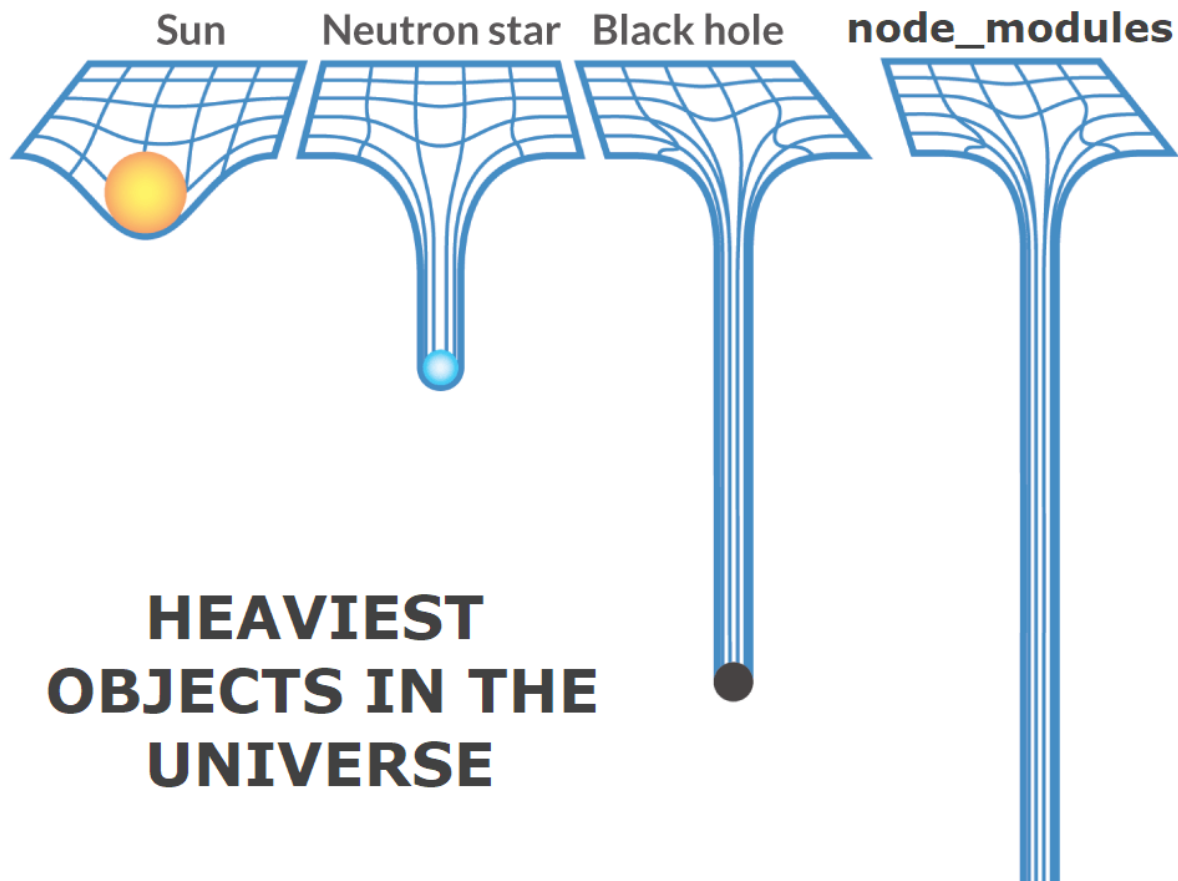


Javascript 😞 😡 😖 🤖

- node_modules



-
- Webpack
- convoluted ecosystem which seems to change every three month
- npm or yarn, and bad package dependencies in general

thursday

About Thursday

`sandropasquali` published 2.0.0 • 4 years ago

tuesday

About Tuesday

`sandropasquali` published 2.0.0 • 4 years ago

wednesday

About Wednesday

`sandropasquali` published 2.0.0 • 4 years ago

april

About April

`sandropasquali` published 2.0.0 • 4 years ago

august

About August

`sandropasquali` published 2.0.0 • 4 years ago

people

Populate a JSON object, save it, change it, send it, validate it, and more

`sandropasquali` published 1.0.0 • 6 years ago

-
- **Javascript**

```

npm http GET https://registry.npmjs.org/grunt-contrib-stylus
npm http 304 https://registry.npmjs.org/grunt-contrib-stylus
npm http 304 https://registry.npmjs.org/grunt-contrib-watch
npm http 304 https://registry.npmjs.org/grunt
npm http GET https://registry.npmjs.org/nib
npm http GET https://registry.npmjs.org/stylus
npm http GET https://registry.npmjs.org/findup-sync
npm http GET https://registry.npmjs.org/coffee-script
npm http GET https://registry.npmjs.org/glob
npm http GET https://registry.npmjs.org/colors
npm http GET https://registry.npmjs.org/minimatch
npm http GET https://registry.npmjs.org/nopt
npm http GET https://registry.npmjs.org/dateformat/1.0.2-1.2.3
npm http GET https://registry.npmjs.org/lodash
npm http GET https://registry.npmjs.org/rimraf
npm http GET https://registry.npmjs.org/hooker
npm http GET https://registry.npmjs.org/which
npm http GET https://registry.npmjs.org/eventemitter2
npm http GET https://registry.npmjs.org/underscore.string
npm http GET https://registry.npmjs.org/async
npm http GET https://registry.npmjs.org/js-yaml
npm http GET https://registry.npmjs.org/getobject
npm http GET https://registry.npmjs.org/exit
npm http GET https://registry.npmjs.org/iconv-lite

```

- lack of a standard build system or a standard library
- • **Javascript**
- dynamically and loosely typed, with almost no restrictions

	true	false	1	0	-1	"true"	"false"	"1"	"0"	"-1"	" "	null	undefined	Infinity	-Infinity	[]	{}	[[]]	[0]	[1]	NaN
true:	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
false:	≠	===	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
1:	≠	≠	===	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
0:	≠	≠	≠	===	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
-1:	≠	≠	≠	≠	===	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
"true":	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
"false":	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
"1":	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
"0":	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
"-1":	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
" ":	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠
null:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠	≠
undefined:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠	≠
Infinity:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠	≠
-Infinity:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠	≠
[]:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠	≠
{}	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠	≠
[[]]:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠	≠
[0]:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠	≠
[1]:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===	≠
NaN:	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	≠	===

≠ Not equal
 ≡ Loose equality
 Often gives "false"
 positives like "1" is
 true; [] is "0"
 === Strict equality
 Mostly evaluates as
 one would expect.

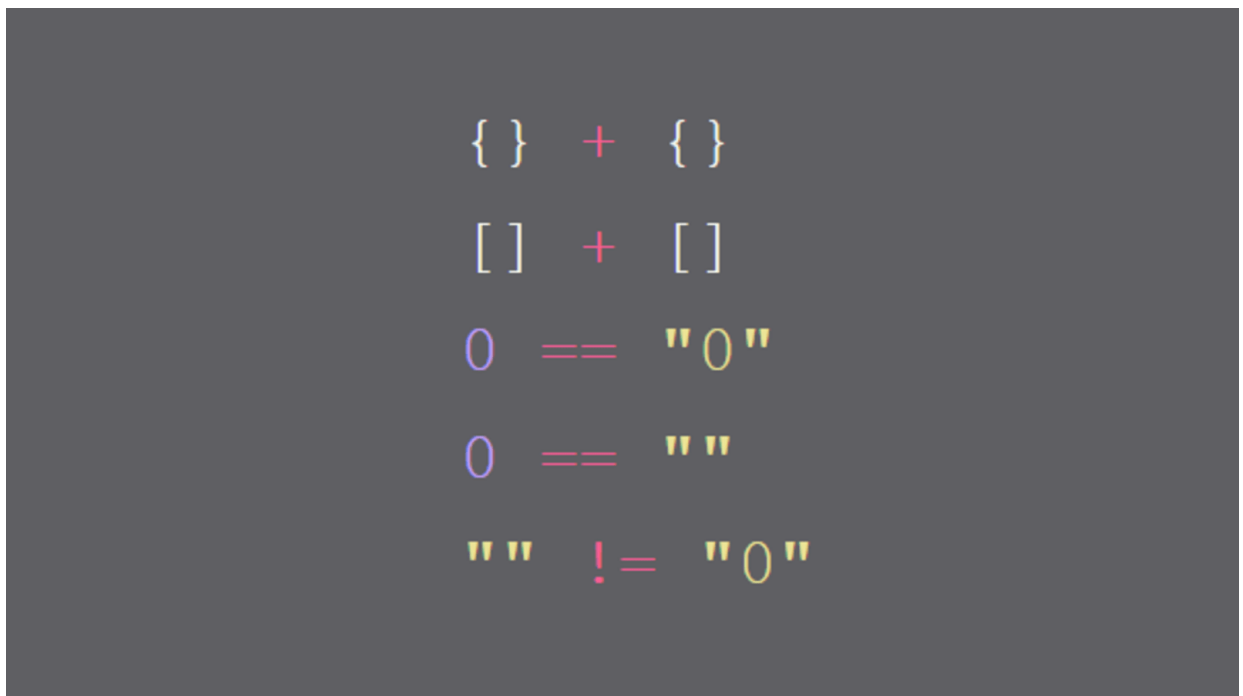
```
> {} + []  
0  
> [] + {}  
'[object Object]'  
> "0" == false  
true  
> true + false  
1  
> [] == ![]  
true  
> '13' + 52 + '46'  
'135246'
```

.

Table 3-2. JavaScript type conversions

Value	Converted to:			
	String	Number	Boolean	Object
undefined	"undefined"	NaN	false	throws <i>TypeError</i>
null	"null"	0	false	throws <i>TypeError</i>
true	"true"	1		new Boolean(true)
false	"false"	0		new Boolean(false)
"" (empty string)		0	false	new String("")
"1.2" (nonempty, numeric)		1.2	true	new String("1.2")
"one" (nonempty, non-numeric)		NaN	true	new String("one")
0	"0"		false	new Number(0)
-0	"0"		false	new Number(-0)
NaN	"NaN"		false	new Number(NaN)
Infinity	"Infinity"		true	new Number(Infinity)
-Infinity	"-Infinity"		true	new Number(-Infinity)
1 (finite, non-zero)	"1"		true	new Number(1)
{ } (any object)	see §3.8.3	see §3.8.3	true	
[] (empty array)	" "	0	true	
[9] (1 numeric elt)	"9"	9	true	
['a'] (any other array)	use <i>join()</i> method	NaN	true	
function () { } (any function)	see §3.8.3	NaN	true	

•



•

- design flaws
 - [deno needs to exist](#)
 - **Javascript**
 - [typescript needs to exist](#)
- [insecure nature](#)
- The pedantic framework wars



-
- .then.then.then.then
 - | [How 2 return variable from promise aaaa](#)
- Javascript

Add a number to another number in JavaScript



hallo

I have got a number in my JavaScript variable! Now how do I add another number to it? Please

[javascript](#)

3 Answers

oldest newest votes



You should definitely use jQuery. It's really great and does all things

[link](#) [edit](#) [flag](#)

answered 11 minutes ago

[I<3jQuery](#)
1,234 ● 2 ● 13

I agree, jQuery is really the best, it solves all kinds of browser problems and is good, as well – [Jsumc0da](#) 8 mins ago

+1 jquery is best quality code ever, if you don't use your a idiot – [Werry_Togan](#) 4 mins ago

[add comment](#)



I think there's a jQuery plugin for that. Google for jQuery basic arithmetic plugin.

[link](#) [edit](#) [flag](#)

answered 5 minutes ago

[Timothy Goatse](#)
4,321 ● 1 ● 12

yeah, jQuery is definately the way to go – [fishnipples](#) 5 mins ago

I used the jQuery diet plugin and lost 10kg in a week – [jfatty](#) 4 mins ago

[add comment](#)



To add numbers together you should use the + operator, for example:

```
var a= 1;
var b= a+2;
alert(b); // 3
```

[link](#) [edit](#) [delete](#) [flag](#)

answered 50 seconds ago

[bobince](#)
some ● ● ●

-1 not enough jQuery – [Jsumc0da](#) 30 secs ago

you suck – [Timothy Goatse](#) 3 secs ago

tagged

[javascript](#) × 18553

asked

a while ago

viewed

some times

latest activity

just now

Wanted: Yet another ASP.NET developer. See this and other great job listings at [jobs.stackoverflow.com](#).

Related

[What is the best number?](#)

[How can I use JavaScript to parse some HTML using regex?](#)

[JavaScript: why is my text content getting mangled when I clone nodes? Obviously I must be doing something wrong as jQuery is perfect](#)

[Stupid JavaScript floating point numbers are broken](#)

[How can I extract number from HTML using a regex without jQuery, singing the song that ends the world?](#)

[Is there a jQuery plugin for making an HTML page appear in the browser?](#)

[Where are my legs?](#)

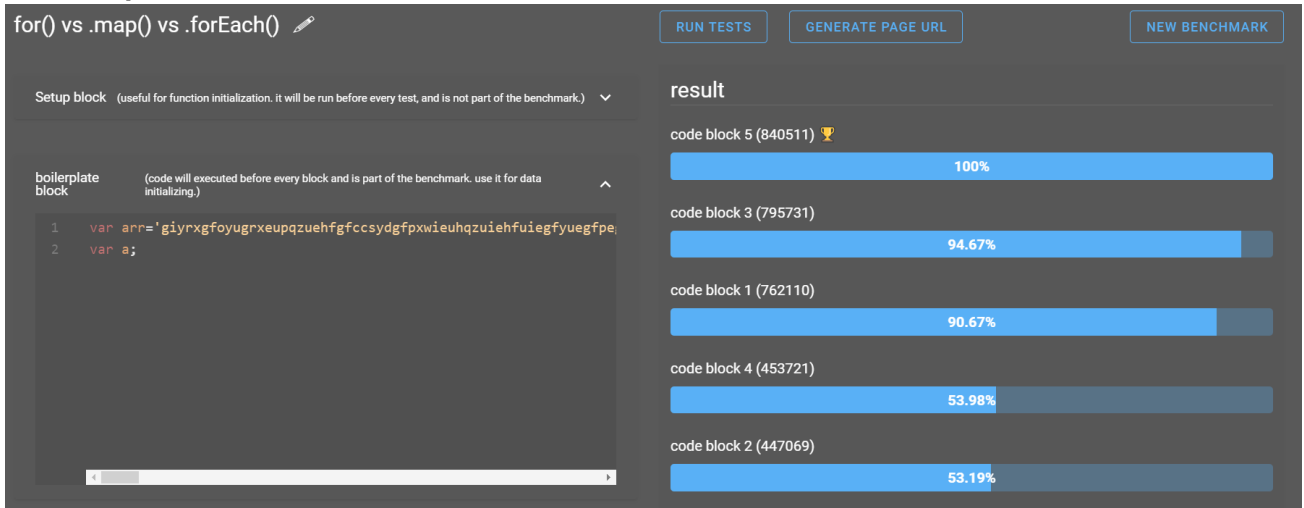
shitcode

```
// accurate debugging
echo "dupa";
```

Latest Browse Top Worst Ad

All 420 [ActionScript](#) 2 [Bash](#) 6 [Brainfuck](#) 1 [LSL](#) 19 [C++](#) 27 [C#](#) 50 [CSS](#) 4 [Pascal](#) 1 [Django](#) 1 [Erlang](#) 1 [GLSL](#) 1 [Go](#) 7 [Groovy](#) 1 [HTML](#) 8 [Java](#) 65 [JavaScript](#) 83 [Less](#) 1 [Lua](#) 2 [Nix](#) 1 [Obj-C](#) 4 [OCaml](#) 1 [PHP](#) 66 [PowerShell](#) 1 [Python](#) 43 [Ruby](#) 2 [Scala](#) 3 [SQL](#) 6 [Swift](#) 4 [TypeScript](#) 7 [VBScript](#) 2

- JavaScript



(<https://jsben.ch/vSgoN>)



It is a deeply flawed language with strange & unpredictable behaviour. It gives a little guarantee of quality bug-free production code unless you have a comprehensive grasp on the language; it is vulnerable to abuse; has weird scoping, and its mutability can cause issues to those who lack experience with such a dynamic language (unless using `const` — ES6).

- <https://medium.com/swlh/typescript-rescued-the-javascript-language-bfc944c0f96b>

im not fucking sorry i needed this rant [For Me, Not You](#) worst programming lang