



Thomas Watson

@wa7son

[github.com/watson](https://github.com/watson)





A large, dark gray, stylized letter 'Q' serves as a background. Inside the white space of the 'Q', there is a white silhouette of a cat's face, including its ears and whiskers. A horizontal red bar is superimposed across the middle of the 'Q', containing the text 'github.com/watson/talks' in white.

[github.com/watson/talks](https://github.com/watson/talks)



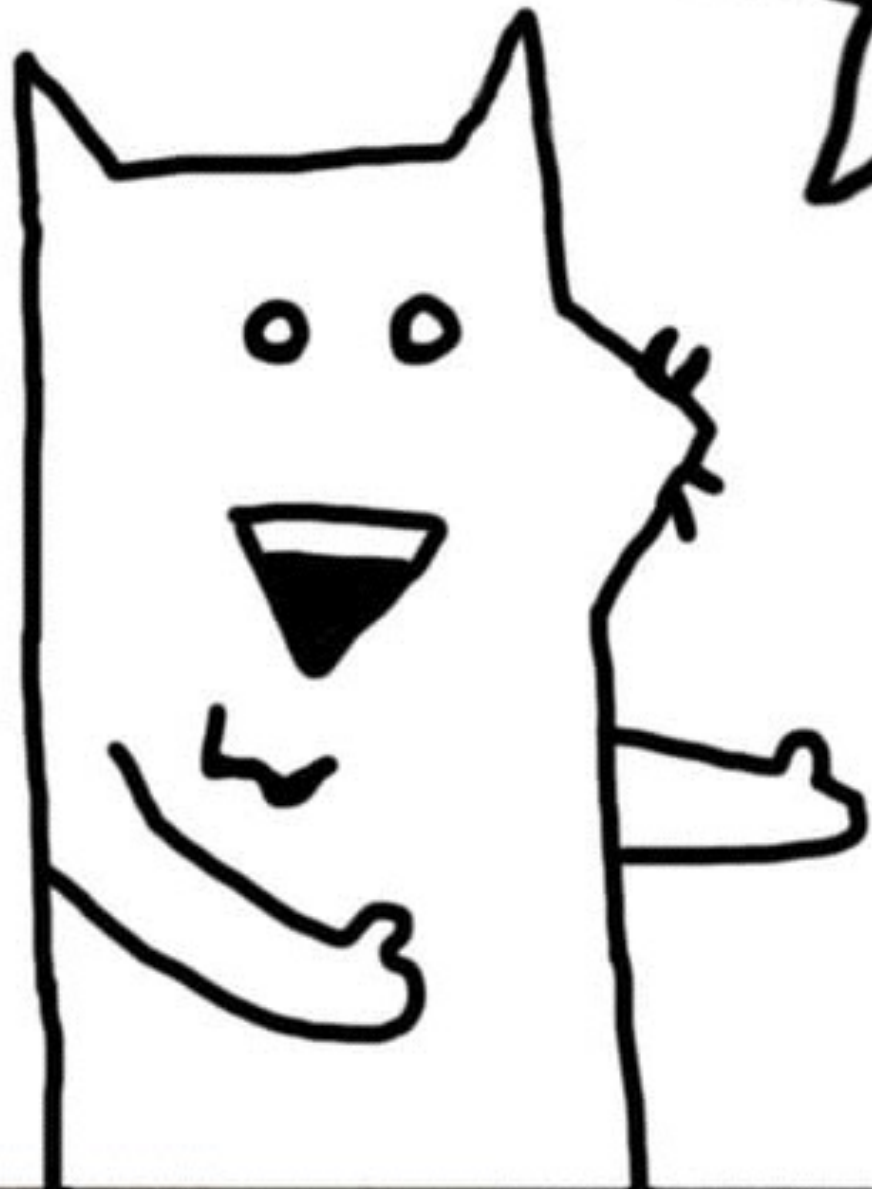
These go to eleven!





How to write code  
that writes code

- why?





performance

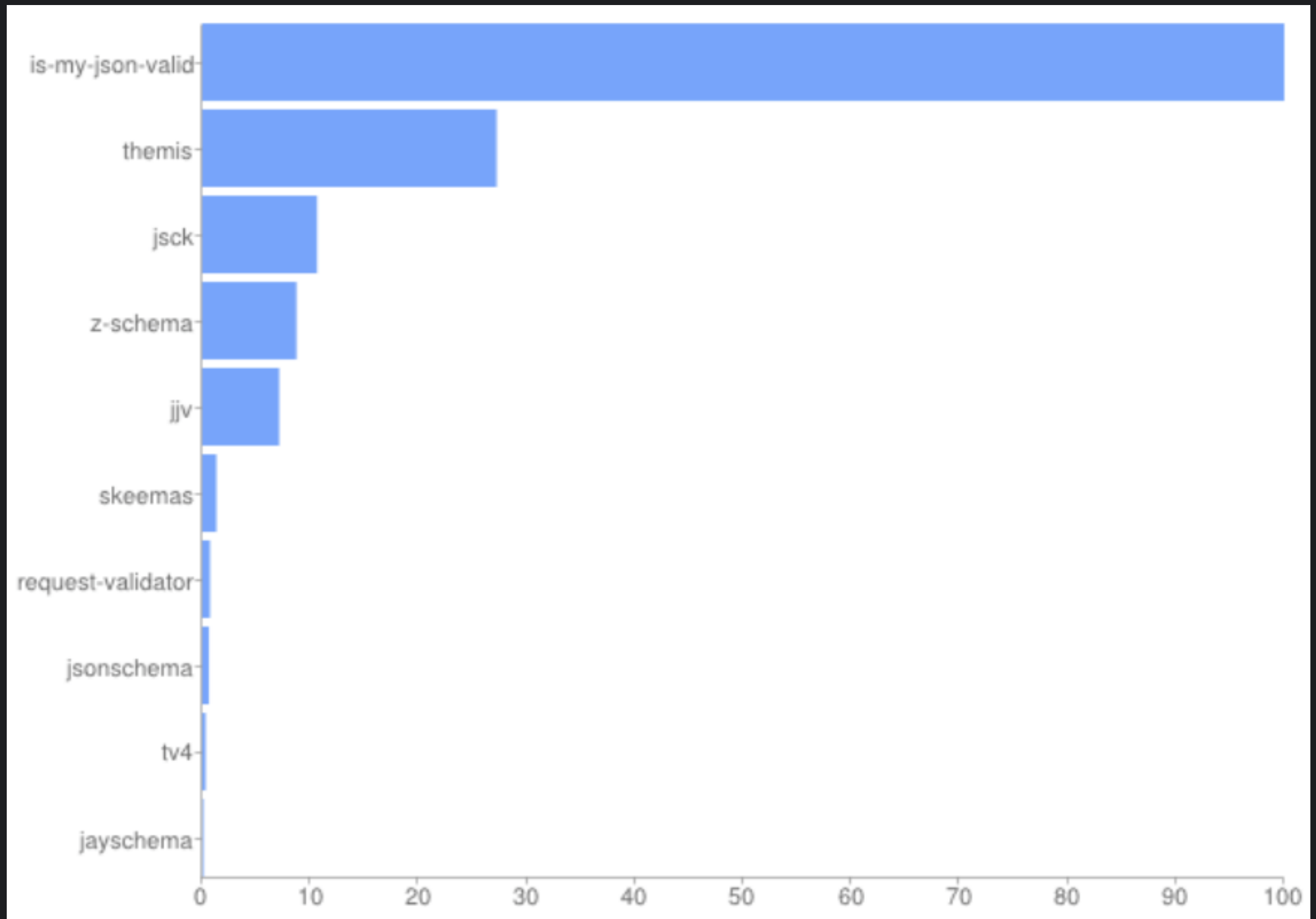




# disclaimer



github.com /  
mafintosh /  
is-my-json-valid





*f(seed, data)*

$$f_2 = f_1(\textit{seed})$$
$$f_2(\textit{data})$$



# An oversimplified example

```
function doMath (a, b) {  
  // ...do stuff with a...  
  return a + b  
}
```

# An oversimplified example

```
function doMath (a, b) {  
  // ...do stuff with a...  
  return a + b  
}
```

```
function prepareMath (a) {  
  // ...do stuff with a...  
  return function doMath (b) {  
    return a + b  
  }  
}
```

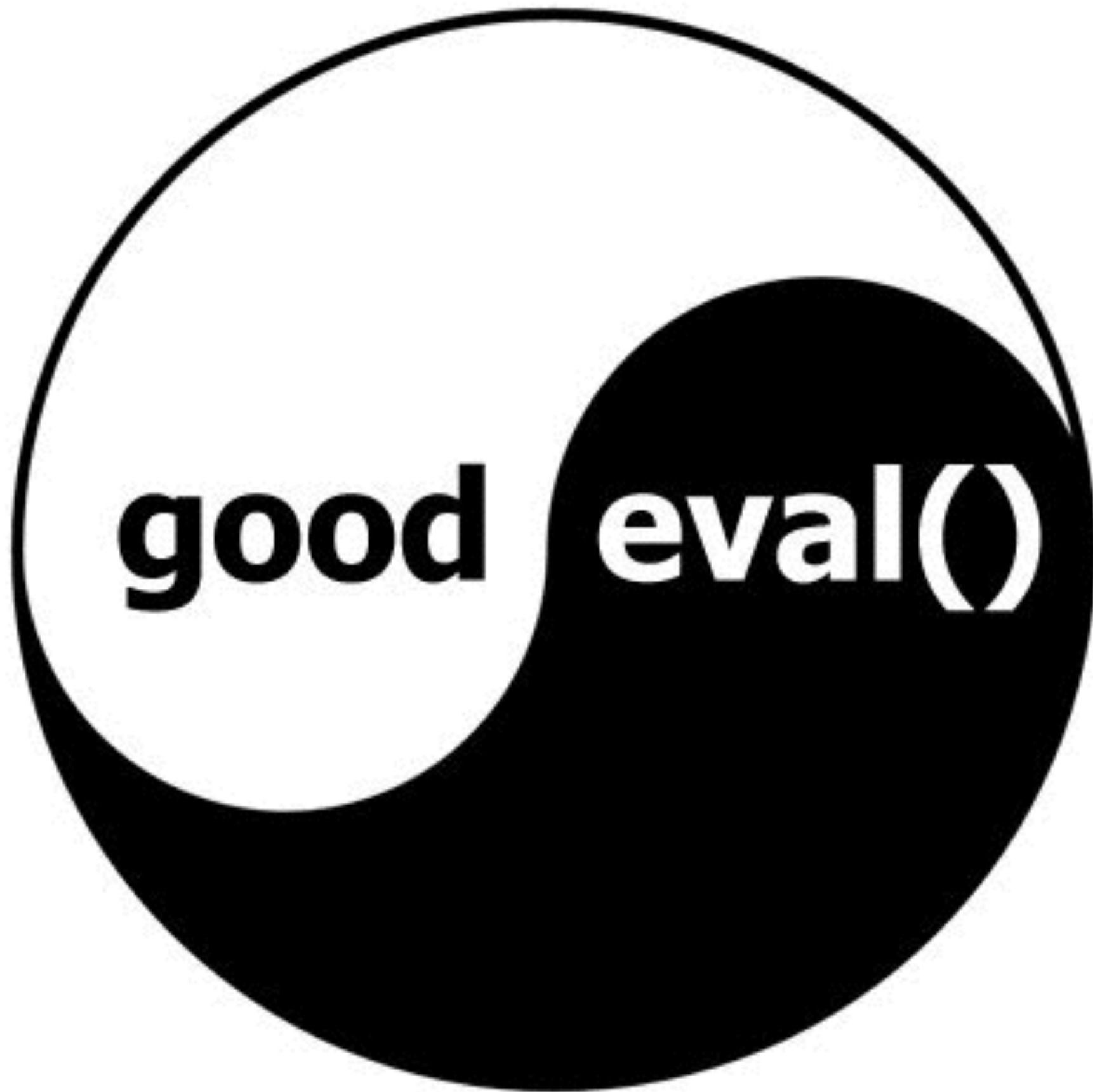


# An oversimplified example

```
function doMath (a, b) {  
  // ...do stuff with a...  
  return a + b  
}
```

```
function prepareMath (a) {  
  // ...do stuff with a...  
  return function doMath (b) {  
    return a + b  
  }  
}
```

```
function prepareMath (a) {  
  // ...do stuff with a...  
  var src = 'return ' + a + ' + b'  
  return new Function ('b', src)  
}
```





```
var admin = false
```

```
eval('var admin = true')
```

```
if (admin) console.log('user is admin')  
else console.log('user is NOT admin')
```

```
$ node example.js  
user is admin
```

```
var admin = false
```

```
var fn = new Function('var admin = true')  
fn()
```

```
if (admin) console.log('user is admin')  
else console.log('user is NOT admin')
```

```
$ node example.js  
user is NOT admin
```

```
var fn = new Function('var admin = true')  
  
console.log(fn.toString())
```

```
$ node example.js  
function anonymous() {  
var admin = true  
}
```



```
var admin = false
```

```
var fn = new Function('admin = true')  
fn()
```

```
if (admin) console.log('user is admin')  
else console.log('user is NOT admin')
```

```
$ node example.js  
user is NOT admin
```

```
global.admin = false
```

```
var fn = new Function('admin = true')  
fn()
```

```
if (global.admin) console.log('user is admin')  
else console.log('user is NOT admin')
```

```
$ node example.js  
user is admin
```

```
var src = '\n"Hello World"\n'  
  
var fn = new Function('return (' + src + ')')  
  
console.log(fn())
```

```
$ node example.js  
Hello World
```

```
var src = 'n + 1'
```

```
var fn = new Function('n', 'return (' + src + ')')
```

```
console.log(fn(1))
```

```
$ node example.js  
2
```



```
var n = 2  
var src = '2 * ' + n  
  
var fn = new Function('return (' + src + ')')  
  
console.log(fn())
```

```
$ node example.js  
4
```

```
var n = 'console.log("owned") '  
var src = '2 * ' + n  
  
var fn = new Function('return (' + src + ')')  
  
console.log(fn())
```

```
$ node example.js  
owned  
NaN
```

```
var n = 'console.log("owned")'  
var src = util.format('2 * %j', n)  
  
var fn = new Function('return (' + src + ')')  
  
console.log(fn())
```

```
$ node example.js  
NaN
```



# use cases



# Regular Expressions

```
var r = new RegExp( 'foo(bar)?' )
```

```
r.test( 'foo' )
```

# Templates

```
<div class="entry">  
  <h1>{{title}}</h1>  
  <div class="body">  
    {{body}}  
  </div>  
</div>
```

# JSON Schema

```
{  
  type: 'object',  
  properties: {  
    aProperty: {  
      type: 'string'  
    }  
  }  
}
```

# MongoDB query language

```
{  
  foo: {  
    $gt: 42,  
    $lt: 100  
  }  
}
```



Let's write a query  
language interpreter

# Our language

```
{  
    $eq: value,  
    $gt: value,  
    $lt: value,  
    $not: {...} // negate this expression  
}
```



# coding time















A black and white photograph of a rock band performing on stage. The band consists of several members, including a lead singer with long, curly hair in the center, and other musicians playing guitars and bass. The background is dark with some stage lights visible. Overlaid on the image is large, bold, purple text that reads "coding" at the top and "time" at the bottom. A red diagonal banner across the middle contains the white text "github.com/watson/talks".

# coding

[github.com/watson/talks](https://github.com/watson/talks)

# time

github.com /  
mafintosh /  
generate-object-property

```
var gen = require('generate-object-property')  
console.log(gen('a', 'b')) // => a.b  
console.log(gen('a', 'foo-bar')) // => a["foo-bar"]
```

github.com /  
mafintosh /  
generate-function



```
var genfun = require('generate-function')

var addNumber = function(val) {
  var fn = genfun()
    ('function add(n) {'
    ('return n + %d', val) // supports format strings
    ('}')

  return fn.toFunction() // will compile the function
}

var add2 = addNumber(2)
console.log('1+2=', add2(1))
console.log(add2.toString()) // logs generated function
```



A scenic view of a harbor at dusk. The water is calm, reflecting the warm lights from the buildings and the cool blue of the twilight sky. Several boats are docked along the quay, their masts reaching up. The buildings are multi-story, with many windows glowing with light. The overall atmosphere is peaceful and picturesque.

# Tak!

@wa7son

[github.com/watson](https://github.com/watson)

 opbeat