

Understanding Ruby's Pros and Cons

“MY TWO CENTS”



Paolo Perrotta

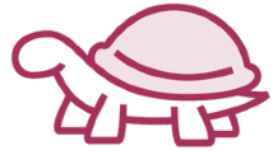
FREELANCE DEVELOPER

@nusco

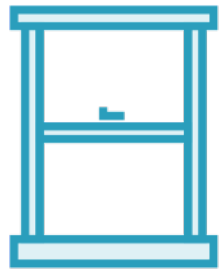
What follows is just my
humble opinion.

Ruby's Shortcomings

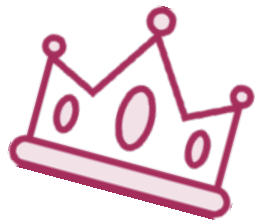
Ruby's Shortcomings



Slow for high-performance applications (realtime, ML...)



Doesn't have the best support for Windows



Not as popular as it used to be



It makes it easy to shoot your own foot

How to Create a Bug

```
def classic_jazz?(album)
  (return album[2] == "Jazz") and album[3].to_i.between?(1950, 1959)
end
```



The diagram illustrates a bug in the function `classic_jazz?`. The function is defined as:

```
def classic_jazz?(album)
  (return album[2] == "Jazz") and album[3].to_i.between?(1950, 1959)
end
```

Two orange arrows point to the conditions in the `and` statement. An orange explosion icon is placed over the `and` operator, indicating a logical error where the second condition is never evaluated due to short-circuiting.

Ruby's Benefits

Ruby's Benefits



Great as a general-purpose scripting language (and for ETL, web apps...)



Productive and convenient for rapid prototyping



Powerful, with expressive and consistent syntax



Optimized for programmer happiness

“Ruby is designed to make
programming not only easy but
also fun.”

Yukihiro Matsumoto

Wrapping up This Training

Summary

We wrote a short Ruby program

- Blocks, selecting from an array...

We touched on Ruby's OOP features

- Mostly everything is an object

We gazed into metaprogramming

- Powerful features and DSLs

We explored Ruby's ecosystem

- Interpreters, libraries, community...

We listed Ruby's pros and cons

- What it's good at, what it's not so good at

Thank You!



Paolo Perrotta
FREELANCE DEVELOPER
@nusco