

②ality – JavaScript and more

[About](#)

[Donate](#)

[Subscribe](#)

[ES2017](#)

[Books \(free online!\)](#)

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](#)”

2012-01-01

The multiple roles of JavaScript objects and arrays

Labels: [dev](#), [javascript](#), [jslang](#)

Both objects and arrays play multiple roles in JavaScript. This blog post explains what those roles are.

Objects and arrays are dual data structures

Objects: records versus maps

A JavaScript object is a combination of two data structures: It is sometimes used as a record, sometimes as a dictionary.

- Record:
 - Fixed shape (keys known beforehand)
 - Every (key, value) entry has a different meaning
 - Literal names: Use names literally. Example: `obj.key`
- Map (dictionary), from strings to arbitrary values:
 - Flexible shape (keys generally unknown beforehand)
 - Every (key, value) entry has a similar meaning
 - Computed names: There is usually one level of indirection, you don't use names directly. Example: `obj[mykey]` (with `mykey` being a variable that holds the actual key)

Arrays: tuples versus lists

In a manner similar to objects, arrays are a combination of two data structures:

Sometimes, they are used as tuples, sometimes as lists:

- Tuple:
 - Fixed length
 - Access elements directly by index
- List:
 - Dynamic length
 - Iterate over all elements

Similarities

Given that an object maps names to values and an array maps natural numbers to values, it is obvious that records and tuples are similar. Their structure and range are the same, only their (functional) domain is different. The same holds for maps and lists.

Mixing the domains “program definition” and “application data”

Objects perform duties in two domains: they sometimes define the program, sometimes hold application data.

- Program definition domain: The object is a data structure holding program logic. One mostly uses them in their capacity as records in this domain.
- Application data domain: The object is a data structure holding application data. Objects are used as both records and maps here. JSON data is part of this domain.

(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: Ecmascript](#)

- [2015 \(65\)](#)
- [2014 \(55\)](#)
- [2013 \(98\)](#)
- ▼ [2012 \(177\)](#)
 - [December \(14\)](#)
 - [November \(14\)](#)
 - [October \(15\)](#)
 - [September \(14\)](#)
 - [August \(17\)](#)
 - [July \(15\)](#)
 - [June \(16\)](#)
 - [May \(1\)](#)
 - [April \(14\)](#)
 - [March \(18\)](#)
 - [February \(20\)](#)
- ▼ [January \(19\)](#)
 - What is {} + {} in JavaScript?

[Trying out Underscore on Node.js](#)

[The increasing pervasiveness of JavaScript – a few...](#)

[Remedies for SOPA causes \(a greedy industry in dec...](#)

[JavaScript myth: JavaScript needs a standard bytec...](#)

[Apple and the online education revolution](#)

[What is JavaScript's typeof operator used for?](#)

[The “no asshole comments” option](#)

[VLC keyboard shortcuts](#)

[The first ECMAScript.next features in Firefox and ...](#)

[Samsung to merge its Bada and Tizen mobile operati...](#)

[Streaming a DVD to another Mac via VLC](#)

[Automatically numbering headings via CSS](#)

[GitHub: serve files to the web, with a single bran...](#)

The overlap of the two domains becomes negatively apparent when you are invoking methods on an object-as-map that can have arbitrary keys. For example, the following function is how you would naively implement a check for the presence of a key:

```
function hasKey(obj, key) {
    return obj.hasOwnProperty(key);
}
```

Normal usage:

```
> hasKey({foo: 123}, "foo")
true
```

However, this function ceases to work properly if the key "hasOwnProperty" is used in obj:

```
> hasKey({hasOwnProperty: 123}, "foo")
TypeError: Property 'hasOwnProperty' of object #<Object> is not a function
```

The fix is as follows (there are other things to watch out for, to be covered in an upcoming blog post):

```
function hasKey(obj, key) {
    return Object.prototype.hasOwnProperty.call(obj, key);
}
```

That is, you refer to the method directly and don't use the property search algorithm to find it.

Mixing domains with the [] operator

The [] operator has been established in mainstream programming language as an application data construct. Only JavaScript mixes domains: It is used for application data in arrays and in objects-as-maps. But it is also used in a meta-programming capacity, to access a property via a computed name. The former use case is so frequent that it makes sense to let user-implemented collection types override the [] operator. To avoid clashing with the latter use case (which does not occur often), methods need to be introduced that perform that task. For example, `Object.getProperty(name)` and `Object.setProperty(name, value)`. The ECMAScript.next proposal "[Object Model Reformation: Decoupling \[\] and Property Access](#)" shows how to do this cleanly.

Related reading

The following emails on the es-discuss mailing list inspired this post:

1. [“decoupling \[\] and property access for collections”](#) (by Allen Wirfs-Brock) points out the two domains program definition and application data.
2. [“An array destructuring specification choice”](#) (by David Herman) mentions record versus map and tuple versus list.

Tweets by @rauschma

 **Axel Rauschma...**
@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...
infoworl...

14h

 **Axel Rauschma...**
@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

JavaScript inheritance by example
Crockford's JSDev: switching off privacy for testi...

The pitfalls of using objects as maps in JavaScript...

Dion Almaer on the history of webOS

The multiple roles of JavaScript objects and array...

- 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)
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)

2 Comments The 2ality blog

 Login

Sort by Best

 Recommend

 Share



Join the discussion...



rwaldron • 5 years ago

The exception can be avoided if you store a reference to `Object.prototype.hasOwnProperty`:

```
var hasOwn = Object.prototype.hasOwnProperty,  
hasKey = function( obj, key ) {  
    return hasOwn.call( obj, key );  
};
```

```
console.log(  
    hasKey({ foo: 123 }, "foo")  
);
```

```
console.log(  
    hasKey({ hasOwnProperty: 123 }, "foo")  
);
```

[see more](#)

[^](#) | [v](#) • Reply • Share >



Axel Rauschmayer Mod → rwaldron • 5 years ago

Good point. I am going to elaborate on the fix in an upcoming blog post, but you are right that I should at least sketch a solution (I edited the post to do so).

[^](#) | [v](#) • Reply • Share >

 [Subscribe](#)  [Add Disqus to your site](#) [Add Disqus Add](#)  [Privacy](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

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)
[pi 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)
[flatr](#) (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)

[pnaci](#) (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)