

JAVASCRIPT *In The Sky* *With TYPES*

ABRAÃO
alves

MATHEUS
albuquerque

cejs 

A screenshot of a web browser showing a presentation slide on Speaker Deck. The slide title is "JAVASCRIPT In The Sky With TYPES". Overlaid on the slide is a large blue text "SLIDES ONLINE AT:" followed by a red URL "bit.ly/js-and-types-cejs". To the right of the slide, there is a sidebar for the speaker, Matheus Brasil, showing his profile picture, name, 3 presentations, 0 stars, published in Programming, 0 views, and sharing options like Twitter, Facebook, Embed, Direct Link, and Download.

speakerdeck.com

Speaker Deck Search... Browse Upload Matheus Brasil

Unpublished

JAVASCRIPT *In The Sky*
With TYPES

ABRAÃO MATHEUS
alves albuquerque

cejs

JavaScript In The Sky With Types by Matheus Brasil

Published June 1, 2016 in Programming

Explore 0 Stars

Matheus Brasil 3 Presentations

Star this Talk 0 Stars

Published in Programming

Stats 0 Views

Share Twitter, Facebook

Embed

Direct Link

Download

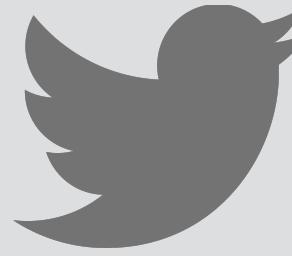
bit.ly/js-and-types-cejs

Slides from the talk Abraão Alves and I gave at CEJS 2016 on the importance of thinking about type systems - and how things can be better with TypeScript + other tools.

Clique para ver mais apresentações by this Speaker Speaker Details



/ytheimulator



@ytheimulator



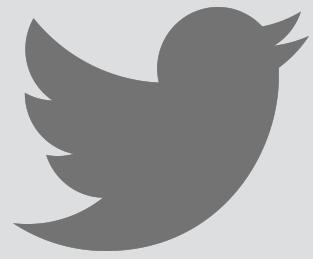
ytheimulator.ME







/AbraaoAlves



@abraao4lves



abraaoalves.GITHUB.IO



nash



A close-up photograph of a man with light brown hair, wearing glasses and a beard. He is looking upwards and slightly to his right with a thoughtful expression. He is wearing a grey zip-up hoodie over a red shirt.

connect the dots



Ju Gonçalves - Reduce seu novo melhor amigo - BrazilJS 2015

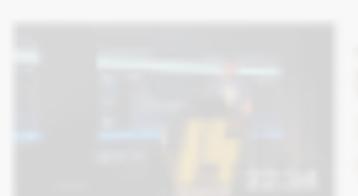


BrazilJS
Mais de 1 milhão de visualizações

+ Adicionar | Partilhar | ...

1.977 visualizações

Pôsteros



Reprodução automática 0

Jonathan Sampson - CSS, o que tem de bacana? - BrazilJS 2015
Visualizar | 107 visualizações

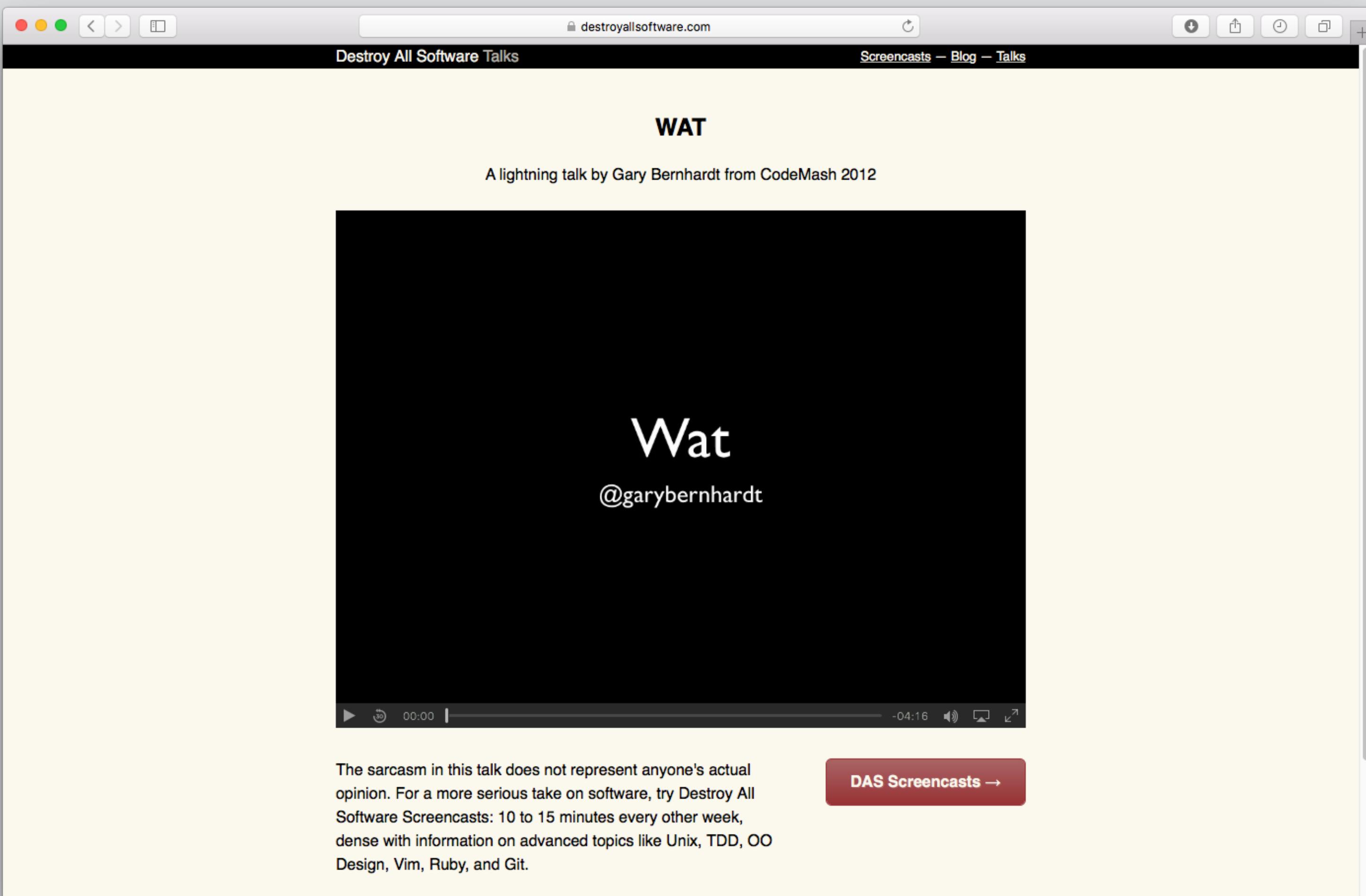


Felipe Ribeiro - JavaScript (playful)



A large audience of people is seated in rows of grey chairs in a lecture hall or auditorium. They are all facing towards the front of the room, where a person in a white shirt and dark trousers is standing, likely giving a presentation. The room has high ceilings and large windows on the right side.

WHAT *brings us to*
CE..JS?



JS Wat by Gary Bernhardt
<http://bit.ly/js-wat>

λ ~ node

> [] + []

''

> [] + {}

'[object Object]'

> {} + []

0

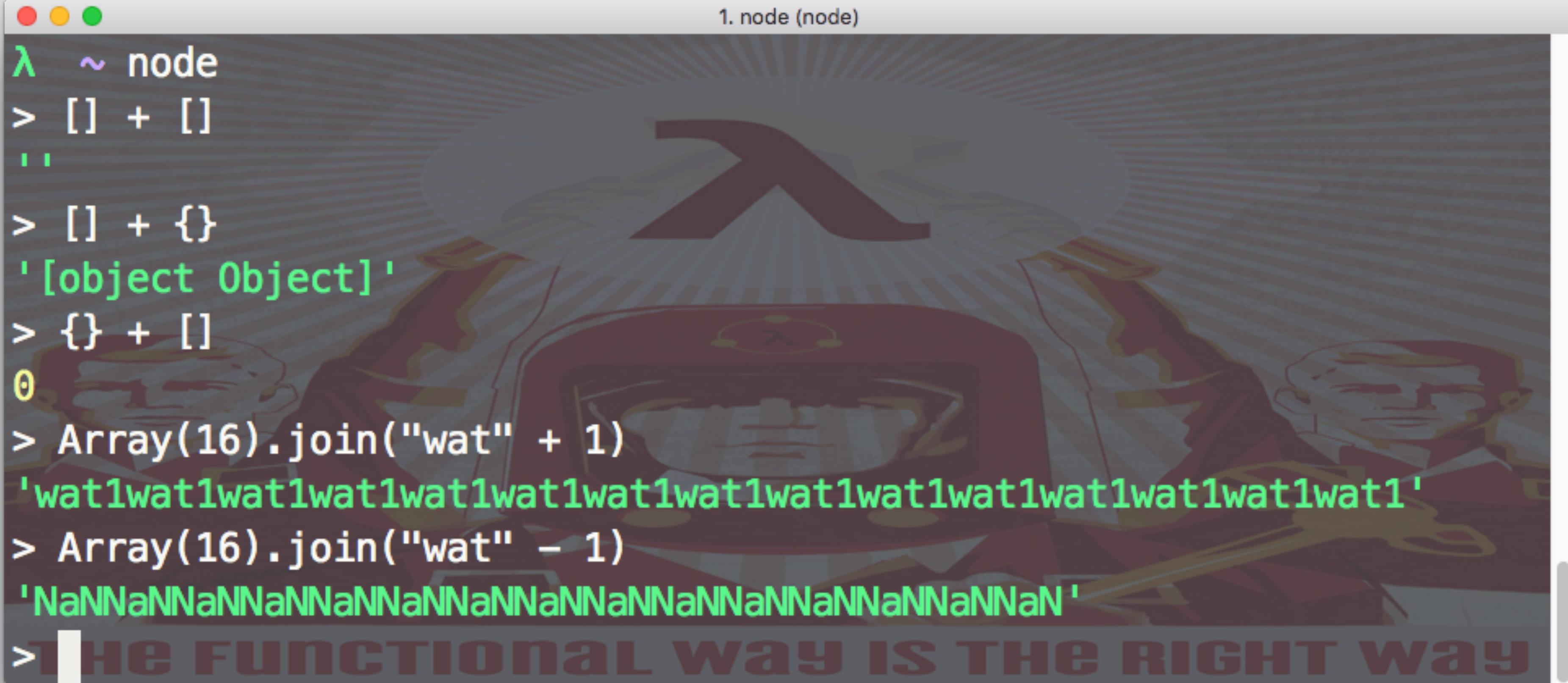
> Array(16).join("wat" + 1)

'wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1wat1'

> Array(16).join("wat" - 1)

'NaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaNNaN'

>**THE FUNCTIONAL WAY IS THE RIGHT WAY**



The image shows a person from the waist up, wearing a dark t-shirt with a large white Lambda symbol on it. They are sitting at a desk with a computer keyboard in front of them. The background is a light-colored wall with some abstract shapes or shadows.

typeof

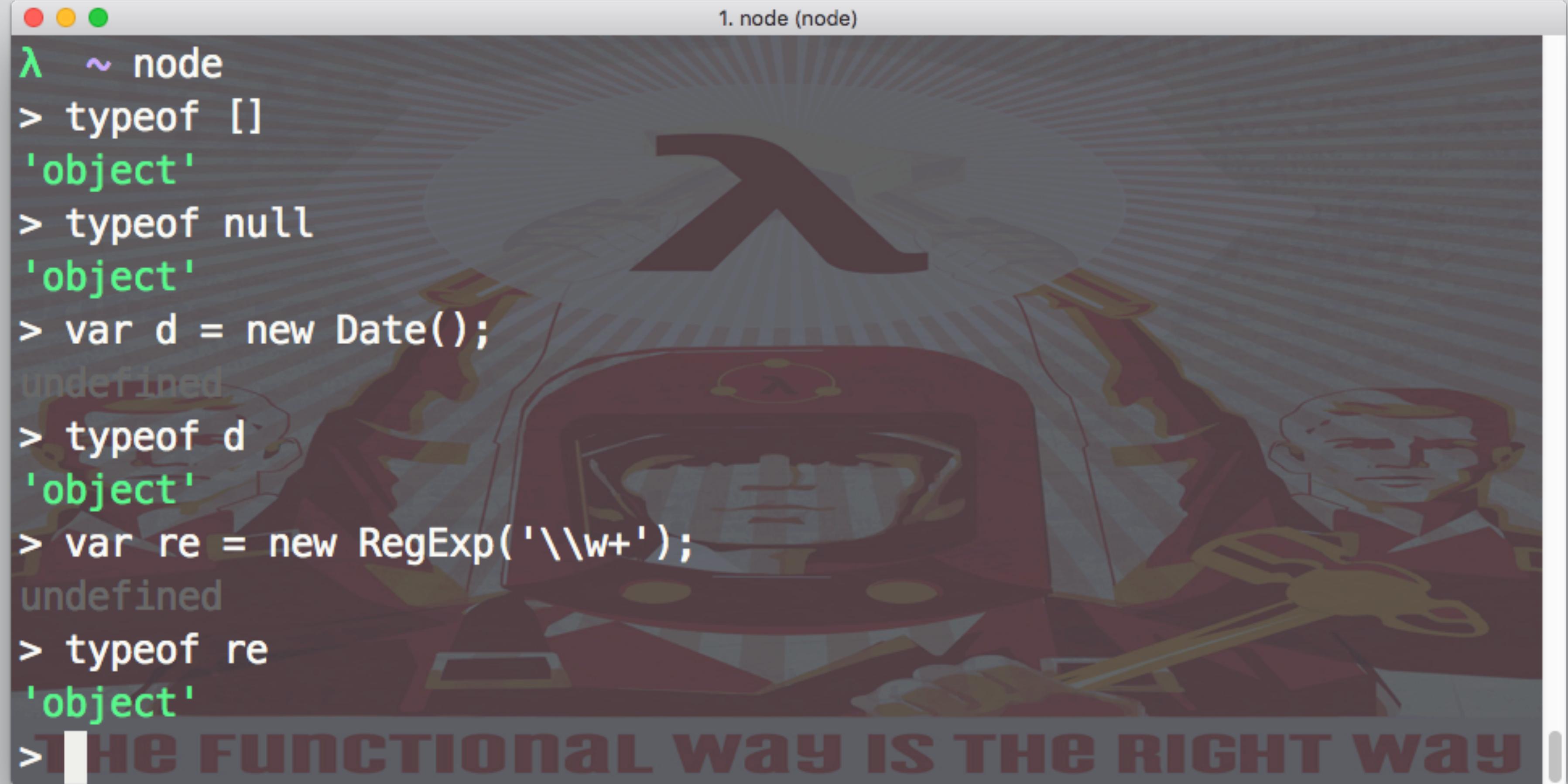
λ ~ node
> typeof "cejs"
'string'
> typeof 2016
'number'
> typeof true
'boolean'
> typeof undefined
'undefined'
> typeof {}
'object'
> typeof function(){}
'function'
>**THE FUNCTIONAL WAY IS THE RIGHT WAY**



The image features a central figure with a large, dark, angular headpiece and a mask. He is wearing a red and gold patterned robe and is holding a long, curved sword. The background consists of concentric, radiating lines in shades of grey and white, creating a sunburst effect. In the top right corner of the slide, there is a small window-like frame containing the text "1. node (node)".

1. node (node)

```
λ ~ node
> typeof []
'object'
> typeof null
'object'
> var d = new Date();
undefined
> typeof d
'object'
> var re = new RegExp('\\w+');
undefined
> typeof re
'object'
> THE FUNCTIONAL WAY IS THE RIGHT WAY
```



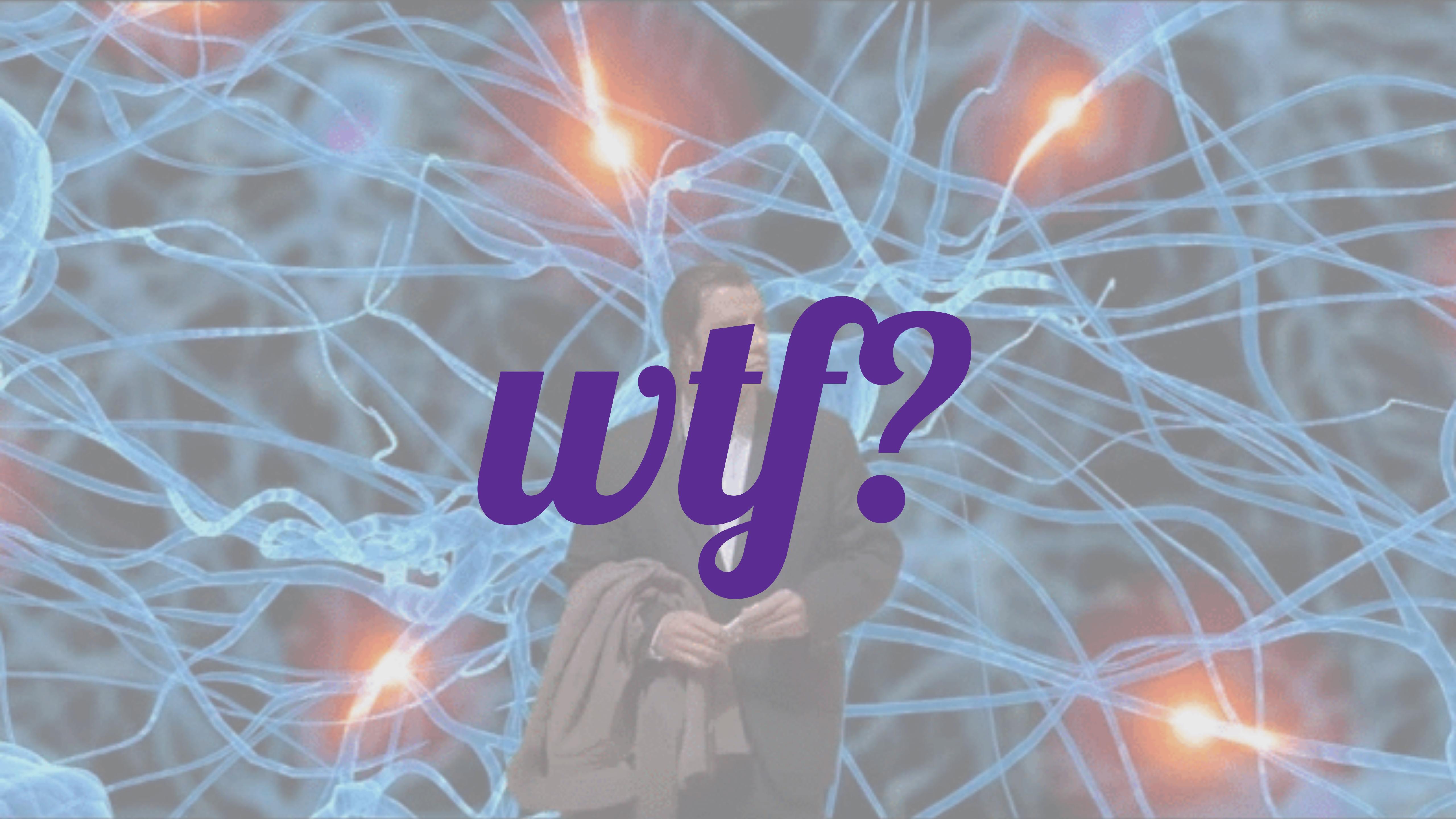


*distinguish between
different kind of
primitive values*



*distinguish between
different kinds of
objects*

*distinguish between
them and objects*



wtf?



LEARNING *what*
I hadn't LEARNT

data

“*Things known - or assumed as FACTS -, which become the BASIS OF REASONING or CALCULATION.*

data

“ QUANTITIES, CHARACTERS or SYMBOLS *of which operations are performed by a computer and stored and transmitted as electrical signals and recorded on magnetic, optical or mechanical recording media.*

Você está usando o YouTube como Matheus Brasil.

Quer alternar a conta? Clique aqui

YouTube Pesquisar Enviar

! ABSTRACTION !

BRAZILJS CONF

2:57 / 31:23

Ju Gonçalves - Reduce seu novo melhor amigo - BrazilJS 2015

BrazilJS Inscrito 3.753

Adicionar a Compartilhar Mais

1.977 visualizações

Próximo Reprodução automática

Jonathan Sampson - ES6, o que tem de bacana? - BrazilJS 2015

BrazilJS 32:34 927 visualizações

Felipe Ribeiro - JavaScript @Spotify -



The act [redacted] over simple ideas, are chiefly these three: 1. Combining several simple ideas into one compound one, and thus all complex ideas are made. 2. The second is bringing two ideas, whether simple or complex, together, and setting them by one another so as to take a view of them at once, without uniting them into one, by which it gets all its ideas of relations. 3. The third is separating them from all other ideas that accompany them in their real existence: this is called abstraction, and thus all its general ideas are made.

John Locke, *An Essay Concerning Human Understanding* (1690)

We are about to study the idea of a *computational process*. Computational processes are abstract beings that inhabit computers. As they evolve, processes manipulate other abstract things called *data*. The evolution of a process is directed by a pattern of rules called a *program*. People create programs to direct processes. In effect, we conjure the spirits of the computer with our spells.

A computational process is indeed much like a sorcerer's idea of a spirit. It cannot be seen or touched. It is not composed of matter at all. However, it is very real. It can perform intellectual work. It can answer questions. It can affect the world by disbursing money at a bank or by controlling a robot arm in a factory. The programs we use to conjure processes are like a sorcerer's spells. They are carefully composed from symbolic expressions in arcane and esoteric *programming languages* that prescribe the tasks we want our processes to perform.

A computational process, in a correctly working computer, executes programs precisely and accurately. Thus, like the sorcerer's apprentice, novice programmers must learn to understand and to anticipate the consequences of their conjuring. Even small errors (usually called *bugs* or *glitches*) in programs can have complex and unanticipated consequences.

Fortunately, learning to program is considerably less dangerous than learning sorcery, because the spirits we deal with are conveniently contained in a secure way. Real-world programming, however, requires care, expertise, and wisdom. A small bug in a computer-aided design program, for example, can lead to the catastrophic collapse of an airplane or a dam or the self-destruction of an industrial robot.

Master software engineers have the ability to organize programs so that they can be reasonably sure that the resulting processes will perform the tasks intended. They can visualize the behavior of their systems in advance. They know how to structure programs so that unanticipated problems do not lead to catastrophic consequences, and when problems do arise, they can *debug* their programs. Well-designed computational systems, like well-designed automobiles or nuclear reactors, are designed in a modular manner, so that the parts can be constructed, replaced, and debugged separately.

Structure and Interpretation of Computer Programs

Second Edition



Harold Abelson and
Gerald Jay Sussman
with Julie Sussman

processes

“ Computational processes are abstract beings that inhabit computers. As they evolve, PROCESSES MANIPULATE OTHER ABSTRACT THINGS CALLED DATA. The evolution of a process is directed by a pattern of rules called a program. PEOPLE CREATE PROGRAMS TO DIRECT PROCESSES.

datatype

“

*A specific representation of some
data.*

type systems

“ *A type system is a syntactic method for automatically CHECKING THE ABSENCE OF CERTAIN ERRONEOUS BEHAVIORS by classifying program phrases according to the kinds of values they compute.*

Benjamin Pierce - Types and Programming Languages

“The fundamental purpose of a type of system is TO AVOID THE OCCURRENCE OF ERRORS DURING THE EXECUTION OF A PROGRAM.

Luca Cardelli - Type Systems



Correctness-by- Design

**“A program that is VALID IN THE
LANGUAGE, should also be A
CORRECT PROGRAM.**





TypeScript

*a STATICALLY TYPED superset of JavaScript
that COMPILES TO PLAIN JAVASCRIPT*

A photograph of Linus Torvalds, a middle-aged man with light brown hair and glasses, wearing a dark suit jacket over a light-colored shirt. He is pointing his right index finger directly at the camera. Overlaid on the image is the text "talk is CHEAP show me the CODE" in a large, bold, sans-serif font. The word "CHEAP" is in blue, while the rest of the sentence is in dark gray.

*talk is CHEAP show me the
CODE*

demotime



thank

ABRAÃO
alves

MATHEUS
albuquerque

cejs



NETFLIX

JAVASCRIPT *In The Sky* *With TYPES*

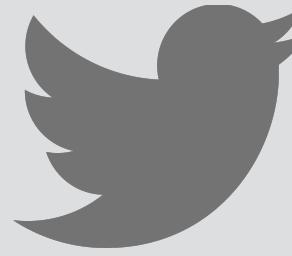
ABRAÃO
alves

MATHEUS
albuquerque

cejs 



/ytheimulator



@ytheimulator

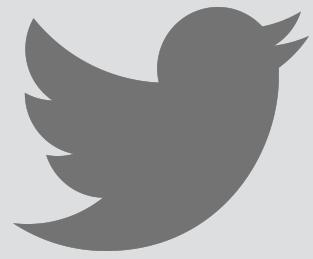


ytheimulator.ME





/AbraaoAlves



@abraao4lves



abraaoalves.GITHUB.IO



TypeScript

1 year total

2,544

contributions

May 23, 2015 — May 23, 2016

Busiest day

40

contributions

May 23, 2016

Longest streak

367

days

null — null

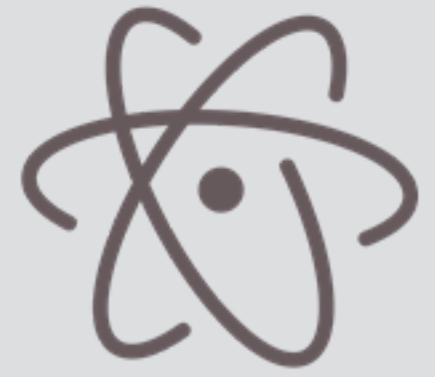
Current streak

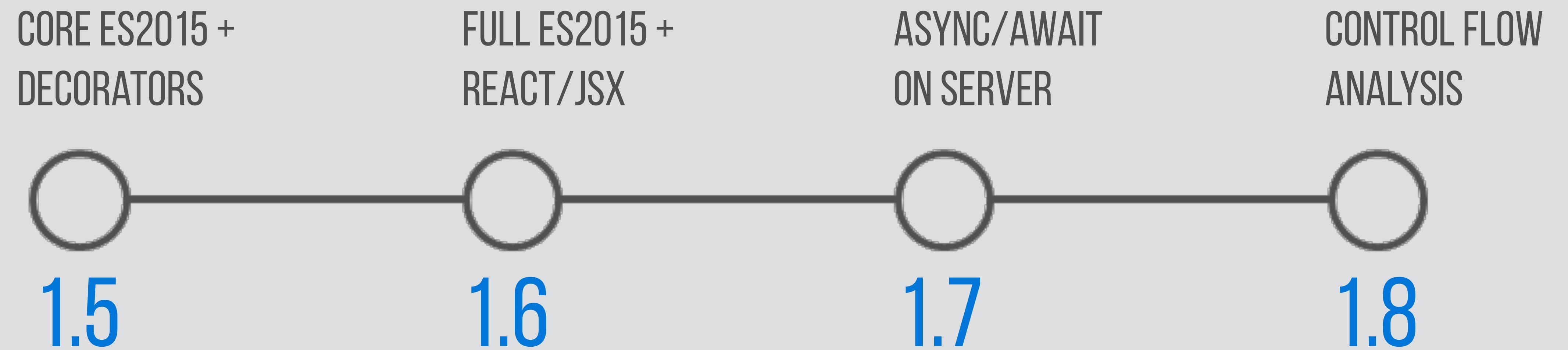
367

days

May 23 — May 23

*is it an ACTIVE
project?*





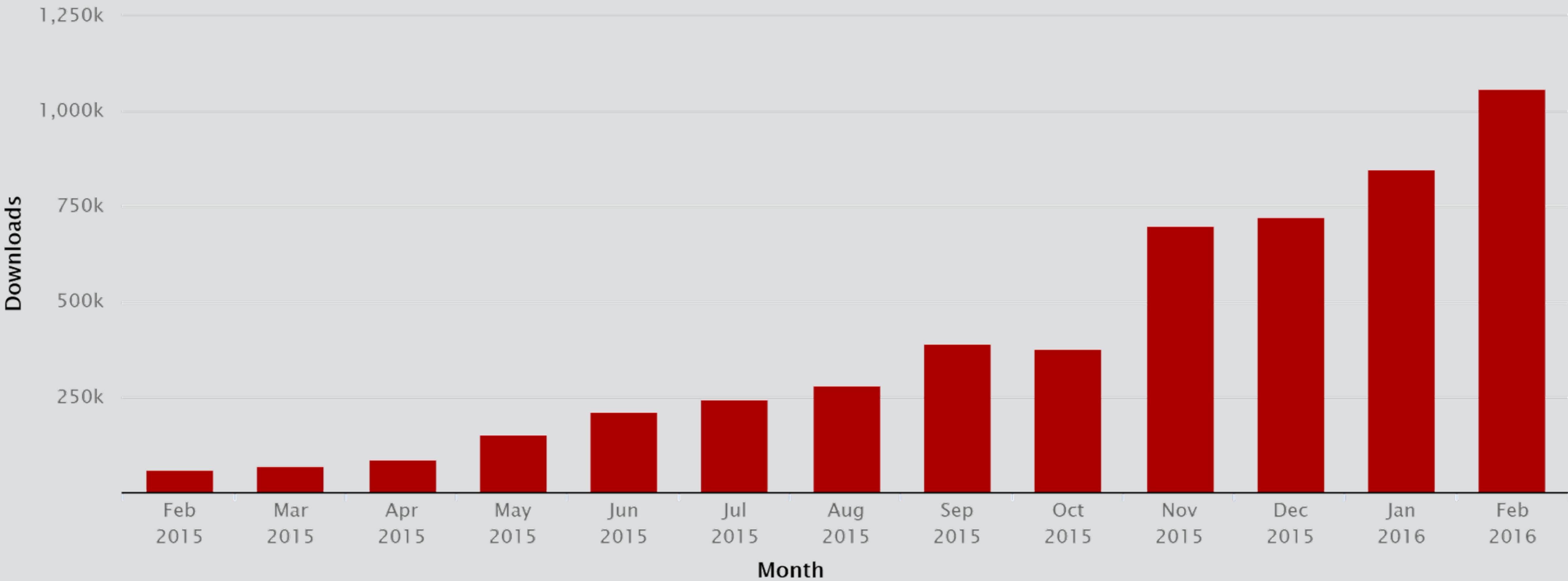
one year, four releases

A photograph of a large crowd of people at a concert or event. Many hands are raised in the air, some holding smartphones to take pictures. The scene is dimly lit with warm stage lights.

do people
USE?

Downloads per month

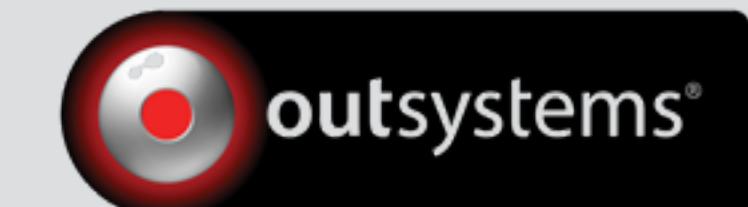
Click and drag in the plot to zoom in





dōjō

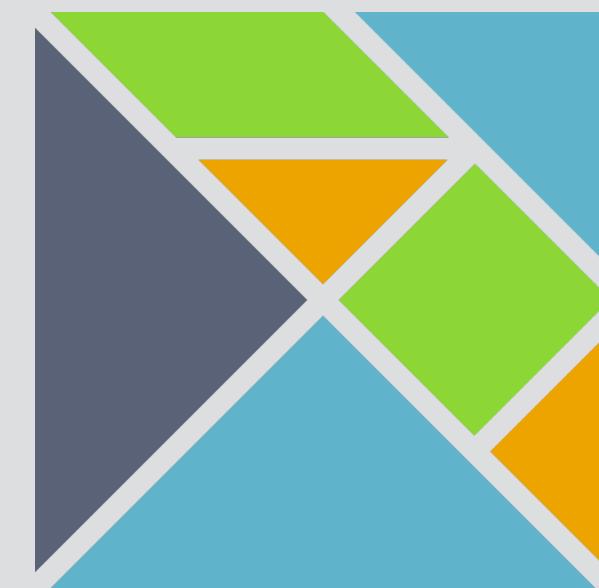




A woman with short brown hair, wearing a denim jacket over a red top, looks down and to her right with a thoughtful expression.

what ELSE?

 PURERESCIPT



 Idris

The screenshot shows a web browser window with the URL equinocios.com in the address bar. The page title is "equinociOS". The main content features a dark background with a starry space theme. On the left, there is a circular logo containing a stylized red and blue bird or feather. The main title "Swift In The Sky With Types" is displayed in large, bold, white font. Below it, a subtitle in Portuguese reads "Ou apenas mais uma conversa sobre linguagens, compiladores, tipos e... Swift!". A smaller text below the subtitle indicates the post was "Postado por Matheus Albuquerque em 28/03/2016". At the bottom of the main content area, there are social sharing icons for Twitter, Facebook, LinkedIn, and Google+. A sidebar on the left contains a note about the post's purpose and a numbered list of guidelines.

SOBRE

Swift In The Sky With Types

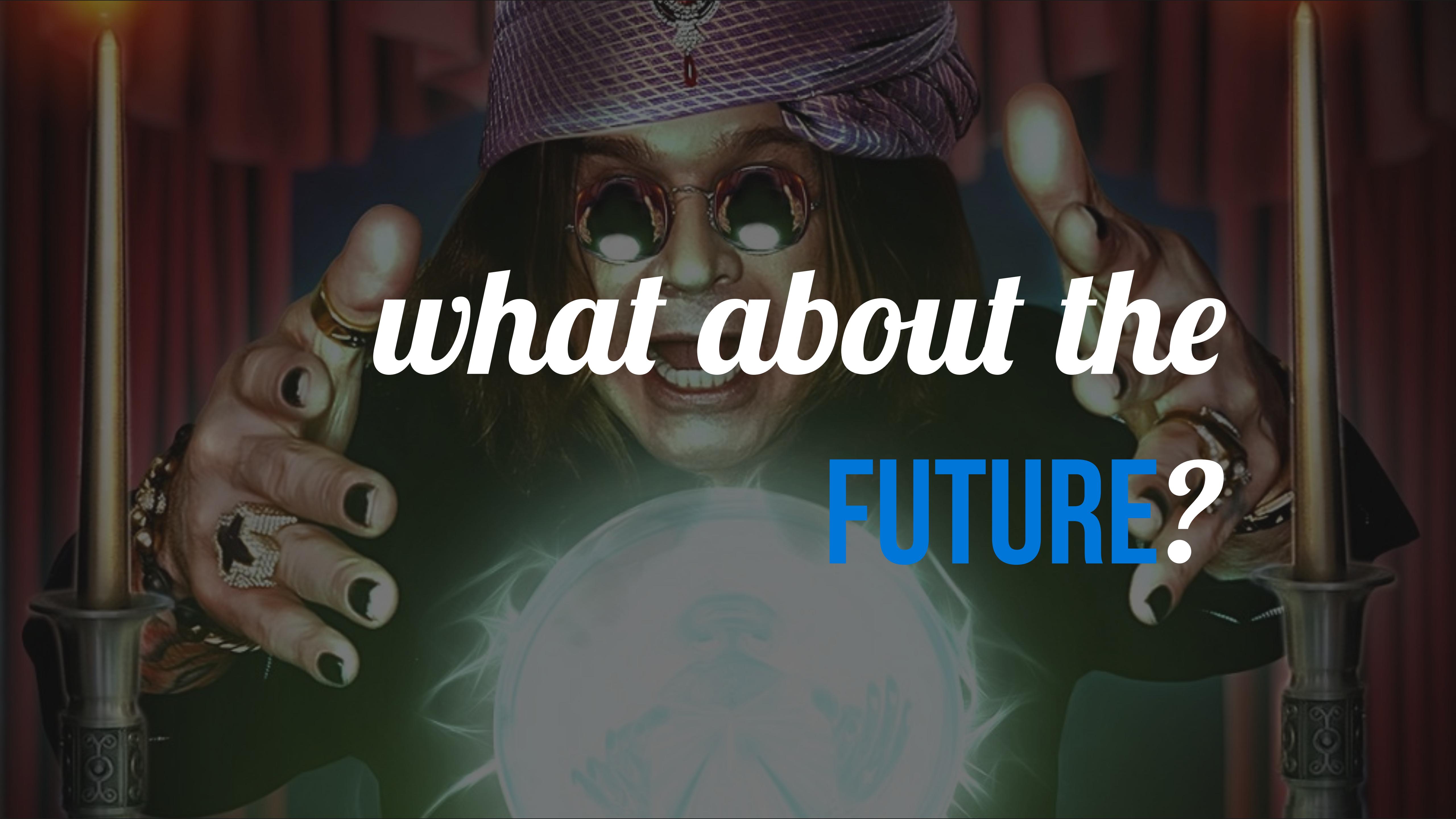
Ou apenas mais uma conversa sobre linguagens, compiladores, tipos e... Swift!

Postado por Matheus Albuquerque em 28/03/2016

Antes de tudo, algumas coisas a se levar em consideração:

1. Esse post é baseado em experiências - e eu não sou nenhum tipo de dono da verdade: se você tem alguma opinião sobre isso - concordando, discordando, complementando - , sinta-se livre para compartilhá-la.

Swift In The Sky With Types
<http://bit.ly/swift-types>

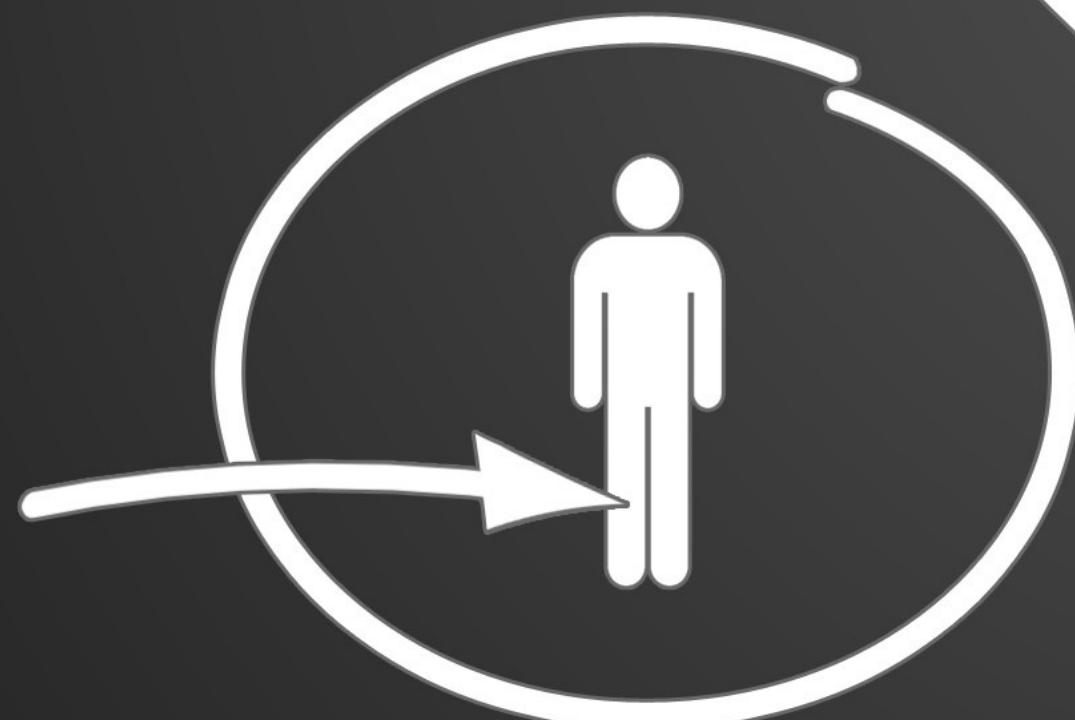
A woman with long brown hair, wearing a purple sequined dress and sunglasses, is looking through a crystal ball. She has a wide, excited smile on her face. Her hands are adorned with rings and bracelets. The background is dark and out of focus.

what about the

FUTURE?

Where the
magic happens

Your
comfort
zone



The screenshot shows the npm package page for 'typescript-closure-tools'. The page has a red header bar with the text 'Never Poke Monkeys' and the npm logo. The main content area features a large title 'typescript-closure-tools' with a 'public' badge, followed by a brief description: 'Command-line tools to convert closure-style JSDoc annotations to typescript, and to convert typescript sources to closure externs files'. Below this is a section titled 'Use it online' which converts 'Closure-style JSDoc type annotations to TypeScript definition files'. It explains the purpose of the tool: to assist developers writing TypeScript definitions on large libraries. The 'Setup' section provides a command-line setup script:

```
git clone git@github.com:fivetran/typescript-closure-tools.git
cd typescript-closure-tools
git submodule update --init
npm install
node definition-generator/src/main.js test/class.js class.d.ts # Run a single example
sudo npm install -g
```

The 'Usage' section shows the command: `closure2ts` [options]. The options listed are `--provides symbols.tsv` and `--globals output/dir/global-declarations.d.ts`.

On the right side of the page, there's a sidebar for 'npm Enterprise' with a call-to-action to 'Start a free trial'. Below that are links for npm commands (`npm i`), developer information ('how? learn more'), and a recent commit by `georgewfraser`. It also lists the latest release (0.0.6), the GitHub repository URL (`github.com/fivetran/typescript-closure-tools`), and sections for 'Collaborators' (with a placeholder icon) and 'Stats' (showing download counts for the last day, week, and month, and GitHub issue and pull request counts).

TypeScript Closure Tools
<http://bit.ly/ts-closure-tools>



lessons
LEARNT

- 1. TYPES CAN SERVE AS DOCUMENTATION**
- 2. THE COMPILER IS YOUR FIRST UNIT TEST**
- 3. THE COMPILER IS YOUR BEST FRIEND**



EASY 'n' FRINDLY?



thank

ABRAÃO
alves

MATHEUS
albuquerque

cejs



NETFLIX