It's all (position:) relative:

Linguistic Relativity and Programming Languages

Jenna Zeigen

BrooklynJS, May 2014

bit.ly/bk-js

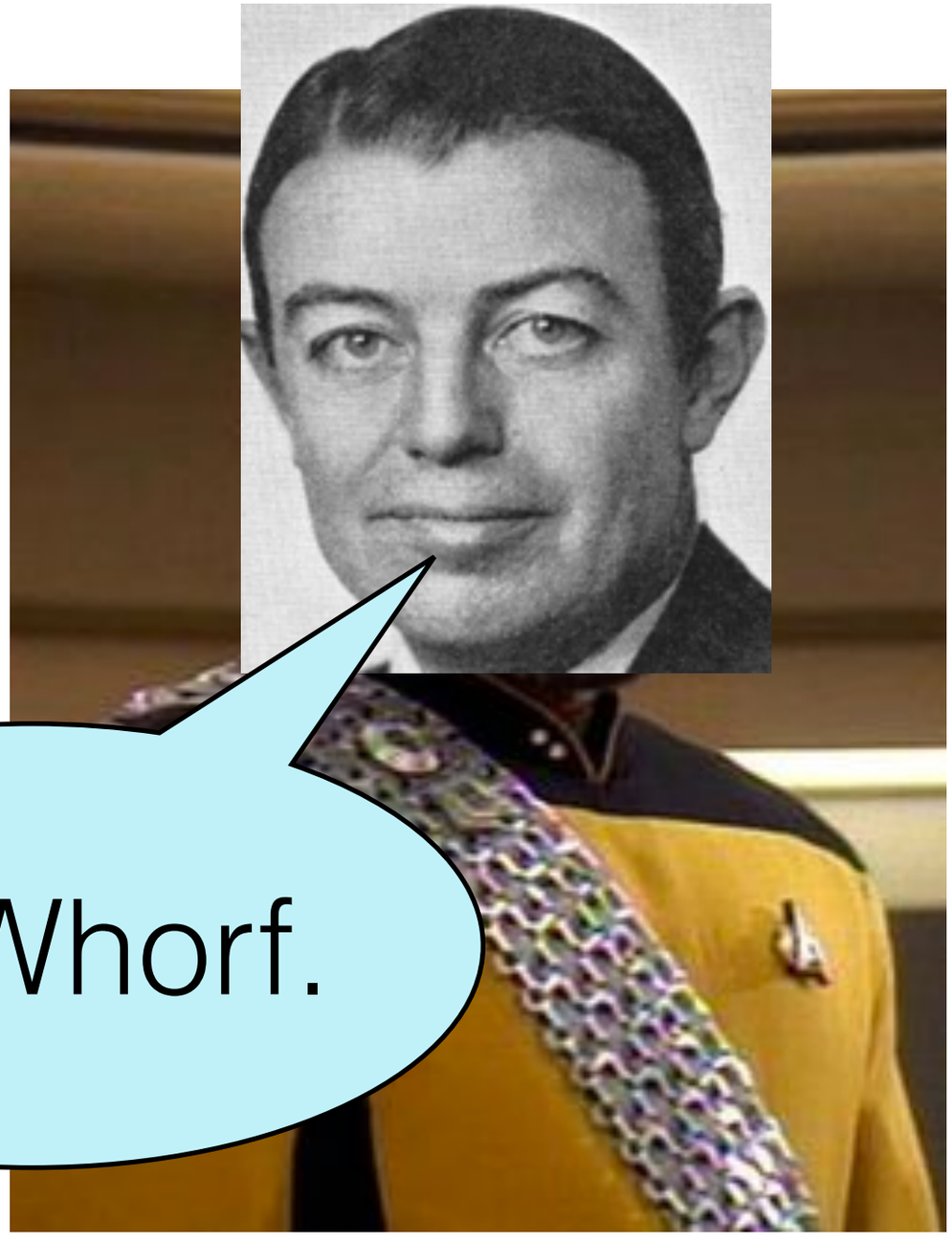
Hacker School Fall 2012

Front-End Engineer at Bitly

@zeigenvector //

jennazee

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.

But what about the JavaScripts ??????????????????

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

Programming languages can **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**. Not only are they universal…, but they are also **executable** and **unambiguous**.”

Kenneth Iverson, “Notation as a Tool of Thought”

(bit.ly/Iverson-NotationAsToolOfThought)

Blub Paradox

Paul Graham, "Beating the Averages"

(bit.ly/blub-paradox)

Blub Paradox

"Some programming languages are more powerful than others."

Weak language

Blub (Exceedingly average)

Super strong language!

(LISP)

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"

(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”

(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."

-*Dijkstra, "How do we tell truths that might hurt"*

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

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

We are only “constrained” by the constructs and idioms of the most powerful programming language we know.

1. We can learn more (powerful) programming languages and program in the idiomatically.

Humans can learn.

iteration

Humans can learn.

iteration

list comprehension

Humans can learn.

iteration

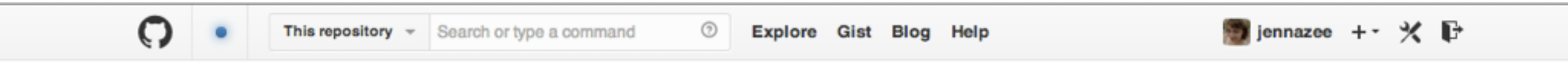
list comprehension

map

2. We can implement the constructs of more powerful languages in other programming languages.

Code can make things.

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



PUBLIC mozilla / sweet.js

Watch 131 Star 2,141 Fork 116

Sweeten your JavaScript. http://sweetjs.org

1,102 commits 8 branches 17 releases 23 contributors

branch: master sweet.js /

Merge pull request #316 from natefaubion/macroclass-where ...

disnet authored 4 days ago latest commit bc28a4244b

_layouts	cleaning up	8 months ago
bin	updating the build	10 months ago

Code
Issues 45
Pull Requests 4
Wiki
Pulse

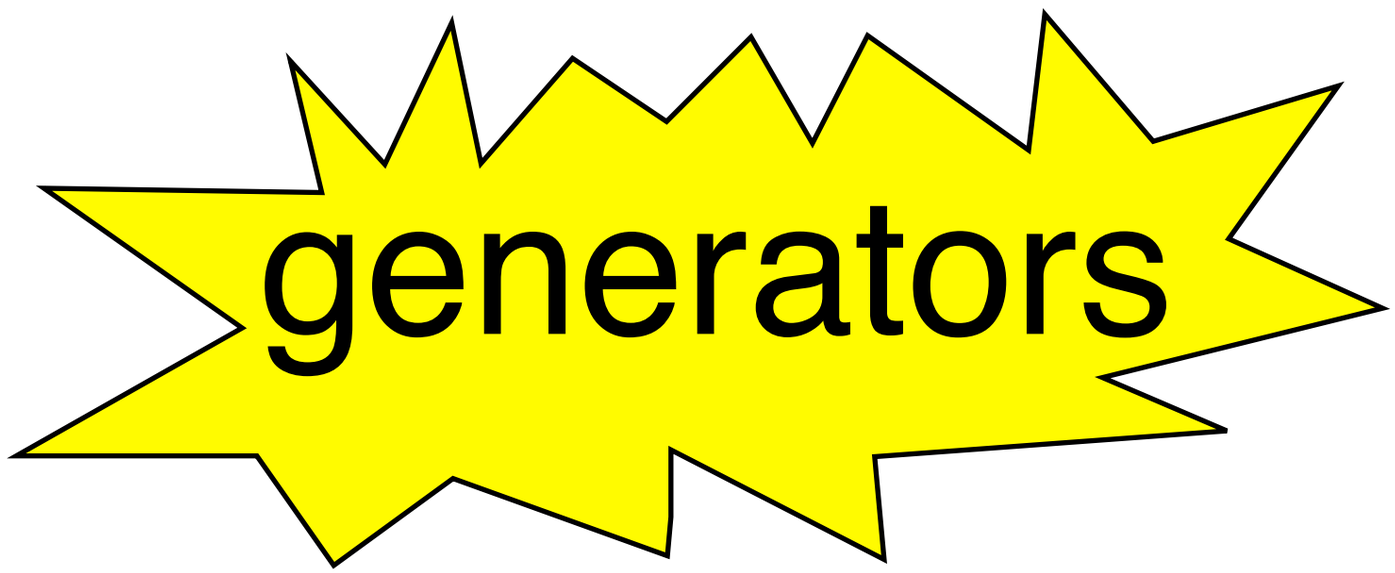
3. Programming languages themselves can change if we want them to.

Synthetic languages can be changed

iteration

list comprehension

map



bit.ly/bk-js

Questions?

@zeigenvector

or

Find me downstairs at beer.js!