Programming

MICHAEL BERNSTEIN
CS 376

Reminder:
project faire II
Wednesday

A Small Matter of Programming

- Software engineering is a highly complex task, a microcosm of many challenges in HCI
- Making software engineering more accessible could empower millions to customize applications and write programs

Research agenda

- Understand the challenges in programming
- Design more effective software engineering interfaces
- Aid novices in learning to program or writing programs
- Abstract best practices into toolkits

Understanding programmers

Information Needs in Programming

[Ko, DeLine and Venolia, ICSE '07]

- Observed 17 developers in 90-minute sessions and transcribed all activities
- Thematic coding of information needs
  - Writing code e.g., how do I use this method?
  - Submitting a change e.g., which files are included?
  - Triaging bugs e.g., is the problem worth fixing?
  - Reproducing failure e.g., what are failure conditions?
  - Understanding execution e.g., what caused this behavior?
  - Design e.g., why is the code implemented this way?
  - Awareness e.g., what are my collaborators working on?
- Most common need: collaborator awareness

Obstacles to learning APIs

[Robillard and DeLine, Empir. Software Engineering 2011]

- Survey and in-person interviews, combined reaching 440 professional software engineers
- Biggest challenge: inadequate documentation
- API intent: how it was intended to be used
  - "Nowhere in there does it say, and we intended to be used for a few graphics of small size because the memory footprint is going to be this."
- Code examples: snippets, tutorials, working apps
- Penetrability: how much detail and implementation to expose?

Web foraging and programming

[Brandt et al., CHI ’09]

- Laboratory study: ask programmers to implement a chat room in PHP
- This paper articulated how programmers make heavy use of the web
  - JIT learning of new skills
  - Clarifying existing skills
  - Reminding themselves of details
- Average participant spent 19% of their programming time on the web

Software engineering interfaces

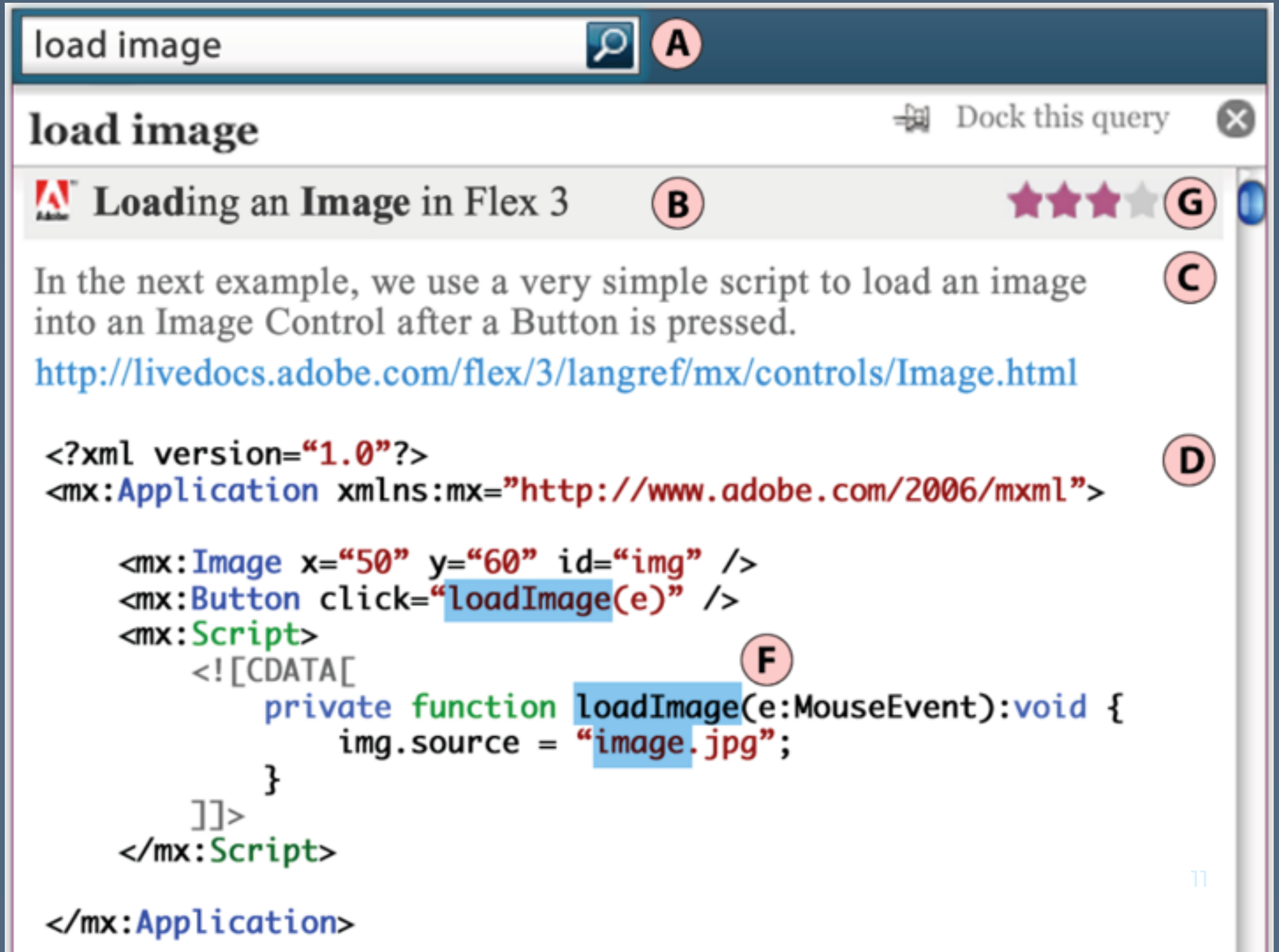
Goals of software engineering interface research

- Design a better toolbench, produce a better programmer
- This research typically assumes that the programming language is static, but the interface of the IDE can be molded

Example-centric programming

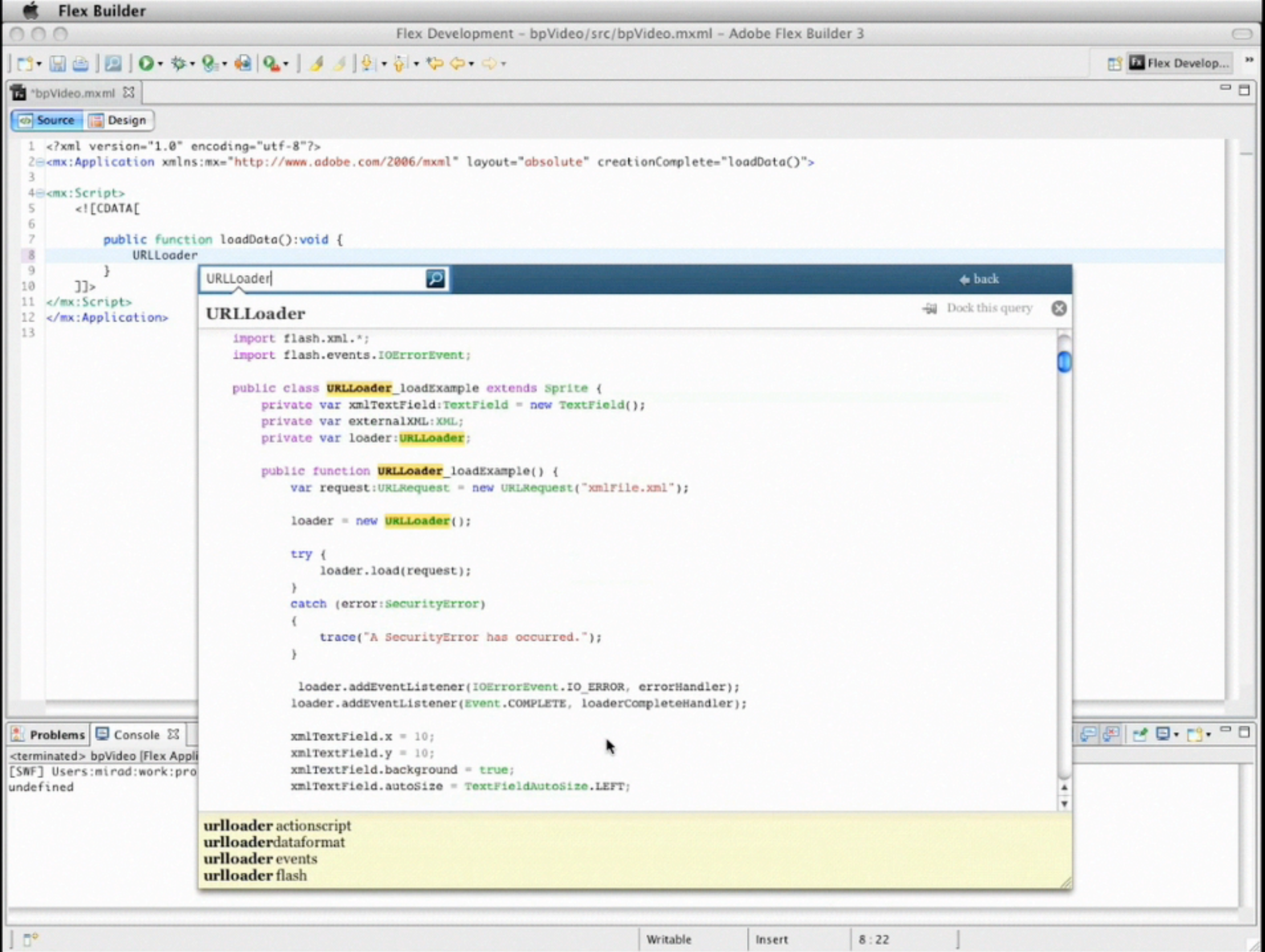
[Brandt et al., CHI '10]

- Close the loop between the development environment and web search
- Autocomplete code via web examples



In the next example, we use a very simple script to load an image into an Image Control after a Button is pressed.
http://livedocs.adobe.com/flex/3/langref/mx/controls/Image.html

```
<?xml version="1.0"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">

    <mx:Image x="50" y="60" id="img" />
    <mx:Button click="loadImage(e)" />
    <mx:Script>
        <![CDATA[
            private function loadImage(e:MouseEvent):void {
                img.source = "image.jpg";
            }
        ]]>
    </mx:Script>

</mx:Application>
```


Flex Builder
Flex Development - bpVideo/src/bpVideo.mxml - Adobe Flex Builder 3
Flex Develop...
*bpVideo.mxml
Source
Design

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" creationComplete="loadData()">

<mx:Script>
    <![CDATA[

        public function loadData():void {
            URLLoader
        }
    ]]>
</mx:Script>
</mx:Application>
```

URLLoader

```
import flash.xml.*;
import flash.events.IOErrorEvent;

public class URLLoader_loadExample extends Sprite {
    private var xmlTextField:TextField = new TextField();
    private var externalXML:XML;
    private var loader:URLLoader;

    public function URLLoader_loadExample() {
        var request:URLRequest = new URLRequest("xmlFile.xml");

        loader = new URLLoader();

        try {
            loader.load(request);
        }
        catch (error:SecurityError)
        {
            trace("A SecurityError has occurred.");
        }

         loader.addEventListener(IOErrorEvent.IO_ERROR, errorHandler);
        loader.addEventListener(Event.COMPLETE, loaderCompleteHandler);

        xmlTextField.x = 10;
        xmlTextField.y = 10;
        xmlTextField.background = true;
        xmlTextField.autoSize = TextFieldAutoSize.LEFT;
```

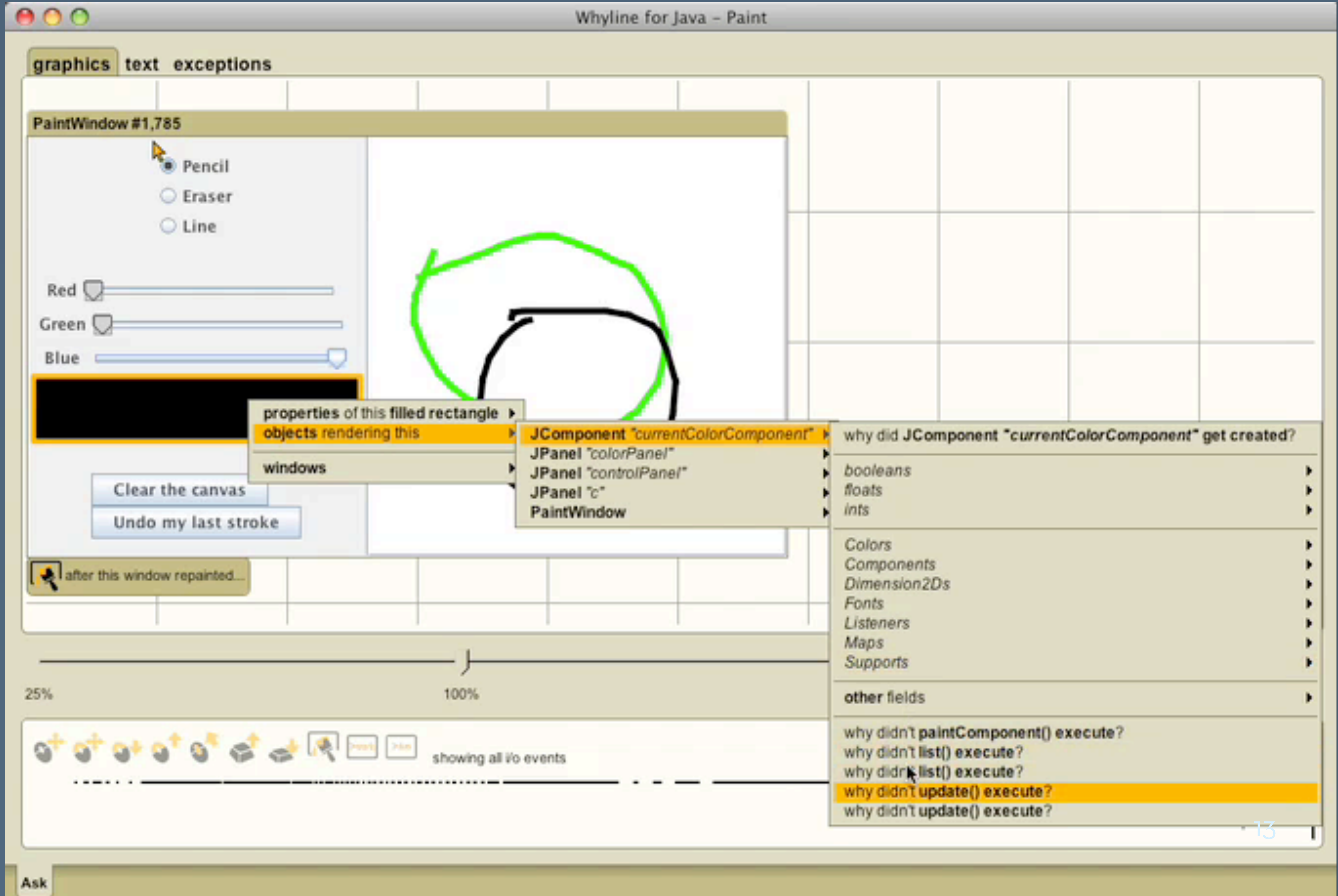
urlloader actionscript
urlloader dataformat
urlloader events
urlloader flash

```
<terminated> bpVideo [Flex Appli
[SWF] Users:mirad:work:pro
undefined
```

Asking ‘why’ questions of code

[Ko and Myers CHI ’04, ICSE ’09]

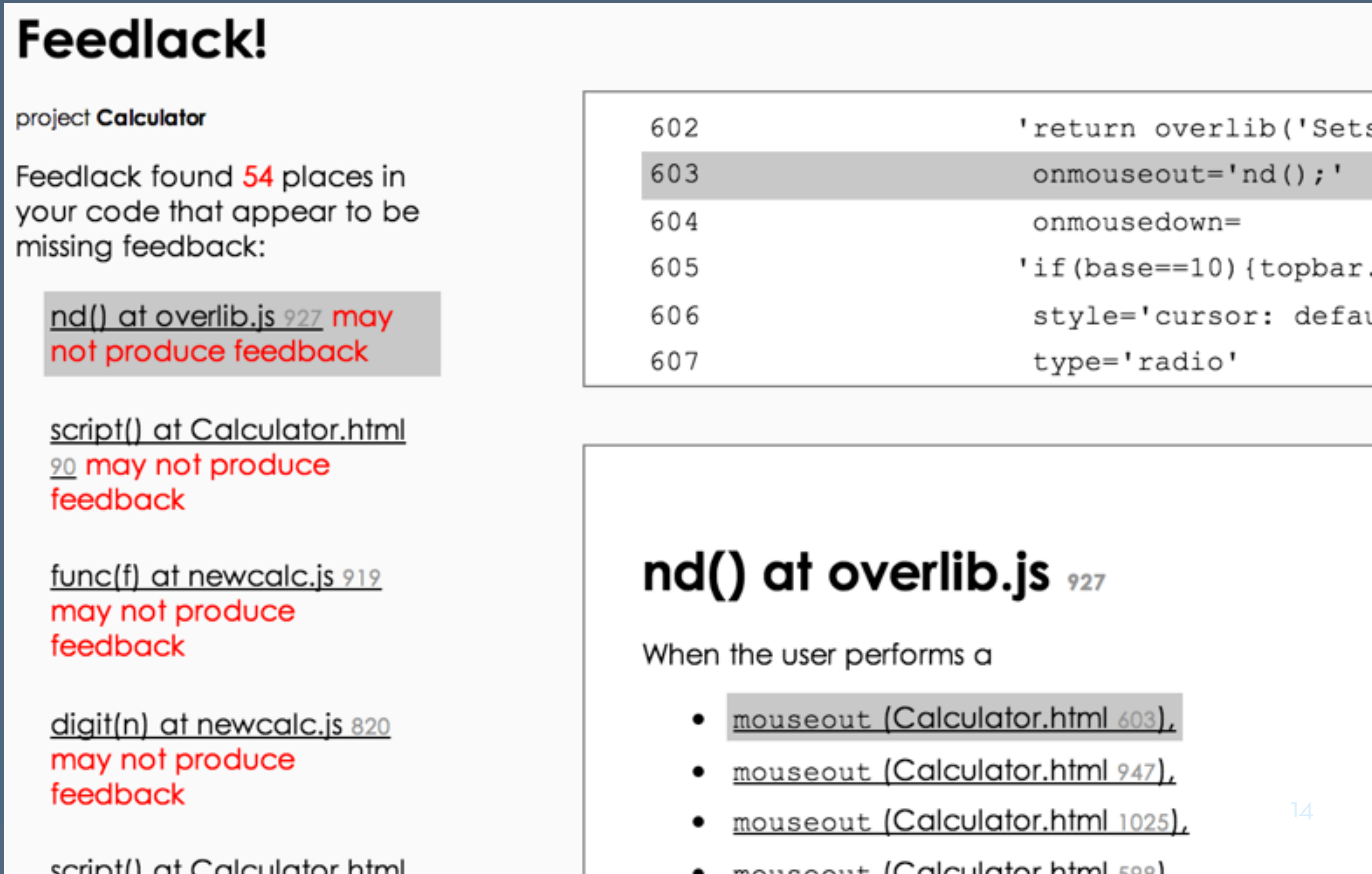
- Debugging problems often reduce to “why” questions
- Analyze program traces to answer them



Missing user-facing feedback

[Ko and Zhang, CHI '11]

- Usability heuristic: all user inputs should produce some form of feedback
- Statically analyze code to identify user inputs that produce no feedback



Feedlack!

project **Calculator**

Feedlack found 54 places in your code that appear to be missing feedback:

nd() at overlib.js 927 may not produce feedback

script() at Calculator.html 90 may not produce feedback

func(f) at newcalc.js 919 may not produce feedback

digit(n) at newcalc.js 820 may not produce feedback

script() at Calculator.html

```
602    'return overlib('Sets
603     onmouseout='nd();'
604     onmousedown=
605    'if(base==10){topbar.
606     style='cursor: defau
607     type='radio'
```

nd() at overlib.js 927

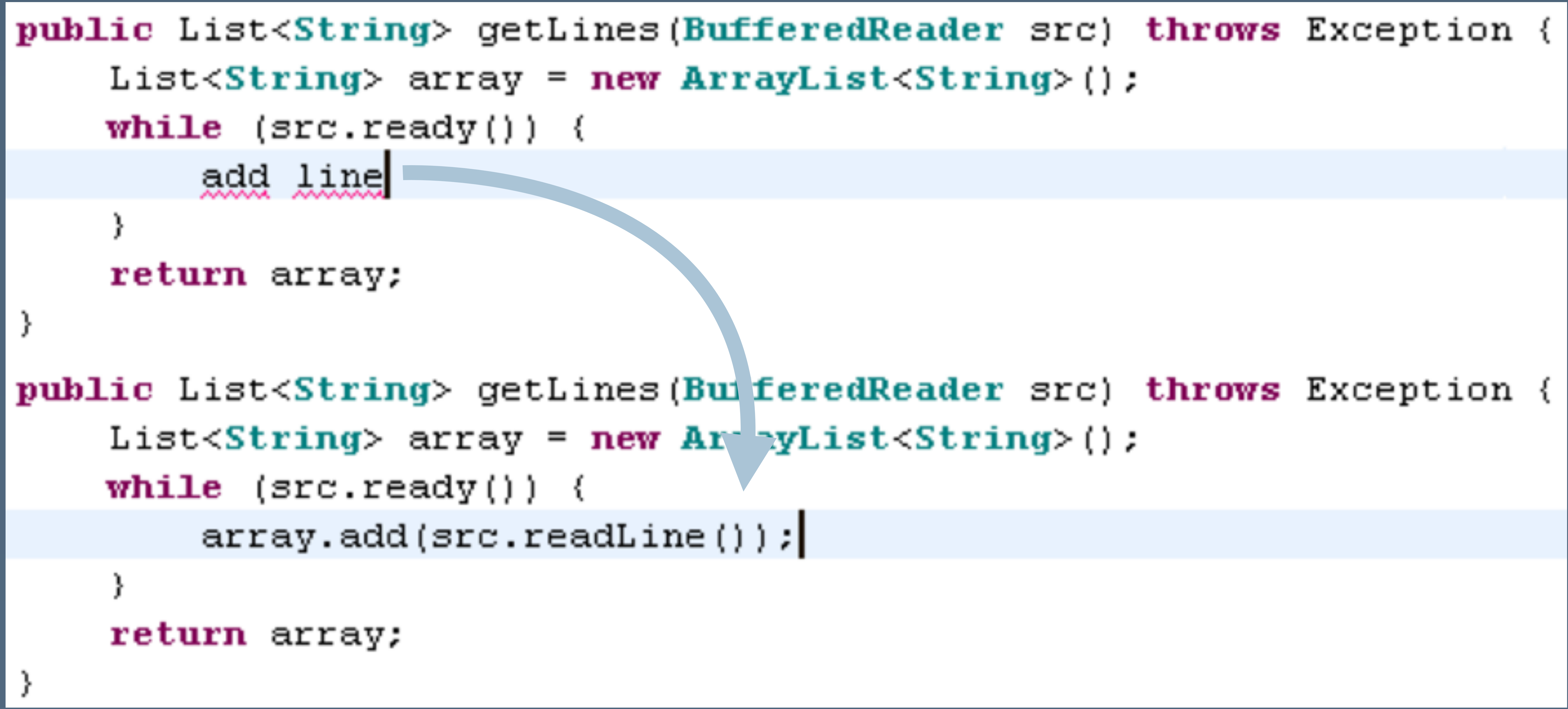
When the user performs a

- mouseout (Calculator.html 603),
- mouseout (Calculator.html 947),
- mouseout (Calculator.html 1025),
- mouseout (Calculator.html 598)

Keyword programming

[Little and Miller, UIST '06, ASE '09]

- Macro programming is difficult to learn
- Allow keyword search over an API: e.g., "click search button" or "left margin 2 inches"



```
public List<String> getLines(BufferedReader src) throws Exception {
    List<String> array = new ArrayList<String>();
    while (src.ready()) {
        add line
    }
    return array;
}
```

```
public List<String> getLines(BufferedReader src) throws Exception {
    List<String> array = new ArrayList<String>();
    while (src.ready()) {
        array.add(src.readLine());
    }
    return array;
}
```

Visual layout of code snippets

[Bragdon et al., CHI '10]

- Most engineering time is spent navigating across multiple related code snippets
- So, design for many small windows into files

ShapeDraw
MainPanel.createMenuBar Undo
Errors (0)
Warnings (0)

```
ShapeDraw ▸ MainPanel ▸
public MainPanel()
{
    this.layoutAsCardinalDirections();

    this.createPropertyButtons();

    Button featureButton = SpecialFeatureButton.
        getInstance(this);

    Button randomShapes = this.
        createRandomShapeButton();

    String[] messages = this.
        generateStatisticsMessages();
    this.handleStatisticsGUI(messages);

    MenuBar menuBar = this.createMenuBar();

    ShapeButton[] shapeButtons = this.
        createShapeButtons();
    Panel shapePanel = this.makeShapeButtonPanel(
        shapeButtons);

    Panel moreFunctionsPanel = new Panel();
    moreFunctionsPanel.layoutAsGrid();
    Label moreFunctionsLabel = new Label(
        "More Functions");
    moreFunctionsLabel.center();
    moreFunctionsPanel.add(moreFunctionsLabel);
    moreFunctionsPanel.add(randomShapes);
    moreFunctionsPanel.add(_deleteShape);
    moreFunctionsPanel.add(_statsButton);
    moreFunctionsPanel.add(featureButton);

    _shapeInfoPanel = new ShapeInfoPanel();
```

```
ShapeDraw ▸ MainPanel ▸
public void createPropertyButtons()
{
    _deleteShape = DeleteButton.getInstance(
        this);
    ((ShapeButton) _deleteShape).storeName(
        "Delete Active Shape");
    _deleteShape.setFocusable(false);

    init();
}
```

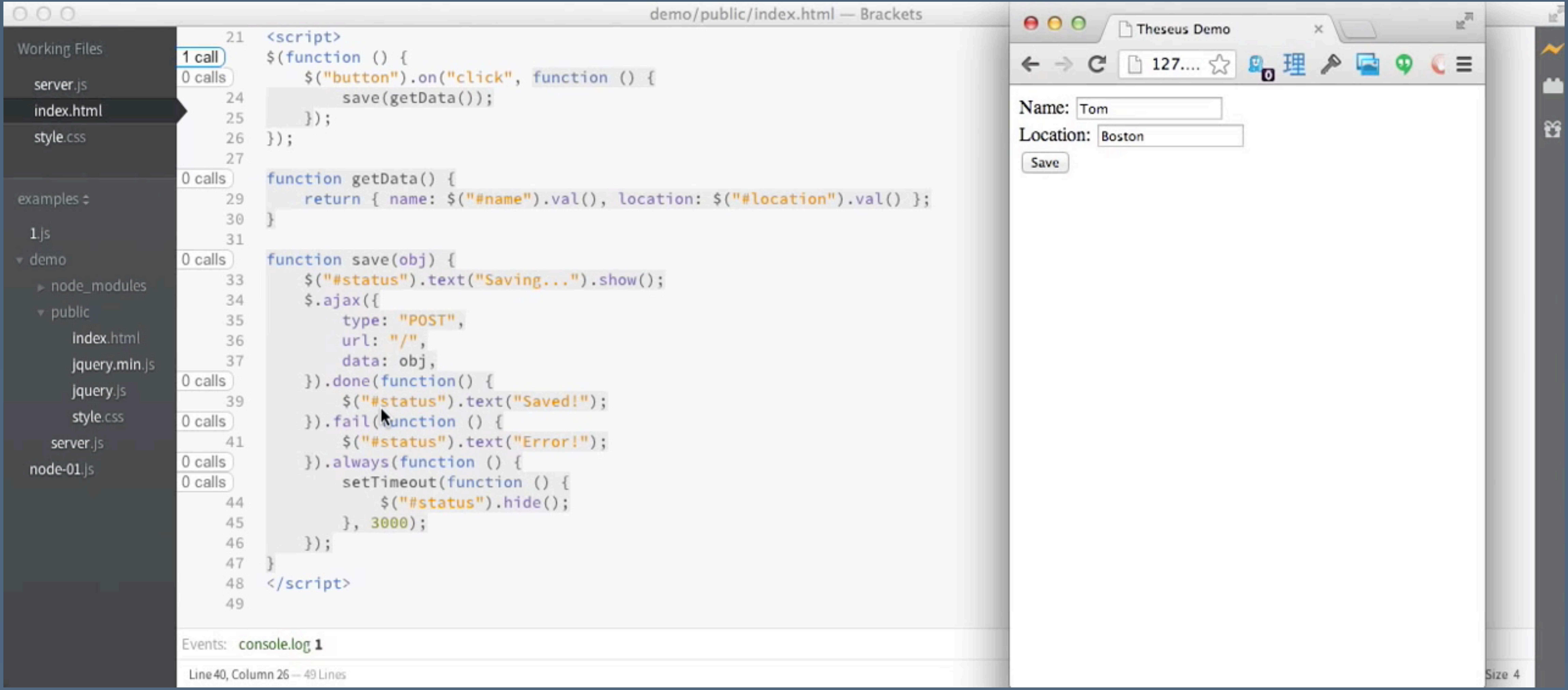
```
ShapeDraw ▸ MainPanel ▸
public Button createRandomShapeButton()
{
    Button button = Dropdown.getInstance(
        this);
    button.setFocusable(false);

    return button;
}
```

```
ShapeDraw ▸ MainPanel ▸
private void createMenu1(Menu m)
{
    MenuItem textInput = TextMenuButton.
        getInstance();
    m.add(textInput);
}
```

```
ShapeDraw ▸ TextMenuButton ▸
public static MenuItem getInstance()
{
    TextMenuButton item= new TextMenuButton();
    item.setText("Text Input");
    return item;
}
```

Debugging with runtime info

[Lieber, Brandt, and Miller, CHI 2014]



```
<script>
$(function () {
    $("button").on("click", function () {
        save(getData());
    });
});

function getData() {
    return { name: $("#name").val(), location: $("#location").val() };
}

function save(obj) {
    $("#status").text("Saving...").show();
    $.ajax({
        type: "POST",
        url: "/",
        data: obj,
    }).done(function() {
        $("#status").text("Saved!");
    }).fail(function () {
        $("#status").text("Error!");
    }).always(function () {
        setTimeout(function () {
            $("#status").hide();
        }, 3000);
    });
}
</script>
```

Events: console.log 1

Line 40, Column 26 — 49 Lines

Size 4

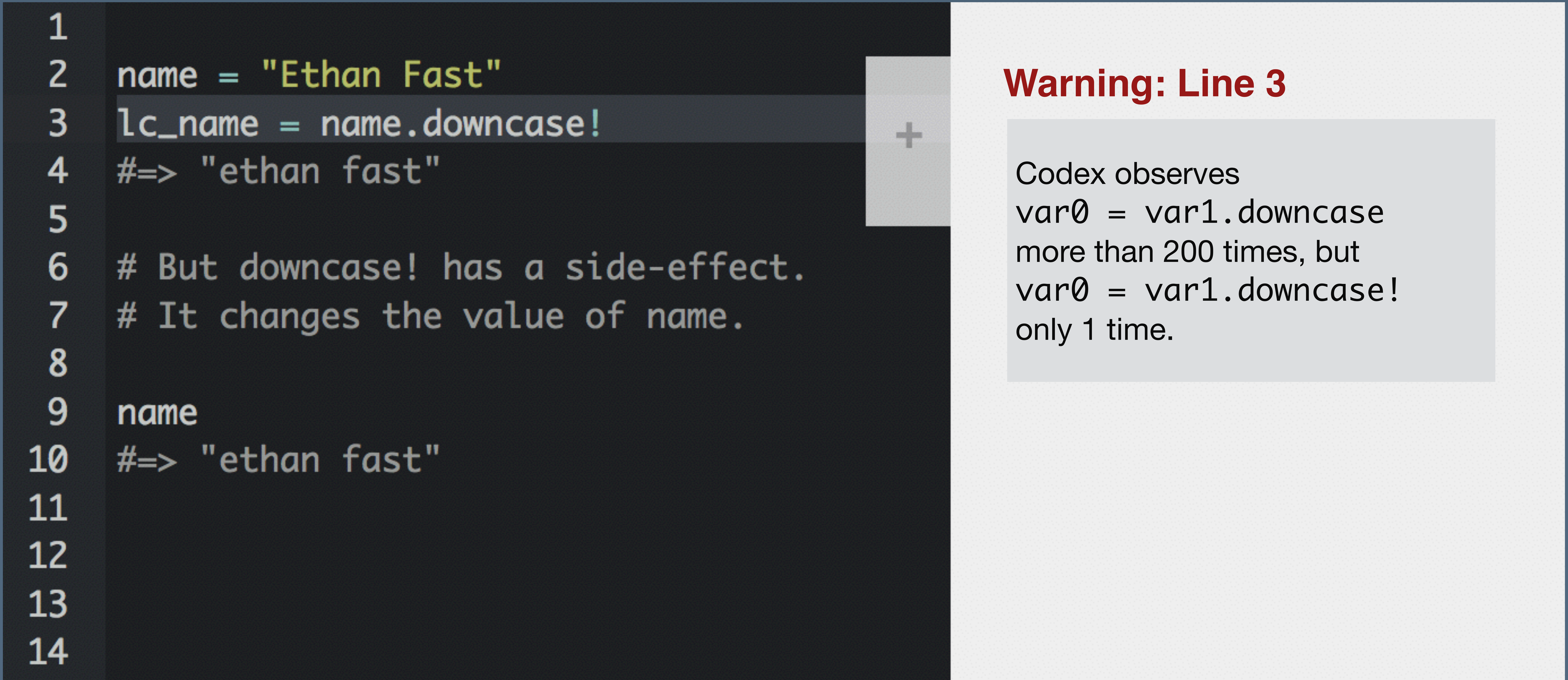
Name: Tom

Location: Boston

Save

Emergent programming practice

[Fast et al., CHI 2014]



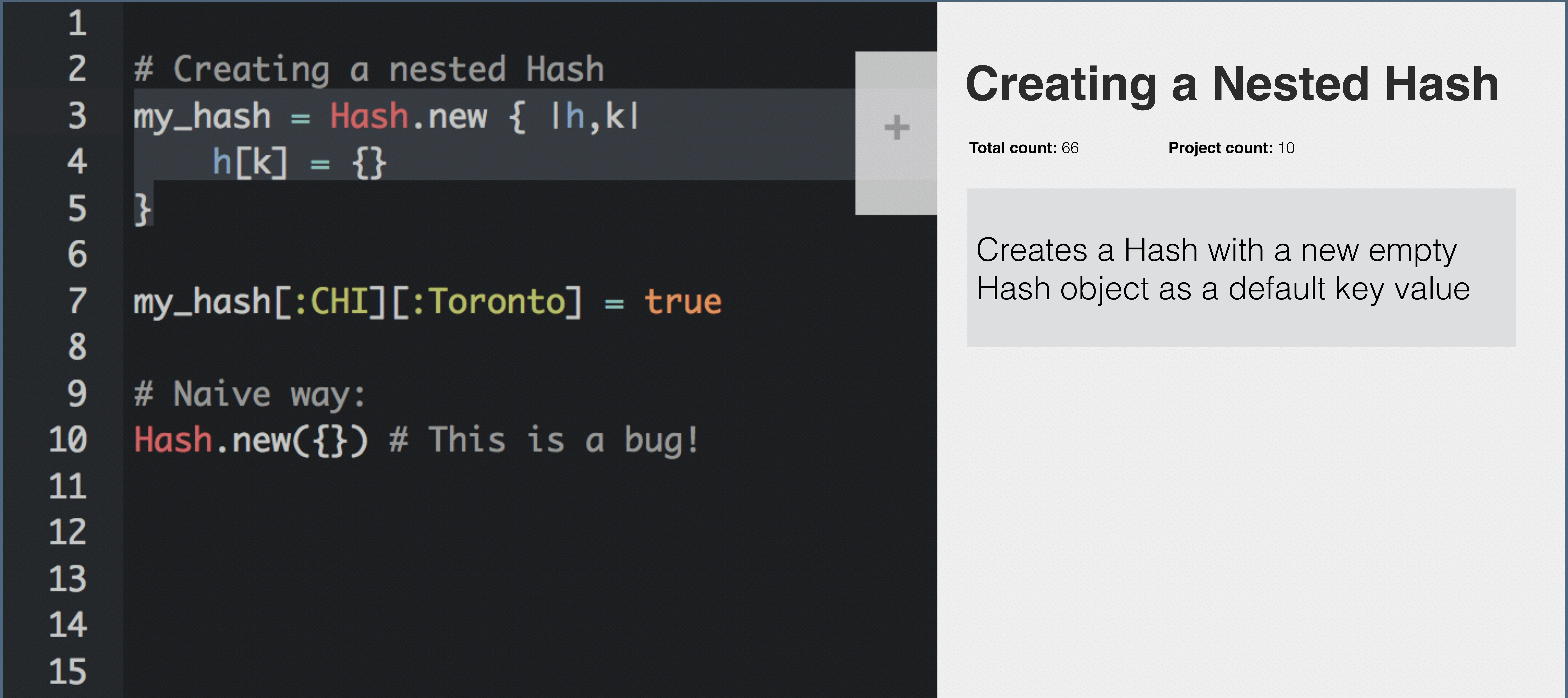
```
 1
 2  name = "Ethan Fast"
 3  lc_name = name.downcase!
 4  #=> "ethan fast"
 5
 6  # But downcase! has a side-effect.
 7  # It changes the value of name.
 8
 9  name
10  #=> "ethan fast"
11
12
13
14
```

Warning: Line 3

Codex observes
`var0 = var1.downcase`
more than 200 times, but
`var0 = var1.downcase!`
only 1 time.

Emergent programming practice

[Fast et al., CHI 2014]



```
# Creating a nested Hash
my_hash = Hash.new { |h,k|
     h[k] = {}
}

my_hash[:CHI][:Toronto] = true

# Naive way:
Hash.new({}) # This is a bug!
```

Creating a Nested Hash

Total count: 66 **Project count:** 10

Creates a Hash with a new empty Hash object as a default key value

Learning programming

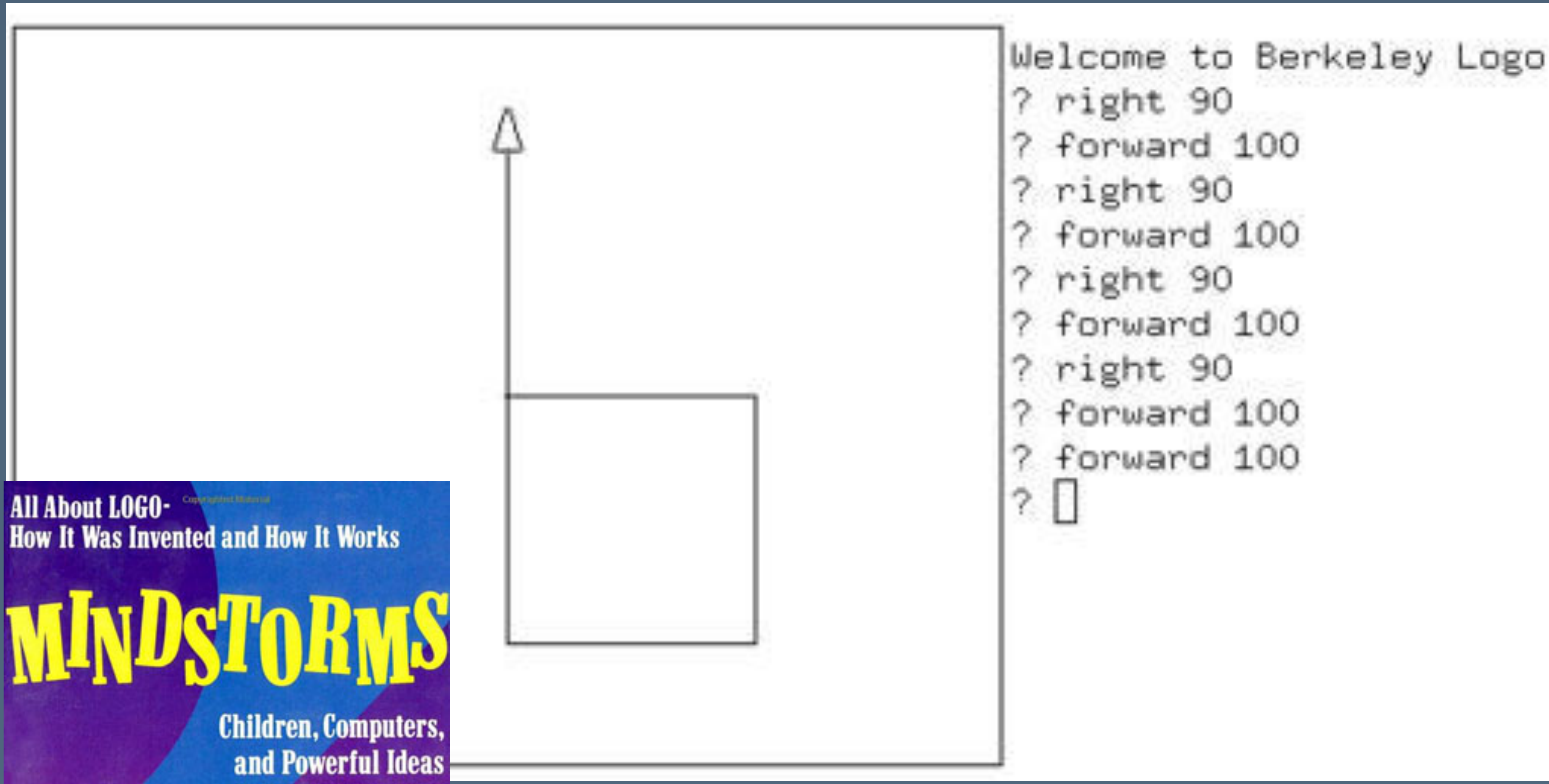
Goals of programming education

- Make programming accessible to new populations: children, scripters, interested amateurs
- Tools and innovations depend on the population

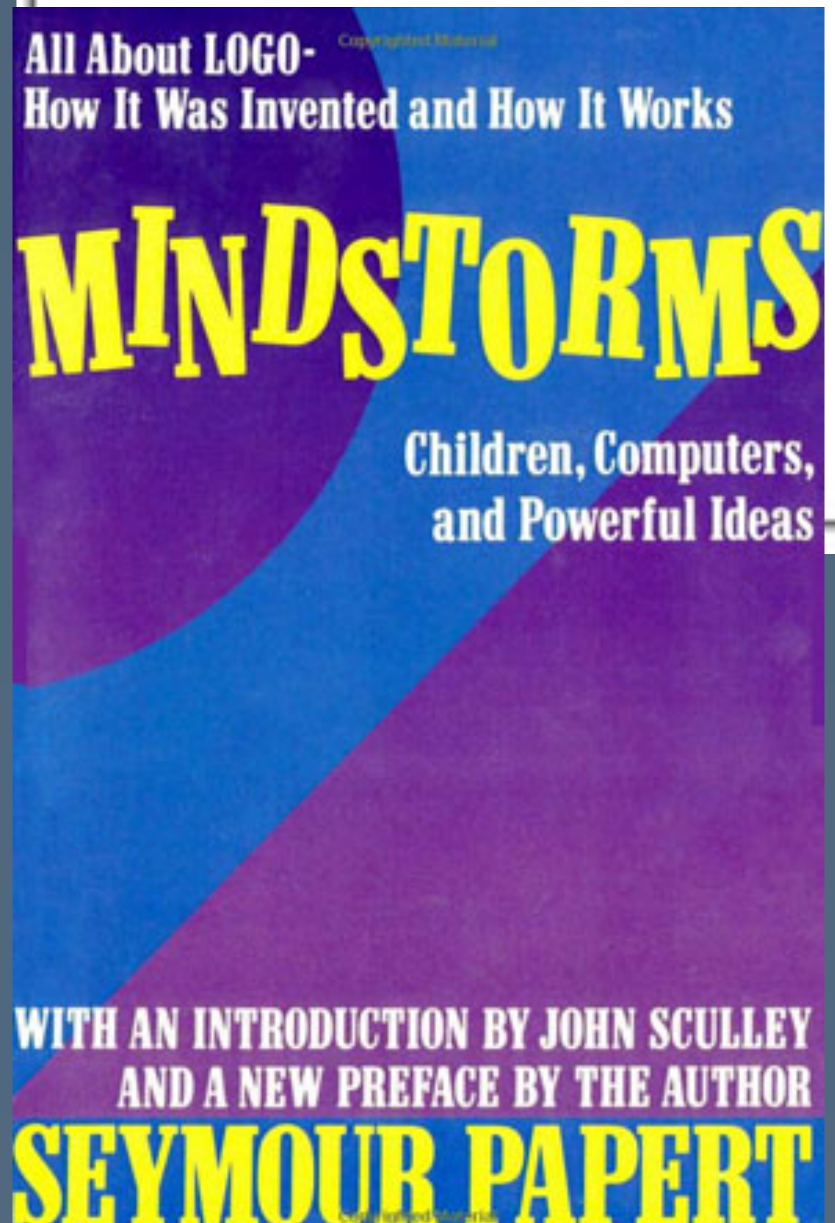
Logo: programming for children

[Papert '93]

- Constructionist learning: learning happens most effectively when people are making tangible objects
- Lego Mindstorms followed this mold and was named after it



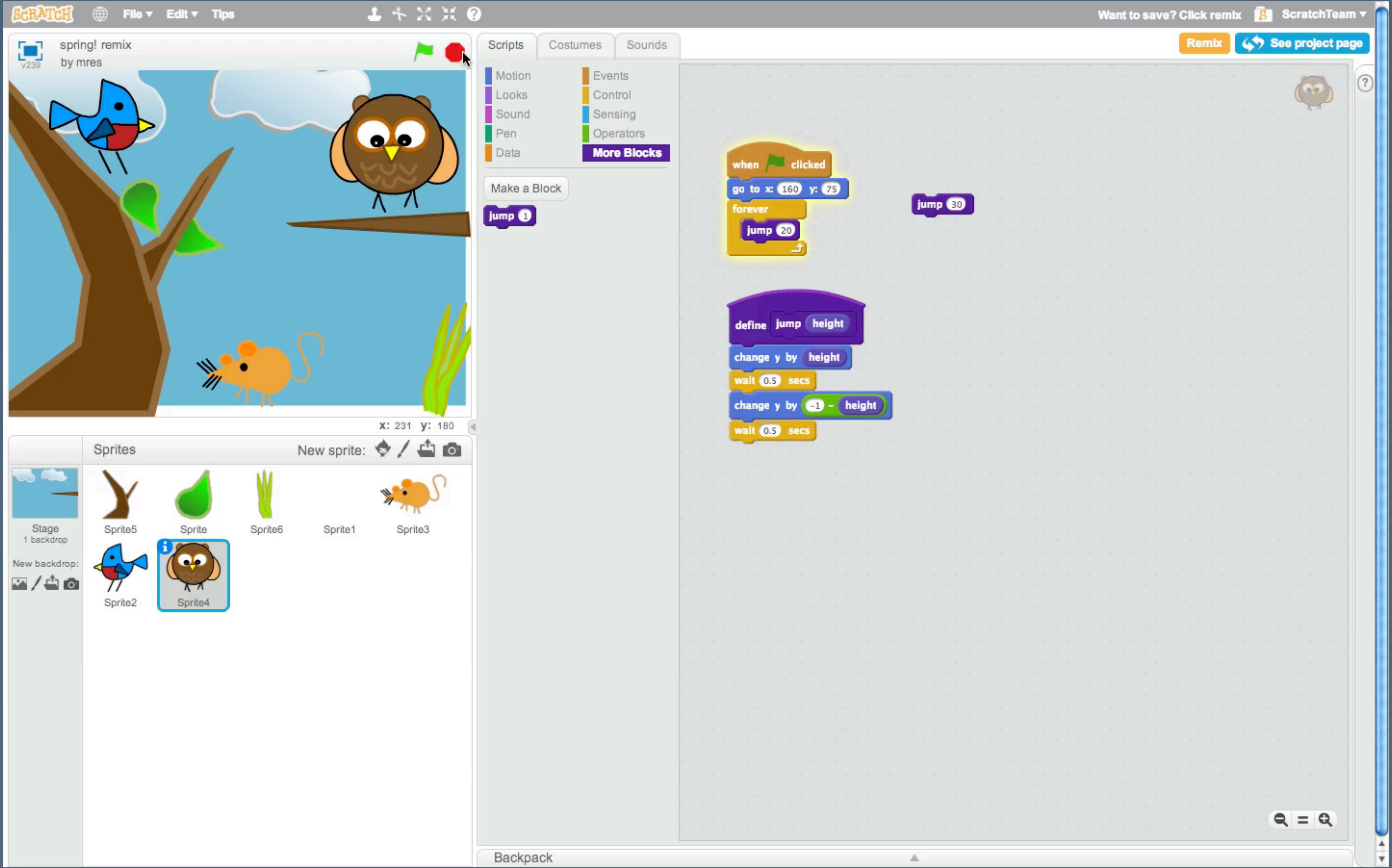
```
Welcome to Berkeley Logo
? right 90
? forward 100
? right 90
? forward 100
? right 90
? forward 100
? right 90
? forward 100
? forward 100
? 
```


All About LOGO-
How It Was Invented and How It Works
MINDSTORMS
Children, Computers, and Powerful Ideas
WITH AN INTRODUCTION BY JOHN SCULLEY
AND A NEW PREFACE BY THE AUTHOR
SEYMOUR PAPERT

Scratch: kids remix and create

[Resnick et al., CACM ’09]

- Social: upload and remix others’ programs
- All programming has been done online. This data has led to many papers on understanding notions of authorship and creative remixing.



```
when [flag] clicked
go to x: 160 y: 75
forever
    jump 20
```

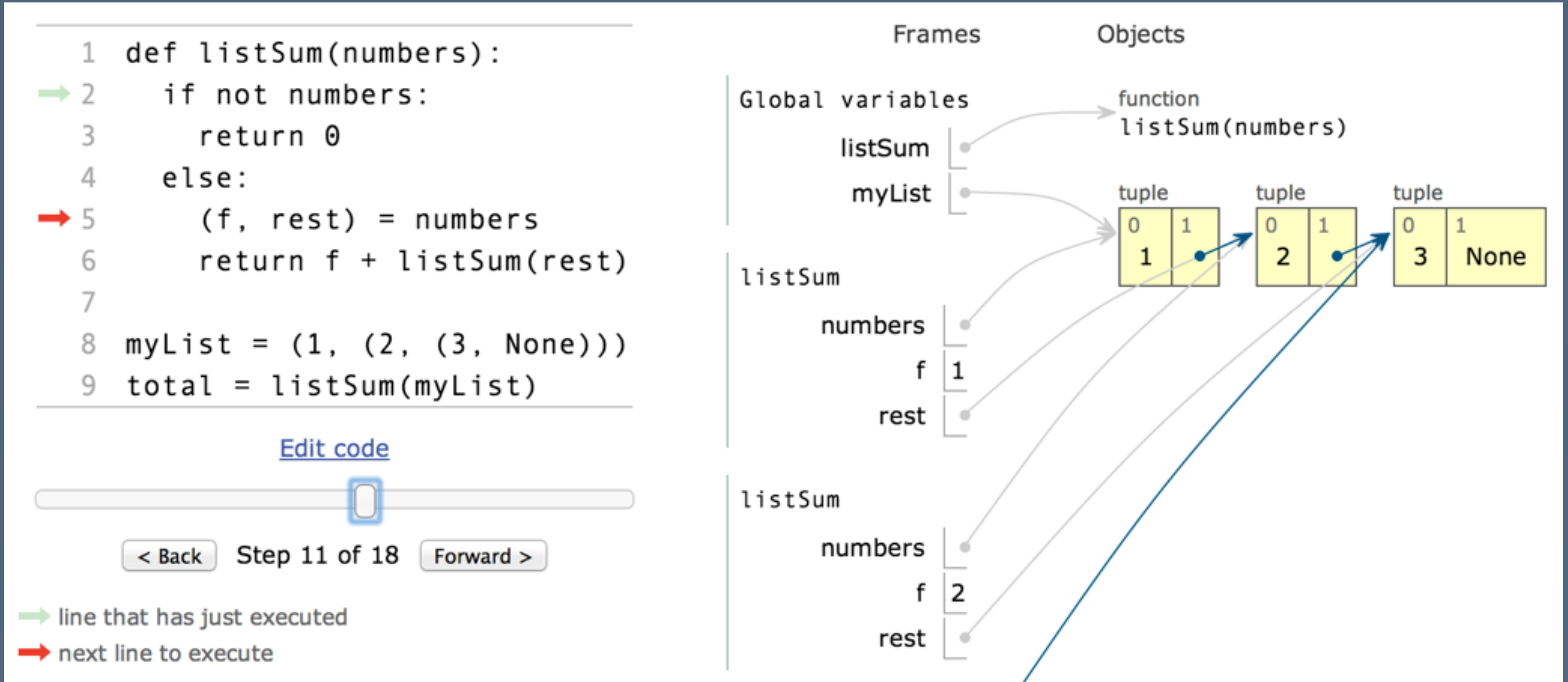
```
jump 30
```

```
define jump height
change y by height
wait 0.5 secs
change y by (-1 - height)
wait 0.5 secs
```

Online python tutor

[Guo, SIGCSE '13]

- Embeddable Python data structure visualization
- Over 200,000 users and a dozen universities using it



```
def listSum(numbers):
  if not numbers:
    return 0
  else:
    (f, rest) = numbers
    return f + listSum(rest)

myList = (1, (2, (3, None)))
total = listSum(myList)
```

Programming by demonstration

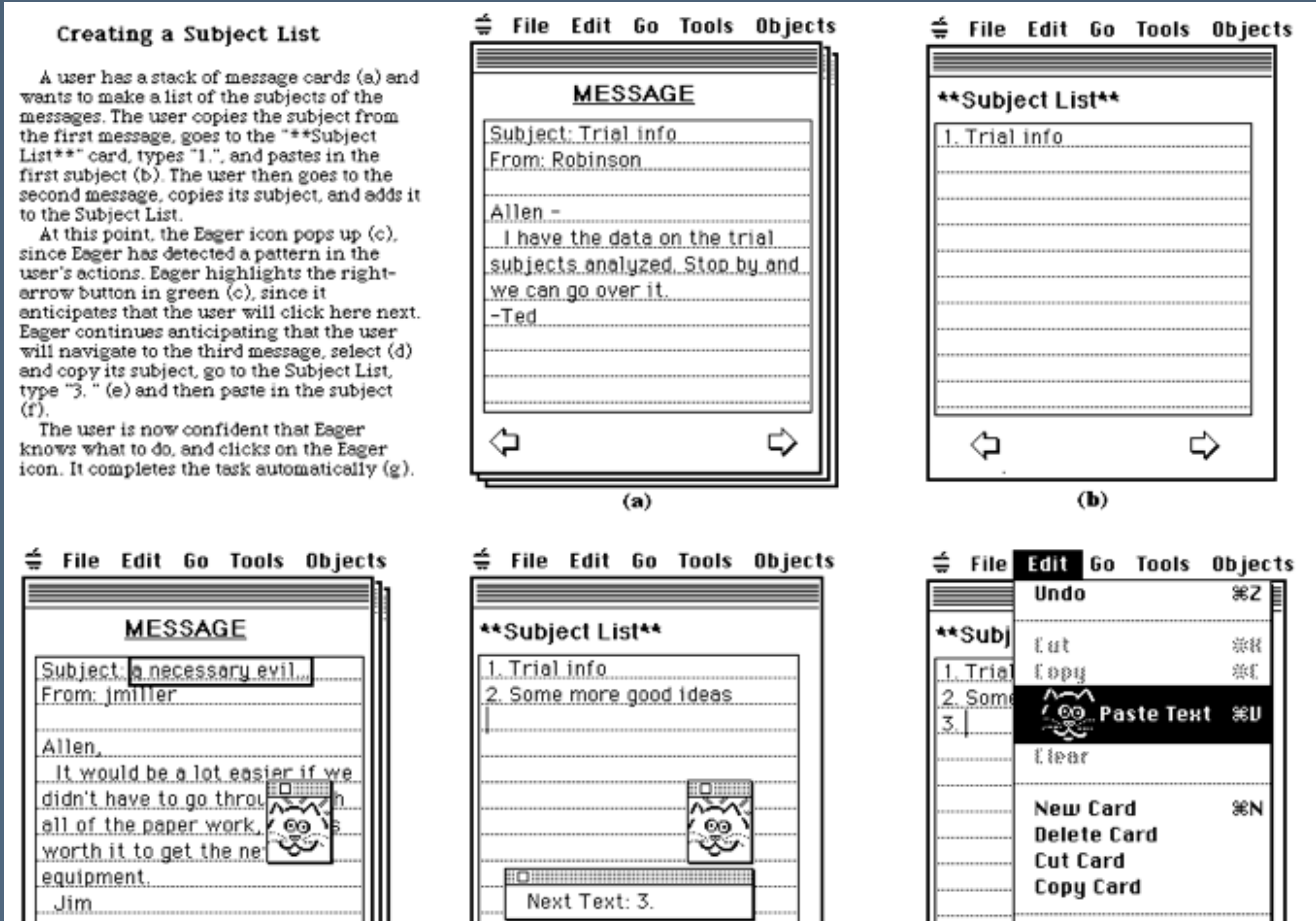
Goals of PBD

- Teach a computer to program simply by demonstrating what should be done
- Challenges
  - There is an infinite, and hugely branching, space of programs that might be inferred
  - Inferred macros can be extremely brittle

Recall: EAGER

[Cypher, CHI '91]

- Infer a macro by watching the user's behavior



Creating a Subject List

A user has a stack of message cards (a) and wants to make a list of the subjects of the messages. The user copies the subject from the first message, goes to the "**Subject List**" card, types "1.", and pastes in the first subject (b). The user then goes to the second message, copies its subject, and adds it to the Subject List.

At this point, the Eager icon pops up (c), since Eager has detected a pattern in the user's actions. Eager highlights the right-arrow button in green (c), since it anticipates that the user will click here next. Eager continues anticipating that the user will navigate to the third message, select (d) and copy its subject, go to the Subject List, type "3. " (e) and then paste in the subject (f).

The user is now confident that Eager knows what to do, and clicks on the Eager icon. It completes the task automatically (g).

MESSAGE

Subject: Trial info
From: Robinson

Allen -
I have the data on the trial subjects analyzed. Stop by and we can go over it.
-Ted

(a)

Subject List

1. Trial info

(b)

MESSAGE

Subject: a necessary evil.
From: jmiller

Allen,
It would be a lot easier if we didn't have to go throu h all of the paper work, s worth it to get the ne equipment.
Jim

Subject List

1. Trial info
2. Some more good ideas

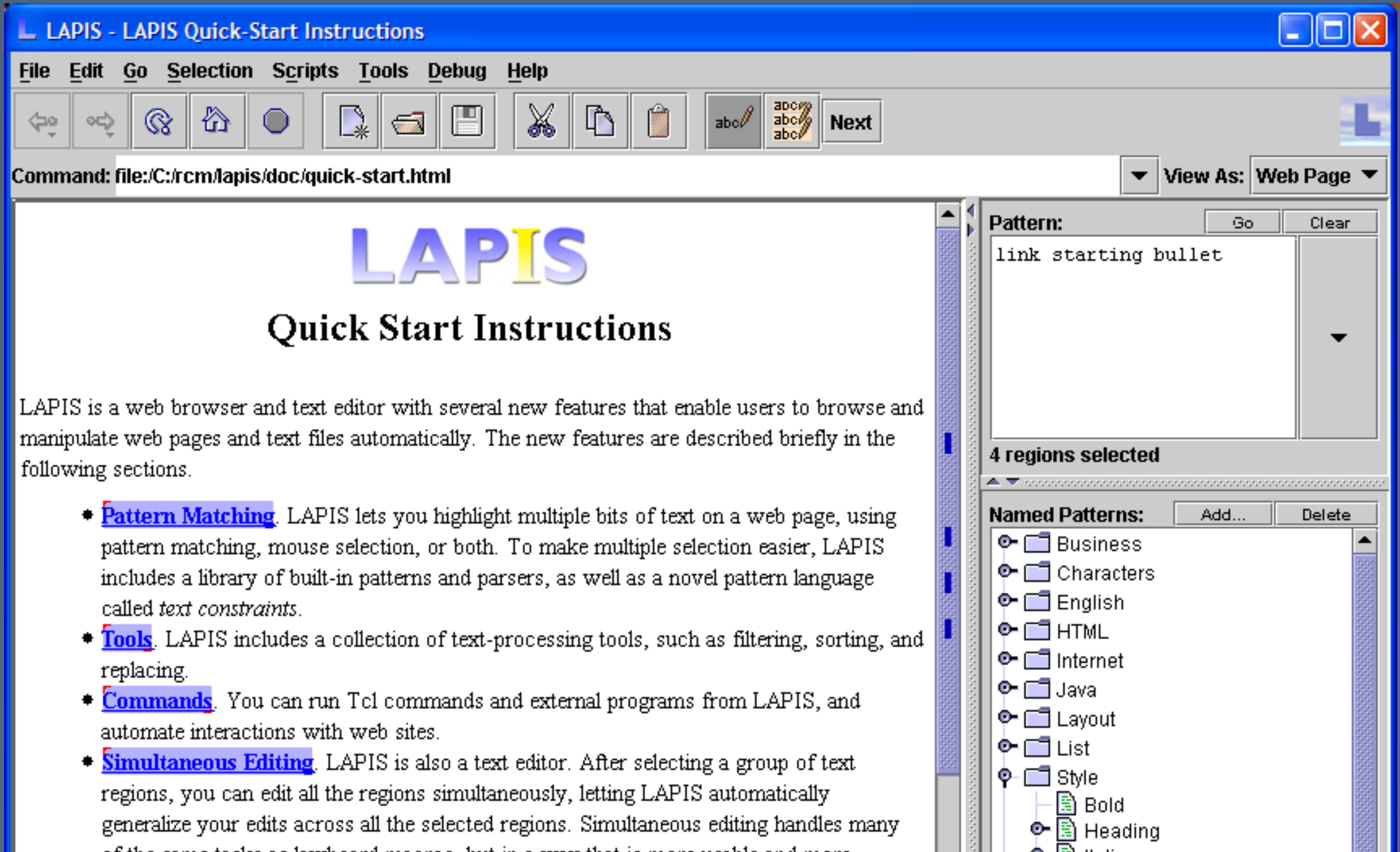
Next Text: 3.

**Subj
1. Trial
2. Some
3.

Simultaneous structured editing

[Miller and Myers, USENIX '01]

- Utilize lightweight structure in text
- Today, versions of this exist in Sublime Text


LAPIS - LAPIS Quick-Start Instructions
File Edit Go Selection Scripts Tools Debug Help
Next
View As: Web Page
Go Clear
Add... Delete

Command: file:/C:/rcm/lapis/doc/quick-start.html

Pattern: link starting bullet

4 regions selected

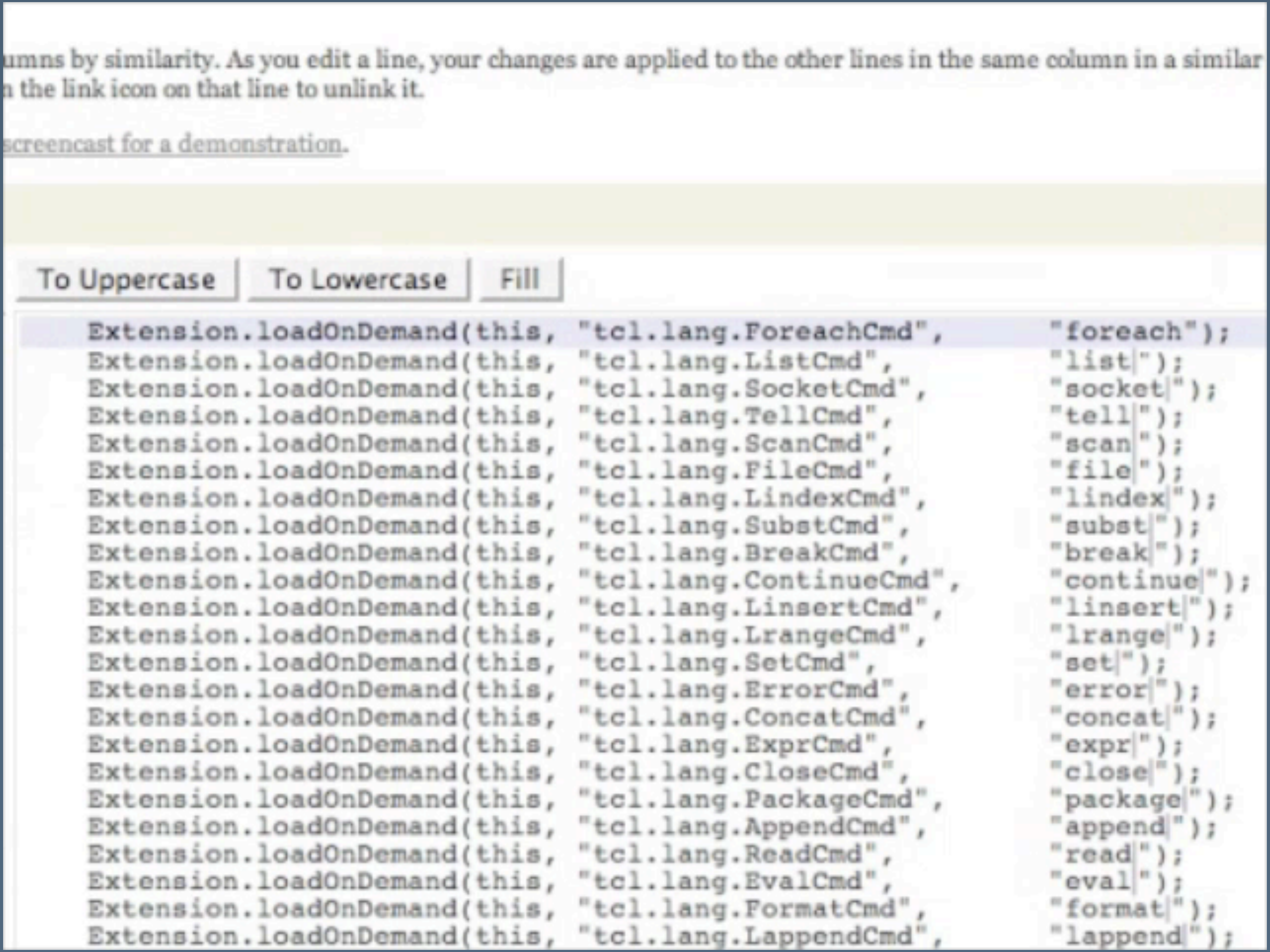
Named Patterns: Business, Characters, English, HTML, Internet, Java, Layout, List, Style, Bold, Heading

LAPIS

Quick Start Instructions

LAPIS is a web browser and text editor with several new features that enable users to browse and manipulate web pages and text files automatically. The new features are described briefly in the following sections.

- Pattern Matching. LAPIS lets you highlight multiple bits of text on a web page, using pattern matching, mouse selection, or both. To make multiple selection easier, LAPIS includes a library of built-in patterns and parsers, as well as a novel pattern language called *text constraints*.
- Tools. LAPIS includes a collection of text-processing tools, such as filtering, sorting, and replacing.
- Commands. You can run Tcl commands and external programs from LAPIS, and automate interactions with web sites.
- Simultaneous Editing. LAPIS is also a text editor. After selecting a group of text regions, you can edit all the regions simultaneously, letting LAPIS automatically generalize your edits across all the selected regions. Simultaneous editing handles many


umns by similarity. As you edit a line, your changes are applied to the other lines in the same column in a similar
n the link icon on that line to unlink it.

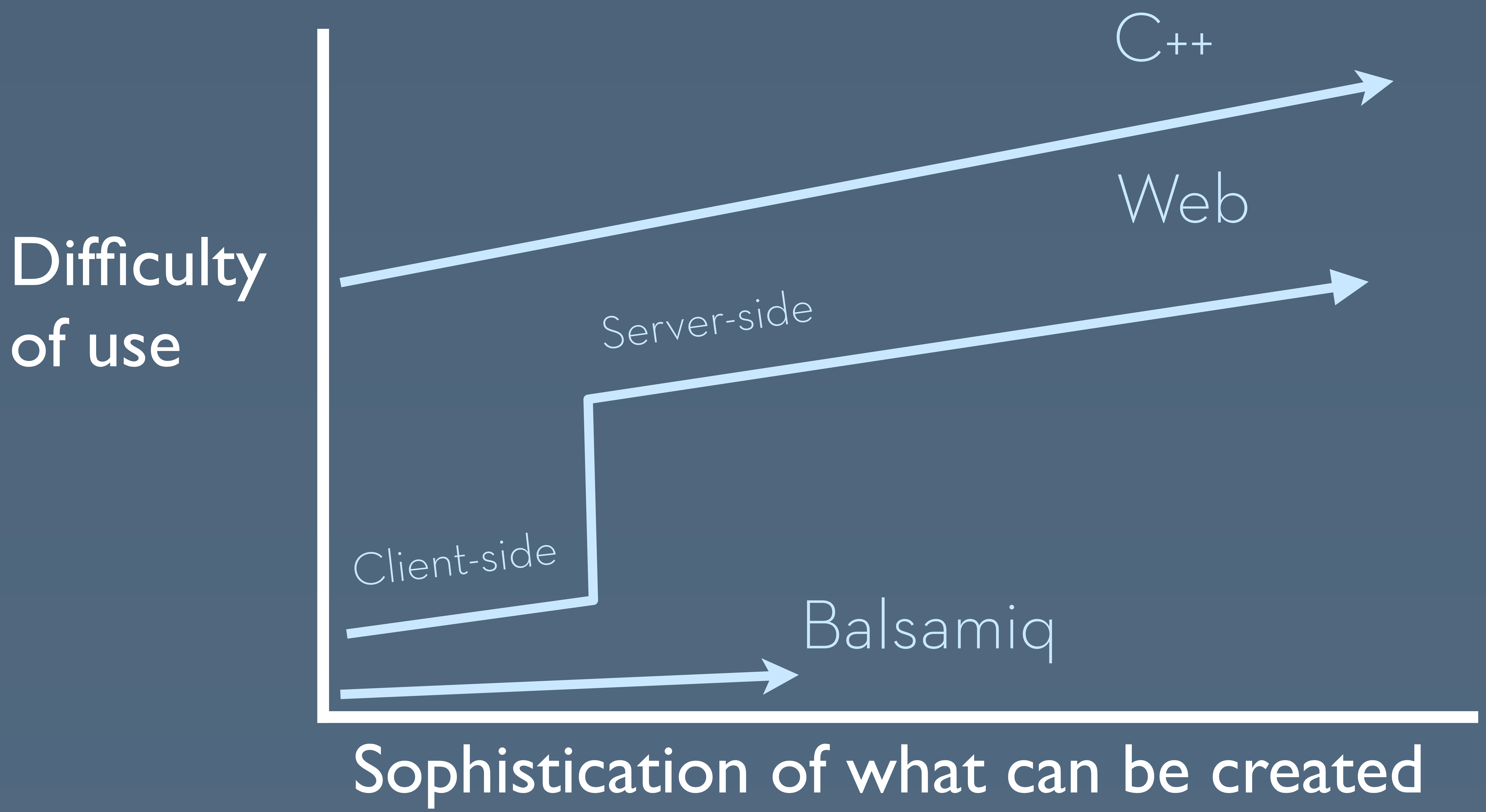
screencast for a demonstration.

```
    Extension.loadOnDemand(this, "tcl.lang.ForeachCmd",       "foreach");
    Extension.loadOnDemand(this, "tcl.lang.ListCmd",          "list");
    Extension.loadOnDemand(this, "tcl.lang.SocketCmd",        "socket");
    Extension.loadOnDemand(this, "tcl.lang.TellCmd",          "tell");
    Extension.loadOnDemand(this, "tcl.lang.ScanCmd",          "scan");
    Extension.loadOnDemand(this, "tcl.lang.FileCmd",          "file");
    Extension.loadOnDemand(this, "tcl.lang.LindexCmd",        "lindex");
    Extension.loadOnDemand(this, "tcl.lang.SubstCmd",         "subst");
    Extension.loadOnDemand(this, "tcl.lang.BreakCmd",         "break");
    Extension.loadOnDemand(this, "tcl.lang.ContinueCmd",      "continue");
    Extension.loadOnDemand(this, "tcl.lang.LinsertCmd",       "linsert");
    Extension.loadOnDemand(this, "tcl.lang.LrangeCmd",        "lrange");
    Extension.loadOnDemand(this, "tcl.lang.SetCmd",           "set");
    Extension.loadOnDemand(this, "tcl.lang.ErrorCmd",         "error");
    Extension.loadOnDemand(this, "tcl.lang.ConcatCmd",        "concat");
    Extension.loadOnDemand(this, "tcl.lang.ExprCmd",          "expr");
    Extension.loadOnDemand(this, "tcl.lang.CloseCmd",         "close");
    Extension.loadOnDemand(this, "tcl.lang.PackageCmd",       "package");
    Extension.loadOnDemand(this, "tcl.lang.AppendCmd",        "append");
    Extension.loadOnDemand(this, "tcl.lang.ReadCmd",          "read");
    Extension.loadOnDemand(this, "tcl.lang.EvalCmd",          "eval");
    Extension.loadOnDemand(this, "tcl.lang.FormatCmd",        "format");
    Extension.loadOnDemand(this, "tcl.lang.LappendCmd",       "lappend");
```

Toolkits

Threshold/Ceiling Tradeoff

[Myers, Hudson and Pausch, TOCHI 2000]



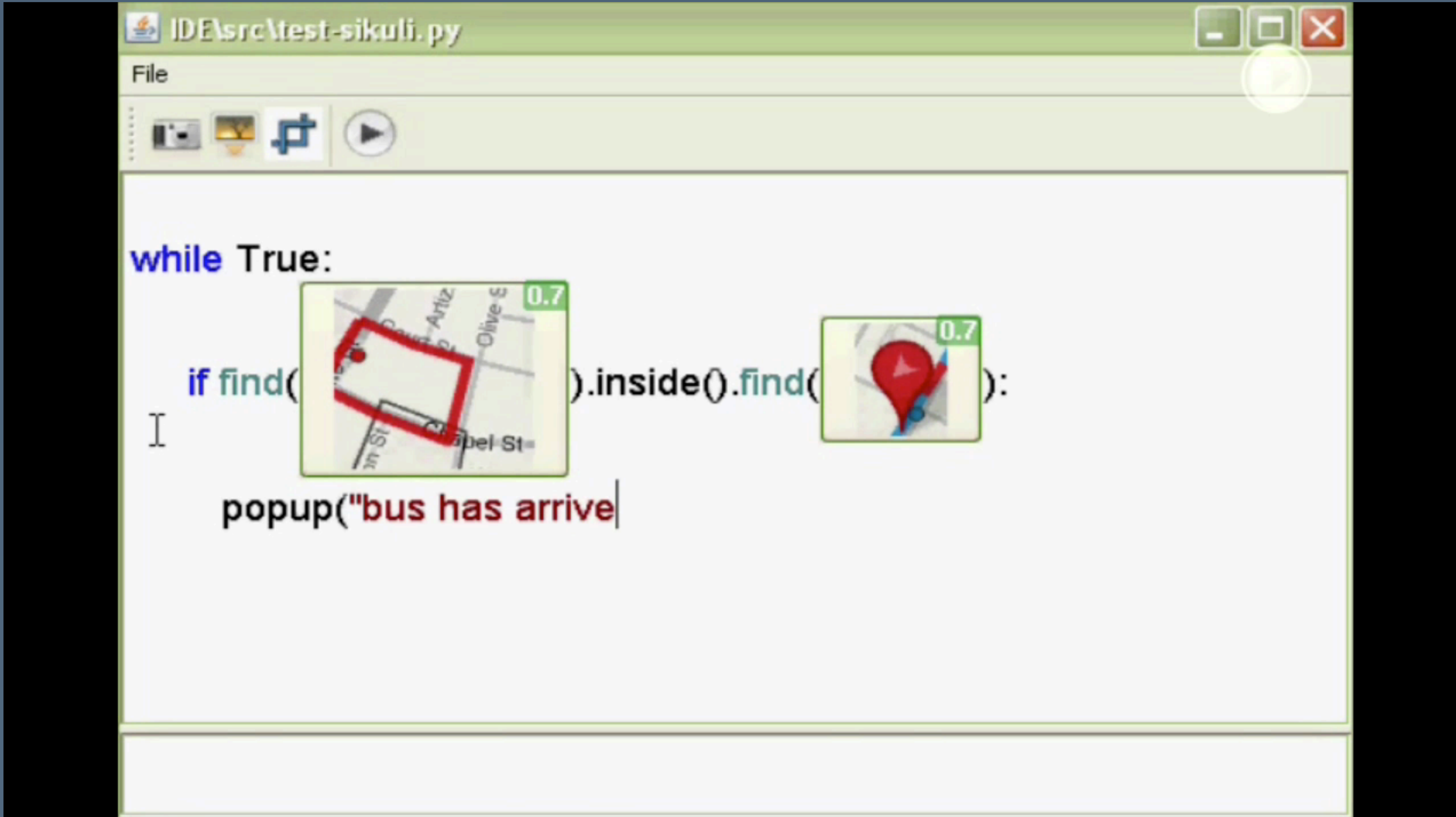
Research agenda: toolkits

- Crystallize and formalize a perspective on a difficult engineering problem
- If successful, shift the entire programming practice for the area

Sikuli: programming with screenshots

[Yeh, Chang, and Miller, UIST '09]

- Visual template search in desktop scripting



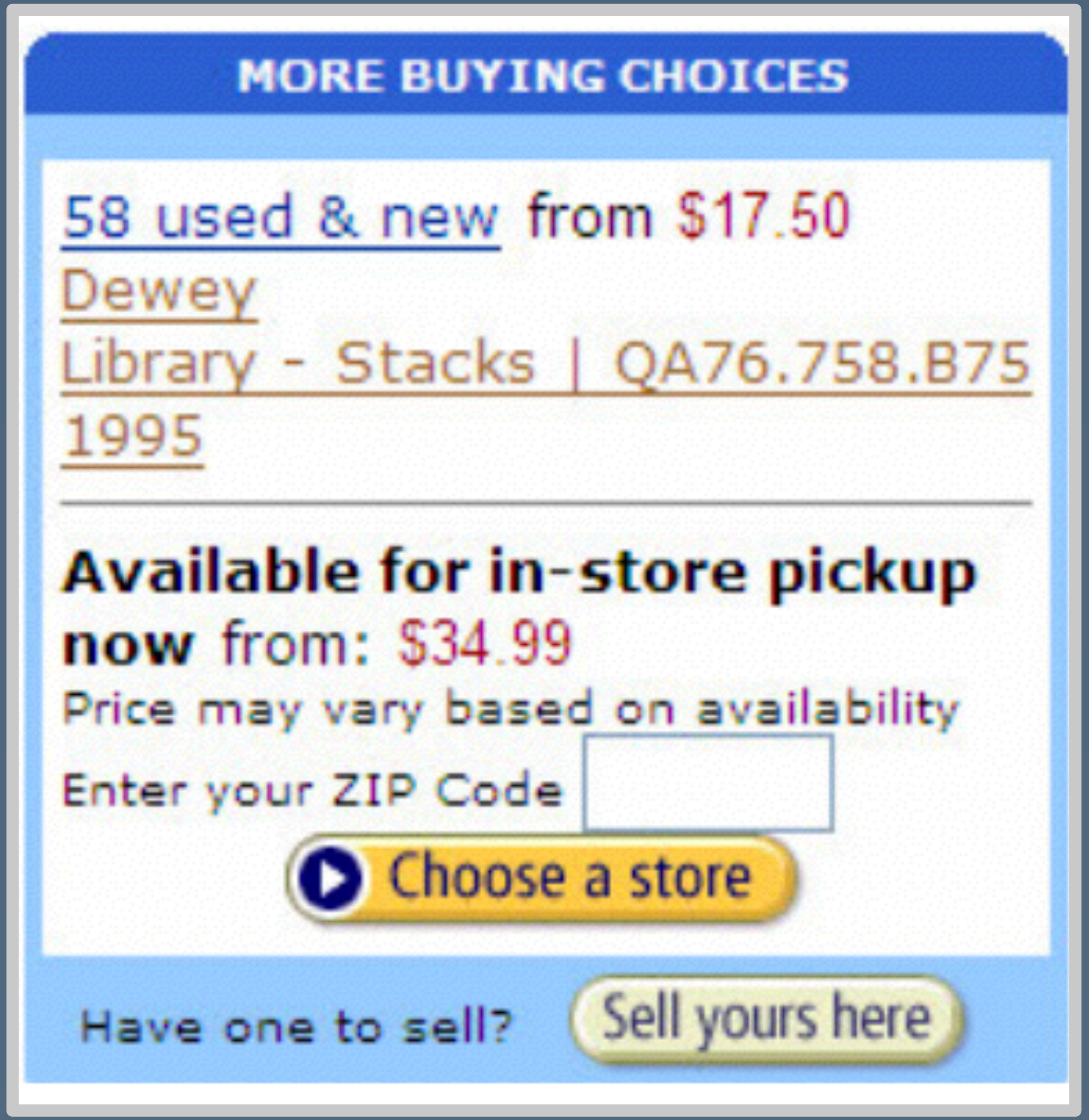
```
while True:
    if find(    ).inside().find(    ):
        popup("bus has arrive
```

Recall: Chickenfoot

[Bolin et al., UIST 2008]

- Lower the threshold to writing programs
- Allow users with little programming skill to author behaviors
  - e.g., Chickenfoot

```
isbn = find('number just after isbn')
with (fetch('libraries.mit.edu')) {
 pick('Keywords’);
 enter(isbn)
 click('Search')
 link=find('link just after Location')
}
// back to Amazon
if (link.hasMatch) {
 insert(before('first rule after "Buying"'),
 link.html)
```



MORE BUYING CHOICES

58 used & new from $17.50

Dewey
Library - Stacks | QA76.758.B75
1995

Available for in-store pickup now from: $34.99

Price may vary based on availability

Enter your ZIP Code

Research agenda: HCI and programming

- Understand the challenges in programming
- Design more effective software engineering interfaces
- Aid novices in learning to program or writing programs
- Abstract best practices into toolkits