Most popular (last 30 days)

ECMAScript 2017: the final feature set

ECMAScript 6 modules: the final syntax

ES proposal: import() – dynamically importing ES modules

Making transpiled ES modules more spec-compliant

ES proposal: Shared memory and atomics

Classes in ECMAScript 6 (final semantics)

Communicating between Web Workers via MessageChannel

Most popular (all time)

ECMAScript 6 modules: the final syntax

Classes in ECMAScript 6 (final semantics)

Iterating over arrays and objects in JavaScript

ECMAScript 6's new array methods

The final feature set of ECMAScript 2016 (ES7)

WebAssembly: a binary format for the web

Basic JavaScript for the impatient programmer

Google Dart to "ultimately ... replace JavaScript"

Six nifty ES6 tricks

Google's Polymer and the future of web UI frameworks

Blog archive

► 2017 (4)

► 2016 (38)

Free email newsletter: "ES.next News"

2013-12-25

ECMAScript 6: the new array methods find() and findIndex()

Labels: dev, esnext, javascript

Update 2014-05-08. Newer version of this post: "ECMAScript 6's new array methods"

Two new Array methods (proposed by Rick Waldron) are in the latest ECMAScript 6 specification draft:

- `Array.prototype.find()`
- `Array.prototype.findIndex()`

This blog post describes them.

1. Array.prototype.find(predicate, thisValue?)

The signature of the callback `predicate()` is

```
predicate(element, index, array)
```

`find()` iterates over the array (while skipping holes) and calls `predicate` for each element. If `predicate` returns `true`, `find()` stops iterating and returns the current element (1). If the callback never returns `true`, `find()` returns `undefined` (2).

This method could be implemented as follows.

```
Array.prototype.find = function (predicate, thisValue) {
    var arr = Object(this);
    if (typeof predicate !== 'function') {
        throw new TypeError();
    }
    for(var i=0; i < arr.length; i++) {
        if (i in arr) {  // skip holes
            var elem = arr[i];
            if (predicate.call(thisValue, elem, i, arr)) {
                return elem;  // (1)
            }
        }
    }
    return undefined;  // (2)
}
```

2. Array.prototype.findIndex(predicate, thisValue?)

This method works like `find()`, but returns `i` in line (1) and `-1` in line (2).

3. Finding NaN via findIndex()

It's a well-known limitation of `Array.prototype.indexOf()` that it can't find `NaN`, because it searches for elements via `===` [1]:

```
> ['a', NaN, 'b'].indexOf(NaN)
-1
```

With `findIndex()`, you can use `Object.is()` [2] and will have no such problem:

(Ad, please don't block.)

90% Unlimited Downloads Choose from Over 300,000 Vectors, Graphics & Photos.

ads via Carbon

Dr. Axel Rauschmayer

Free online books by Axel

Speaking JavaScript [up to ES5]

Exploring ES6

JavaScript training:
Ecmanauten

► 2015 (65)
► 2014 (55)
▼ 2013 (98)
▼ December (6)
Printing Markdown files on GitHub
JavaScript gains support for SIMD
ECMAScript 6: the new array methods find() and fin...
ParallelJS: data parallelism for JavaScript
Using strict mode in the Node.js REPL
Why are there so many array-like objects in the DO...
► November (3)
► October (4)
► September (6)
► August (6)
► July (8)
► June (11)
► May (11)
► April (12)
► March (8)
► February (10)
► January (13)
► 2012 (177)
► 2011 (380)
► 2010 (174)
► 2009 (68)
► 2008 (46)
► 2007 (12)
► 2006 (1)
► 2005 (2)

Labels

dev (592)
javascript (400)
computers (316)
life (194)
jslang (179)
esnext (156)
apple (107)
webdev (95)
mobile (83)
scitech (50)
hack (49)
mac (47)


```
> ['a', NaN, 'b'].findIndex(y => Object.is(NaN, y))
1
```

You can also adopt a more general approach, by creating a helper function `elemIs`:

```
> function elemIs(x) { return Object.is.bind(null, x) }
> ['a', NaN, 'b'].findIndex(elemIs(NaN))
1
```

4. Availability

As you can see in Kangax' ECMAScript 6 compatibility table, recent versions of Firefox and Chrome already support the `Array.prototype.find*()` methods.

Furthermore, Paul Millr's es6-shim has backported the methods to ECMAScript 5.

5. References

[1] NaN and Infinity in JavaScript

[2] Stricter equality in JavaScript

3 Comments The 2ality blog Login

Recommend 1 Share Sort by Best

Join the discussion…

Satheesh Kumar • a year ago

I would appreciate if you could use simple words and examples and how exactly to make it work... i am really frustrated that all these geeks are writing as if they themselves are the readers and all others have the same thought process as theirs

• Reply • Share ›

Axel Rauschmayer Mod ➚ Satheesh Kumar • a year ago

Check out the new version of this blog post:
http://www.2ality.com/2014/05/...
Still difficult to understand?

1 • Reply • Share ›

Satheesh Kumar ➚ Axel Rauschmayer • a year ago

Thank You

• Reply • Share ›

Subscribe Add Disqus to your site Add Disqus Add Privacy

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Tweets by @rauschma

Axel Rauschmayer @rauschma

Enjoying "Troll Hunters":
– "Let's call him 'Gnome Chomsky'"
– "Juliet dies in this? Nooo!"

10h

Axel Rauschmayer Retweeted

Jonathan Creamer @jcreamer898

A nice little shout out to @rauschma...
infoworld.com/article/316483… #es2017 #async

JavaScr…
ECMASc…
infoworld…

13h

Axel Rauschmayer @rauschma

Not a physics book!
twitter.com/ManningBooks/s…

02 Feb

Axel Rauschmayer Retweeted

Jordan Harband @ljharb

Making our React components forbid extra props has caught SO many bugs. I highly recommend it.

npmjs.com/airbnb-prop-ty…

npm: air…
Custom …
npmjs.com

02 Feb

Axel Rauschmayer Retweeted

Seth Petry-Johnson @spetryjohnson

Open strong. Be bold. Tell a story. Do something to get me interested. Opening w/ the "obligatory 'about me' slide" puts me to sleep :)

02 Feb

Google

google (39)

java (37)

ios (33)

business (32)

video (32)

clientjs (31)

hci (27)

entertainment (26)

nodejs (26)

society (26)

browser (25)

firefox (25)

html5 (24)

ipad (24)

movie (23)

psychology (22)

2ality (18)

tv (18)

android (17)

social (17)

chrome (16)

fun (16)

jsmodules (16)

tablet (16)

humor (15)

politics (15)

web (15)

cloud (14)

hardware (14)

microsoft (14)

software engineering (13)

blogging (12)

gaming (12)

eclipse (11)

gwt (11)

numbers (11)

programming languages (11)

app store (10)

media (10)

nature (10)

security (10)

semantic web (10)

software (10)

twitter (10)

webos (10)

12quirks (9)

education (9)

jstools (9)

photo (9)

webcomponents (9)

windows 8 (9)

async (8)

idea (8)

iphone (8)

itunes (8)

scifi-fantasy (8)

app (7)

babel (7)

bookmarklet (7)

chromeos (7)

english (7)

es proposal (7)

fringe (7)

html (7)

jsint (7)

jsshell (7)

thunderbolt (7)

webapp (7)

advancedjs (6)

blogger (6)

crowdsourcing (6)

latex (6)

lion (6)

promises (6)

ted (6)

book (5)

environment (5)

gadget (5)

googleio (5)

intel (5)

jsarrays (5)

jshistory (5)

layout (5)

light peak (5)

michael j. fox (5)

music (5)

pdf (5)

polymer (5)

shell (5)

tc39 (5)

template literals (5)

underscorejs (5)

vlc (5)

__proto__ (4)

coffeescript (4)

concurrency (4)

dart (4)

facebook (4)

gimp (4)

googleplus (4)

health (4)

howto (4)

hp (4)

javafx (4)

kindle (4)

leopard (4)

macbook (4)

motorola (4)

münchen (4)

occupy (4)

pl fundamentals (4)

presenting (4)

publishing (4)

series (4)

textbook (4)

web design (4)

amazon (3)

asmjs (3)

back to the future (3)

bitwise_ops (3)

css (3)

es2016 (3)

flattr (3)

fluentconf (3)

food (3)

foreign languages (3)

house (3)

icloud (3)

info mgmt (3)

jsfuture (3)

jsstyle (3)

linux (3)

mozilla (3)

python (3)

regexp (3)

samsung (3)

tizen (3)

traffic (3)

typedjs (3)

unix (3)

adobe (2)

angry birds (2)

angularjs (2)

astronomy (2)

audio (2)

comic (2)

design (2)

dom (2)

ecommerce (2)

eval (2)

exploring es6 (2)

facebook flow (2)

facets (2)

flash (2)

free (2)

futurama (2)

guide (2)

history (2)

hyena (2)

internet explorer (2)

iteration (2)

journalism (2)

jquery (2)

jsengine (2)

jslib (2)

law (2)

lightning (2)

markdown (2)

math (2)

meego (2)

month (2)

nike (2)

nokia (2)

npm (2)

programming (2)

raffle (2)

repl (2)

servo (2)

sponsor (2)

steve jobs (2)

travel (2)

typescript (2)

usb (2)

winphone (2)

wwdc (2)

airbender (1)

amdefine (1)

aol (1)

app urls (1)

architecture (1)

atscript (1)

basic income (1)

biology (1)

blink (1)

bluetooth (1)

canada (1)

clip (1)

coding (1)

community (1)

cross-platform (1)

deutsch (1)

diaspora (1)

distributed-social-network (1)

dsl (1)

dvd (1)

dzone (1)

emacs (1)

emberjs (1)

energy (1)

esnext news (1)

esprop (1)

example (1)

facetator (1)

feedback (1)

firefly (1)

firefoxos (1)

fritzbox (1)

german (1)

git (1)

guest (1)

guice (1)

h.264 (1)

home entertainment (1)

hosting (1)

htc (1)

ical (1)

jsdom (1)

jsmyth (1)

library (1)

location (1)

marketing (1)

mars (1)

meta-data (1)

middle east (1)

mpaa (1)

msl (1)

mssurface (1)

netflix (1)

nsa (1)

obama (1)

openoffice (1)

opinion (1)

oracle (1)

organizing (1)

philosophy (1)

pixar (1)

pnacl (1)

prism (1)

privacy (1)

proxies (1)

puzzle (1)

raspberry pi (1)

read (1)

rodney (1)

rust (1)

safari (1)

sponsoring (1)

star trek (1)

static generation (1)

talk (1)

technique (1)

theora (1)

thunderbird (1)

typography (1)

unicode (1)

v8 (1)

voice control (1)

webassembly (1)

webkit (1)

webm (1)

webpack (1)

yahoo (1)

Powered by Blogger.