

Code Style Sheets^{*}

CSS For Code

Sam Cohen and Ravi Chugh

February 17, 2025

*Much of the material in this presentation is from our upcoming paper to OOPSLA 25

Code Displays Haven't Changed Much



Plain Text

A screenshot of a code editor window. The window title is "convoc.c". The code is displayed in a monospaced font with syntax highlighting. The code includes a function definition for "mprConvoEval" and several conditional blocks. The status bar at the bottom shows "Ln 160, Col 1", "Spaces: 4", "UTF-8", "LF", and "Plain Text".

```
Users > sam > Coursework > SciViz > code > samcohen-scivis-2025 > p2mapr > C convoc.c
99  mprConvoEval(mprCtx *ctx, real xw, real yw) {
139
140      if(mprVerbose > 10) {
141          mprPrintf("xi0: %g, xi1: %g\n", xi0, xi1);
142          mprPrintf("n0: %d, n1: %d\n", n0, n1);
143          mprPrintf("a0: %g, a1: %g\n", a0, a1);
144          mprPrintf("iLo: %d, iHi: %d\n", iLo, iHi);
145      }
146
147      if(iLo + n0 < 0 ||
148         iLo + n1 < 0 ||
149         iHi + n0 >= (int)ctx->image->size[0] ||
150         iHi + n1 >= (int)ctx->image->size[1]) {
151          ctx->outside = true;
152          return;
153      }
154
155      // Evaluate the kernel everywhere we need to.
156      real k0Evald[s];
157      real dk0Evald[s];
158      real k1Evald[s];
```

Code Displays Haven't Changed Much

Syntax Highlighting



Plain Text

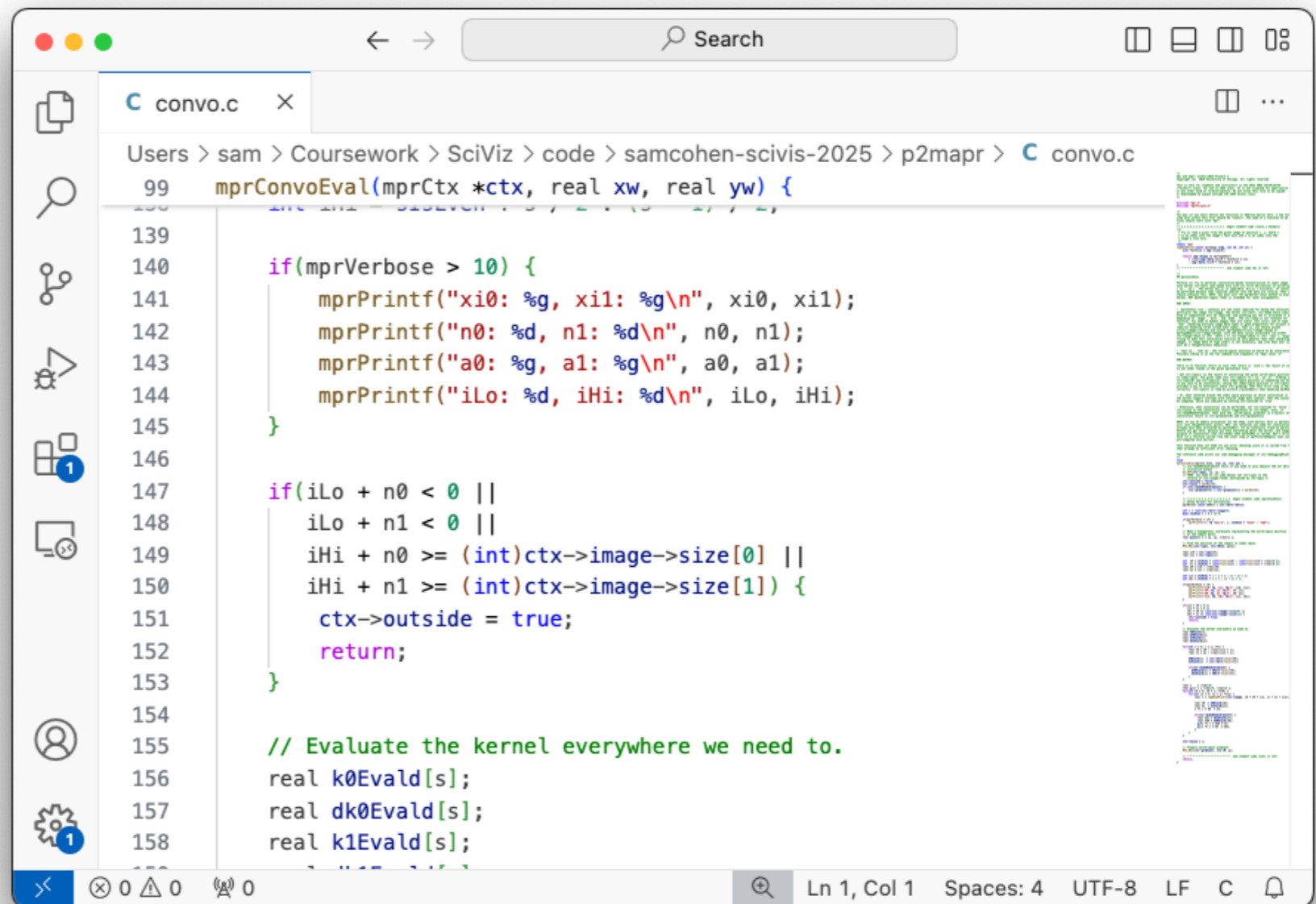
A screenshot of a code editor window titled "convoc.c". The code is syntax-highlighted with various colors: keywords in blue, identifiers in black, strings in red, and comments in green. The code is as follows:

```
99 mprConvoEval(mprCtx *ctx, real xw, real yw) {
139
140     if(mprVerbose > 10) {
141         mprPrintf("xi0: %g, xi1: %g\n", xi0, xi1);
142         mprPrintf("n0: %d, n1: %d\n", n0, n1);
143         mprPrintf("a0: %g, a1: %g\n", a0, a1);
144         mprPrintf("iLo: %d, iHi: %d\n", iLo, iHi);
145     }
146
147     if(iLo + n0 < 0 ||
148        iLo + n1 < 0 ||
149        iHi + n0 >= (int)ctx->image->size[0] ||
150        iHi + n1 >= (int)ctx->image->size[1]) {
151         ctx->outside = true;
152         return;
153     }
154
155     // Evaluate the kernel everywhere we need to.
156     real k0Evald[s];
157     real dk0Evald[s];
158     real k1Evald[s];
```

Code Displays Haven't Changed Much

And styles are applied in a monolithic fashion, independent of task

- Writing
- Debugging
- Reading
- Refactoring
- Presenting

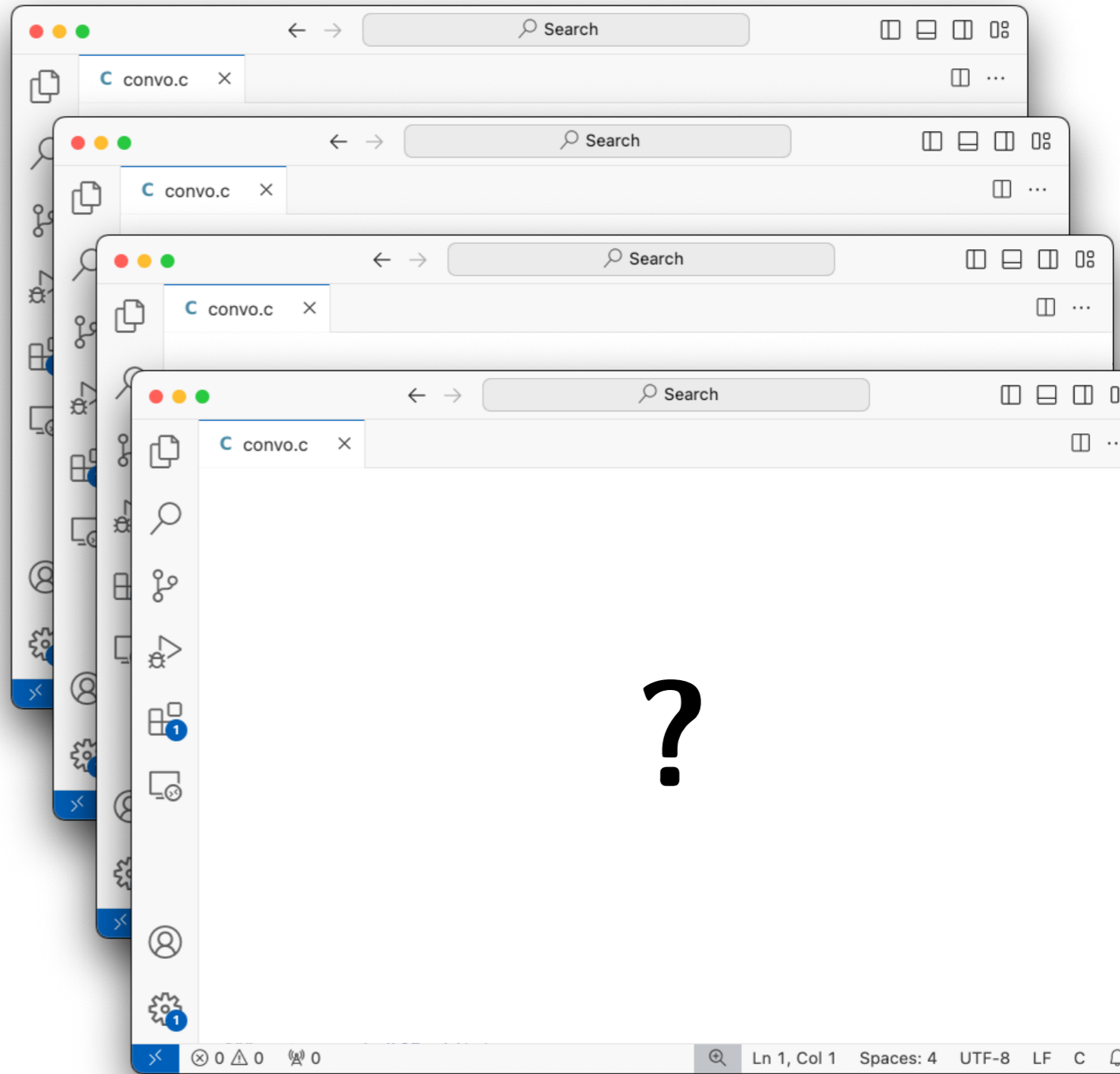
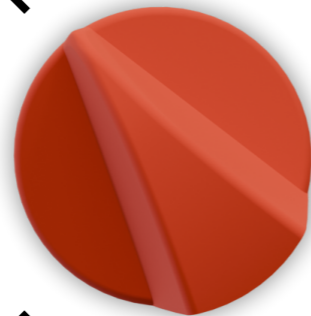


The screenshot shows a code editor window titled 'convo.c'. The code is displayed with a monolithic styling scheme where all text is in a single color (black or green), regardless of its syntactic role. The code includes a function `mprConvoEval` with several `printf` statements and conditional logic. The editor interface includes a search bar, a file explorer on the left, and a status bar at the bottom showing 'Ln 1, Col 1 Spaces: 4 UTF-8 LF C'.

```
99  mprConvoEval(mprCtx *ctx, real xw, real yw) {
139
140      if(mprVerbose > 10) {
141          mprPrintf("xi0: %g, xi1: %g\n", xi0, xi1);
142          mprPrintf("n0: %d, n1: %d\n", n0, n1);
143          mprPrintf("a0: %g, a1: %g\n", a0, a1);
144          mprPrintf("iLo: %d, iHi: %d\n", iLo, iHi);
145      }
146
147      if(iLo + n0 < 0 ||
148         iLo + n1 < 0 ||
149         iHi + n0 >= (int)ctx->image->size[0] ||
150         iHi + n1 >= (int)ctx->image->size[1]) {
151          ctx->outside = true;
152          return;
153      }
154
155      // Evaluate the kernel everywhere we need to.
156      real k0Evald[s];
157      real dk0Evald[s];
158      real k1Evald[s];
```

Idea: Code Style Sheets

- Writing
- Debugging
- Reading
- Refactoring
- Presenting



A Motivating Example

```
main =  
  getContents  
  >>= print  
    . length  
    . filter $ isPrefixOf "--"  
    . lines
```

```
-- Anonymous Authors  
module Main where  
  
-- Print a greeting  
main =  
  putStrLn "Hello, PLATEAU!"
```

Expected output:
4

A Motivating Example

```
main =  
  getContents  
  >>= print  
    . length  
    . filter $ isPrefixOf "--"  
    . lines
```

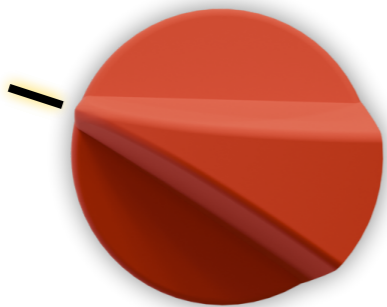
```
-- Anonymous Authors  
module Main where  
  
-- Print a greeting  
main =  
  putStrLn "Hello, PLATEAU!"
```

Expected output:
4

Can you spot the bug?

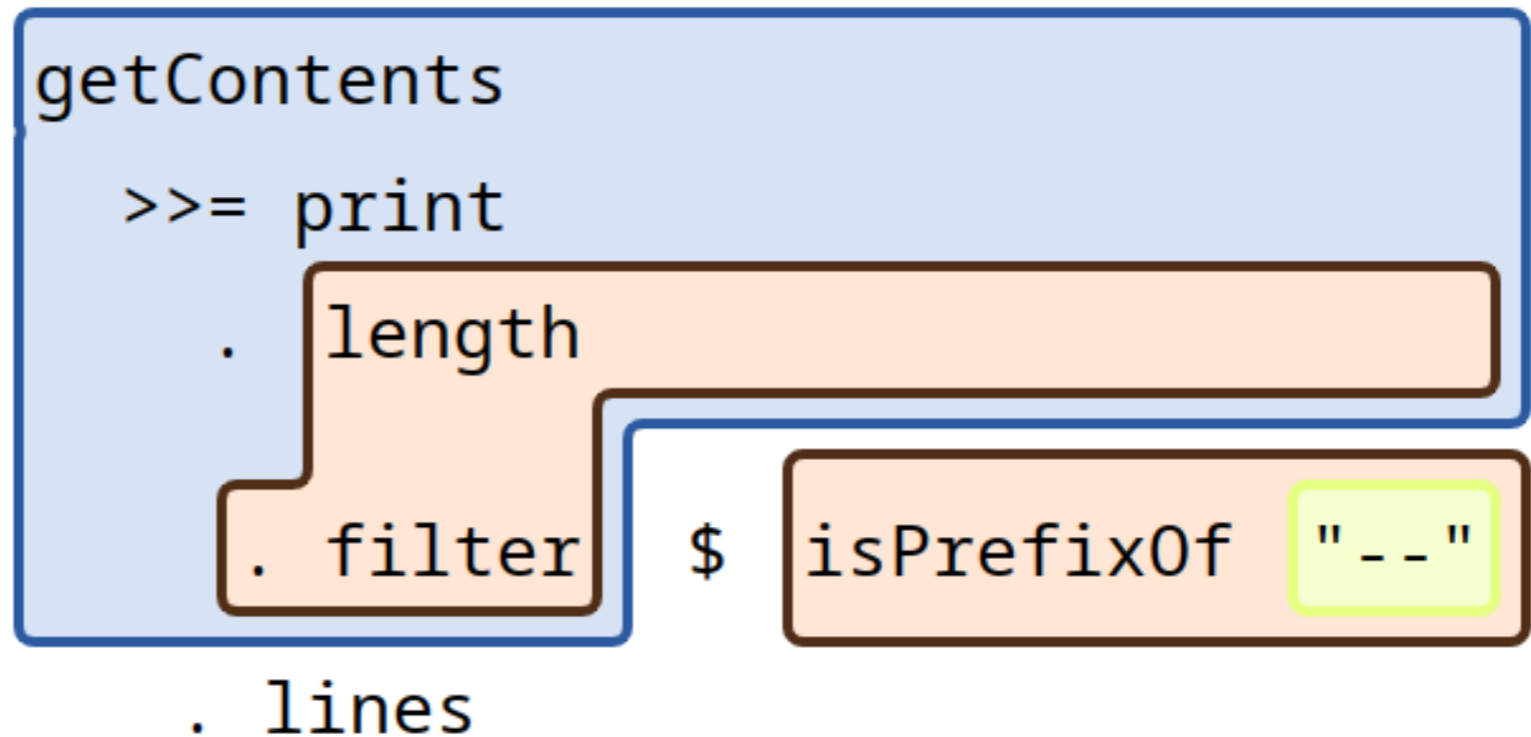
```
main =  
  getContents  
  >>= print  
    . length  
    . filter $ isPrefixOf "--"  
    . lines
```

**Syntax
Highlighting**



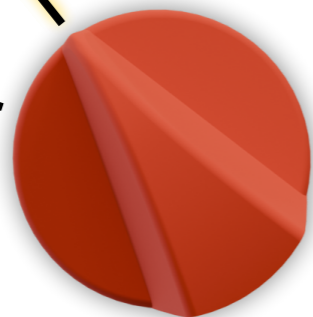

```
main =
  getContents
  >>= print
  . length
  . filter $ isPrefixOf "--"
  . lines
```

main =



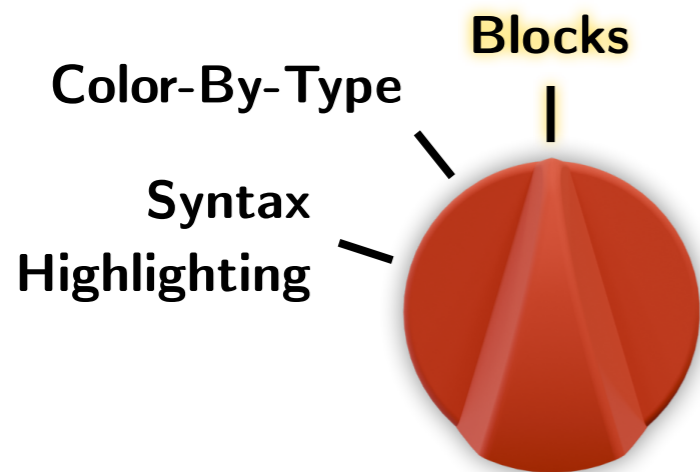
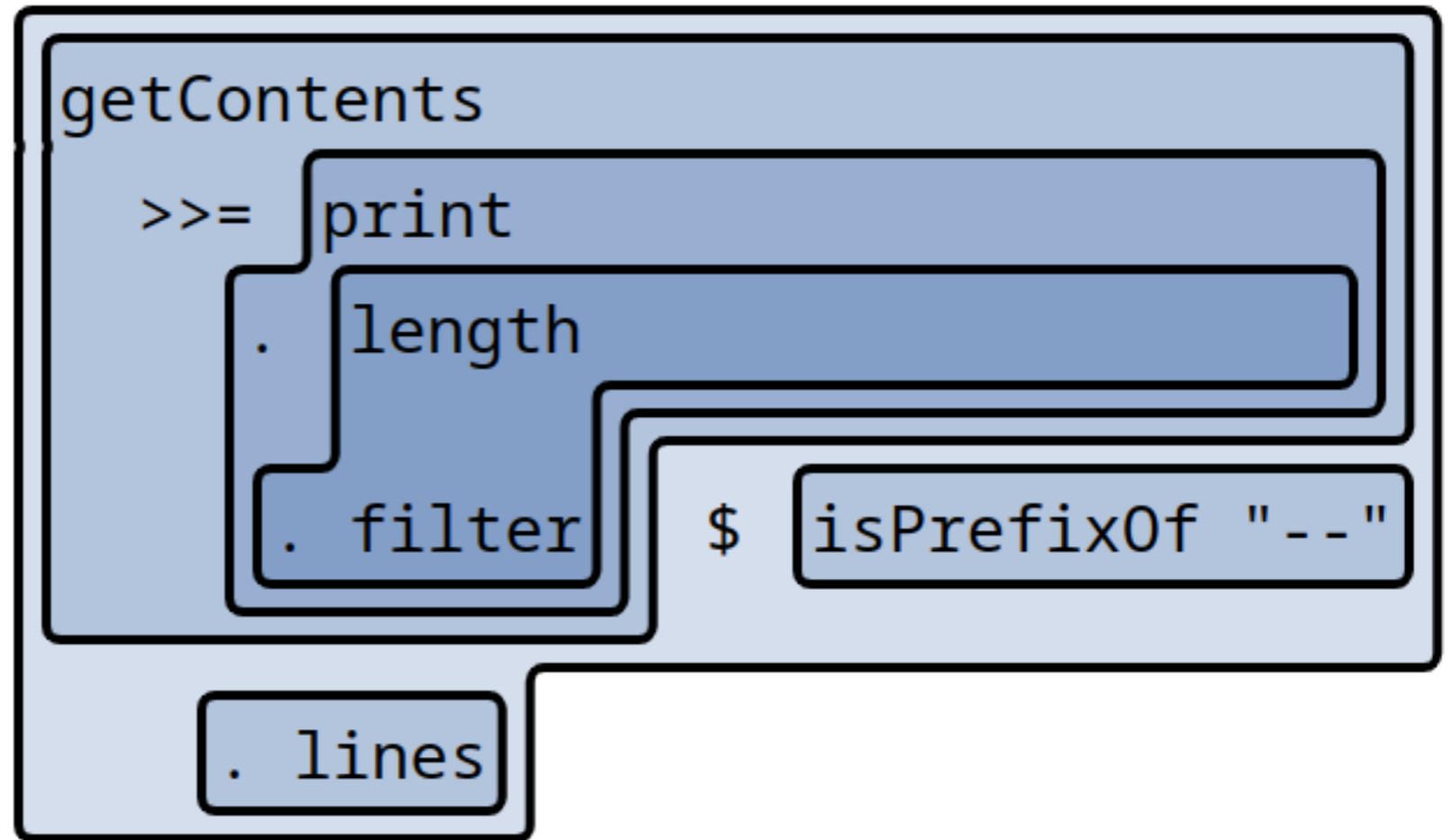
Color-By-Type

Syntax
Highlighting



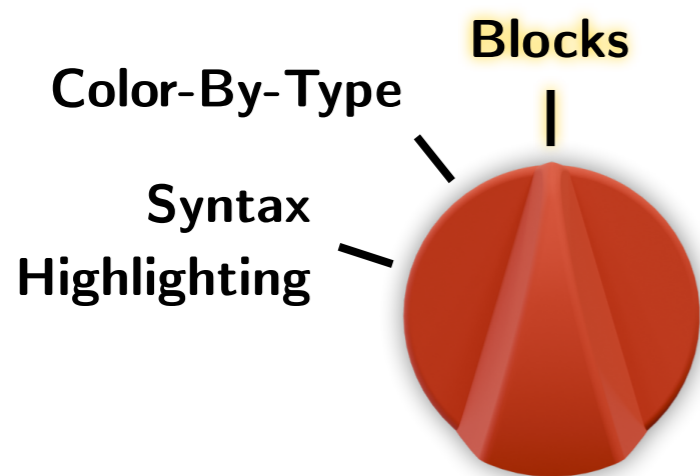
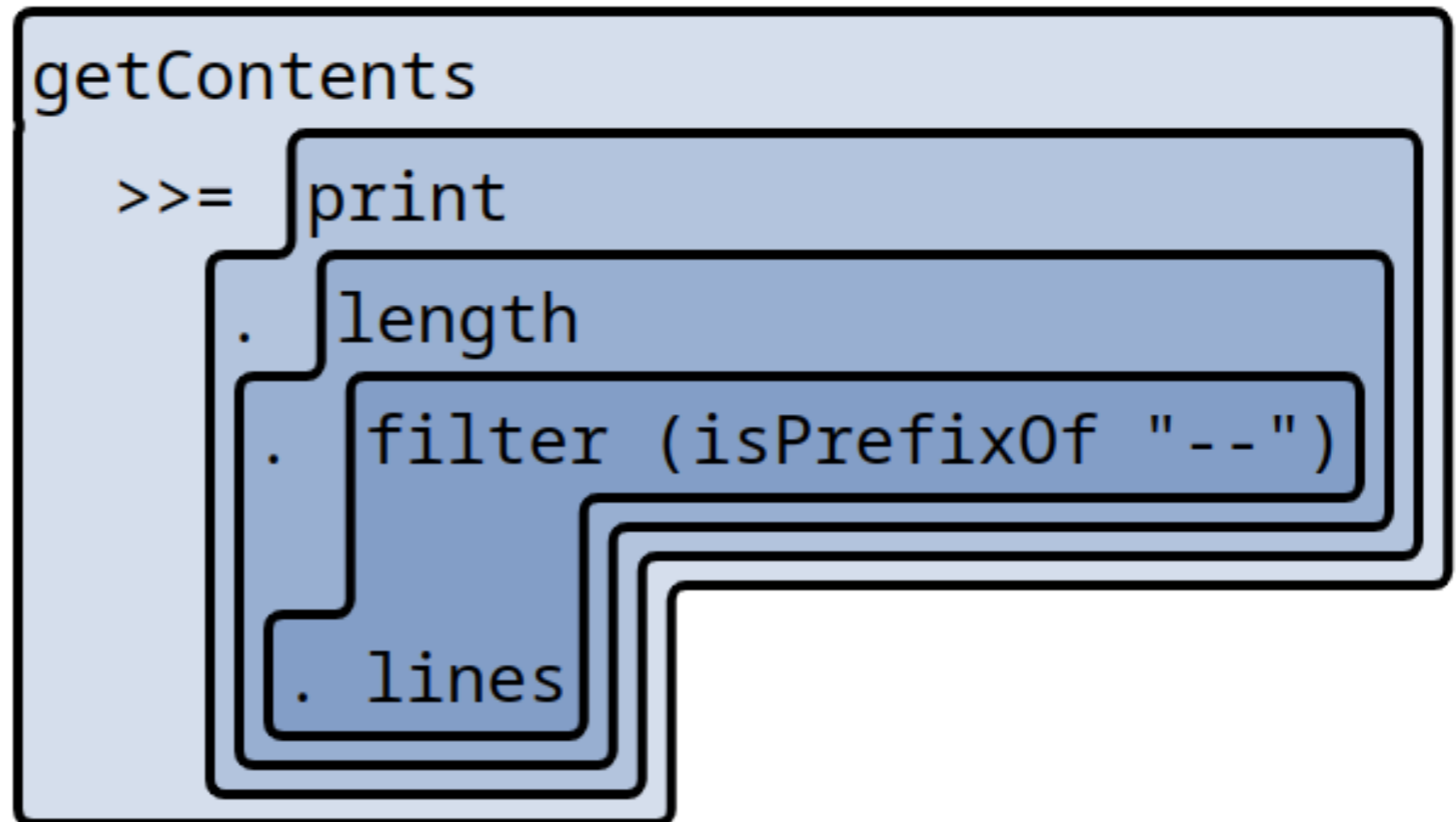
```
main =
  getContents
  >>= print
  . length
  . filter $ isPrefixOf "--"
  . lines
```

```
main =
```



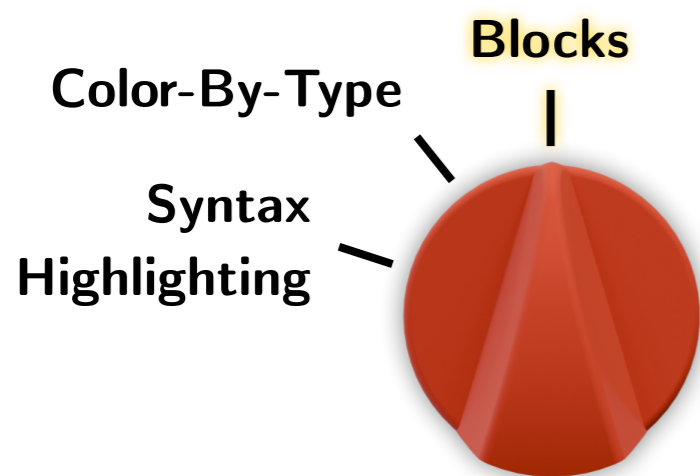
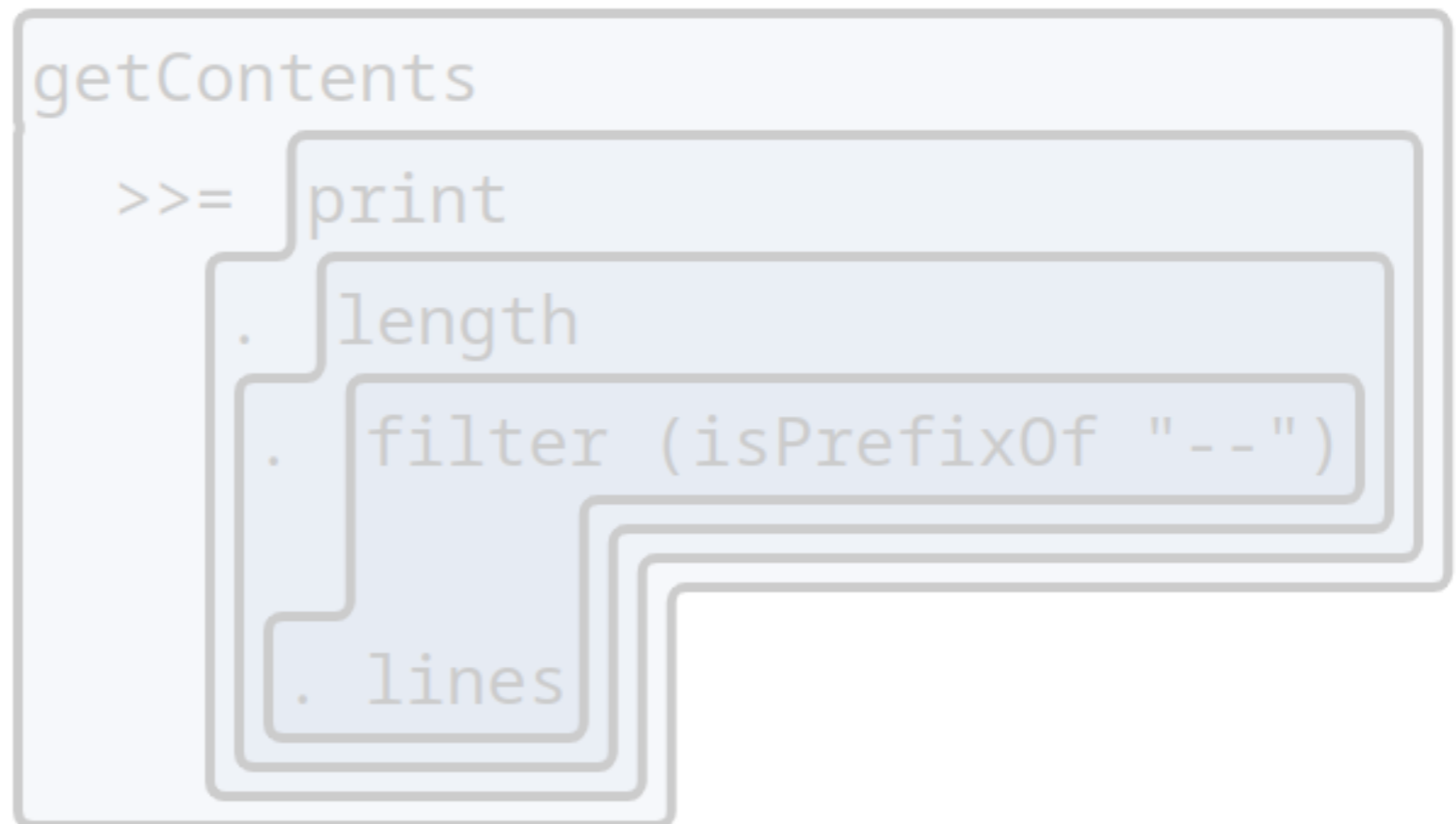
```
main =
  getContents
  >>= print
  . length
  . filter (isPrefixOf "--")
  . lines
```

```
main =
```



```
main =
  getContents
  >>= print
  . length
  . (\ls -> trace ((filter (isPrefixOf "--")) ls))
  . lines
```

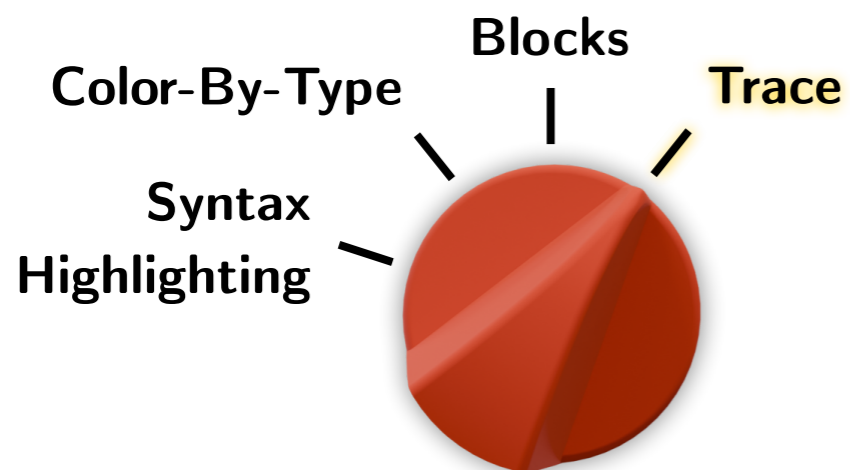
```
main =
```



```
main =
  getContents
  >>= print
  . length
  . (\ls -> trace ((filter (isPrefixOf "--")) ls))
  . lines
```

```
main =
  getContents
  >>= print
  . length
  . (\ls -> trace ((filter (isPrefixOf "--")) ls) )
  . lines
```

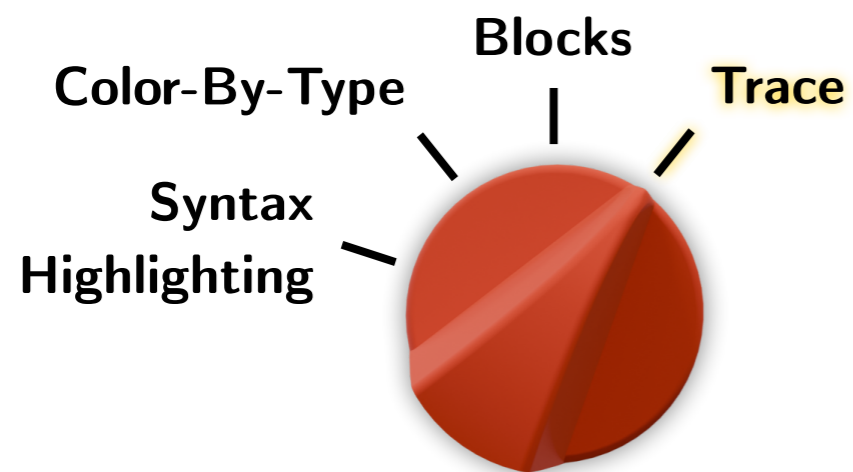
```
value
[ "-- Anonymous Authors"
, "-- Print a greeting" ]
```



```
main =
  getContents
  >>= print
  . length
  . filter (not . isPrefixOf "--")
  . lines
```

```
main =
  getContents
  >>= print
  . length
  . (\ls -> trace ((filter (isPrefixOf "--")) ls)
  . lines
```

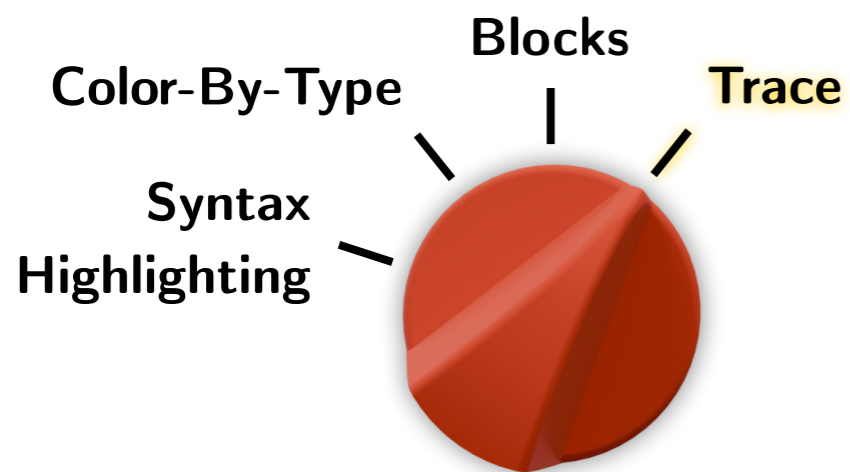
value
["-- Anonymous Authors"
,"-- Print a greeting"]



```
main =
  getContents
  >>= print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```

```
main =
  getContents
  >>= print
    . length
    . (\ls -> trace ((filter (isPrefixOf "--")) ls)
    . lines
```

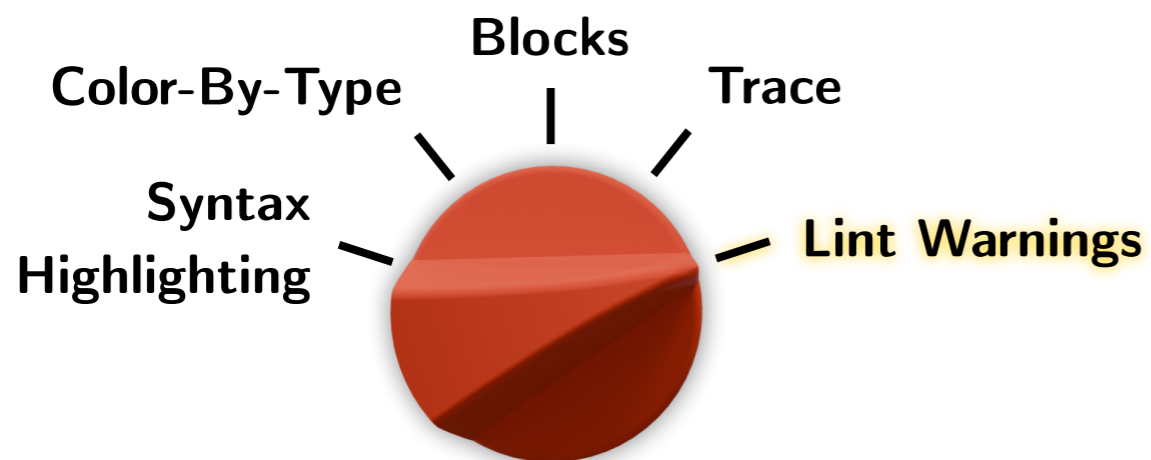
value
["-- Anonymous Authors"
,"-- Print a greeting"]



```
main =
  getContents
  >>= print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```

```
main =
```

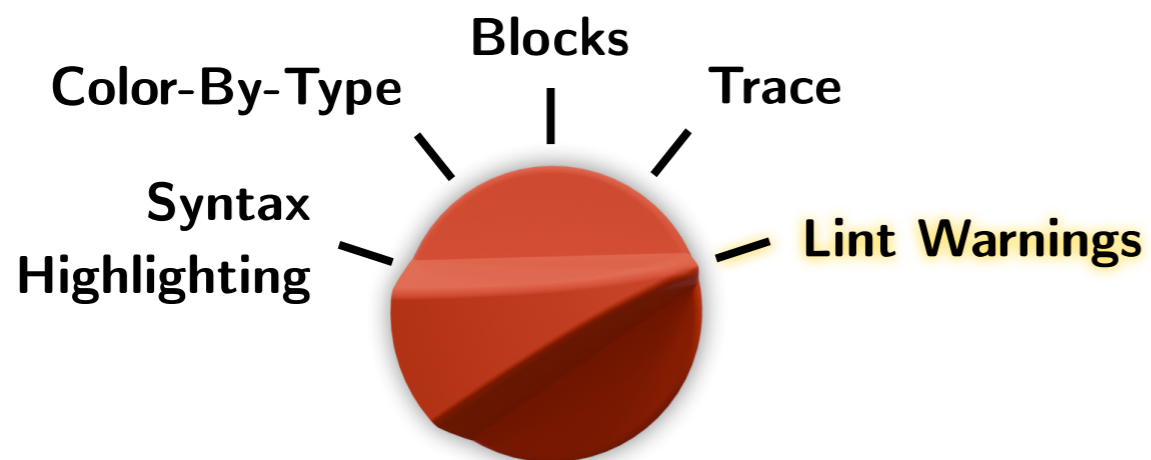
```
getContents
  >>= print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```



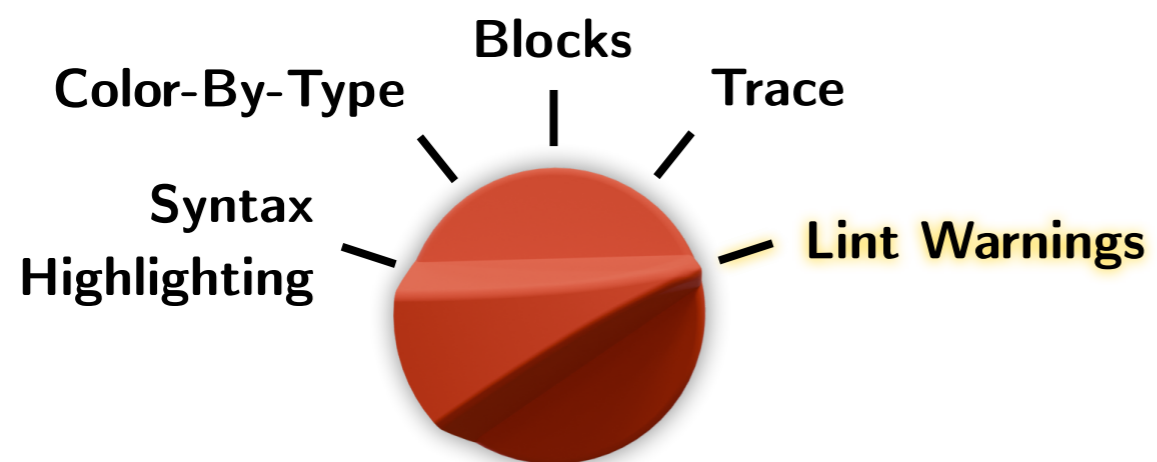

```
main =
  getContents
  >>= print
  >>> length
  >>> filter (not . isPrefixOf "--")
  >>> lines
```

```
f >>> g = \a -> g (f a)
```

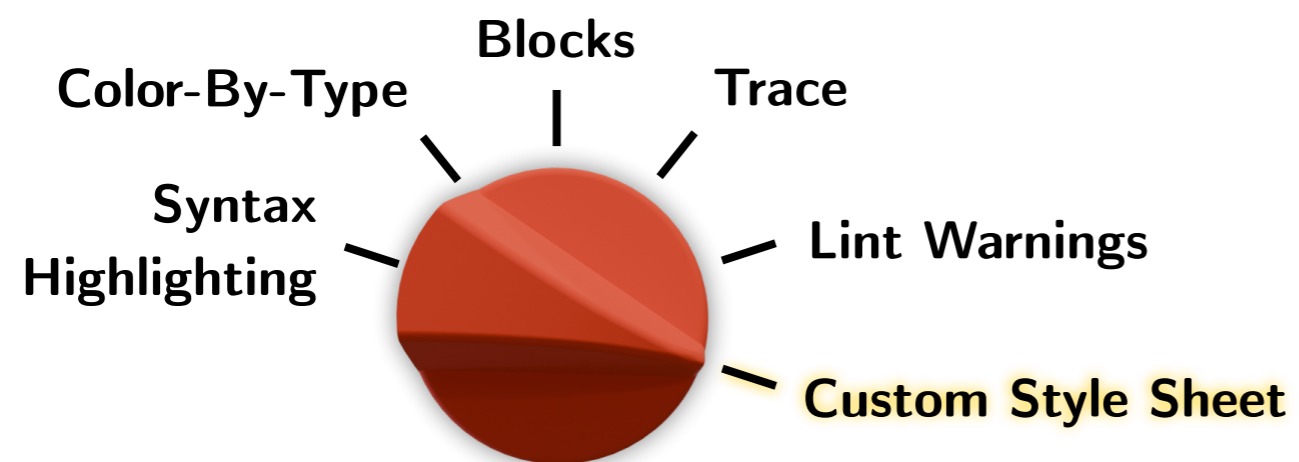
```
main =
  getContents
  >>= lines
  >>> filter (not . isPrefixOf "--")
  >>> length
  >>> print
```



```
main =
  getContents
  >>= lines
  >>> filter (not . isPrefixOf "--")
  >>> length
  >>> print
```



```
main =  
  getContents  
  >>= lines  
  >>> filter (not . isPrefixOf "--")  
  >>> length  
  >>> print
```



```
main =
  getContents
    >>= lines
    >>> filter (not . isPrefixOf "--")
    >>> length
    >>> print

x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}

x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}

EBinop (_, Located _ x@(Op op), _) if isLR op ->
x {
  font-weight: bold;
  color: indigo;
}

EBinop (_, x@(Op op), _) if isRL op ->
x {
  font-weight: bold;
  color: orange;
}
```

Highlighting

Custom Style Sheet

```
main =
  getContents
    >>= lines
    >>> filter (not . isPrefixOf "--")
    >>> length
    >>> print
```

```
x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}

x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}

EBinop (_, Located _ x@(Op op), _) if isLR op ->
x {
  font-weight: bold;
  color: indigo;
}

EBinop (_, x@(Op op), _) if isRL op ->
x {
  font-weight: bold;
  color: orange;
}
```

Highlighting

Custom Style Sheet

```
main =
  getContents
    >>= lines
    >>> filter (not isPrefixOf "--")
    >>> length
    >>> print
```

```
x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}

x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}

EBinop (_, Located _ x@(Op op), _) if isLR op ->
x {
  font-weight: bold;
  color: indigo;
}

EBinop (_, x@(Op op), _) if isRL op ->
x {
  font-weight: bold;
  color: orange;
}
```

Highlighting

Custom Style Sheet

```
main =
  getContents
    >>= lines
    >>> filter (not . isPrefixOf "--")
    >>> length
    >>> print

x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}

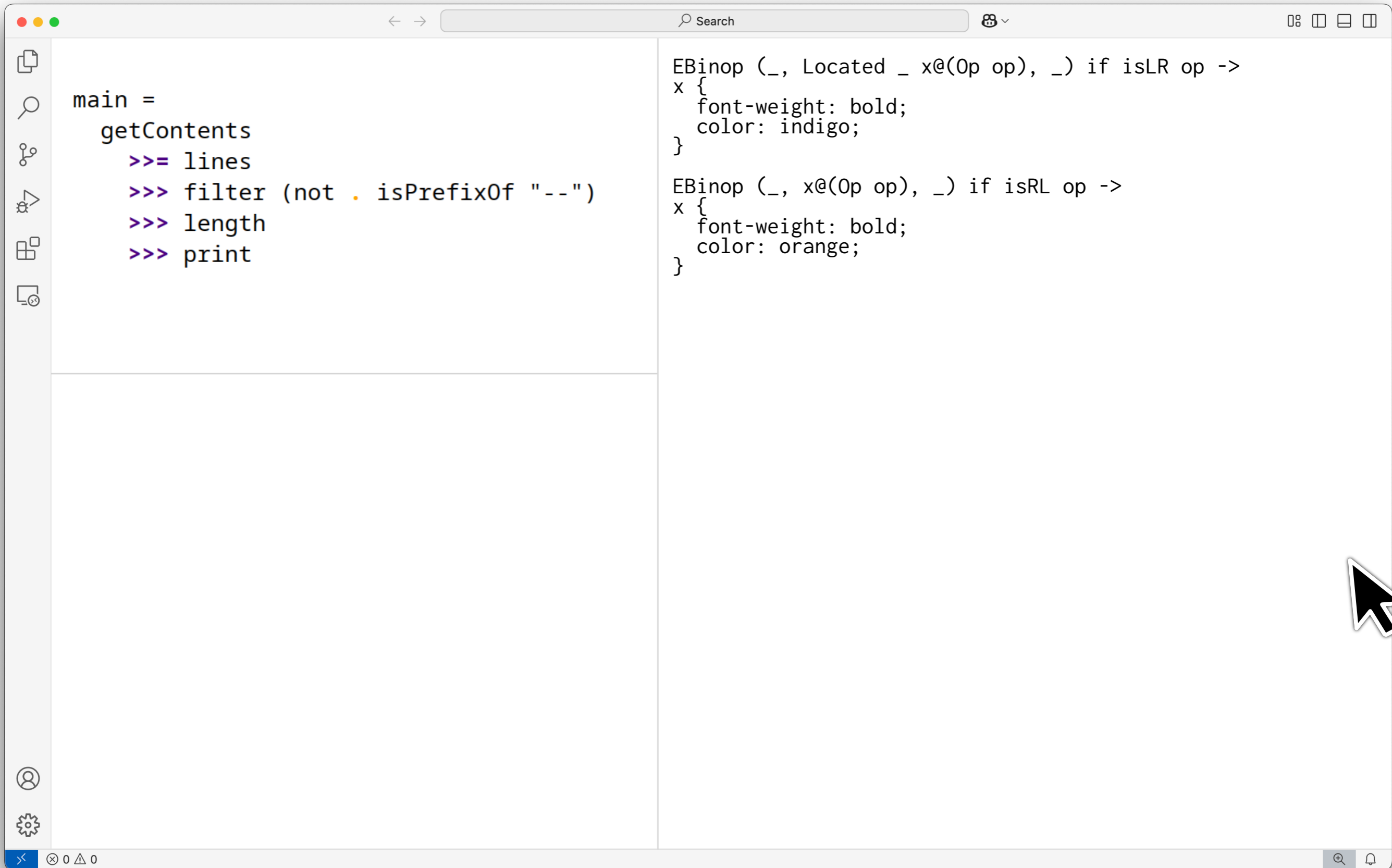
x@(EBinop (_, Op op1, y@(EBinop (_, Op op2, _))))
  if isDirUnEq op1 op2 ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: orange;
  background-color: papayawhip;
}
y {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  border-color: indigo;
  background-color: lavender;
}

EBinop (_, Located _ x@(Op op), _) if isLR op ->
x {
  font-weight: bold;
  color: indigo;
}

EBinop (_, x@(Op op), _) if isRL op ->
x {
  font-weight: bold;
  color: orange;
}
```

Highlighting

Custom Style Sheet



The screenshot shows a code editor window with a search bar at the top and a sidebar on the left. The main editor area is split into two panes. The left pane contains code with syntax highlighting: 'main =' is in grey, 'getContents' is in grey, '>>=' is in purple, '>>>' is in purple, and the rest of the code is in black. The right pane contains code with custom styling: 'EBinop' is in grey, 'Located' is in grey, 'x@(Op op)' is in grey, 'if isLR op ->' is in grey, 'x {' is in grey, 'font-weight: bold;' is in bold black, 'color: indigo;' is in indigo, and 'EBinop' is in grey. The right pane also shows 'EBinop' in grey, 'x@(Op op)' in grey, 'if isRL op ->' in grey, 'x {' in grey, 'font-weight: bold;' in bold black, and 'color: orange;' in orange. A mouse cursor is visible on the right side of the window.

```
main =
  getContents
    >>= lines
    >>> filter (not . isPrefixOf "--")
    >>> length
    >>> print

EBinop (_, Located _ x@(Op op), _) if isLR op ->
x {
  font-weight: bold;
  color: indigo;
}

EBinop (_, x@(Op op), _) if isRL op ->
x {
  font-weight: bold;
  color: orange;
}
```

Highlighting

— Custom Style Sheet

The image shows a code editor interface. On the left, a file tree shows a file named 'main'. The main content area displays code with syntax highlighting. The code includes comments and CSS-like styling for code blocks.

```
main =
  getContents
    >>= lines
    >>> filter (not . isPrefixOf "--")
    >>> length
    >>> print
```

```
EBinop (_, Located _ x@(Op op), _) if isLR op ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  font-weight: bold;
  border-color: indigo;
  background-color: lavender;
}
EBinop (_, x@(Op op), _) if isRL op ->
x {
  border-width: 2; padding: 2; margin: 2; border-radius: 3;
  font-weight: bold;
  border-color: orange;
  background-color: papayawhip;
}
```

Highlighting

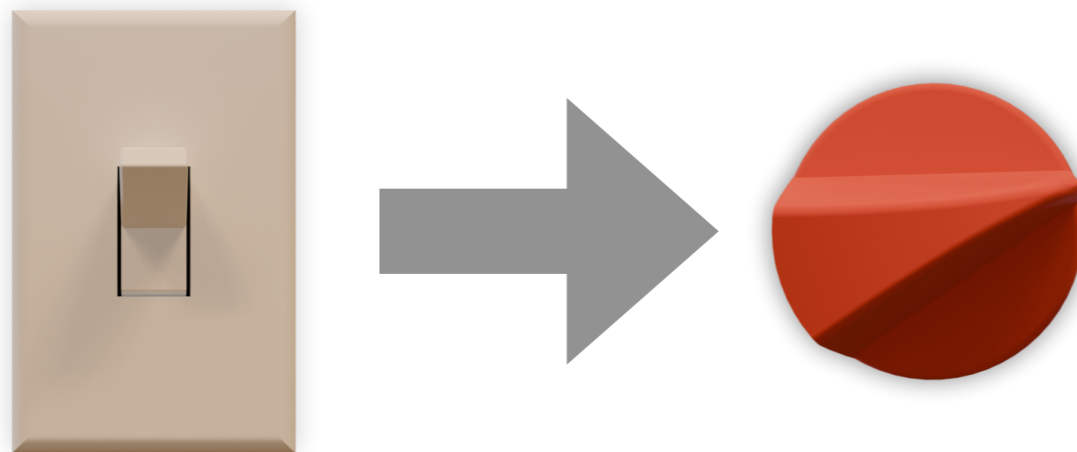
Custom Style Sheet

Takeaways

- Views are of the *program source code*. (i.e. they are projections of the program's annotated AST.)

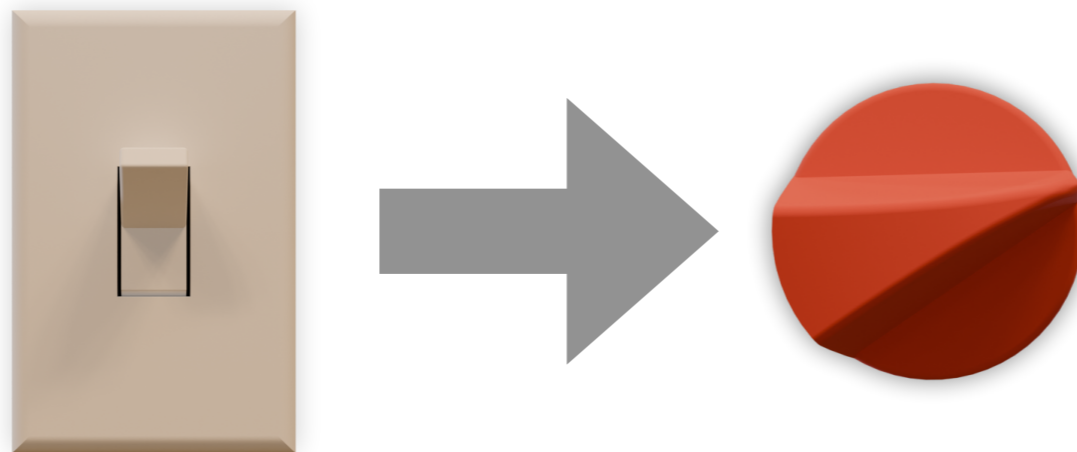
Takeaways

- Views are of the *program source code*. (i.e. they are projections of the program's annotated AST.)
- Utilized *task-specific* views of the same program



Takeaways

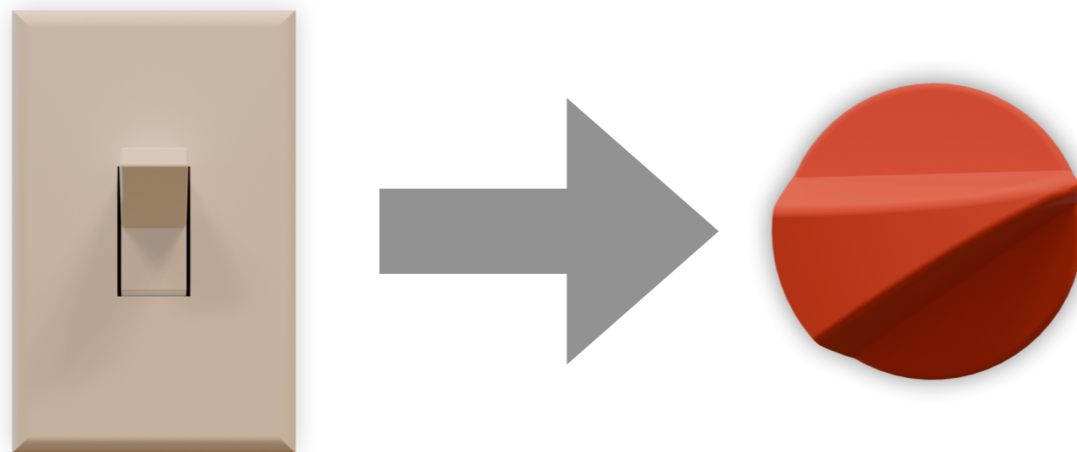
- Views are of the *program source code*. (i.e. they are projections of the program's annotated AST.)
- Utilized *task-specific* views of the same program



- User customized a view to suit their taste

Takeaways

- Views are of the *program source code*. (i.e. they are projections of the program's annotated AST.) → **Document**
- Utilized *task-specific* views of the same program → **Style Sheets**

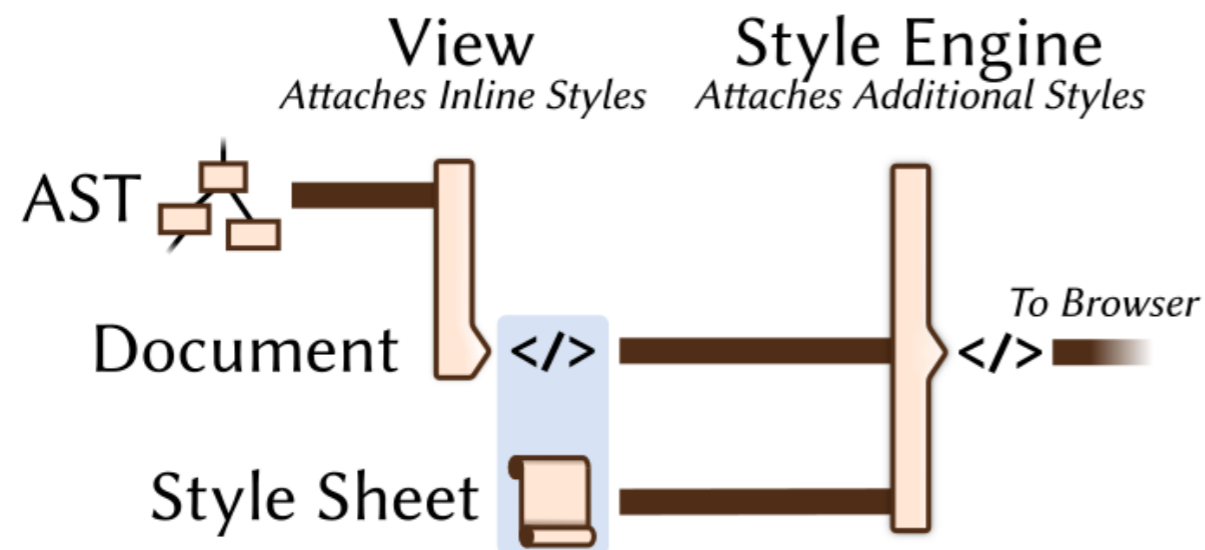


- User customized a view to suit their taste → **DOM Inspector**

So we can just use CSS/HTML?

Challenges

“Two Trees” Problem



Layout Problem

```
main =  
  getContents  
  >>= print  
    . length  
    . filter (not . isPrefixOf "--")  
    . lines
```

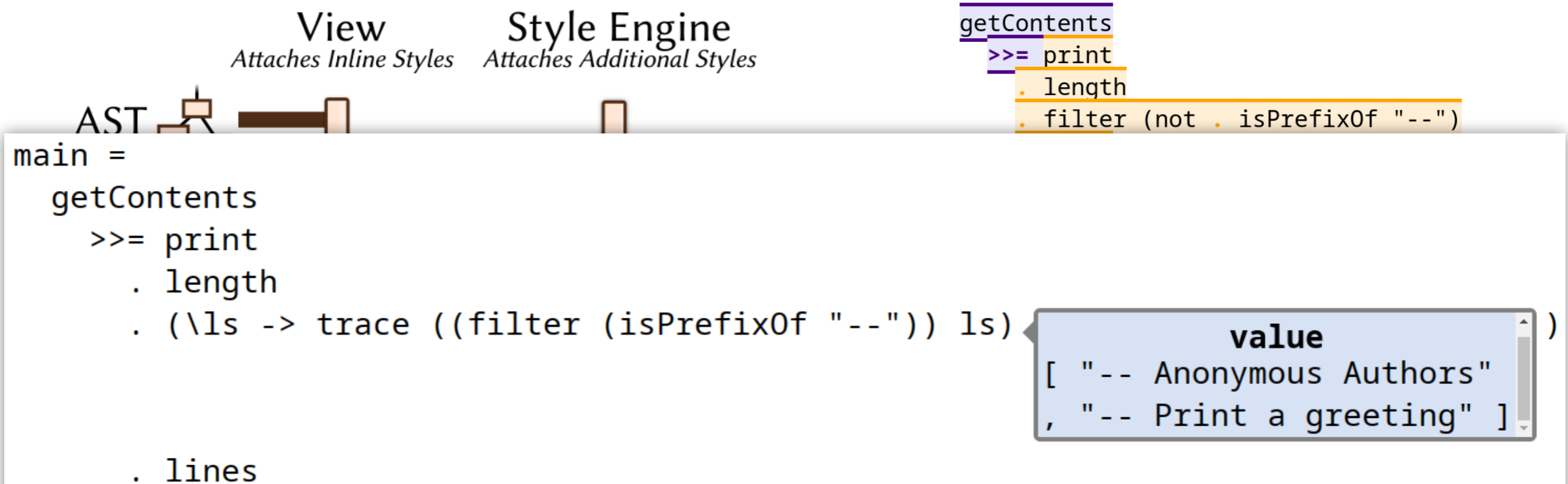
or

```
main =  
  getContents  
  >>=  
    print  
    . length  
    . filter (not . isPrefixOf "--")  
    . lines
```

Challenges

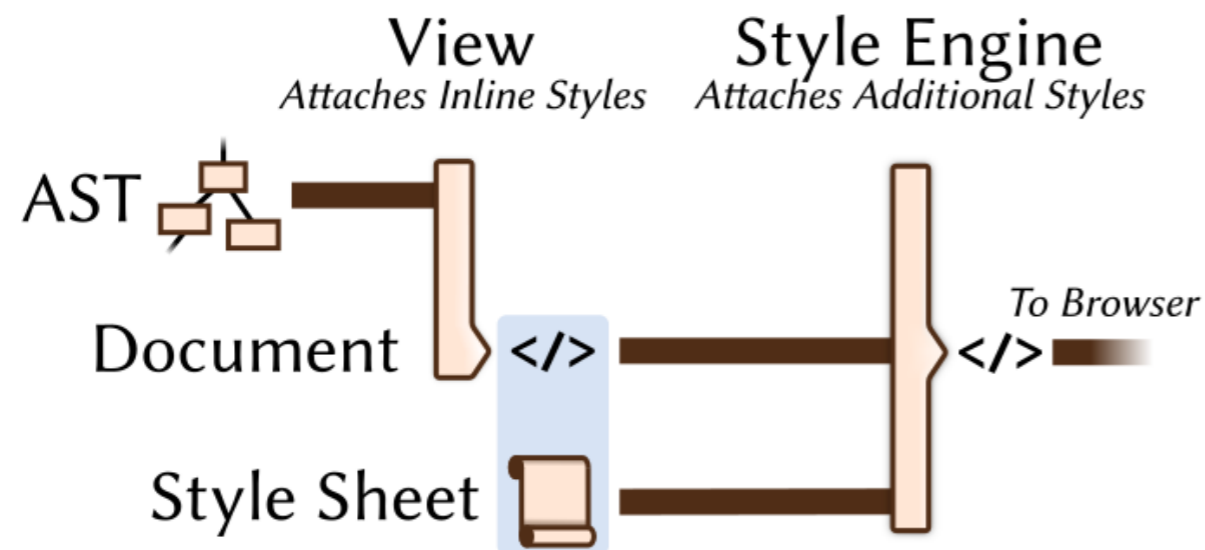
“Two Trees” Problem

Layout Problem



Challenges

“Two Trees” Problem



Layout Problem

```
main =  
  getContents  
  >>= print  
    . length  
    . filter (not . isPrefixOf "--")  
    . lines
```

or

```
main =  
  getContents  
  >>=  
    print  
    . length  
    . filter (not . isPrefixOf "--")  
    . lines
```


Layout Problem

Want

main =

```
getContents
  >>= print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```

Have

main =

```
getContents
  >>= print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```

or

main =

```
getContents
  >>=
    print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```

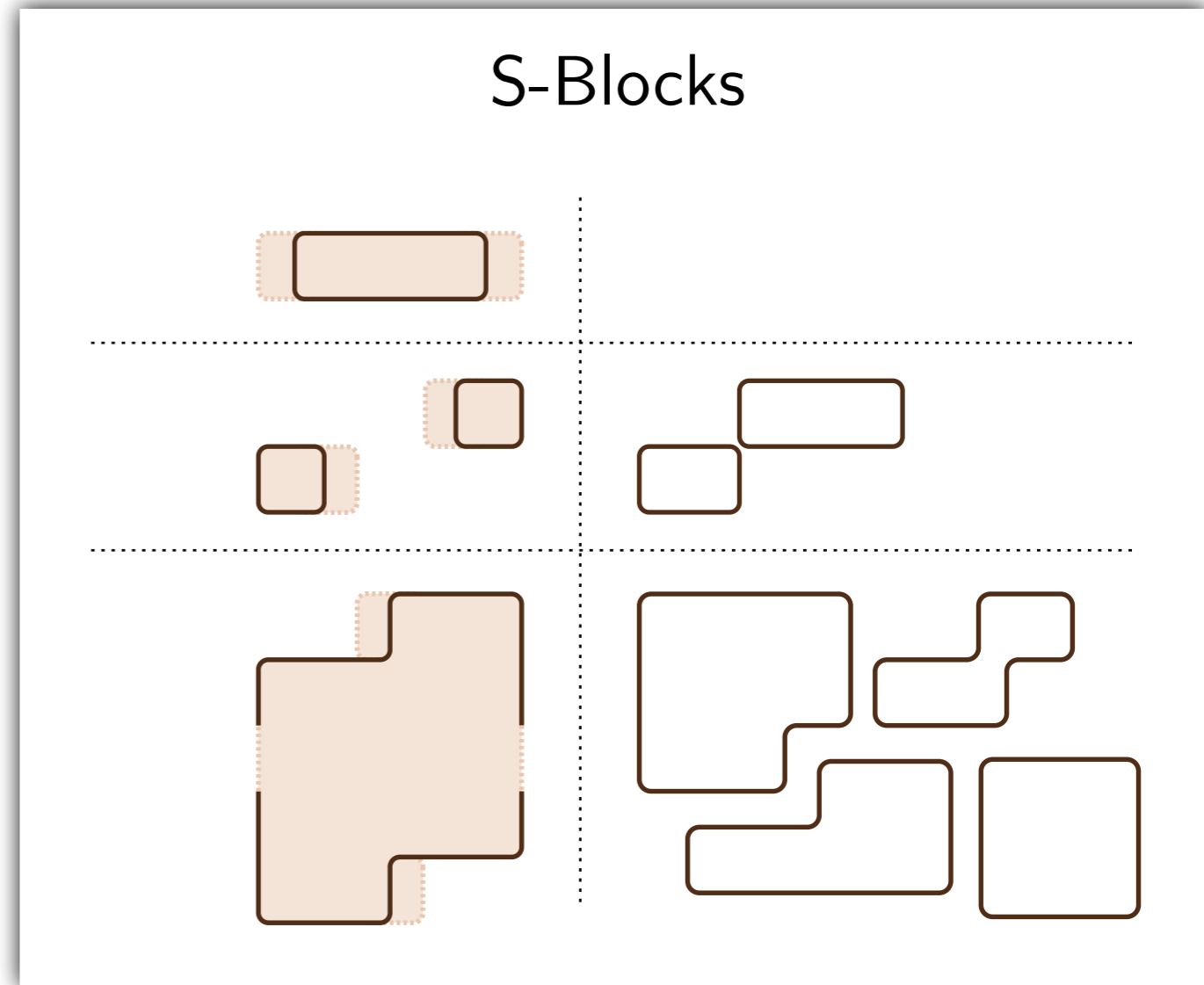
Need to give up boxes → S-blocks

Layout Problem

Want

main =

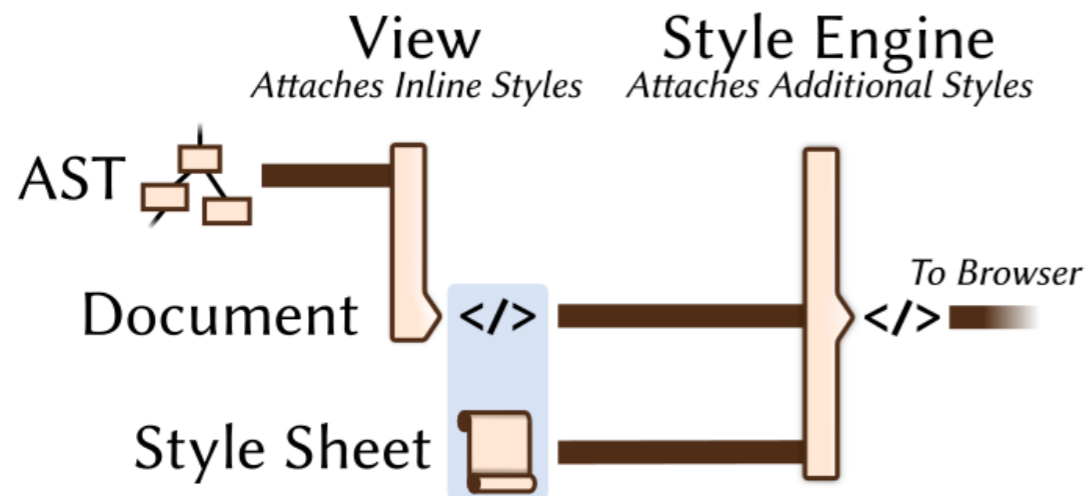
```
getContents
  >>= print
    . length
    . filter (not . isPrefixOf "--")
    . lines
```



Need to give up boxes → S-blocks

“Two Trees” Problem

Existing Systems



Our System

