

# Command Performance

the why and whether of  
small-scale optimizations

David A. Black  
Director, Ruby Power and Light, LLC

Philly ETE 2011

# The familiar quotation....

Premature optimization is the root of all evil.

-- Donald Knuth and/or Sir Tony Hoare

- Never seems *quite* right or *quite* wrong
- Already discussed at enormous length elsewhere
  - including analysis of misunderstandings
- Taking as suggestive point of departure
- Not trying to prove or disprove the idea itself

# Four “chapters”

- Prematurity and software development
- Optimization vs. melioration
- Optimization as a reflex
- Small-scale optimizations in Ruby

*First code demo*

# Prematurity and software development

# Software and soap opera

- Soap operas are “all middle” (cf. Robert Allen, *Speaking of Soap Operas*)
- Beginnings often lost in mists of time
- End nowhere in sight
  - and ending often viewed as a kind of failure

# Iterative vs. singulative

- Gérard Genette, literary theorist
- Iterative: habitual, repeated actions or events
- Singulative: unique events
- Software cycles: a bit of both



# (Pre)maturity and software

- How do you know?
- What is maturity in an iterative process?

*Second code demo*

# Optimization vs. melioration

# Meliorism

- Possibly coined by novelist George Eliot
- Between optimism and pessimism
  - things can get better and be made better
- As a coding philosophy, cf. Bob Martin et al.
  - Leave code better than how you found it

*Third code demo*

Optimization as a reflex

Given the choice between two ways of writing a line of code, both with similar readability, writability, and maintainability properties, and yet interestingly different performance profiles, don't be a bozo: choose the performant approach.

-- Joe Duffy

# Optimization by attrition

```
def my_method  
  a = 1  
  b = a  
  c = b  
  return c * 10  
end
```

```
def my_method  
  10  
end
```



*Fourth code demo*

# Small-scale optimizations in Ruby

- A bit of wishful thinking
  - Could there be a set of performant idioms we can all gravitate to?
- Some idioms seem inherently faster
  - but this runs aground on different interpreters
- Fallback: measure everything
  - (but don't obsess about optimizations where they really won't matter)

Questions? Discussion?

# Thank you!

David A. Black

[dblack@rubypal.com](mailto:dblack@rubypal.com)

[@david\\_a\\_black](#)