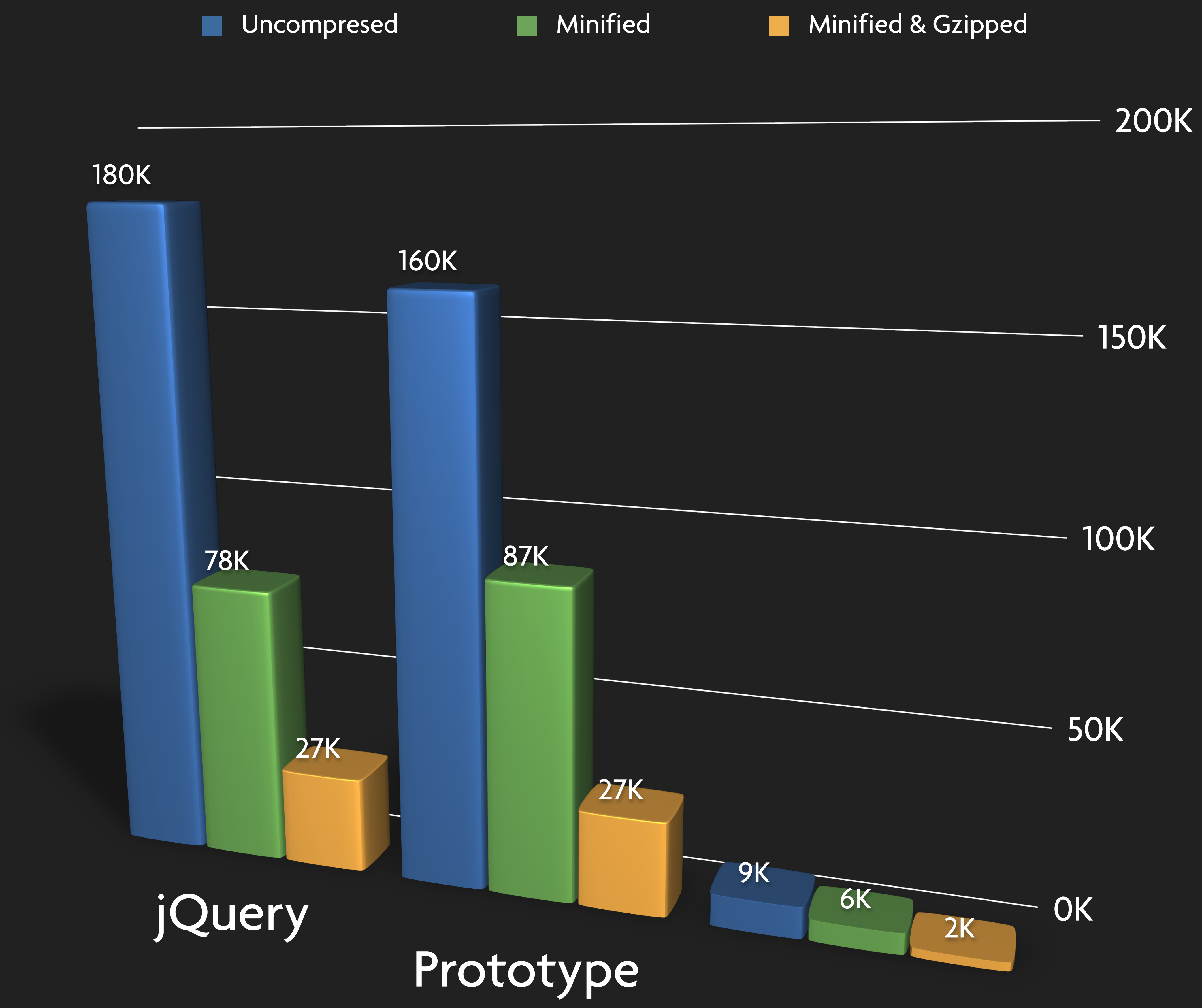
🟦 Uncompresed 🟩 Minified 🟧 Minified & Gzipped

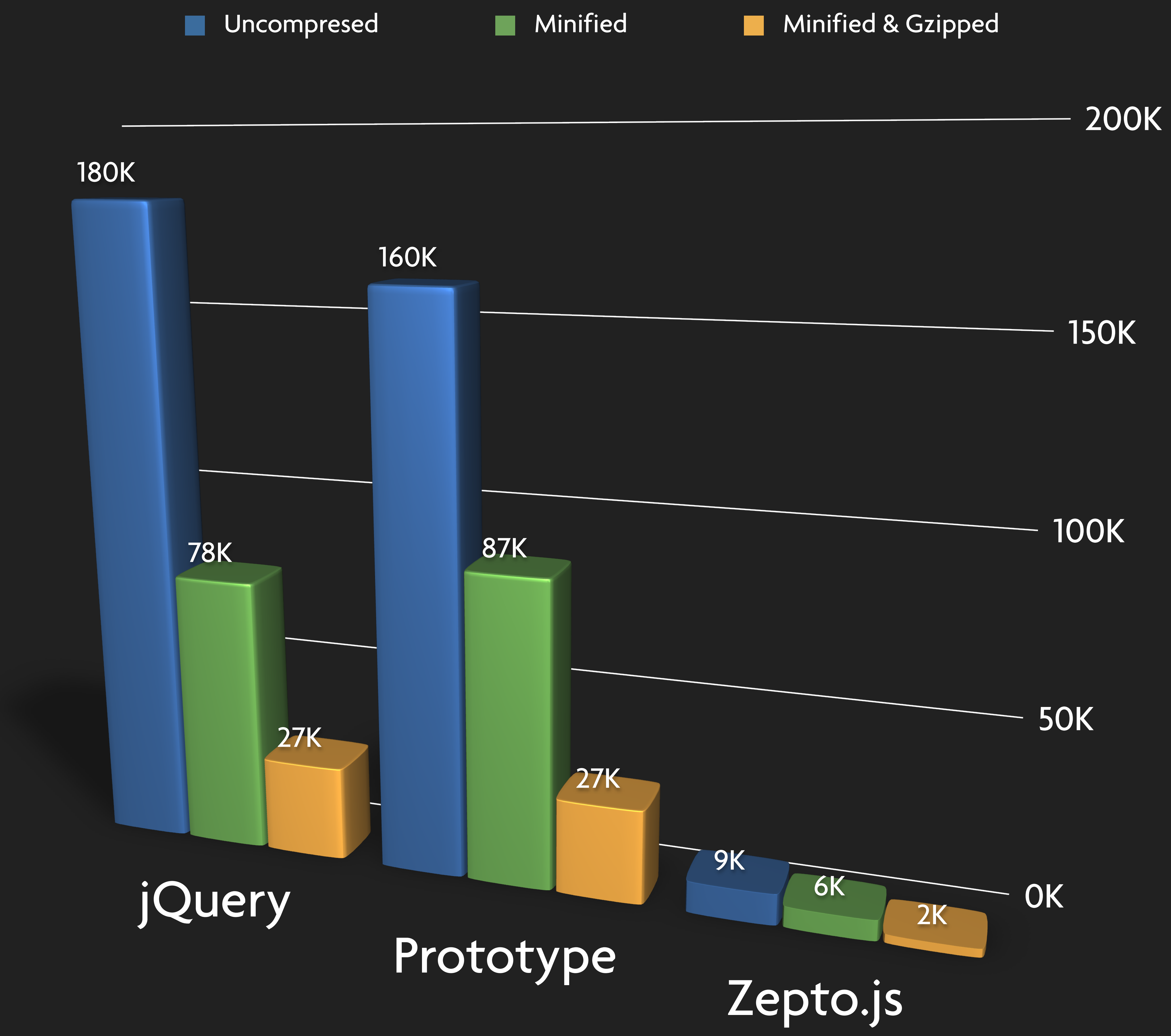


Uncompresed
Minified
Minified & Gzipped
200K
150K
100K
50K
0K
180K
78K
27K
160K
87K
27K
9K
6K
2K



Uncompresed
Minified
Minified & Gzipped
200K
150K
100K
50K
0K
180K
78K
27K
160K
87K
27K
9K
6K
2K
jQuery


Uncompresed
Minified
Minified & Gzipped
200K
150K
100K
50K
0K
180K
78K
27K
160K
87K
27K
9K
6K
2K
jQuery
Prototype



Uncompresed
Minified
Minified & Gzipped
200K
150K
100K
50K
0K
180K
78K
27K
160K
87K
27K
9K
6K
2K
jQuery
Prototype
Zepto.js

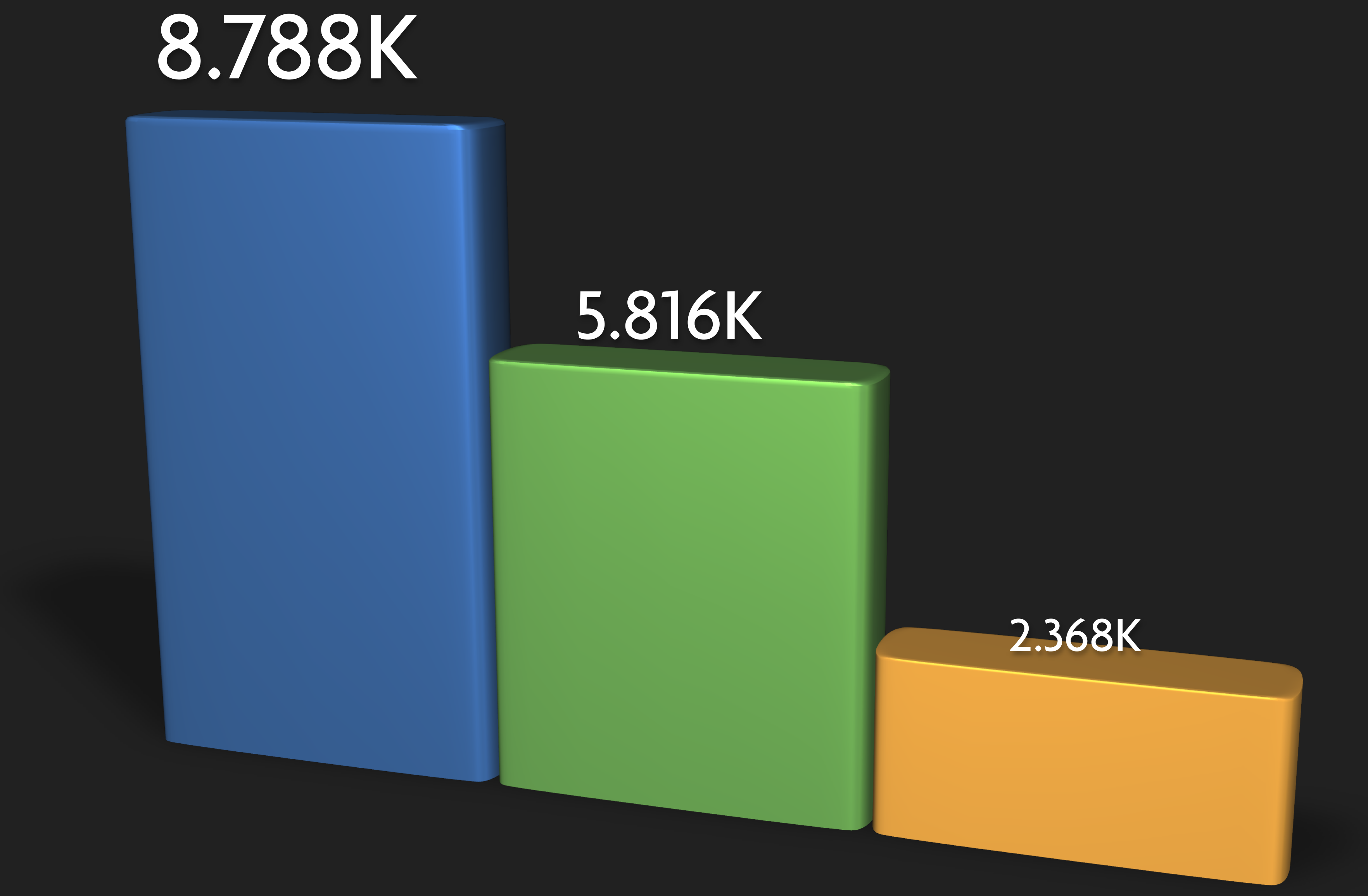
zepto.JS

the aerogel-weight
mobile javascript framework

🟦 Uncompresed 🟩 Minified 🟧 Minified & Gzipped

zepto.JS

the aerogel-weight
mobile javascript framework



🟦 Uncompresed 🟩 Minified 🟧 Minified & Gzipped

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}

$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}

$.css = function(style){
  this.dom.forEach(function(el){ el.style.cssText += ';'+style }); return this;
}
```

Core implementation (simplified)

$('some CSS selector')

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}

$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}

$.css = function(style){
  this.dom.forEach(function(el){ el.style.cssText += ';'+style }); return this;
}
```

returns a zepto.js object

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}

$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}

$.css = function(style){
  this.dom.forEach(function(el){ el.style.cssText += ';'+style }); return this;
}
```

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}

$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}

$.css = function(style){
  this.dom.forEach(function(el){ el.style.cssText += ';'+style }); return this;
}
```

html('new html') sets the contents of one or more elements

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}

$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}

$.css = function(style){
  this.dom.forEach(function(el){ el.style.cssText += ';'+style }); return this;
}
```

css('style') sets the style of one or more elements

How $() works

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}
```

How $() works

select elements on the page as per the user-specified CSS selector

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}
```

```
$('p')
```

How $() works

```
var $ = function(selector){
  return { dom: Array.prototype.slice.apply(document.querySelectorAll(selector)),
    anim: $.anim, css: $.css, html: $.html };
}
```

return a "zepto" object

```
{
  dom: [/* element 1, element 2, element 3, etc.*/],
  css: $.css, html: $.html
}
```

How a chainable function works

```
$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}
```

How a chainable function works

```
$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}
```

this.dom refers to the nodes selected by the call to $

How a chainable function works

```
$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}
```

forEach iterates over all the nodes (nodelist was converted to an array)

How a chainable function works

```
$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}
```

set the contents of the node to
some specificed html

How a chainable function works

```
$.html = function(html){
  this.dom.forEach(function(el){ el.innerHTML = html }); return this;
}
```

return the “zepto” object
for chaining

Inlining FTW

```
<!DOCTYPE html>
<html>
<head>
  <title>Zepto.js</title>
  <script>
    // stick all JS stuff in here
  </script>
</head>
<body>
  <p>
    Blah
  </p>
  <p>
    Blub
  </p>
  <script>
    $('p').html('test').css('color:red');
  </script>
</body>
</html>
```

```
function $$(el, selector){
  return slice.call(el.querySelectorAll(selector))
}

function compact(array){
  return array.filter(function(el){
    return el !== void 0 && el !== null
  })
}

function $(_, context){
  if(context !== void 0) return $(context).find(_);
  function fn(_){ return fn.dom.forEach(_), fn }
  fn.dom = compact((typeof _ == 'function' && 'dom' in _) ?
    _.dom : (_ instanceof Array ? _ :
      (_ instanceof Element ? [_] :
        $$(d, fn.selector = _))));
  $.extend(fn, $.fn);
  return fn;
}
```

zepto.JS

the aerogel-weight
mobile javascript framework

- ✓ Minimalist, jQuery-ish framework
- ✓ Use WebKit features as much as possible
- ✓ Only targets mobile browsers
- ✓ Can be “upgraded”
- ✓ Inlineable

http://zeptojs.com/