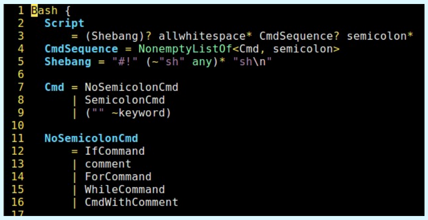
RAinBoWs

"Code the rainbow"

In traditional programming languages, the code determines the colors.

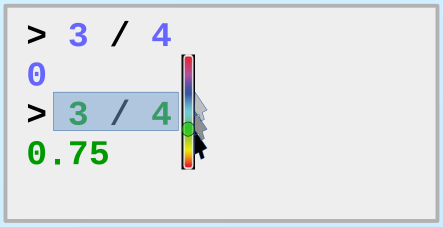


```
Bash {
  Script
      = (Shebang)? allwhitespace* CmdSequence? semicolon*
  CmdSequence = NonemptyListOf<Cmd, semicolon>
  Shebang = "#!" (~"sh" any)* "sh\n"

  Cmd = NoSemicolonCmd
      | SemicolonCmd
      | ("" ~keyword)

  NoSemicolonCmd
      = IfCommand
      | comment
      | ForCommand
      | WhileCommand
      | CmdWithComment
```

This isn't very natural though. In real life, the colors are part of the objects themselves

Colors are often manipulated in art to give the piece different meaning. Programming is art too!

If variable types are represented by colors of text, changing behavior is as easy as picking a different color on a slider.



```
> 3 / 4
0
> 3 / 4
0.75
```

Type checking is as easy as looking for matching colors.

```
foo = bar(a, b);

foo2 = bar(c, d);
```

Code & data are easy to distinguish, eliminating the need for quotes. This completely eliminates the security problem of escaping quotes.

```
shell("python -c \"print \\\"Hello world\\\"\"");
```

↓

```
shell(python -c "print \"Hello world\"");
```

Custom types get custom colors, as denoted by their constructors.

```
class Circle {
  constructor(x, y, rad);
}
```