

Free Formatter For You, Jim

Free Formatter For You, Jim
(You gotta install it but the formatter is free)



Style rules in linters are bad
(when you can just have a formatter)

You don't need a tool that tells you "Ooops, forgot a semicolon there!" when it can just remove it for you.

DEMO

(...) for me it's about freeing up the brain/focus space by not having to worry about formatting or semicolons or whitespace or any of that, my focus can remain on the problem I'm solving.

— *@brianloveswords*

Working in environments without an automatic code formatter makes me feel like I have to keep a context (that's unrelated to the problem space) in my head at all times of satisfying a linter or style guide or whatever.

— [*@brianloveswords*](#)

Wait, where did it come from?



Joyce Kilmer and most
computer scientists agree:
there is no poem as lovely as
a tree.

— *Philip Wadler, "A prettier printer" (2003)*

Abstract Syntax Trees

```
>> Parser::CurrentRuby.parse("2 + 4")  
=> s(:send,  
    s(:int, 2), :+,  
    s(:int, 4))
```

Abstract Syntax Trees

```
>> Parser::CurrentRuby.parse("foo.bar.baz")  
=> s(:send,  
    s(:send,  
        s(:send, nil, :foo), :bar), :baz)
```

Abstract Syntax Trees

```
>> Parser::CurrentRuby.parse("foo\n.bar\n.baz")  
=> s(:send,  
    s(:send,  
        s(:send, nil, :foo), :bar), :baz)
```

Abstract Syntax Trees

```
>> Parser::CurrentRuby.parse("foo.bar().baz()")  
=> s(:send,  
    s(:send,  
        s(:send, nil, :foo), :bar), :baz)
```

FUTURE

FUTURE

for ASTs in our lives

FUTURE

for Ruby formatters

Future for Ruby formatters

```
class Foo
  # ...

  def bar
    # ...

    baz
    # What is `baz`?
    # A variable? A method call?
    # Should I put parens after it?
  end
end
```

Future for Ruby formatters

Prefer `do...end` over `{...}` for procedural blocks.

Good: return value is not used out of scope

```
map do |x|  
  x  
end
```

Prefer `{...}` over `do...end` for functional blocks.

Good: return value is used/assigned

```
foo = map { |x|  
  x  
}  
map { |x|  
  x  
}.inspect
```

```
yarn add prettier  
prettier.io
```