


It's all (position:) relative:

# 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`

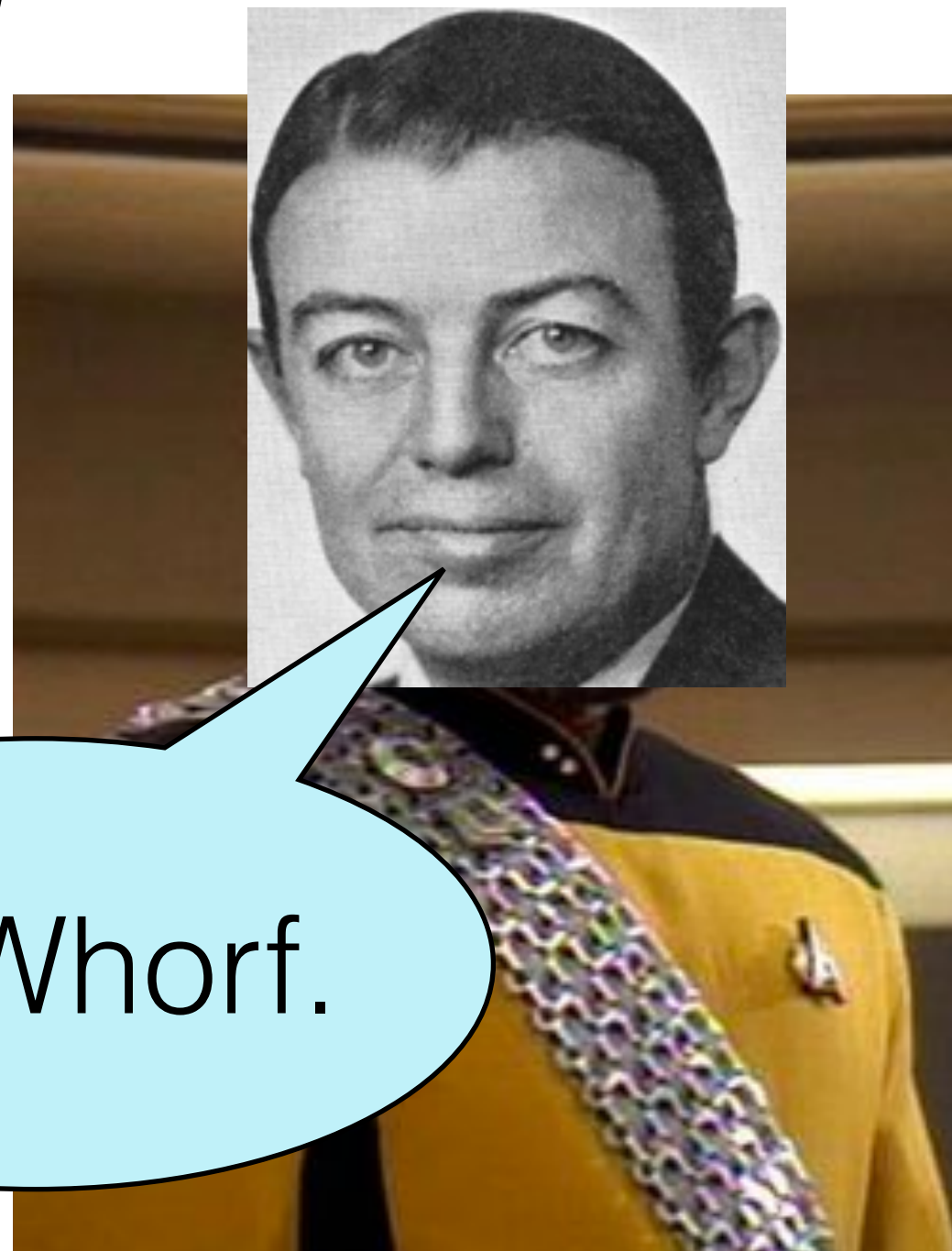
[bit.ly/jz-jsconfeu](http://bit.ly/jz-jsconfeu)

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

I'm Sapir.



I'm Whorf.



# Sapir-Whorf Hypothesis

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

# Sapir-Whorf Hypothesis

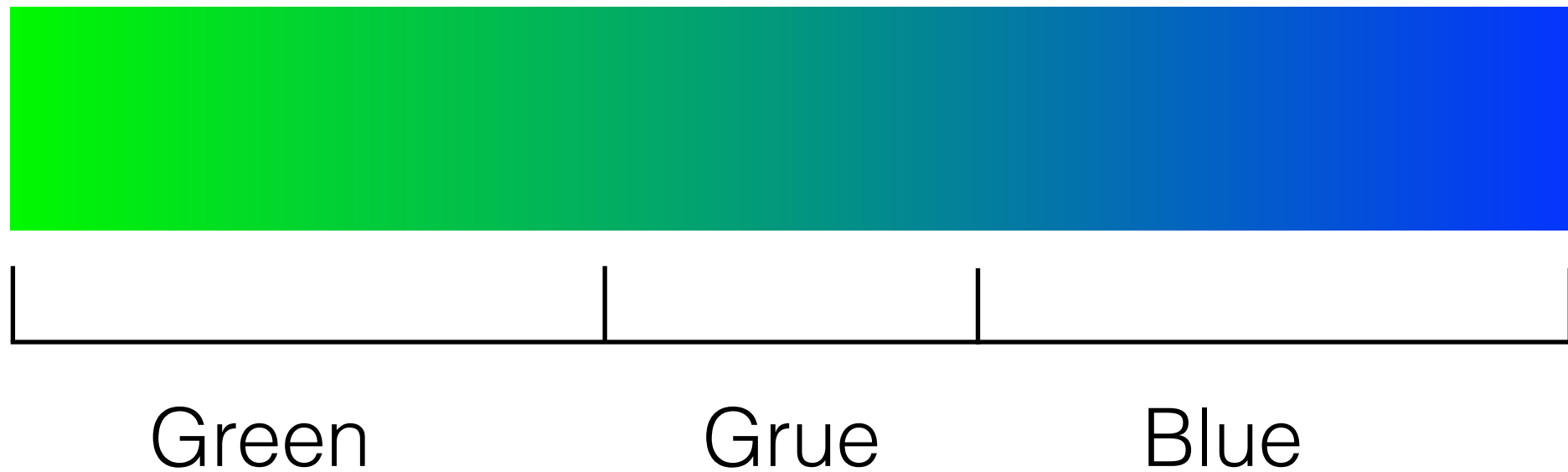
The languages you speak  
~~{determine |~~ influence} the way  
you think.

# 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](https://bit.ly/Iverson-NotationAsToolOfThought))

# Blub Paradox



Some  
programming languages  
are more powerful than  
others.

*Paul Graham, “Beating the Averages” (2003)*

([bit.ly/blub-paradox](http://bit.ly/blub-paradox))

# Blub Paradox

“Some programming languages are more powerful than others.”

Weak  
language

Blub (average)



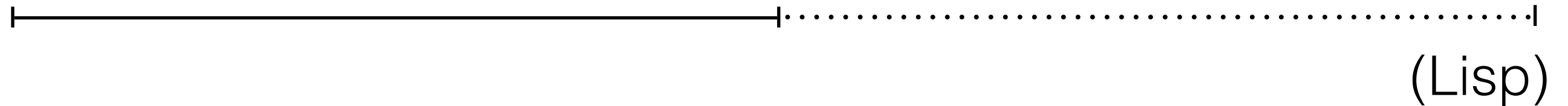
# Blub Paradox

“Some programming languages are more powerful than others.”

Weak  
language

Blub (average)

Super strong  
language!





# 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](http://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](http://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](http://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](http://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.”

The screenshot shows the GitHub repository page for `mozilla/sweet.js`. The repository is public and has 131 watches, 2,141 stars, and 116 forks. It features 1,102 commits, 8 branches, 17 releases, and 23 contributors. A recent merge pull request #316 from `natefaubion/macroclass-where` is highlighted, authored by `disnet` 4 days ago. The latest commit is `bc28a4244b`. The repository description is "Sweeten your JavaScript. <http://sweetjs.org>". The right sidebar shows links to Code, Issues (45), Pull Requests (4), Wiki, and Pulse.

GitHub interface showing the repository `mozilla / sweet.js`. The repository is public and has 131 watches, 2,141 stars, and 116 forks. The repository description is "Sweeten your JavaScript. <http://sweetjs.org>". The repository statistics show 1,102 commits, 8 branches, 17 releases, and 23 contributors. A recent merge pull request #316 from `natefaubion/macroclass-where` is highlighted, authored by `disnet` 4 days ago. The latest commit is `bc28a4244b`. The repository structure shows files like `_layouts` and `bin`.

Code can make things.

The future  
can be now  
(kinda).

3. Programming languages are synthetic and can change.



Synthetic languages can be changed

iteration

map

Synthetic languages can be changed

iteration

map

array comprehension

generators

Synthetic languages can be changed

Languages that can't easily  
grow will die

*Guy Steele, "Growing a Language" (1998)*

CLJS

The logo consists of the letters 'CLJS' in a bold, white, sans-serif font, centered within a solid magenta rectangular background.

CLJS

or

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

[bit.ly/jz-jsconfeu](https://bit.ly/jz-jsconfeu)



zeigenvector