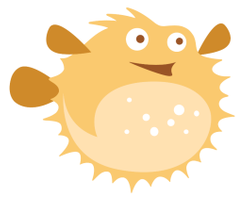
THE LINGUISTIC RELATIVITY OF PROGRAMMING LANGUAGES

Jenna Zeigen

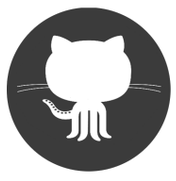
JSConf EU 2014

Hacker School Fall 2012 

Front-End Engineer at Bitly 

Brooklyn, NY, USA

 zeigenvector

 jennazee

bit.ly/jz-jsconfeu

1. Linguistic Relativity
2. … and Programming Languages?
3. Previous allusions
4. How it applies
5. Implications

(p.s. this isn't a coincidence)

SAPIR-WHORF HYPOTHESIS

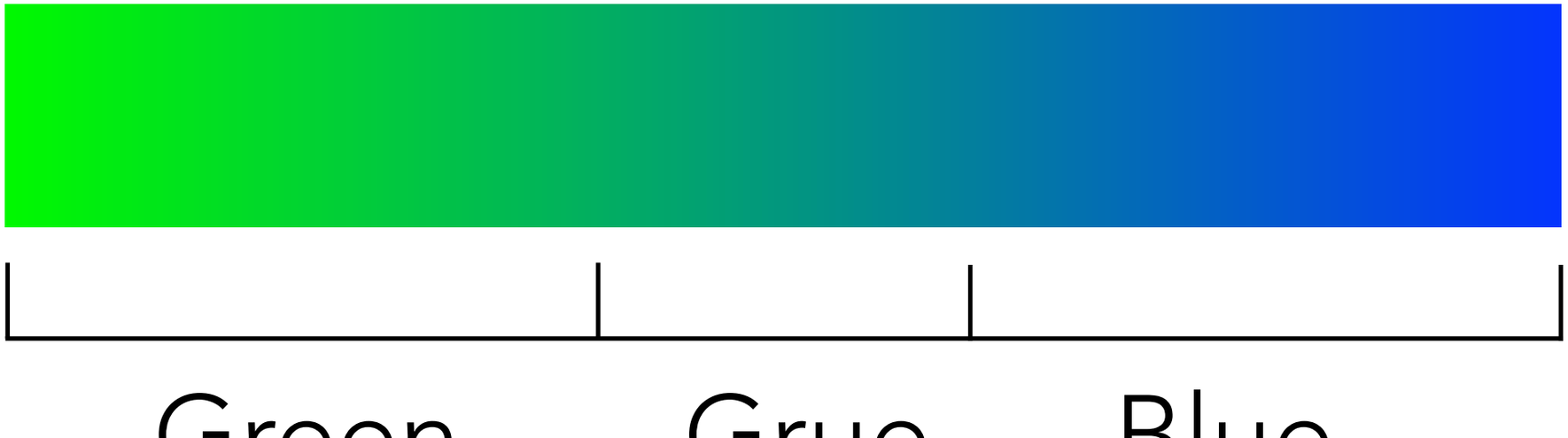
The languages you speak {determine | influence} the way you think.

SAPIR-WHORF HYPOTHESIS

The languages you speak {~~determine~~ | influence} the way you think.

SAPIR-WHORF HYPOTHESIS

SAPIR-WHORF HYPOTHESIS



But what about the

JavaScripts

??????????????

Proposal: The programming languages we know **strongly influence** the way we think about programming.

Programming languages **create** and **manipulate** the space, rather than just describe it.

"Programming languages, because they were designed for the purpose of directing computers, offer important advantages as **tools of thought.**"

Kenneth Iverson, "Notation as a Tool of Thought" (1979)

(bit.ly/Iverson-NotationAsToolOfThought)

BLUB PARADOX

Paul Graham, "Beating the Averages" (2003)

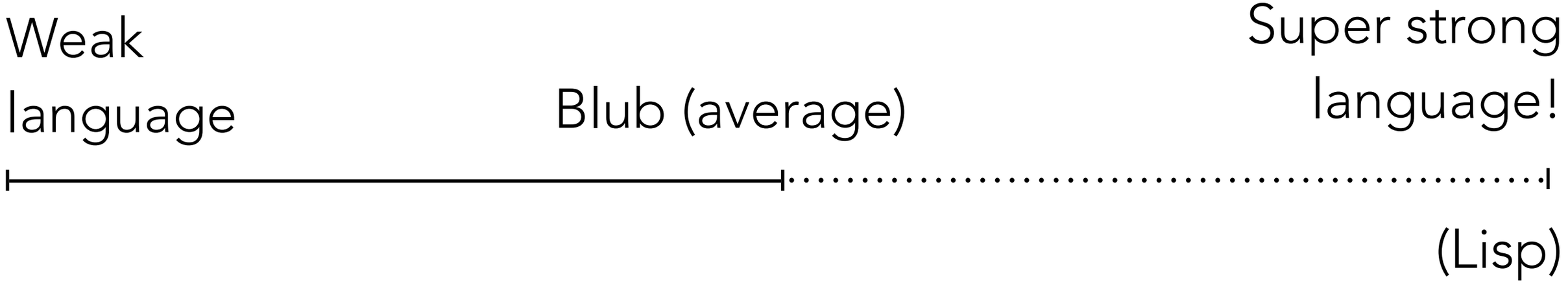
(bit.ly/blub-paradox)

BLUB PARADOX

"Some programming languages are more powerful than others."

BLUB PARADOX

"Some programming languages are more powerful than others."



BLUB PARADOX

"I look at [Python, Java, C, and Perl]. How can you get anything done in them, I think, without macros?"

Paul Graham, "Beating the Averages" (2003)

(bit.ly/blub-paradox)

BLUB PARADOX

"They're satisfied with whatever language they happen to use, because it **dictates** the way they think about programs."

Paul Graham, "Beating the Averages" (2003)

(bit.ly/blub-paradox)

BLUB PARADOX

"I know this from my own experience, as a high school kid writing programs in Basic. That language didn't even support recursion… but I didn't miss it at the time. **I thought in Basic.**"

Paul Graham, "Beating the Averages" (2003)

(bit.ly/blub-paradox)

SCARRED FOR LIFE?

"It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are **mentally mutilated beyond hope** of regeneration."

-Edsger Dijkstra, "How do we tell truths that might hurt" (1975)

(http://bit.ly/dijkstra-truths)


MENTALLY MUTILATED
BEYOND HOPE?!?11/?
memegenerator.net

We are influenced by the **constructs** and **idioms** of the most powerful programming language we know, not the languages themselves, or the language we are using at the time.

(http://bit.ly/cutest-red-panda)

1. We can learn more (powerful) programming languages and how to program in them idiomatically.

HUMANS CAN LEARN

iteration

HUMANS CAN LEARN

iteration

list comprehension

HUMANS CAN LEARN

iteration

list comprehension

map

2. We can implement the constructs of more powerful languages in whatever language we use.

CODE CAN MAKE THINGS

"We should now think of a language design for being a pattern for language designs, a tool for making more tools of the same kind."

Guy Steele, "Growing a Language"
(ACM OOPSLA 1998)

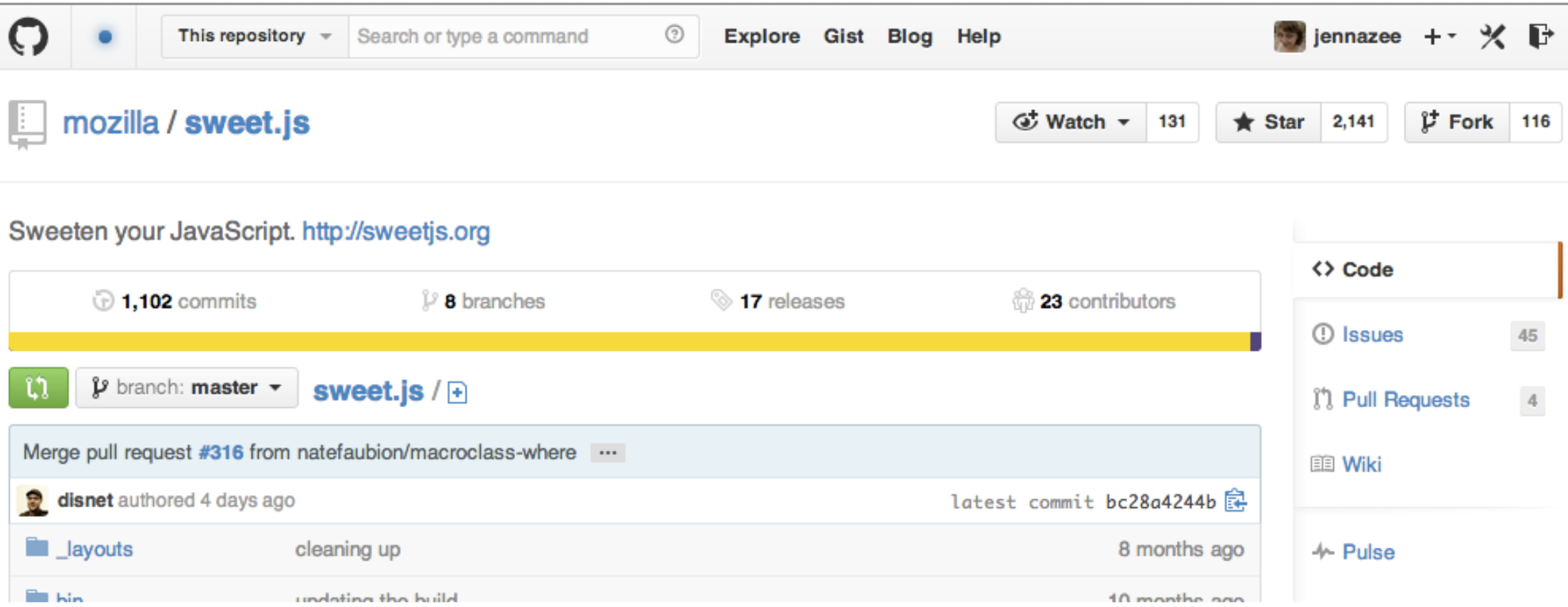
CODE CAN MAKE THINGS

```
_.map(array, function(el) {
    console.log(el);
})
```

```
array.map(function(el) {
    console.log(el);
})
```

CODE CAN MAKE THINGS

"Sweet.js brings the hygienic macros of languages like Scheme and Rust to JavaScript. Macros allow you to… **craft the language you've always wanted.**"



CODE CAN MAKE THINGS

The future
can be now!
(kinda…)

3. Programming languages are synthetic and can change if we want them to.

SYNTHETIC LANGUAGES CAN BE CHANGED

iteration

map

SYNTHETIC LANGUAGES CAN BE CHANGED

iteration

map

array comprehension

generators

Languages that can’t easily grow will die

Guy Steele, “Growing a Language” (1998)


CLJS

or

Learn new languages, find cool things, bring them back, and share!

Slides: bit.ly/jz-jsconfeu

Blog Post: bit.ly/ling-rel-prog

zeigenvector

Thanks!

↑

Me