

Direct Manipulation Programming



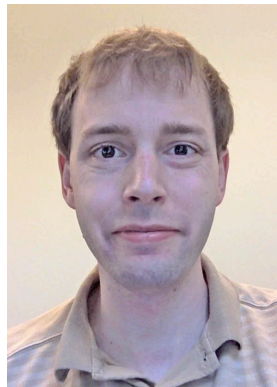
Sketch-n-Sketch



Ravi
Chugh



Nick
Collins



Brian
Hempel



Justin
Lubin

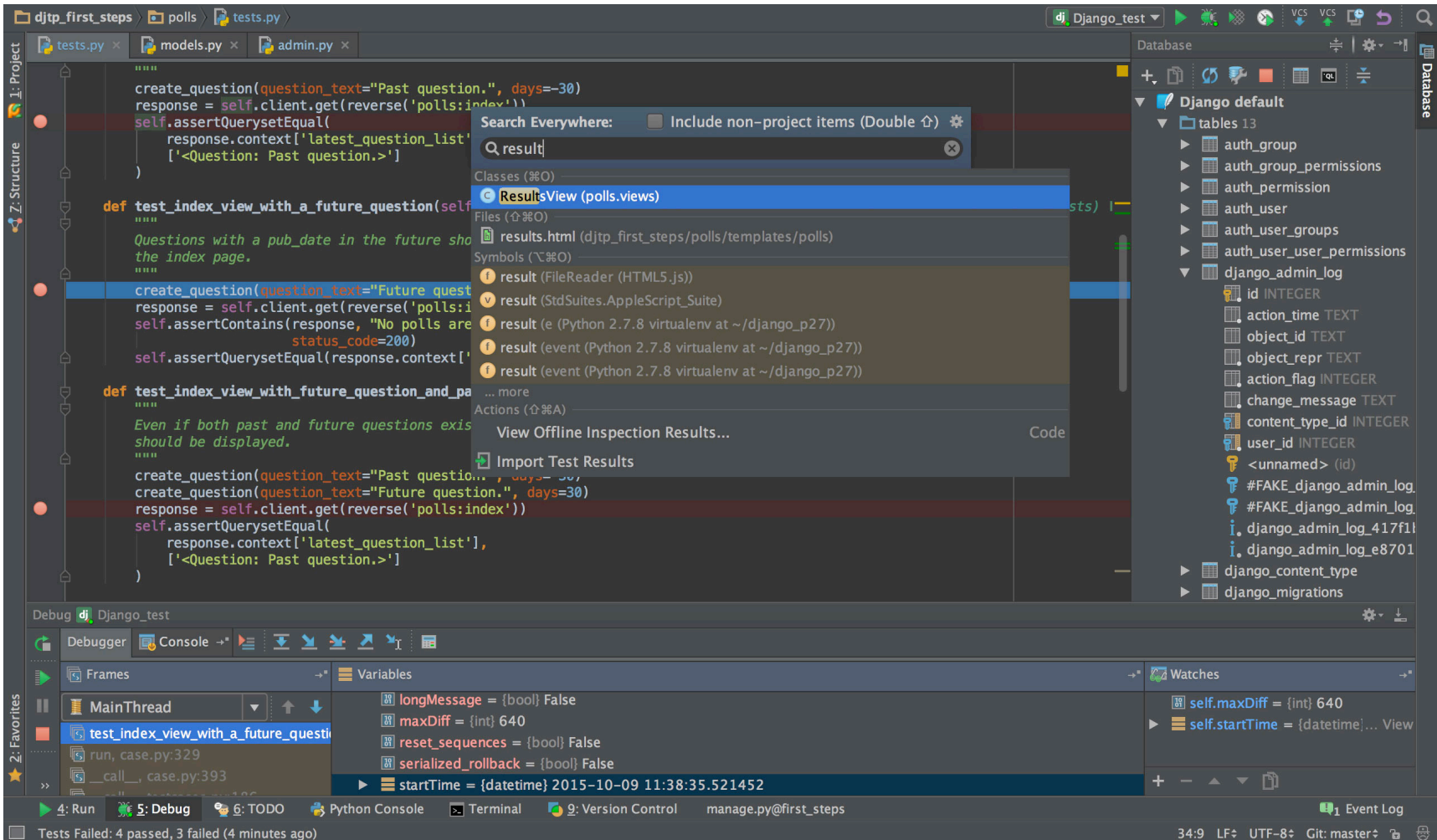


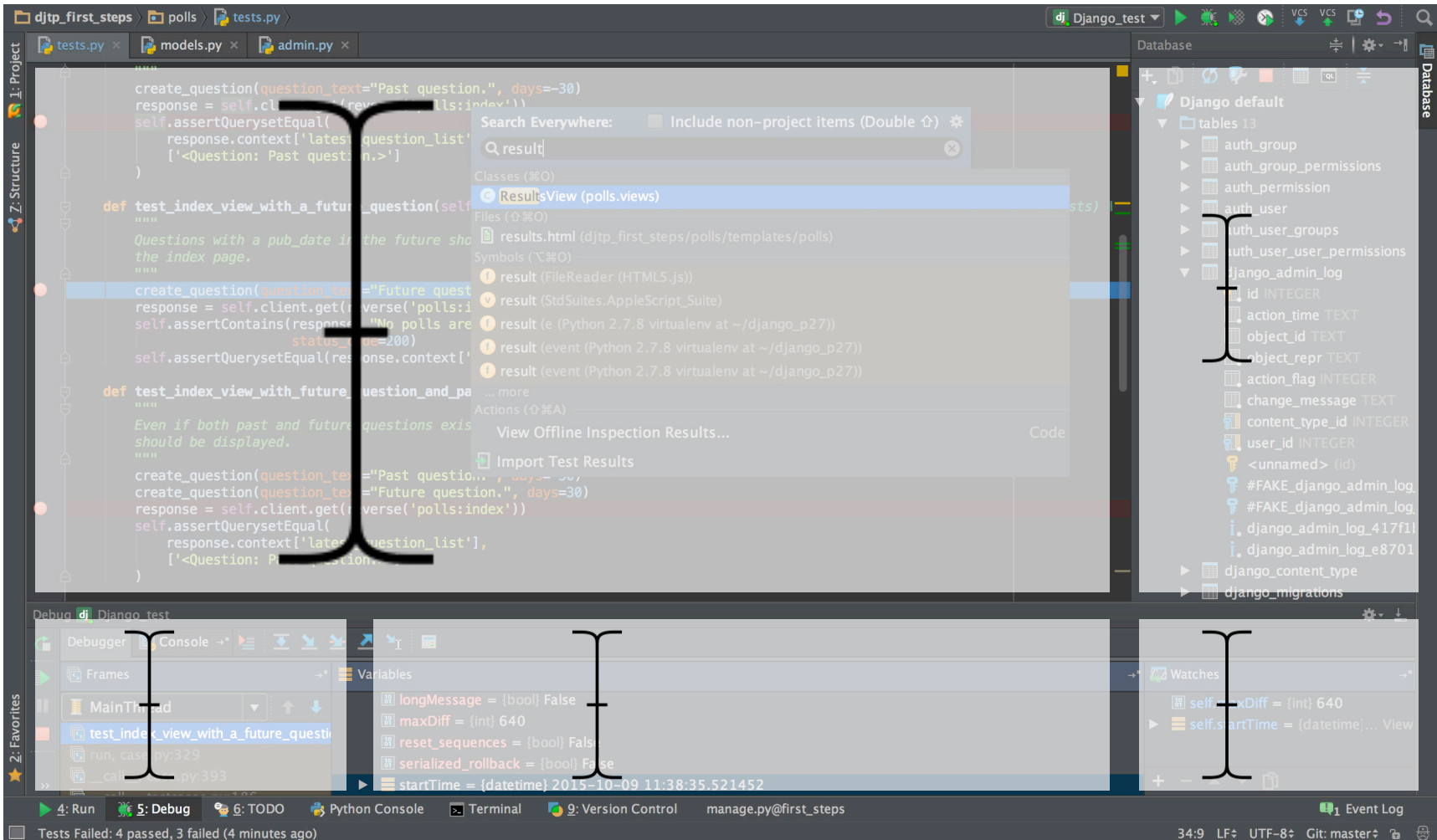
Mikaël
Mayer



THE UNIVERSITY OF
CHICAGO

ICFP Tutorial
09.27.2018







+



Sketch-n-Sketch

The screenshot shows a web browser window titled "Sketch-n-Sketch" displaying a code editor and a rendered logo. The code editor shows the following code:

```
1 type LogoParams
2   = TopLeft {x:Num, y:Num, size:Num}
3   | Center {cx:Num, cy:Num, rad:Num}
4
5 logo : String -> LogoParams -> Svg
6 logo fill logoParams =
7   let
8     {x, y, size} =
9       case logoParams of
10        TopLeft data ->
11          data
12        Center {cx, cy, rad} ->
13          {x=cx-rad, y=cy-rad, size=2*rad}
14
15     (cx, cy) =
16       (x + 0.5*size, y + 0.5*size)
17   in
18     [ rect fill x y size size
19       , line "white" 10 x y (x + size) (y + size)
20       , line "white" 10 x (y + size) cx cy
21     ]
22
23 main =
24   svg <| logo "gray" <| TopLeft {x=100, y=130, size=200}
```

The rendered logo is a gray square with a white diagonal line from the top-left to the bottom-right. The application interface includes a menu bar with "Sketch-n-Sketch", "File", "Code Tools", "View", and "Options". There are also navigation buttons for "Previous Example" and "Next Example". A toolbar on the right side contains various drawing tools like a mouse cursor, text, lines, shapes, and a star. The bottom status bar shows icons for zooming and a "Raw Stretchy Sticky" button.



Sketch-n-Sketch

[PLDI 2016] [OOPSLA 2018]

Bidirectional Programming

A screenshot of the Sketch-n-Sketch IDE. The left pane shows Scala code for a logo generator. The right pane shows a visual output of a gray square with a white diagonal line. A hand cursor is positioned over the visual output. Two red arrows point from the code to the visual output and vice versa, illustrating bidirectional programming. The code in the left pane is:

```
1 type LogoParams
2   = TopLeft {x:Num, y:Num, size:Num}
3   | Center {cx:Num, cy:Num, rad:Num}
4
5 logo : String -> LogoParams -> Svg
6 logo fill logoParams =
7   let
8     {x, y, size} =
9       case logoParams of
10         TopLeft data ->
11           data
12         Center {cx, cy, rad} ->
13           {x=cx-rad, y=cy-rad, size=2*rad}
14
15     (cx, cy) =
16       (x + 0.5*size, y + 0.5*size)
17   in
18     [ rect fill x y size size
19       , line "white" 10 x y (x + size) (y + size)
20       , line "white" 10 x (y + size) cx cy
21     ]
22
23 main =
24   svg <| logo "gray" <| TopLeft {x=100, y=130, size=200}
```



Sketch-n-Sketch

[PLDI 2016] [OOPSLA 2018]

Bidirectional Programming

The screenshot shows the Sketch-n-Sketch IDE interface. On the left, a code editor displays the following code:

```
1 type LogoParams
2   = TopLeft {x:Num, y:Num, size:Num}
3   | Center {cx:Num, cy:Num, rad:Num}
4
5 logo : String -> LogoParams -> Svg
6 logo fill logoParams =
7   let
8     {x, y, size} =
9       case logoParams of
10        TopLeft data ->
11          data
12        Center {cx, cy, rad} ->
13          {x=cx-rad, y=cy-rad, size=2*rad}
14
15     (cx, cy) =
16       (x + 0.5*size, y + 0.5*size)
17   in
18     [ rect fill x y size size
19       , line "white" 10 x y (x + size) (y + size)
20       , line "white" 10 x (y + size) cx cy
21     ]
22
23 main =
24   svg <| logo "gray" <| TopLeft {x=100, y=130, size=200}
```

On the right, a visual output shows a gray square with a white diagonal line from the top-left to the bottom-right. A hand cursor is pointing at the output. A red double-headed arrow connects the code to the output. A purple arrow points from the output back to the code, with a hand cursor pointing at it.

Output-Directed Programming

[UIST 2016] [wip]



[PLDI 2016] [OOPSLA 2018]

Bidirectional Programming

Structured
Text Editing

Output-Directed
Programming



[ICSE 2018] [wip]

[UIST 2016] [wip]

Sketch-n-Sketch

```
1 type LogoParams
2   = TopLeft {x:Num, y:Num, size:Num}
3   | Center {cx:Num, cy:Num, rad:Num}
4
5 logo : String -> LogoParams -> Svg
6 logo fill logoParams =
7   let
8     {x, y, size} =
9       case logoParams of
10         TopLeft data ->
11           data
12         Center {cx, cy, rad} ->
13           {x=cx-rad, y=cy-rad, size=
14
15     (cx, cy) =
16       (x + 0.5*size, y + 0.5*size)
17   in
18     [ rect fill x y size size
19       , line "white" 10 x y (x + size)
20       , line "white" 10 x (y + size) c
21     ]
22
23 main =
24   svg <| logo "gray" <| TopLeft {x=100, y=130, size=200}
```

Justin Lubin
~now-1:15pm



Sketch-n-Sketch

[PLDI 2016] [OOPSLA 2018]

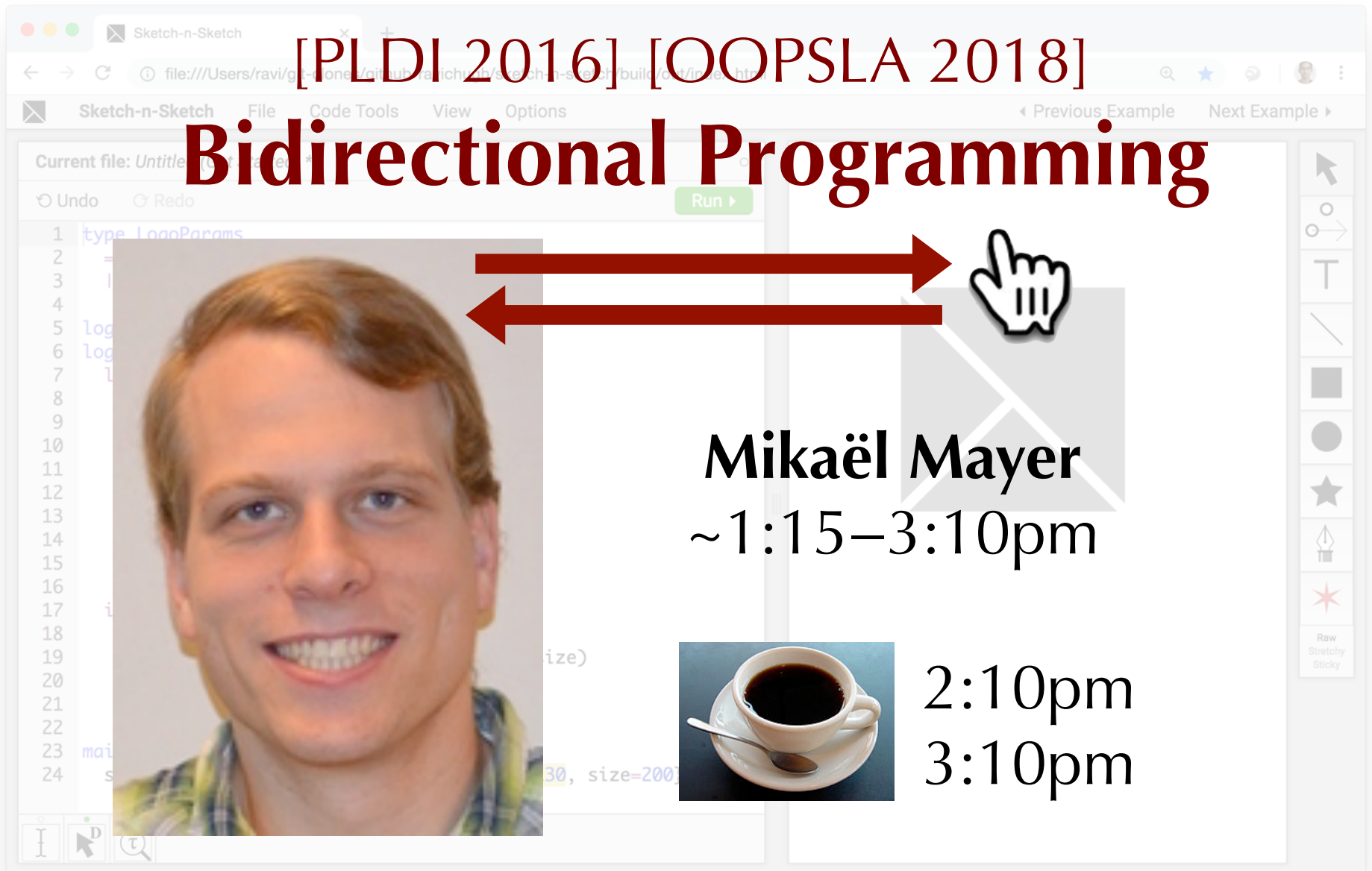
Bidirectional Programming



Mikaël Mayer
~1:15–3:10pm



2:10pm
3:10pm





Sketch-n-Sketch

A screenshot of the Sketch-n-Sketch web application. The browser window shows the URL 'h-n-sketch/build/out/index.html'. On the left, there is a code editor with a 'Run' button. The main area displays a portrait of a man, Brian Hempel. A large purple arrow points from the text 'Output-Directed Programming' to the portrait. A hand cursor icon is positioned over the arrow. The interface includes navigation buttons for 'Previous Example' and 'Next Example', and a toolbar on the right with various drawing tools.

h-n-sketch/build/out/index.html

Previous Example Next Example

Run

1 ty
2
3
4
5 lo
6 lo
7
8
9
10
11
12
13
14
15
16
17
18
19
20 , line "white" 10 x (y + size) cx cy
21
22
23 main =
24 svg <!-- logo "gray" --> <!-- Topleft {x=100, y=130, size=200} -->

Raw Stretchy Sticky

Output-Directed Programming

Brian Hempel

~3:30–3:45pm

[UIST 2016] [wip]



Sketch-n-Sketch

```
Sketch-n-Sketch x +
file:///Users/ravi/git-clones/github-ravichugh/sketch-n-sketch/build/out/index.h
Sketch-n-Sketch File Code Tools View Options
Current file: Untitled (Get Started) *
Undo Redo Run
1 type LogoParams
2   = TopLeft {x:Num, y:Num, size:Num}
3   | Center {cx:Num, cy:Num, rad:Num}
4
5 logo : String -> LogoParams -> Svg
6 logo fill logoParams =
7   let
8     {x, y, size} =
9     case logoParams of
10    | TopLeft {x, y, size} ->
11    | Center {cx, cy, rad} ->
12      Center {cx, cy, rad} ->
13      {x=cx-rad, y=y-rad, size=*rad}
14    | _ ->
15      (cx, cy) =
16      (x + 0.5*size, y + 0.5*size)
17  in
18  [ red fill x y (x + size) (y + size)
19    , lightgray fill (x + size) (y + size) cx cy
20    , lightgray fill (x + size) (y + size) cx cy
21  ]
22
23 main =
24   svg <| logo "gray" TopLeft {x=100, y=130, size=200}
```

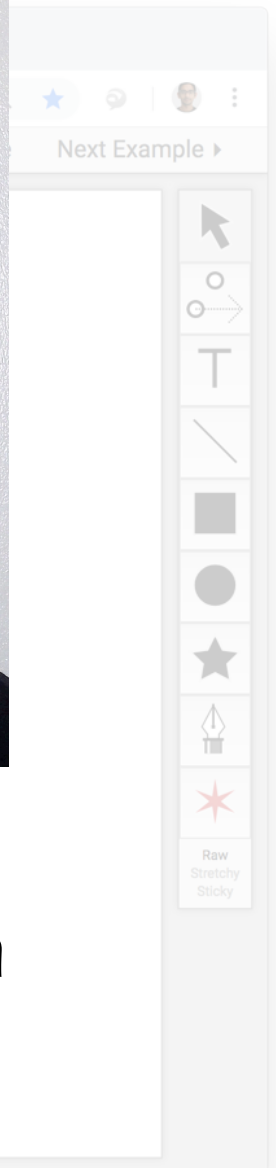
Structured
Text Editing



[ICSE 2018] [wip]



Nick Collins
~3:45–4:00pm



[PLDI 2016] [OOPSLA 2018]

Bidirectional Programming

Structured
Text Editing

Output-Directed
Programming

[ICSE 2018] [wip]

[UIST 2016] [wip]

