Back to the Future

Lessons from the Past for Today's Web Developers

Bill Scott
2008 Yahoo! Front End Developer's Summit
10.08.2008

“‘There's an old saying about those who forget history. I don't remember it, but it's good.’’

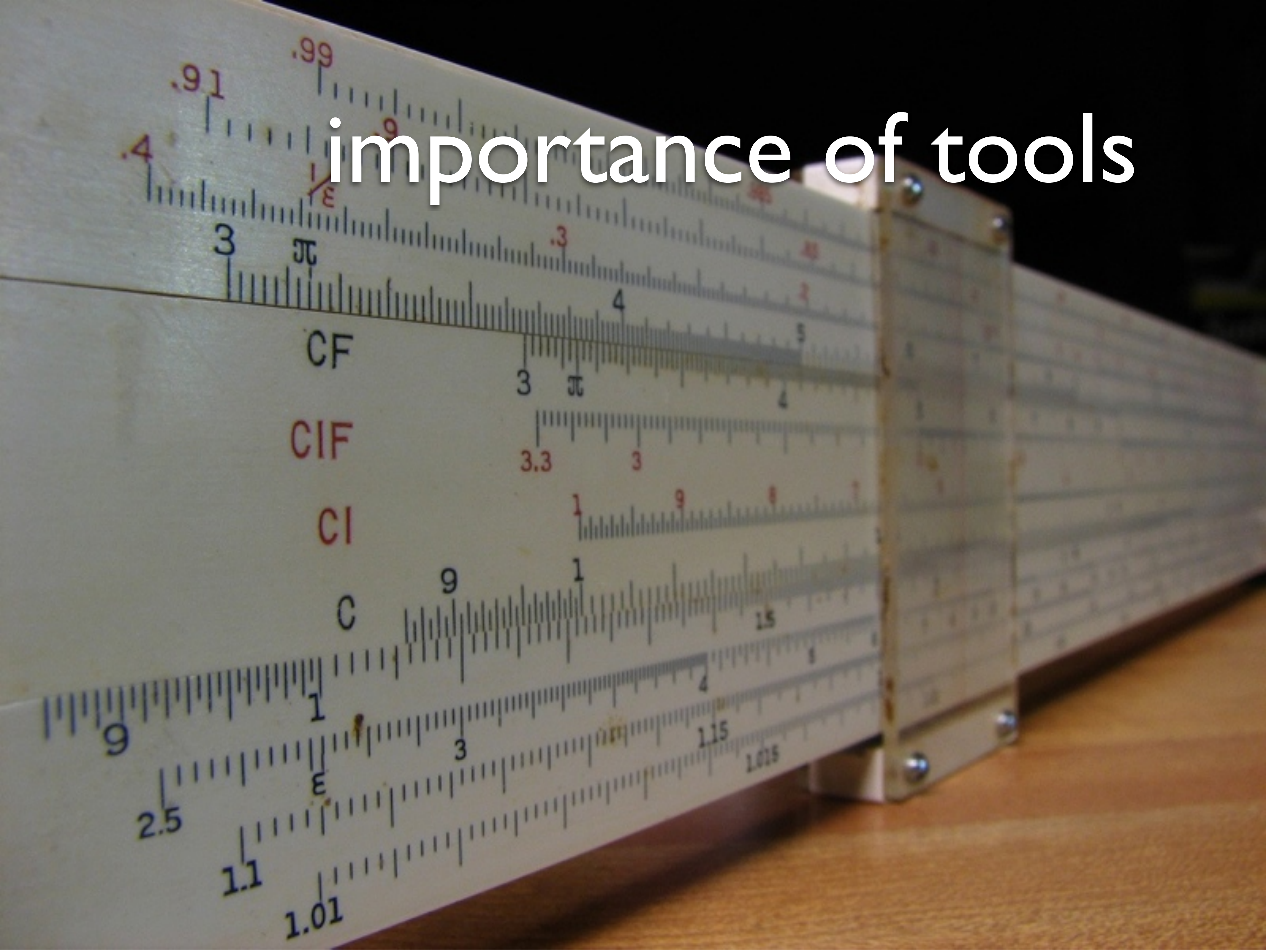
Stephen Colbert, The Colbert Report, March 10, 2008

Lessons from the Past

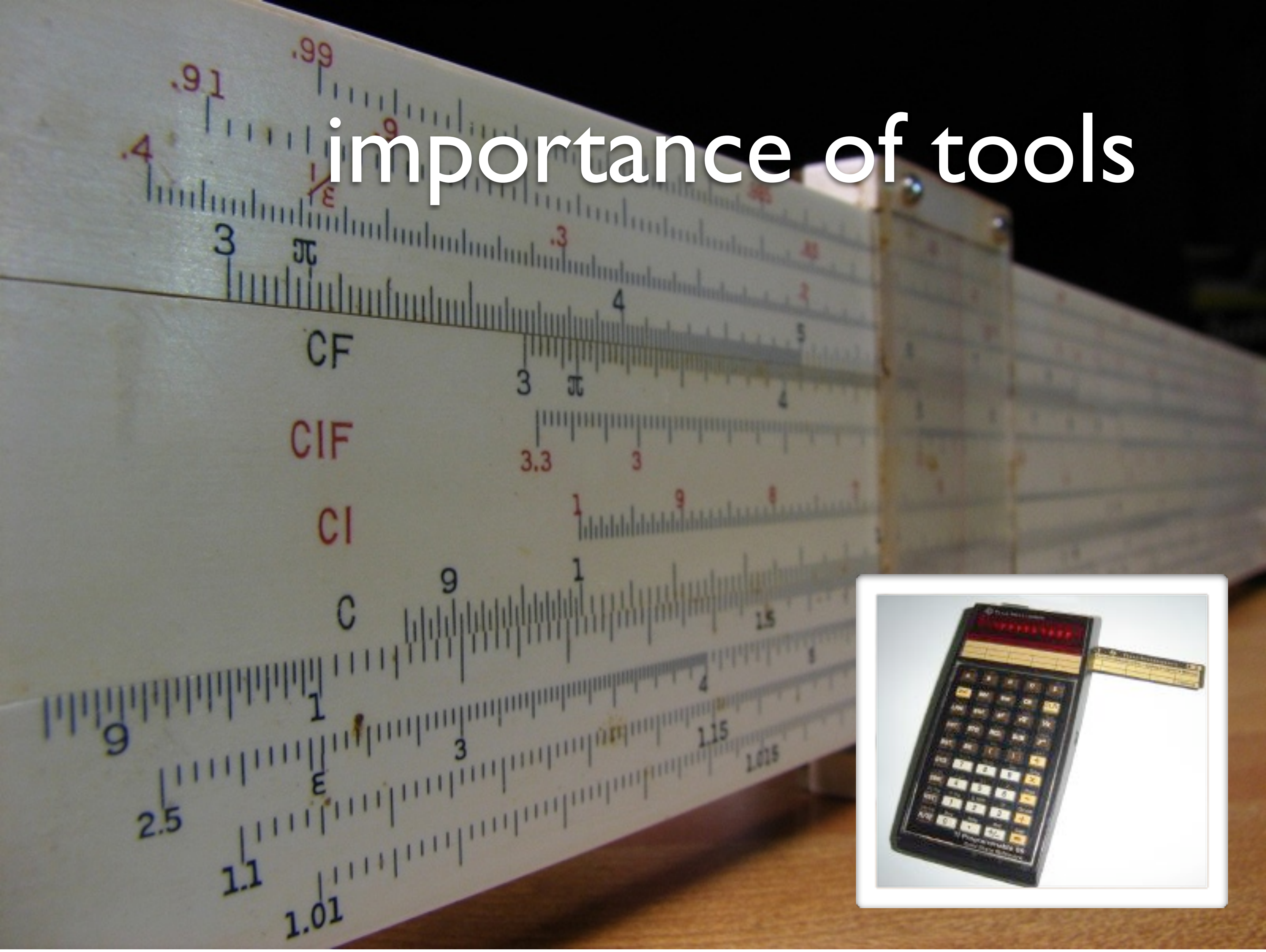
- Importance of Tools
- Proficiency in Debugging
- Understanding the Leaky Abstraction
- Being Pixel Retentive
- Inspired by Problems
- Students of Beautiful Code

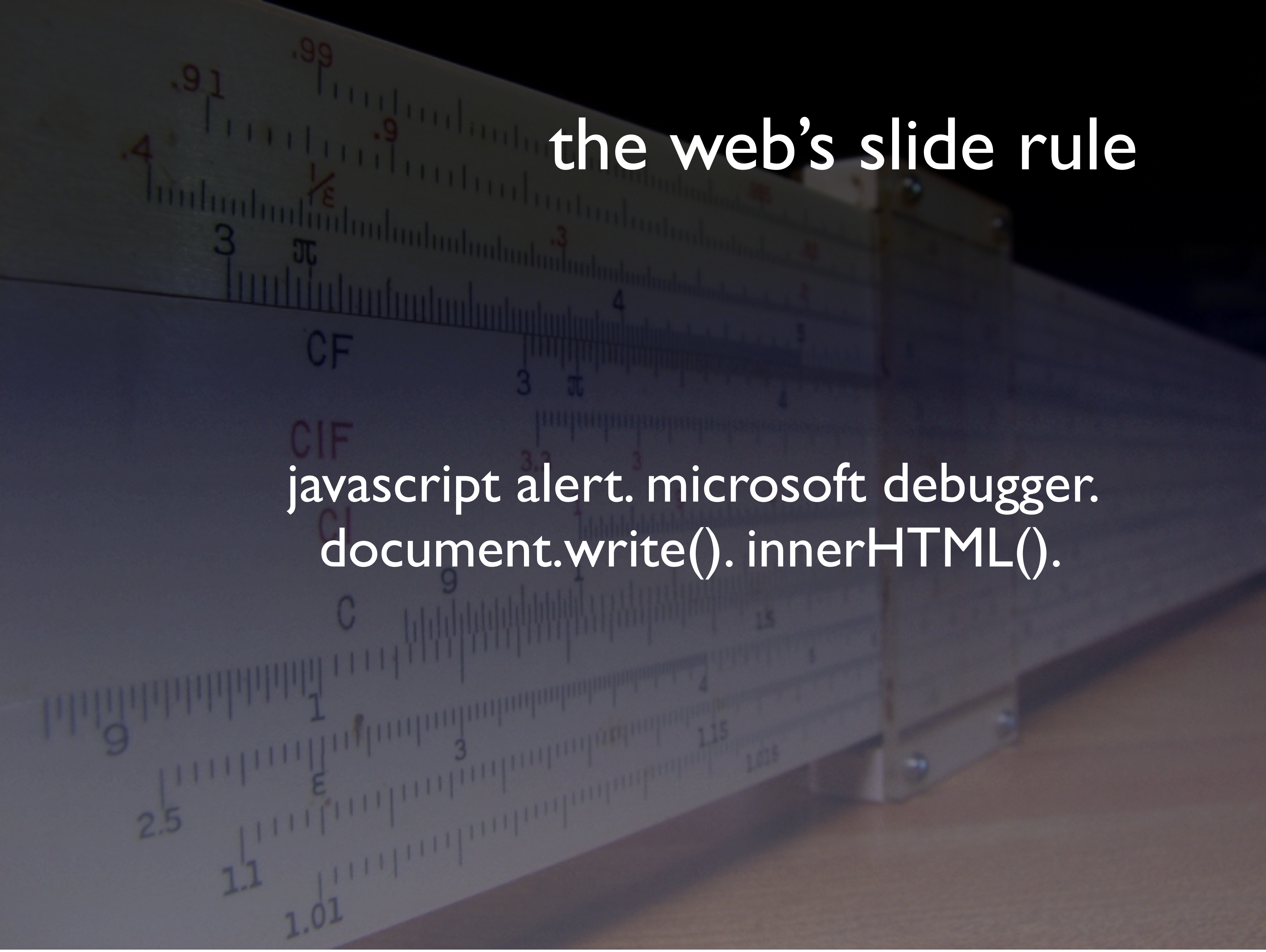
my history

importance of tools



importance of tools





the web's slide rule

javascript alert. microsoft debugger.
document.write(). innerHTML().

explosion of tools

firebug debugger. yslow. hammerhead. developer toolbar. drosera. webkit web inspector. webkit network timeline. fiddler. charles. httpwatch. firefox throttle. firebug profiler. jiffy. episodes. cuzillion. ua profiler. greasemonkey. dom inspector. html validator. live http headers. tamper data. venkman. firecookie. selenium. ie8 debugger. chrome debugger. foxyproxy. firephp. pixel perfect. dragonfly. debugbar. modify headers. xray. design bookmarklet. jsmin. jslint.

missing tools

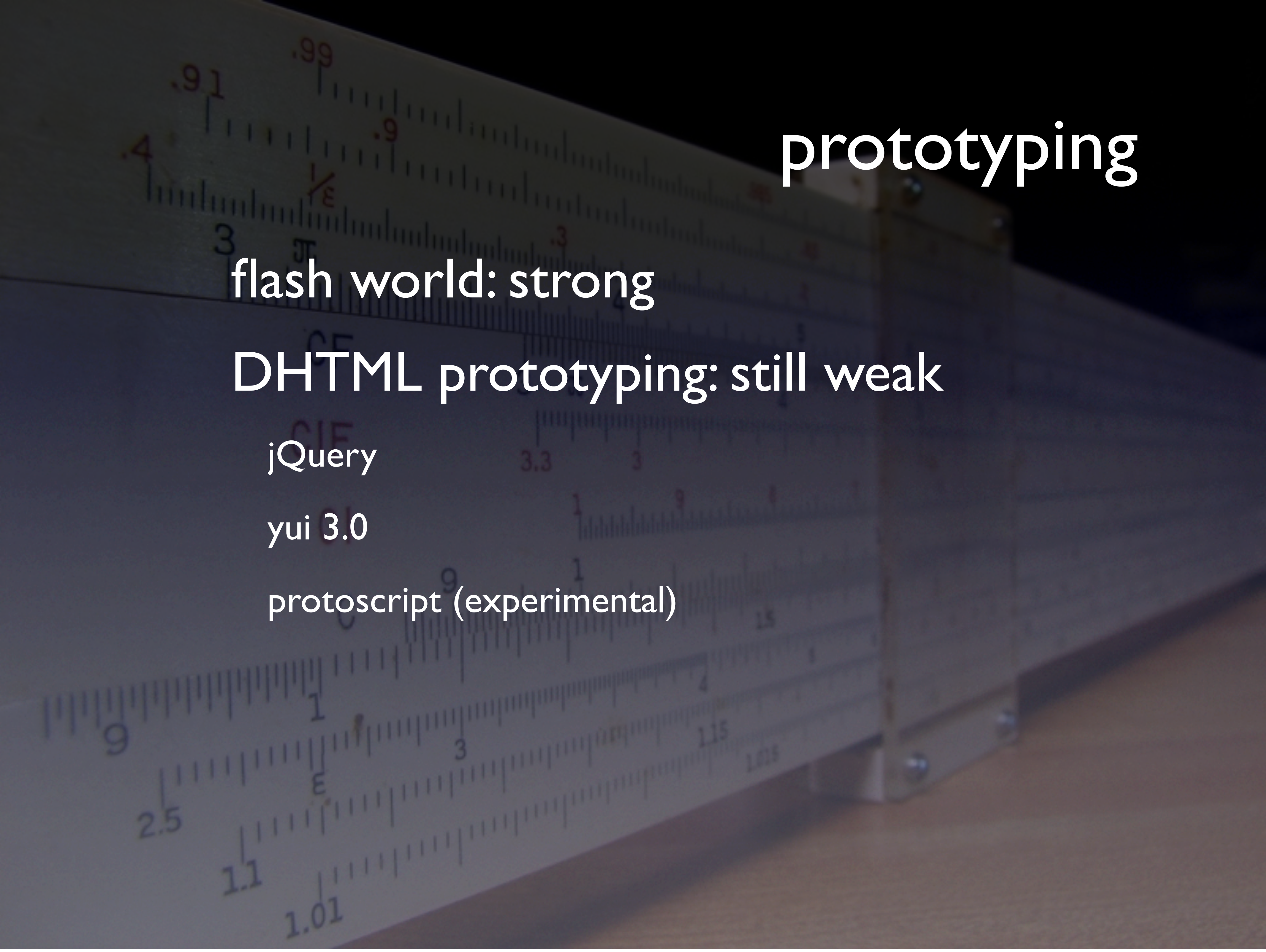
simple prototyping

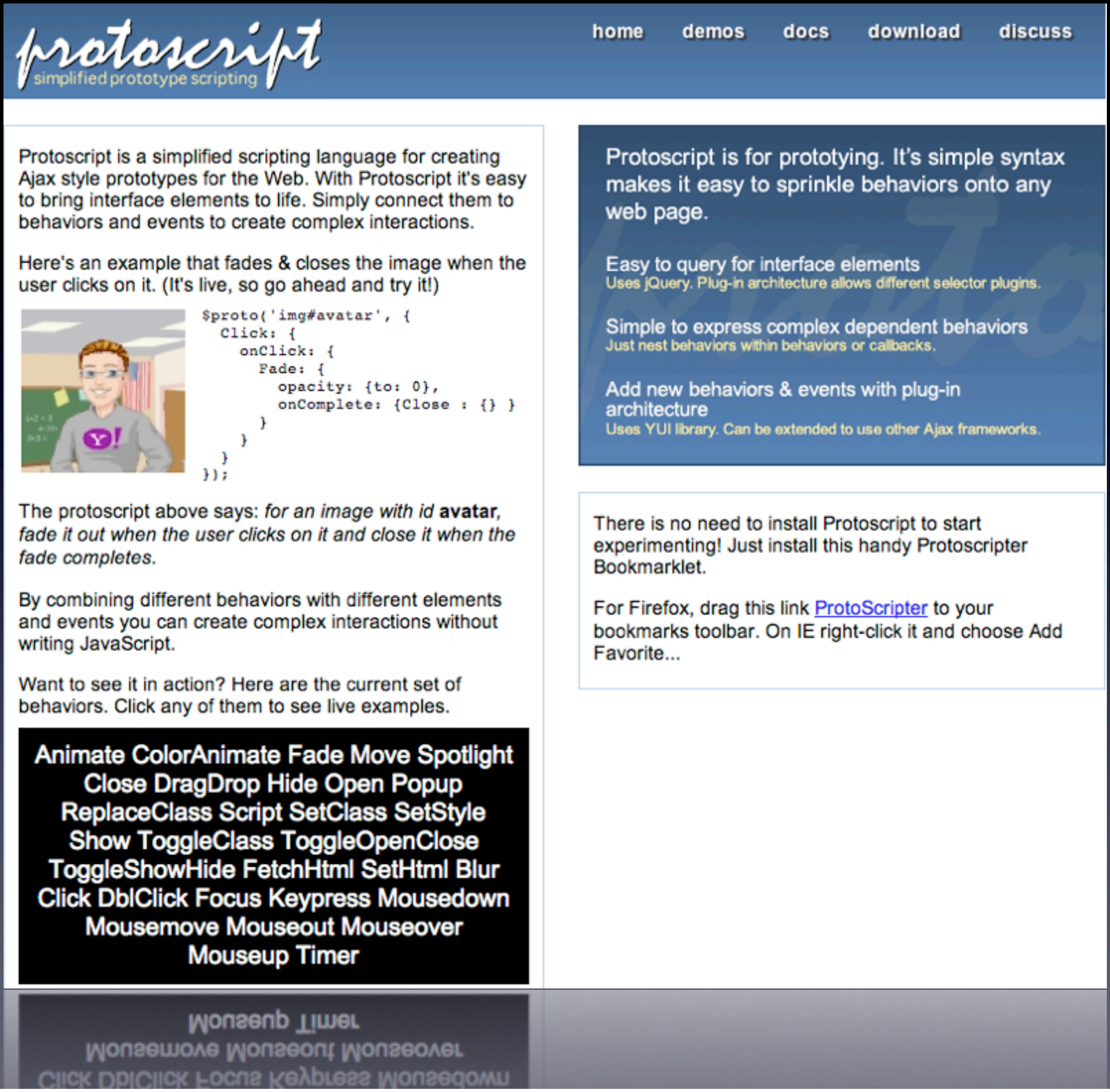
visibility into browser engine

css layout & refactoring tools

prototyping

- flash world: strong
- DHTML prototyping: still weak
  - jQuery
  - yui 3.0
  - protoscript (experimental)




protoscript
simplified prototype scripting

home demos docs download discuss

Protoscript is a simplified scripting language for creating Ajax style prototypes for the Web. With Protoscript it's easy to bring interface elements to life. Simply connect them to behaviors and events to create complex interactions.

Here's an example that fades & closes the image when the user clicks on it. (It's live, so go ahead and try it!)

```
$proto('img#avatar', {
  Click: {
    onClick: {
      Fade: {
        opacity: {to: 0},
        onComplete: {Close : {} }
      }
    }
  }
});
```

The protoscript above says: *for an image with id* **avatar**, *fade it out when the user clicks on it and close it when the fade completes.*

By combining different behaviors with different elements and events you can create complex interactions without writing JavaScript.

Want to see it in action? Here are the current set of behaviors. Click any of them to see live examples.

Animate ColorAnimate Fade Move Spotlight Close DragDrop Hide Open Popup ReplaceClass Script SetClass SetStyle Show ToggleClass ToggleOpenClose ToggleShowHide FetchHtml SetHtml Blur Click DblClick Focus Keypress Mousedown Mousemove Mouseout Mouseover Mouseup Timer

Protoscript is for prototyping. It's simple syntax makes it easy to sprinkle behaviors onto any web page.

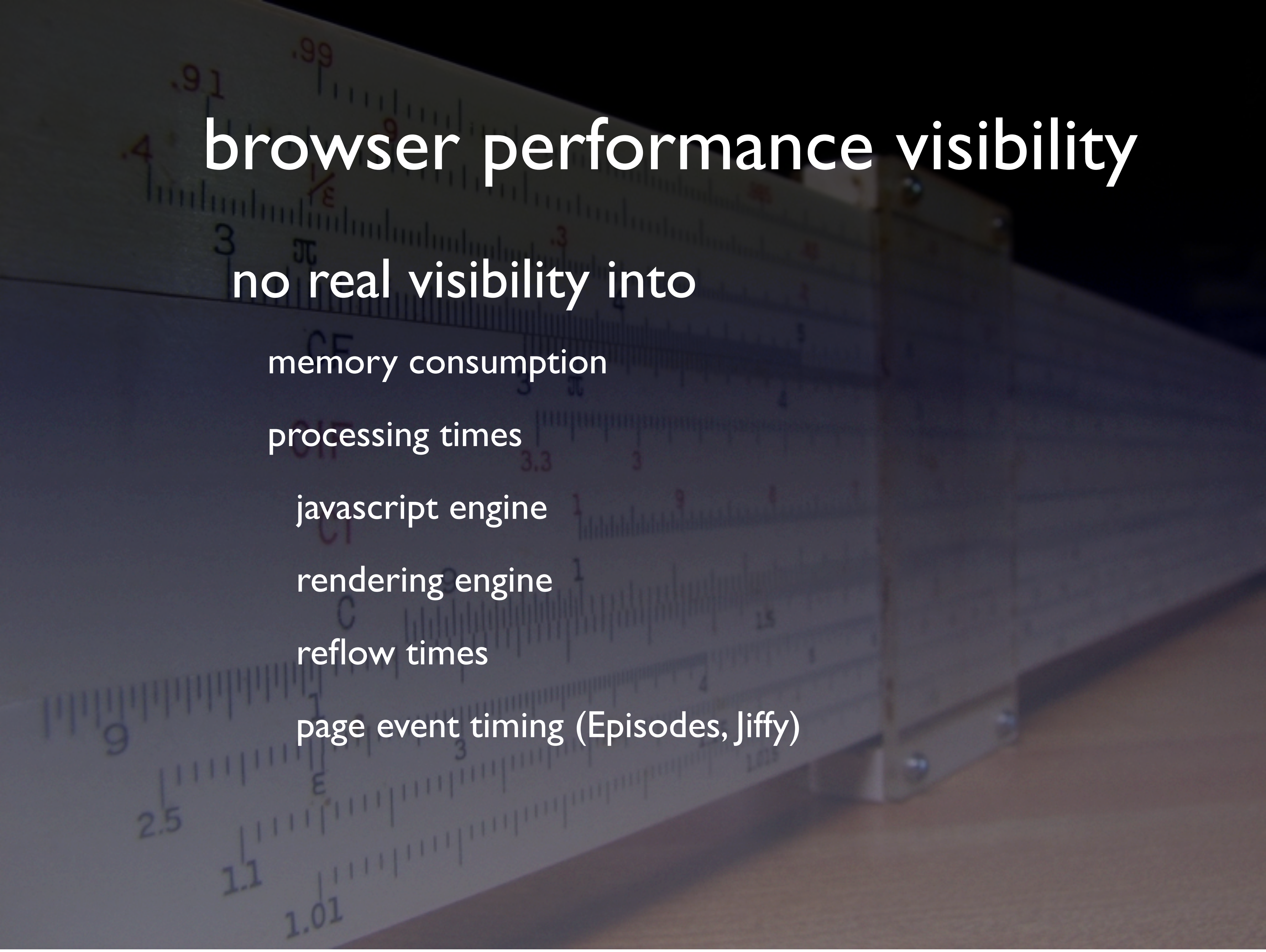
Easy to query for interface elements
Uses jQuery. Plug-in architecture allows different selector plugins.

Simple to express complex dependent behaviors
Just nest behaviors within behaviors or callbacks.

Add new behaviors & events with plug-in architecture
Uses YUI library. Can be extended to use other Ajax frameworks.

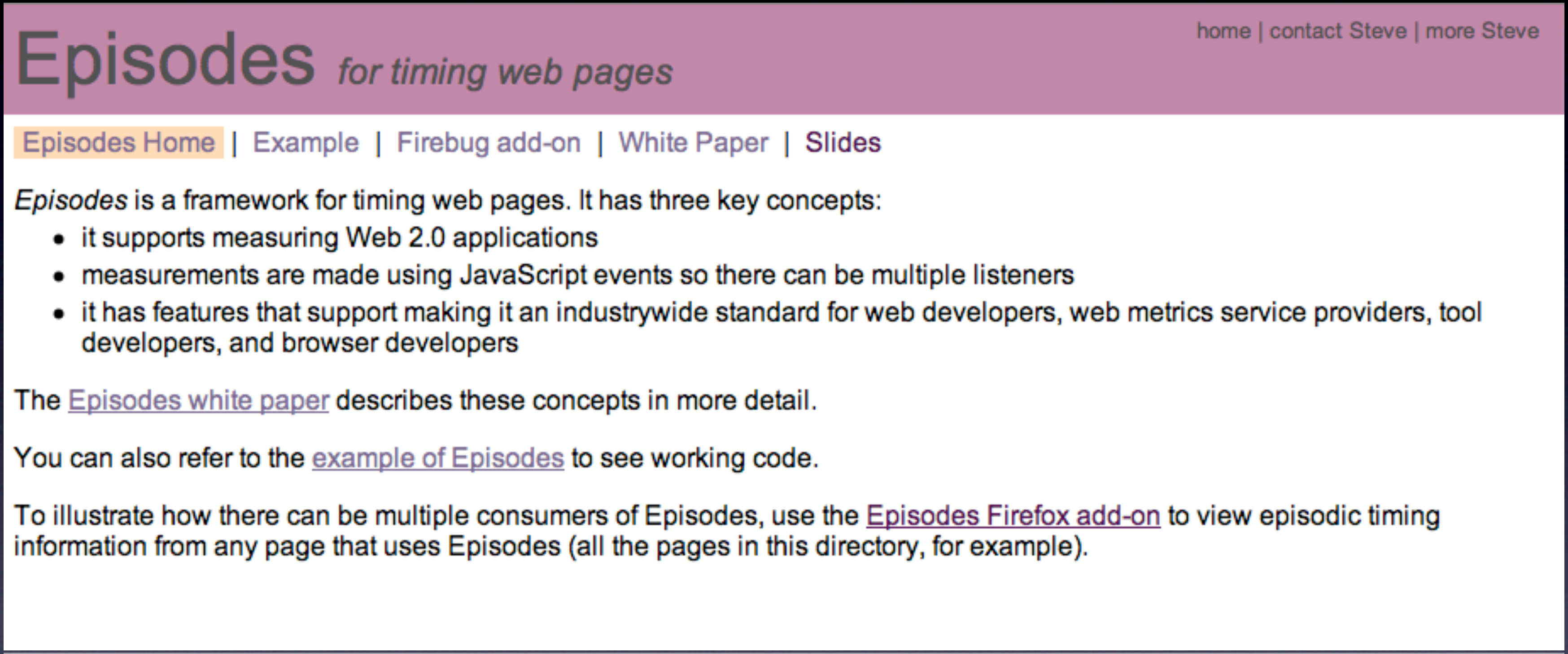
There is no need to install Protoscript to start experimenting! Just install this handy Protoscripter Bookmarklet.

For Firefox, drag this link ProtoScripter to your bookmarks toolbar. On IE right-click it and choose Add Favorite...



browser performance visibility

- no real visibility into
  - memory consumption
  - processing times
    - javascript engine
    - rendering engine
    - reflow times
    - page event timing (Episodes, Jiffy)



home | contact Steve | more Steve

Episodes *for timing web pages*

Episodes Home | Example | Firebug add-on | White Paper | Slides

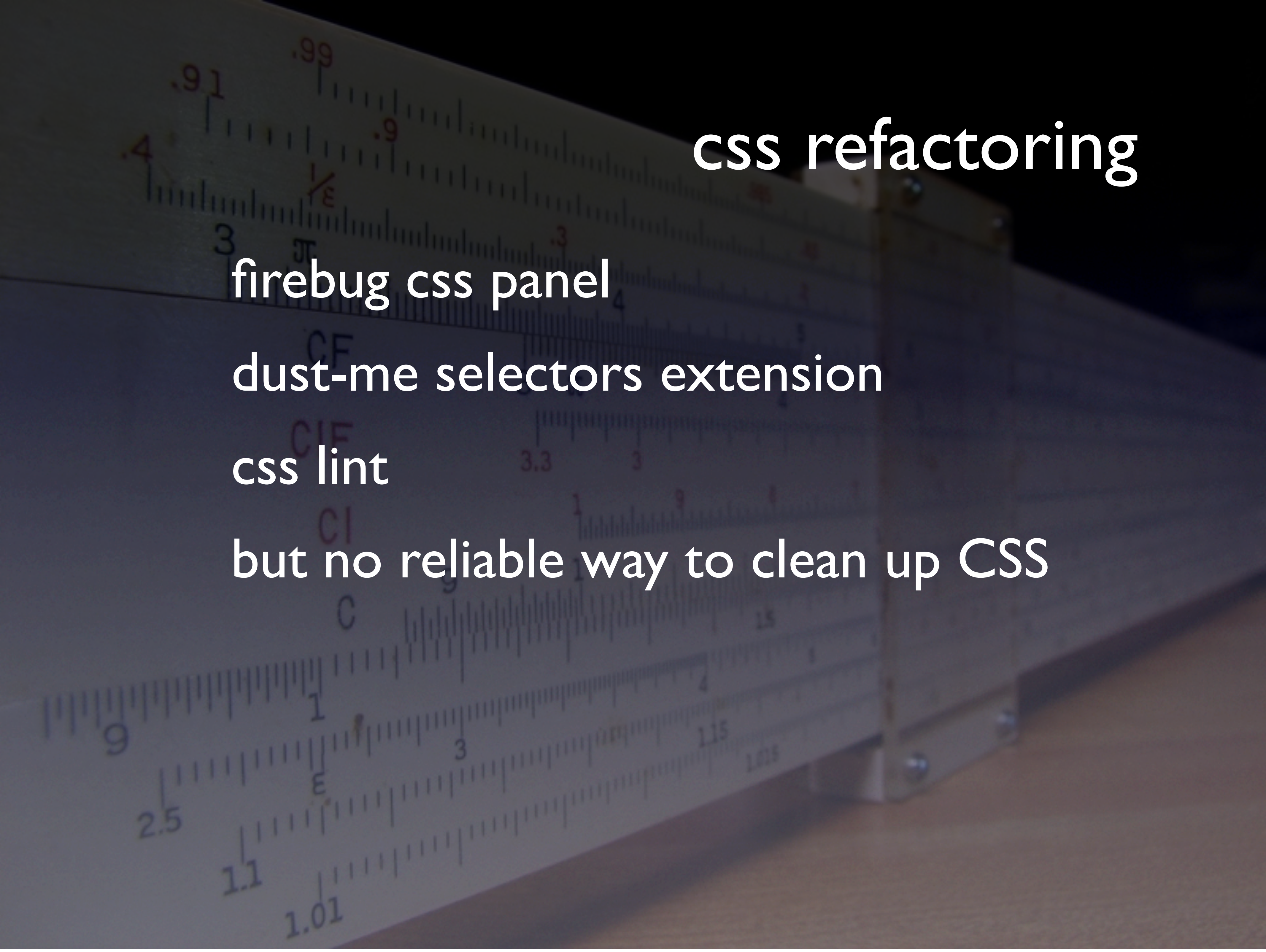
Episodes is a framework for timing web pages. It has three key concepts:

- it supports measuring Web 2.0 applications
- measurements are made using JavaScript events so there can be multiple listeners
- it has features that support making it an industrywide standard for web developers, web metrics service providers, tool developers, and browser developers

The Episodes white paper describes these concepts in more detail.

You can also refer to the example of Episodes to see working code.

To illustrate how there can be multiple consumers of Episodes, use the Episodes Firefox add-on to view episodic timing information from any page that uses Episodes (all the pages in this directory, for example).



css refactoring

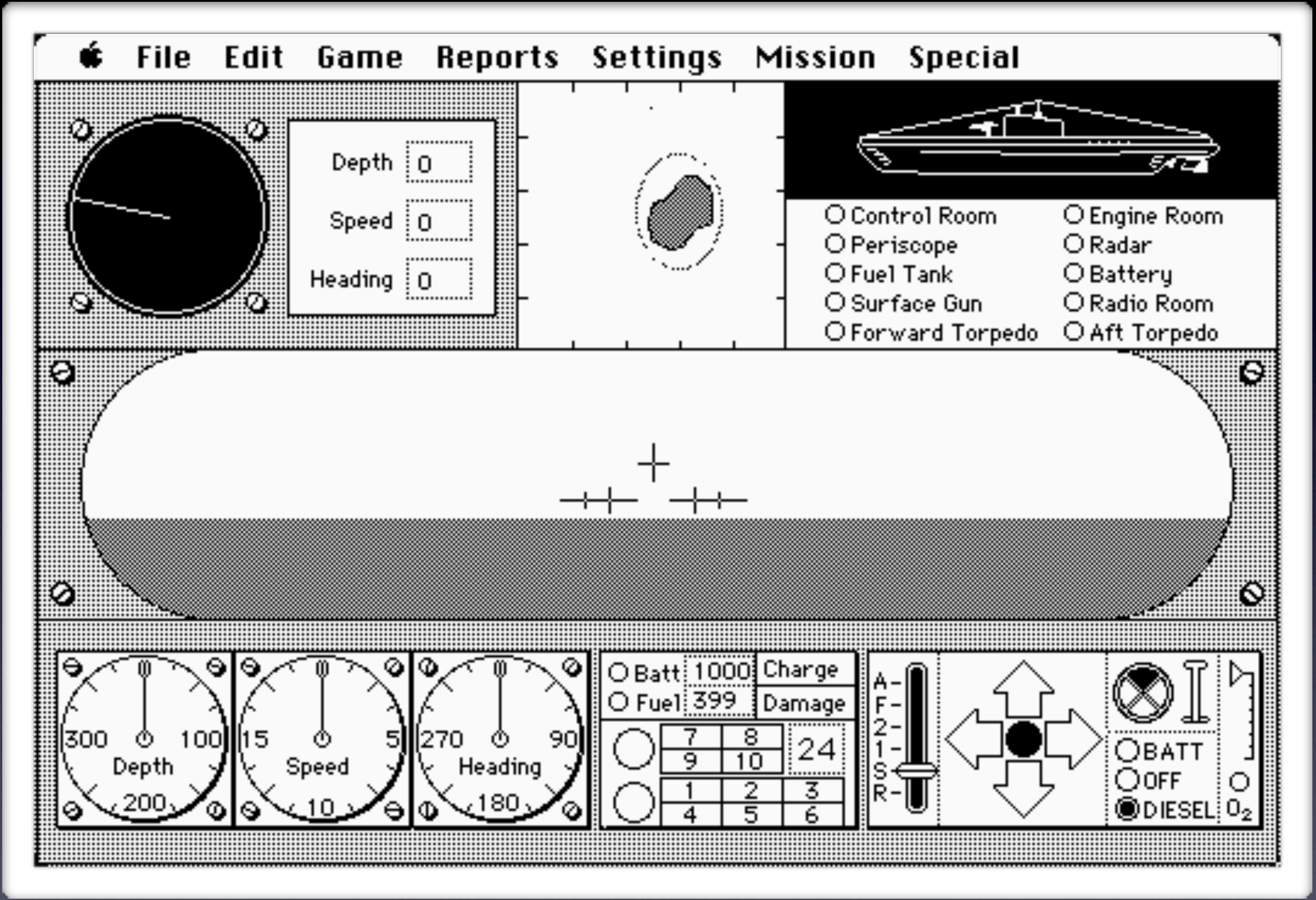
firebug css panel

dust-me selectors extension

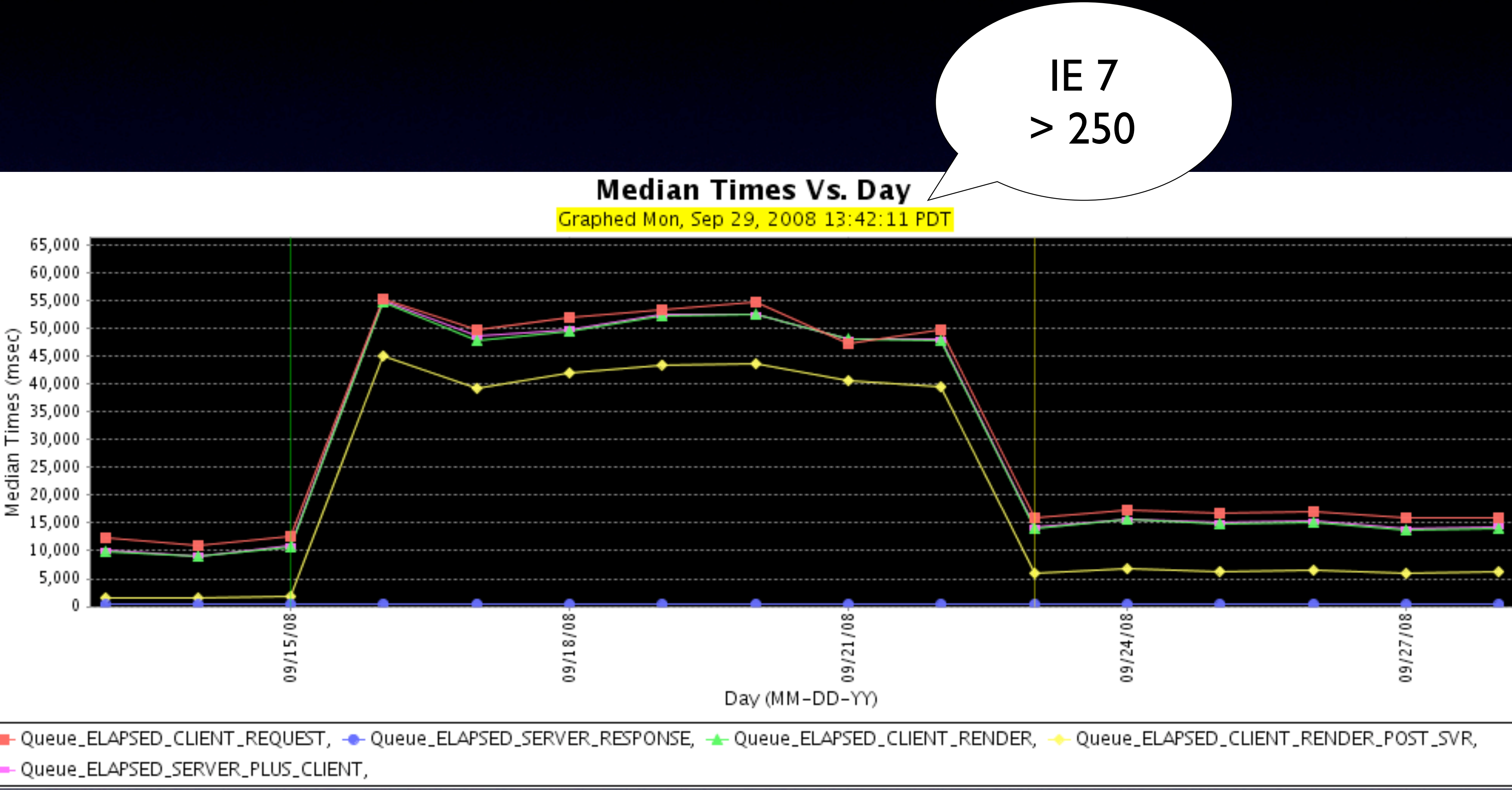
css lint

but no reliable way to clean up CSS

proficiency in debugging

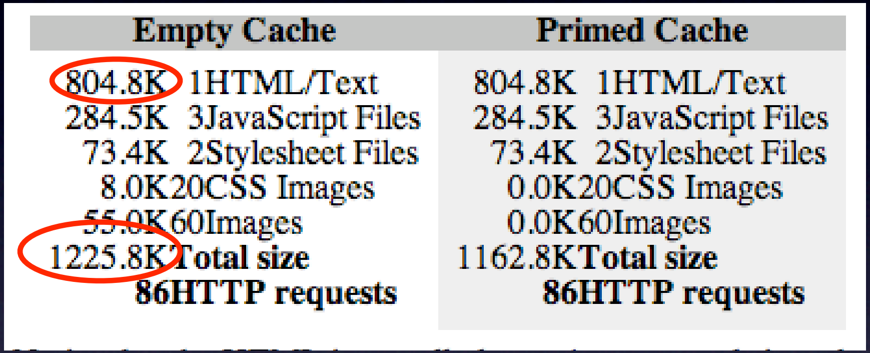

File
Edit
Game
Reports
Settings
Mission
Special
Depth 0
Speed 0
Heading 0
Control Room
Periscope
Fuel Tank
Surface Gun
Forward Torpedo
Engine Room
Radar
Battery
Radio Room
Aft Torpedo
Depth
Speed
Heading
Batt 1000
Fuel 399
Charge
Damage
24
BATT
OFF
DIESEL
O2

example: netflix Q performance

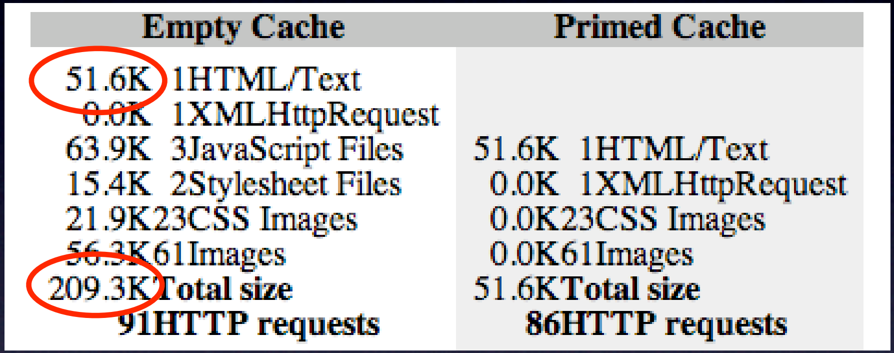


example: gzip breaks safari

page weight dropped by 6x

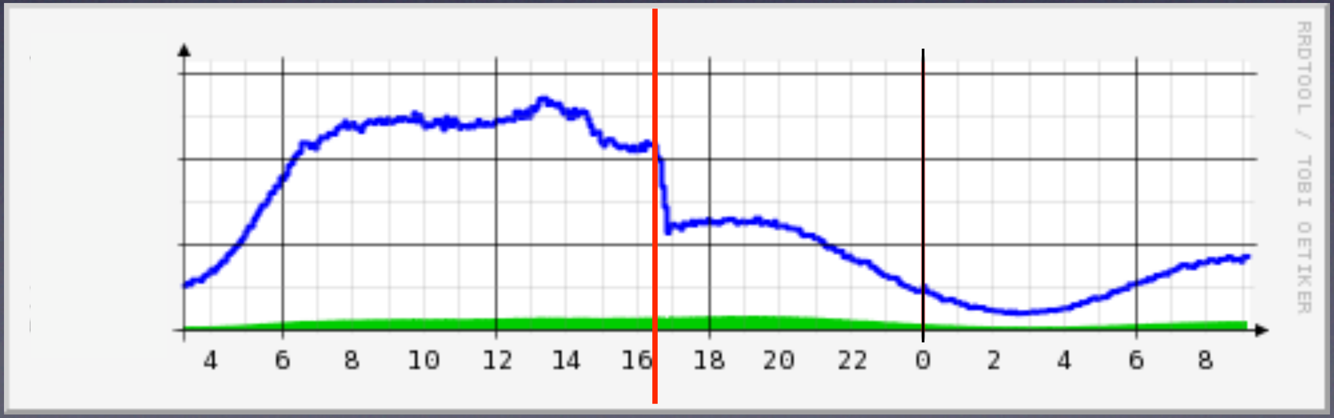


Empty Cache		Primed Cache	
804.8K	1HTML/Text	804.8K	1HTML/Text
284.5K	3JavaScript Files	284.5K	3JavaScript Files
73.4K	2Stylesheet Files	73.4K	2Stylesheet Files
8.0K	20CSS Images	0.0K	20CSS Images
55.0K	60Images	0.0K	60Images
1225.8K	**Total size**	1162.8K	**Total size**
	86HTTP requests		**86HTTP requests**



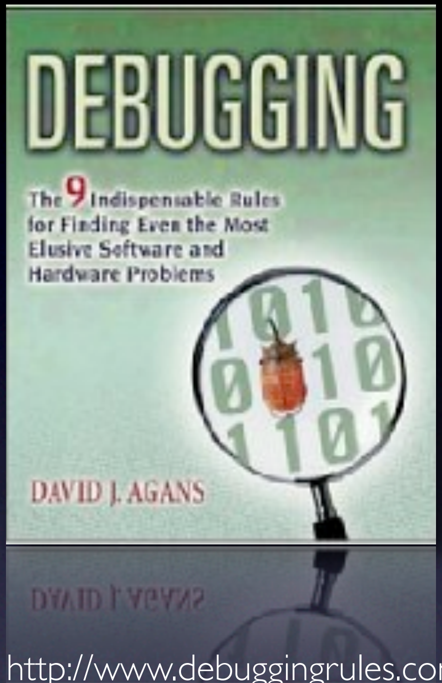
Empty Cache		Primed Cache	
51.6K	1HTML/Text		
0.0K	1XMLHttpRequest		
63.9K	3JavaScript Files	51.6K	1HTML/Text
15.4K	2Stylesheet Files	0.0K	1XMLHttpRequest
21.9K	23CSS Images	0.0K	23CSS Images
56.3K	61Images	0.0K	61Images
209.3K	**Total size**	51.6K	**Total size**
	91HTTP requests		**86HTTP requests**

outbound network traffic dropped in half



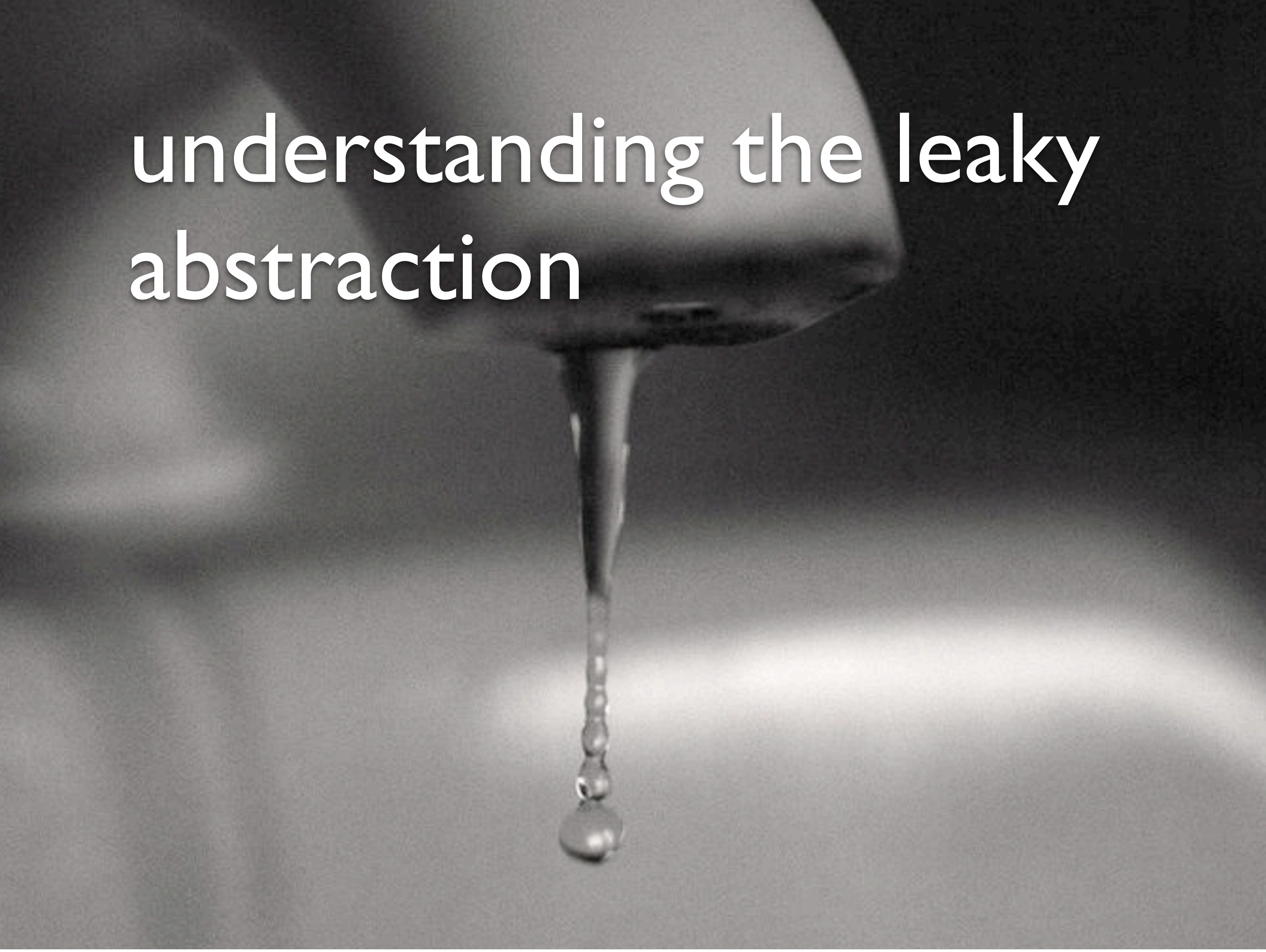
rules of debugging

1. understand the system
2. make it fail
3. quit thinking and look
4. divide and conquer
5. change one thing at a time
6. keep an audit trail
7. check the plug
8. get a fresh view
9. if you didn't fix It, it ain't fixed


DEBUGGING
The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems
DAVID J. AGANS

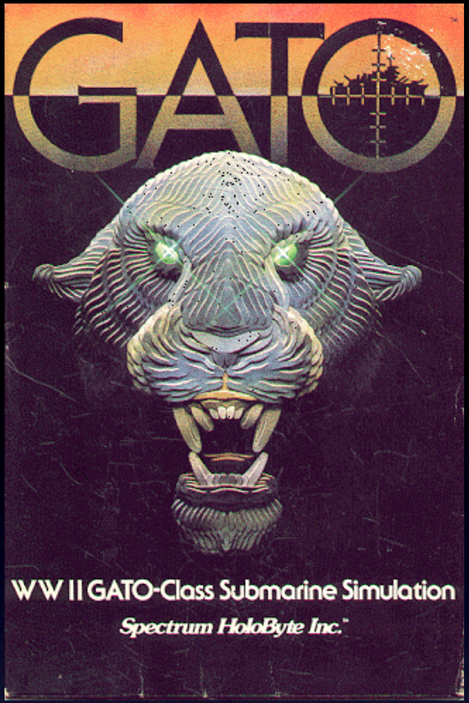
http://www.debuggingrules.com/

http://www.whyprogramsfail.com/toc.php



understanding the leaky abstraction

understanding deep magic



- story of GATO development
  - barely understood the language
  - barely understood mac development
- afterward determined to understand the deep magic
  - disassembled ROM, commented all the code
  - wrote numerous utilities
  - asked what happens from click to render

building on abstractions

boosts productivity

but what happens when you have to look underneath?

the “leaky abstraction” syndrome

it’s easy to just fiddle till it works but not stop to ask why

example: closures

- Most candidates we interview
  - cannot accurately explain a closure
  - cannot detect common errors with closures
- If you use closures you should understand closures
- When something goes wrong you have to understand it

```
var alertFuncs = [];
for(var i = 0; i < 3; i++) {
     var alertFunc = function(value) {
          return function() {
             alert(value);
          }
     }(i);
     alertFuncs.push(alertFunc);
}

for (var i = 0; i < alertFuncs.length; i++) {
     alertFuncs[i]();
}
```

example: toolkits

libraries have been a big boost to our community

however, it’s easy to end up knowing a library but not really what the library does for them

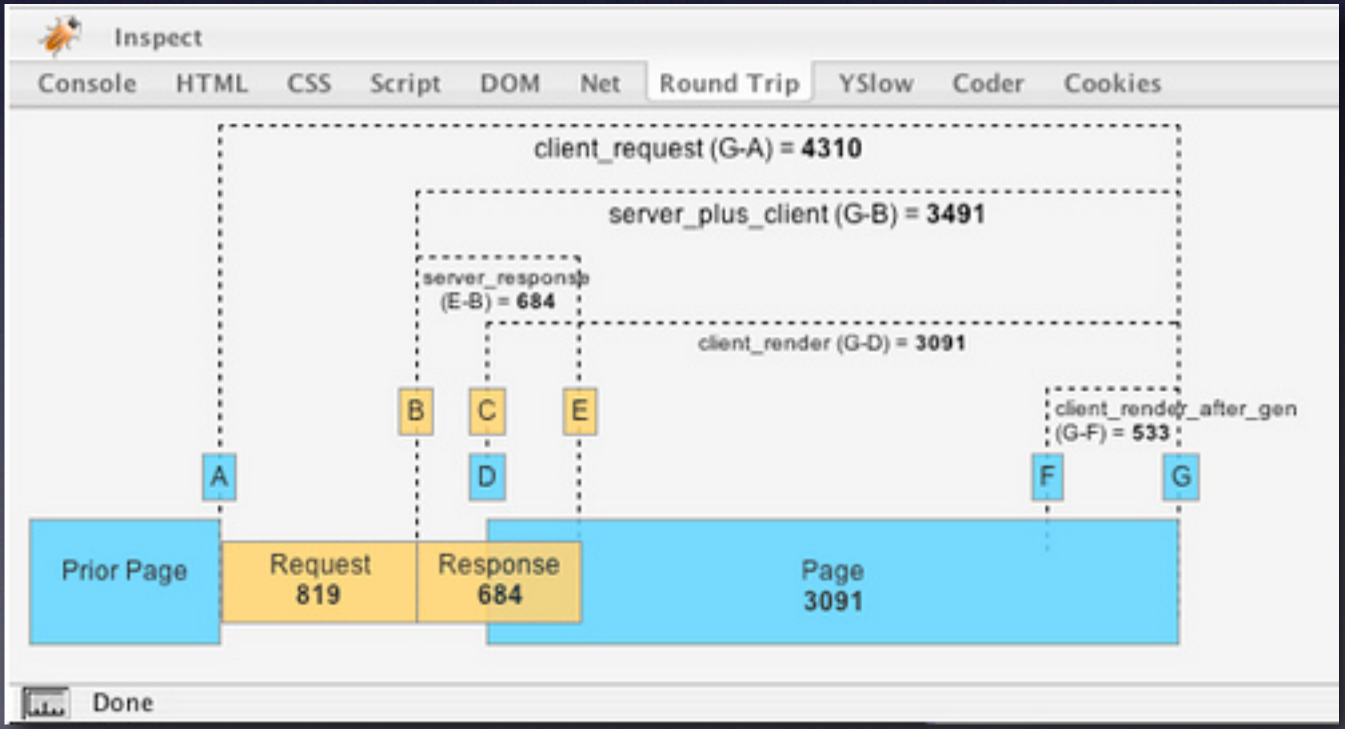
it’s ok to focus on the usage of a library, but a little dose of curiosity a day will teach you a lot

example: end to end

http request to http response is basic

but do we understand what happens and when?

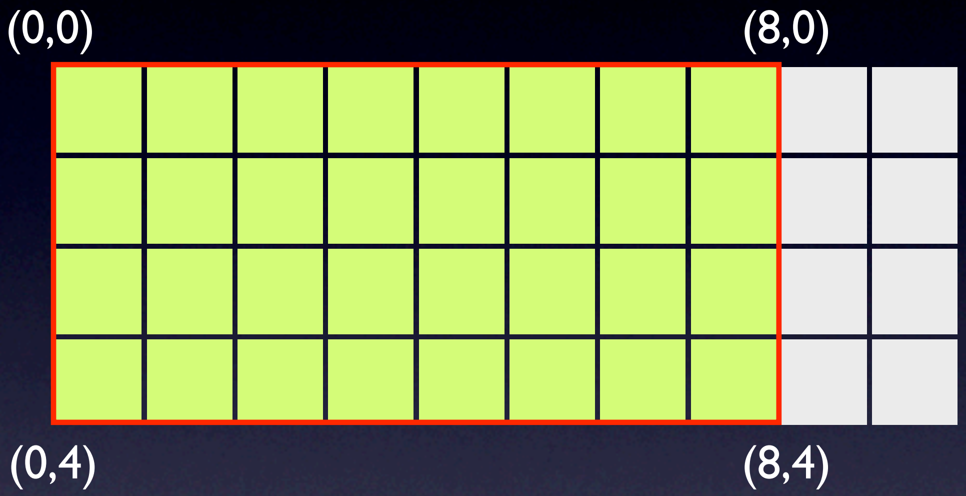
getting a picture of the overall picture helps us track down performance issues


Inspect
Console
HTML
CSS
Script
DOM
Net
Round Trip
YSlow
Coder
Cookies
client_request (G-A) = 4310
server_plus_client (G-B) = 3491
server_response (E-B) = 684
client_render (G-D) = 3091
client_render_after_gen (G-F) = 533
A
B
C
D
E
F
G
Prior Page
Request 819
Response 684
Page 3091
Done

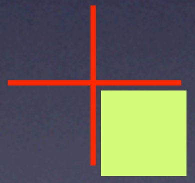
being pixel retentive

end-point paranoia (quickdraw)

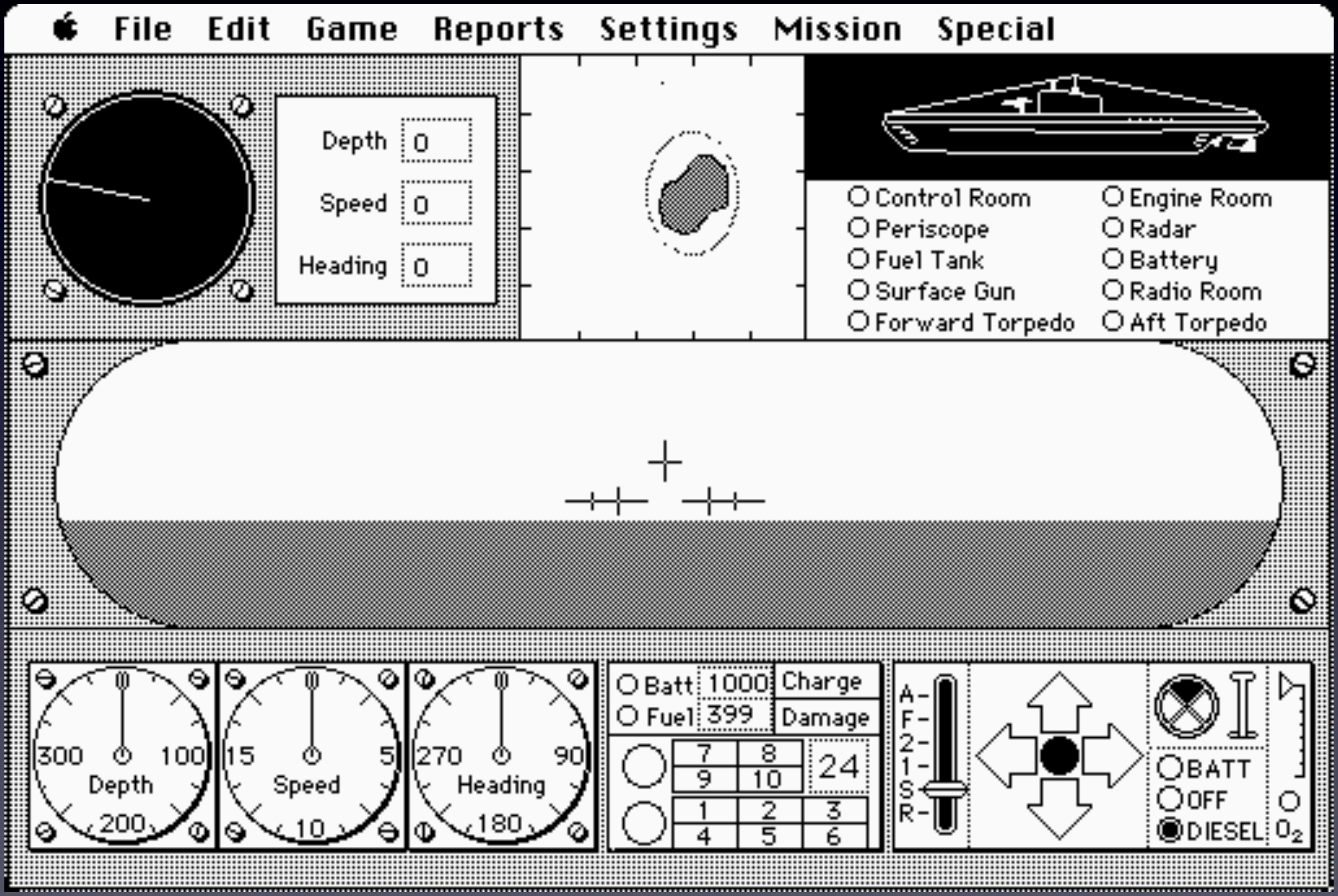
coordinate system infinitely thin lines between pixels



filling or framing a rectangle: 0,0,8,4



points anchor to thin grid, to right & below


File
Edit
Game
Reports
Settings
Mission
Special
Depth 0
Speed 0
Heading 0
Control Room
Engine Room
Periscope
Radar
Fuel Tank
Battery
Surface Gun
Radio Room
Forward Torpedo
Aft Torpedo
300 Depth 100 200
15 Speed 5 10
270 Heading 90 180
Batt 1000 Charge
Fuel 399 Damage
7 8 9 10
24
1 2 3 4 5 6
A F 2 1 S R
BATT
OFF
DIESEL
O2

css reset

css frameworks

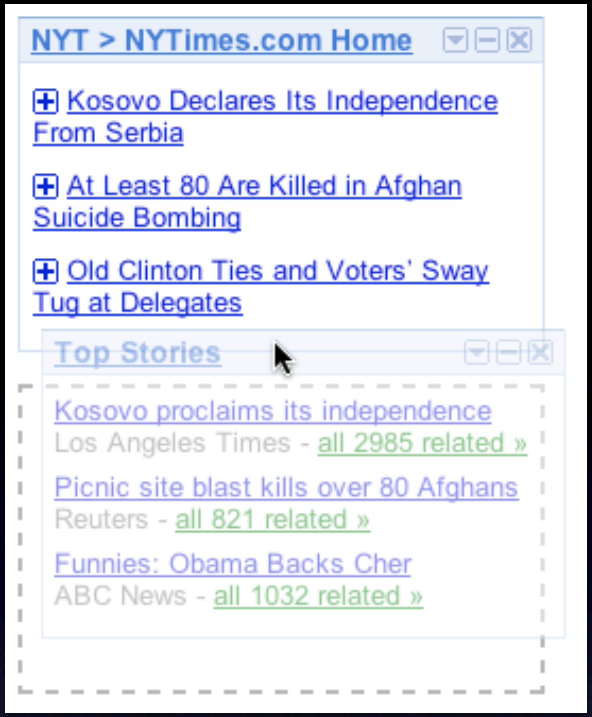
YUI
setX(), setY()

interactive intelligence

drag and drop

interesting moments grid

subtlety of drag & drop


NYT > NYTimes.com Home

Kosovo Declares Its Independence From Serbia

At Least 80 Are Killed in Afghan Suicide Bombing

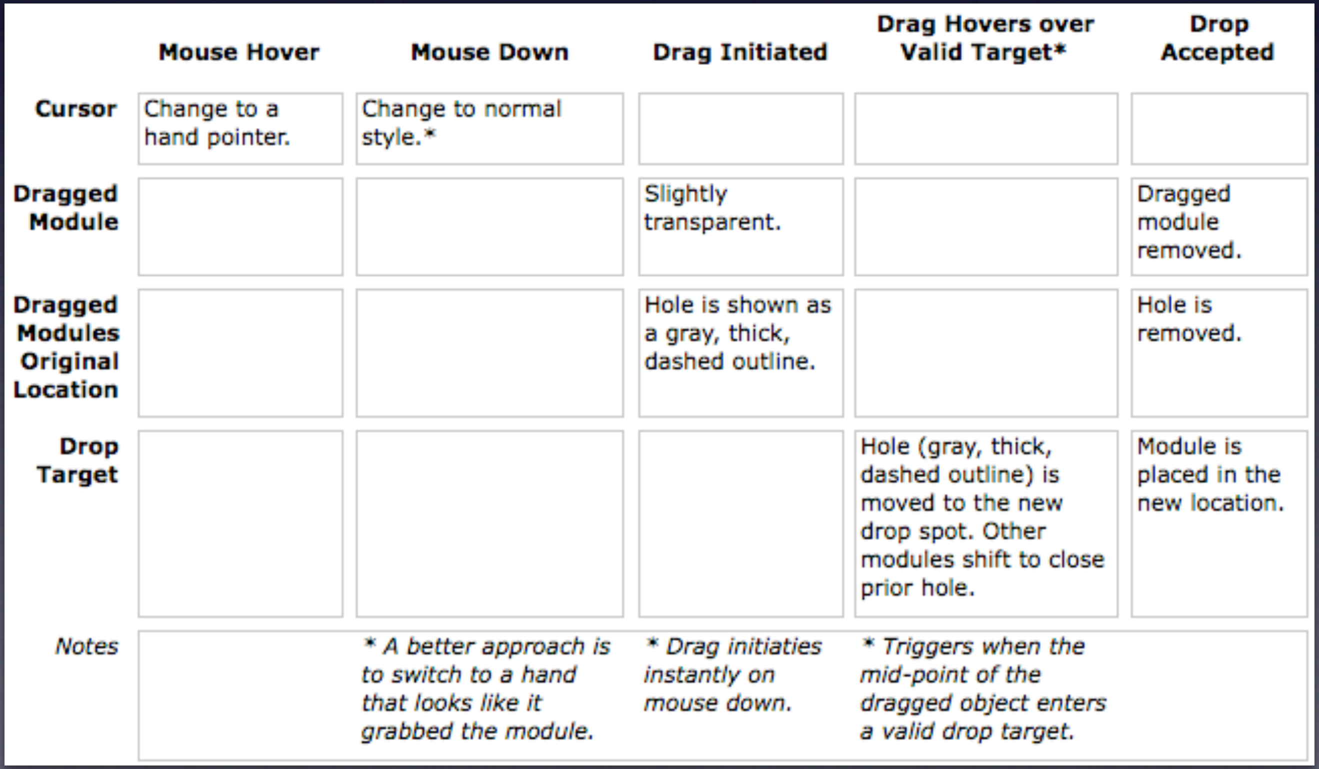
Old Clinton Ties and Voters' Sway Tug at Delegates

Top Stories

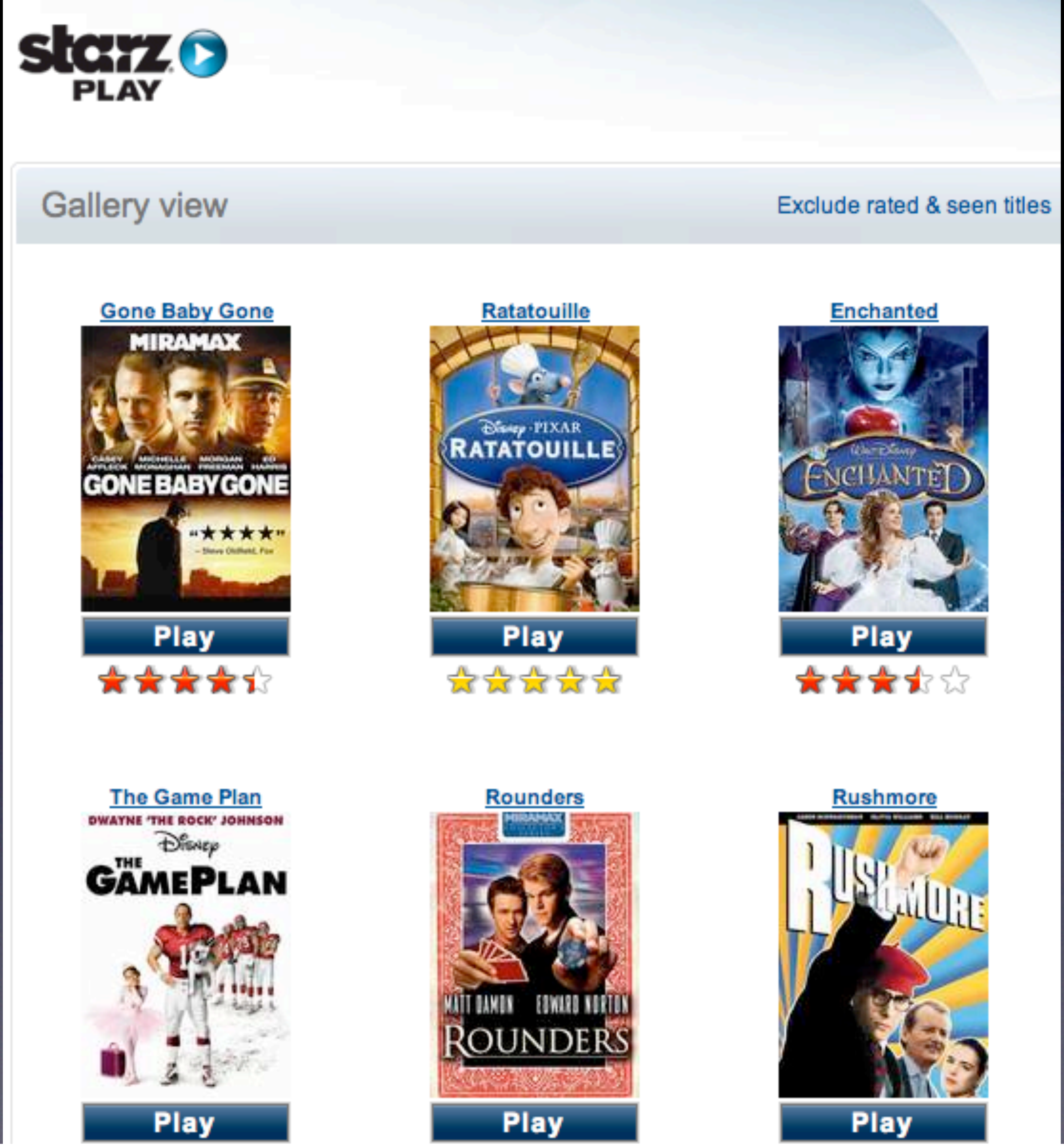
Kosovo proclaims its independence
Los Angeles Times - all 2985 related »

Picnic site blast kills over 80 Afghans
Reuters - all 821 related »

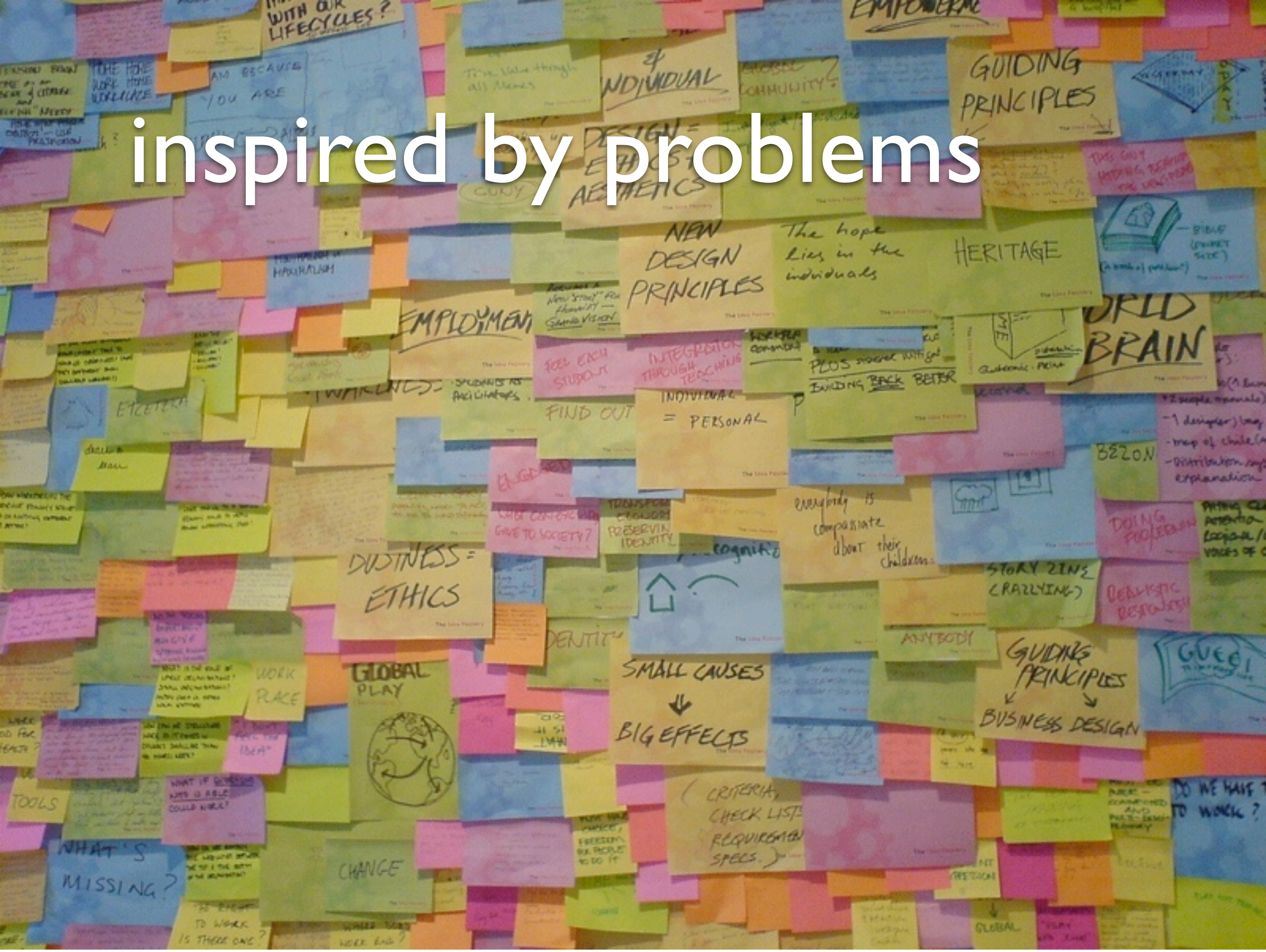
Funnies: Obama Backs Cher
ABC News - all 1032 related »



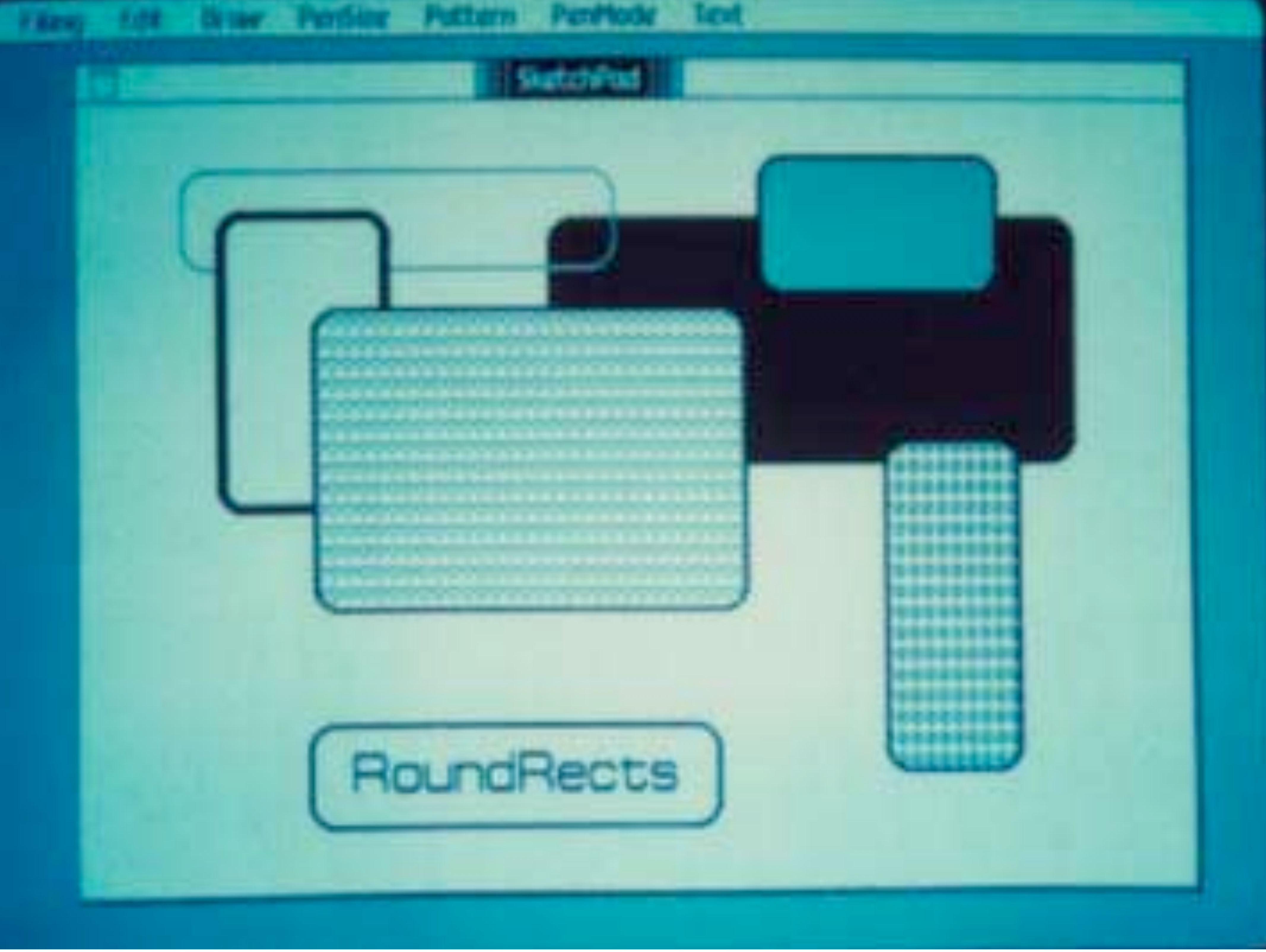
	Mouse Hover	Mouse Down	Drag Initiated	Drag Hovers over Valid Target*	Drop Accepted
Cursor	Change to a hand pointer.	Change to normal style.*			
Dragged Module			Slightly transparent.		Dragged module removed.
Dragged Modules Original Location			Hole is shown as a gray, thick, dashed outline.		Hole is removed.
Drop Target				Hole (gray, thick, dashed outline) is moved to the new drop spot. Other modules shift to close prior hole.	Module is placed in the new location.
Notes		** A better approach is to switch to a hand that looks like it grabbed the module.*	** Drag initiaties instantly on mouse down.*	** Triggers when the mid-point of the dragged object enters a valid drop target.*	


starz PLAY
Gallery view
Exclude rated & seen titles
Gone Baby Gone
Play
Ratatouille
Play
Enchanted
Play
The Game Plan
Play
Rounders
Play
Rushmore
Play

inspired by problems



GUIDING PRINCIPLES
NEW DESIGN PRINCIPLES
The hope lies in the individuals
HERITAGE
EMPLOYMENT
BUSINESS ETHICS
SMALL CAUSES BIG EFFECTS
GLOBAL PLAY
GUIDING PRINCIPLES BUSINESS DESIGN
everybody is compassionate about their children
CRITERIA, CHECK LISTS REQUIREMENTS SPECS.
WHAT'S MISSING?
CHANGE
TOOLS

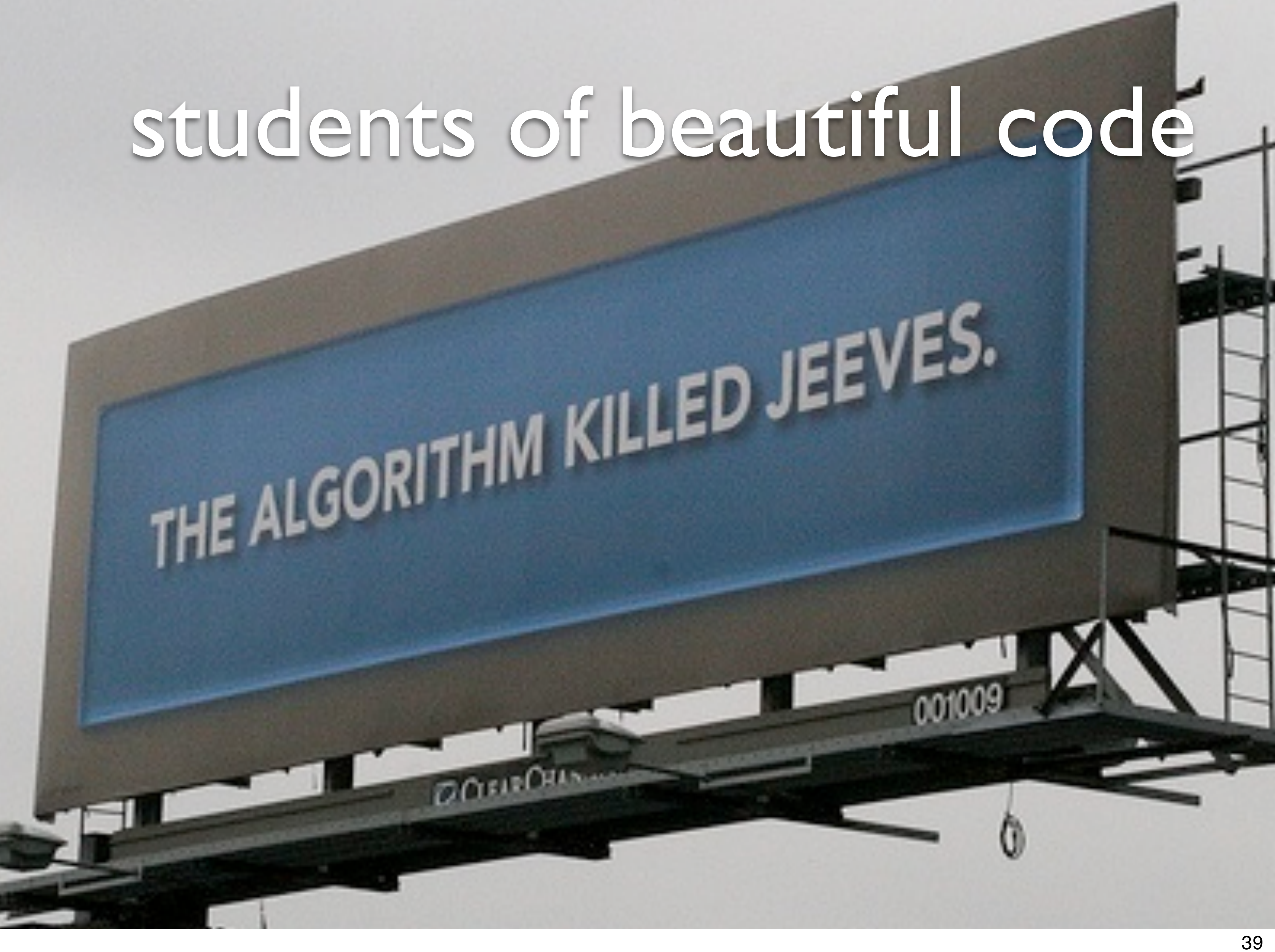

File
Edit
Draw
PenSize
Pattern
PenMode
Text
SketchPad
RoundRects

chasing ideas

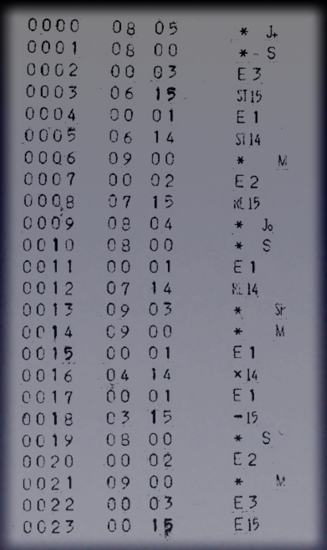
belief in creative process

saying yes

students of beautiful code



wang calculator



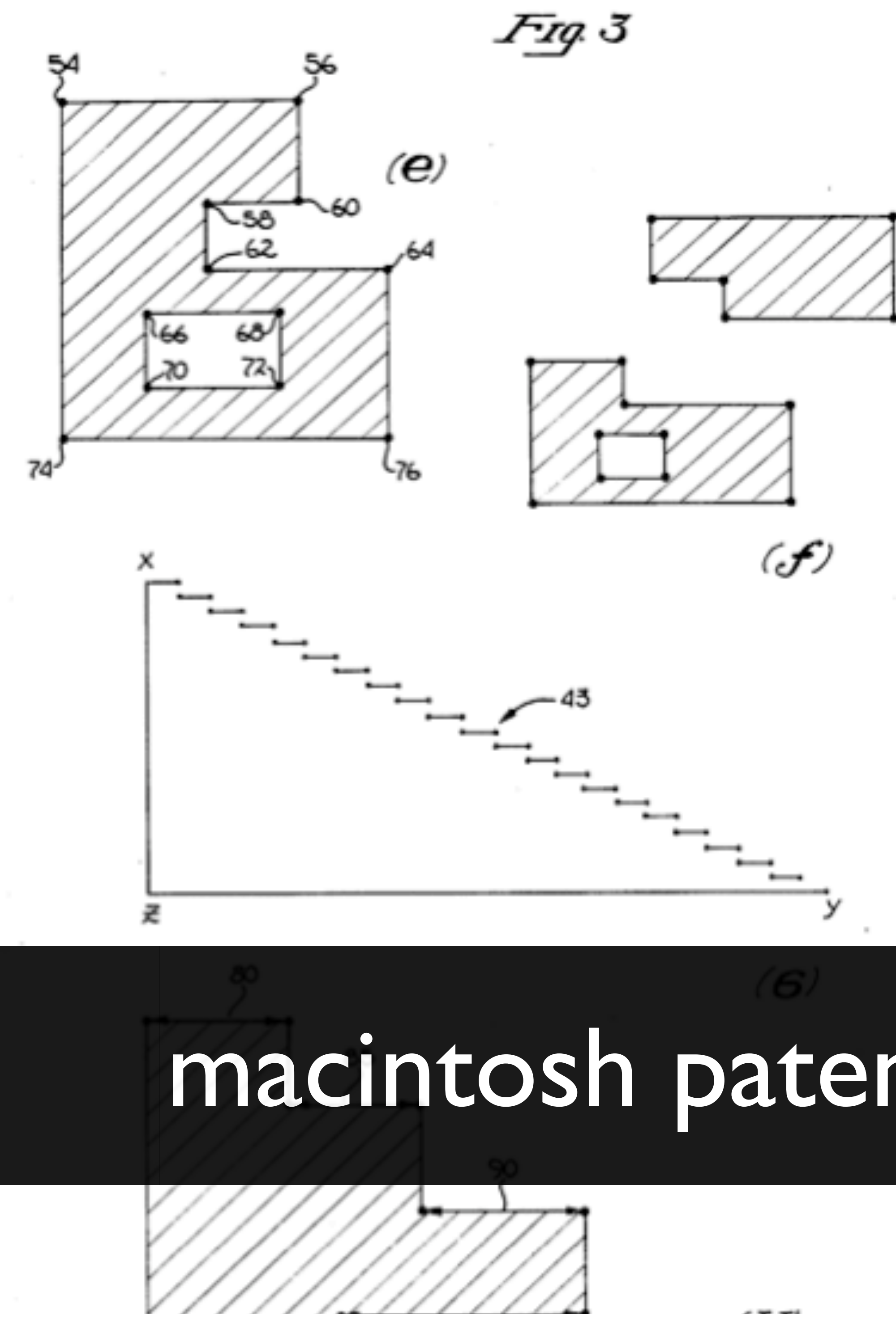
```
0000   08 05   *  J+
0001   08 00   *  S
0002   00 03   E 3
0003   06 15   ST15
0004   00 01   E 1
0005   06 14   ST14
0006   09 00   *     M
0007   00 02   E 2
0008   07 15   RL15
0009   08 04   *  Jo
0010   08 00   *  S
0011   00 01   E 1
0012   07 14   RL14
0013   09 03   *     Sr
0014   09 00   *     M
0015   00 01   E 1
0016   04 14   ×14
0017   00 01   E 1
0018   03 15   -15
0019   08 00   *  S
0020   00 02   E 2
0021   09 00   *     M
0022   00 03   E 3
0023   00 15   E15
```

Find out if the given number was prime

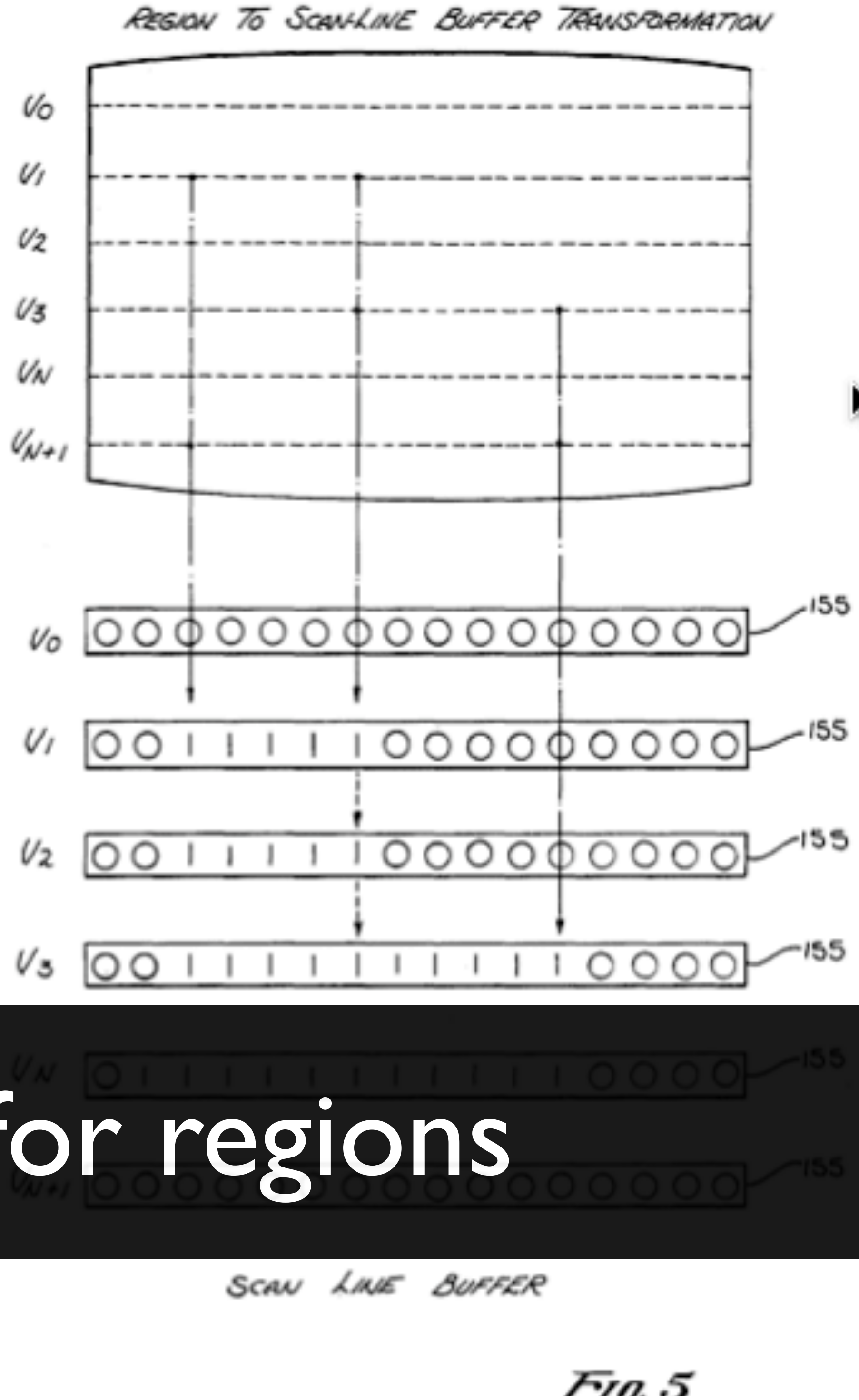
Instructions: J if 0, J if +, J if <>0, J if Err, Store, Recall, Mark, Search, Set PC, Indir

Sieve of Eratosthenes

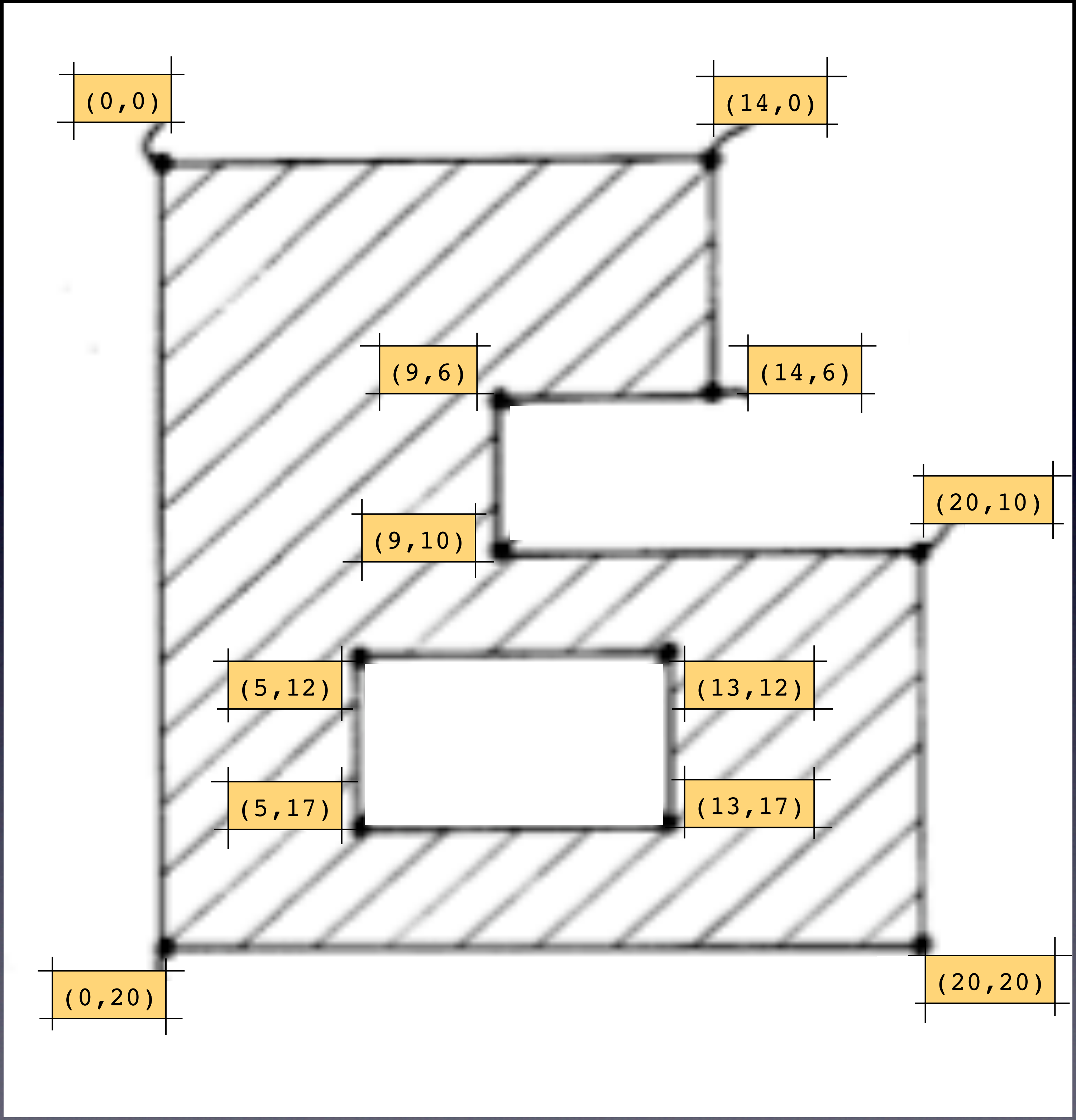
U.S. Patent Nov. 11, 1986 Sheet 3 of 7 4,622,545



U.S. Patent Nov. 11, 1986 Sheet 5 of 7 4,622,545



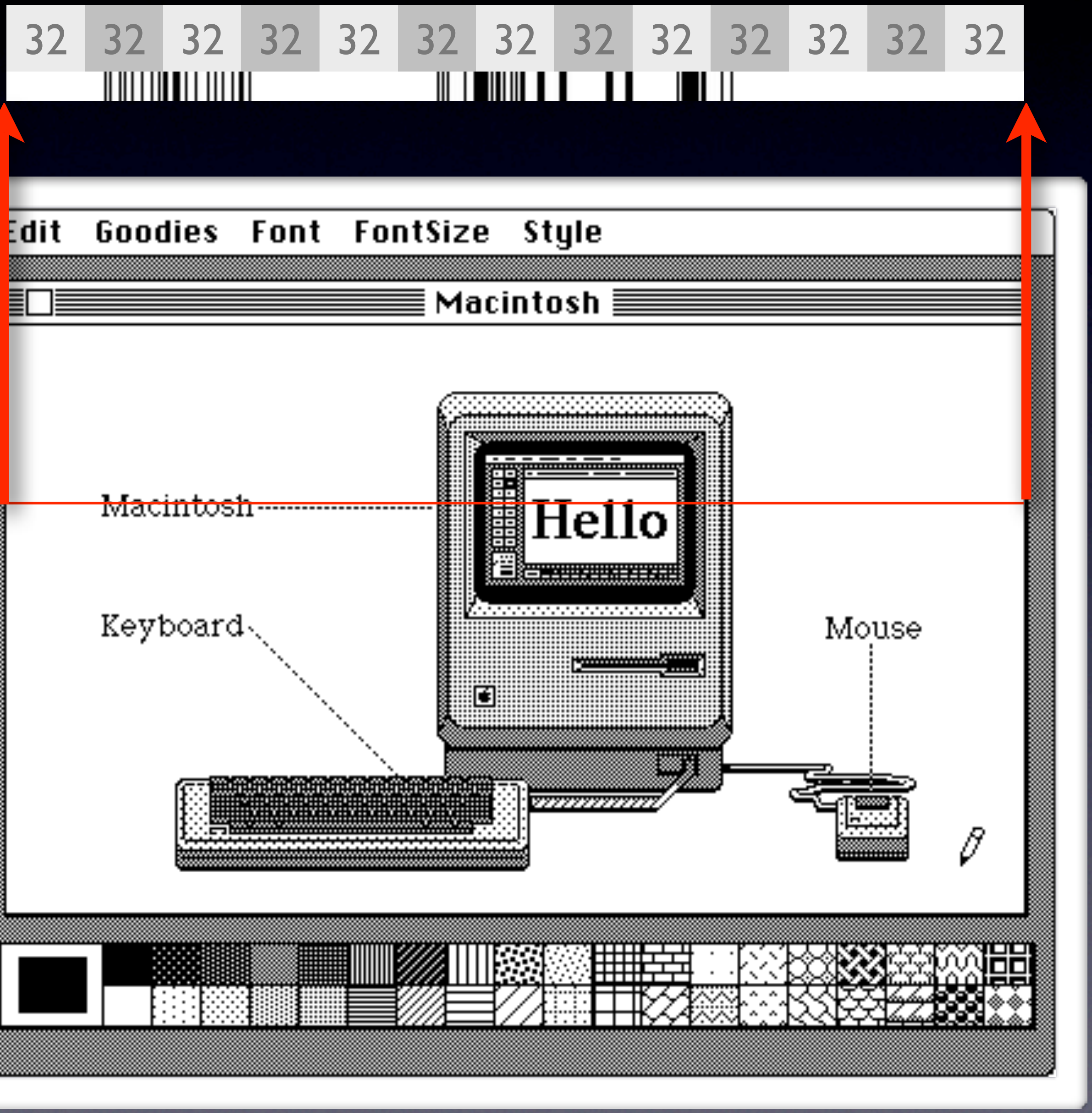
macintosh patent for regions



```
0, 0, 14, 32678,
6, 9, 14, 32768,
10, 9, 20, 32768,
12, 5, 13, 32768,
17, 5, 13, 32768,
20, 0, 20, 32768
```

macpaint

MC68000 MOVEM



html*

hacker?

event-driven

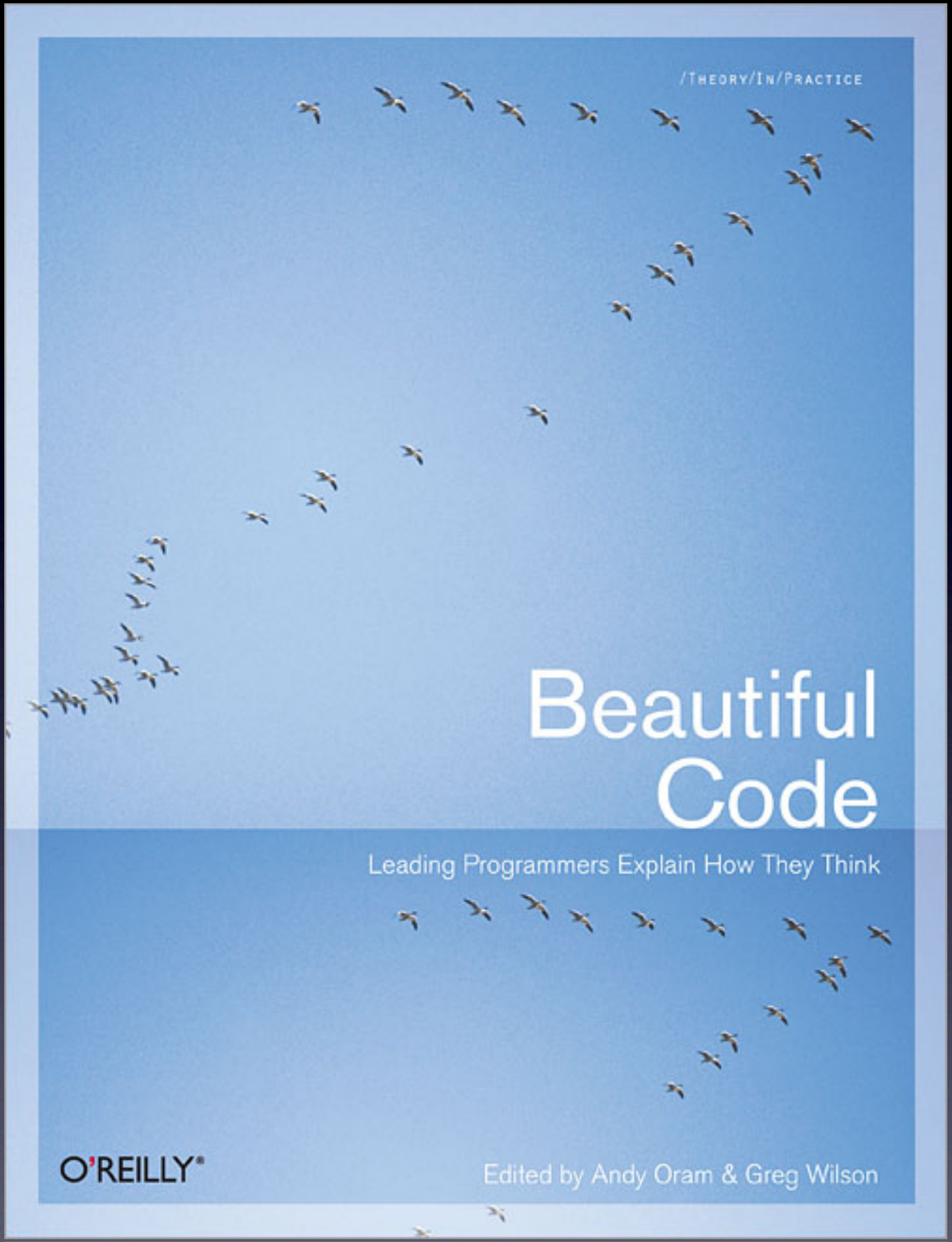
objects

separation of concerns

model-view-controller

style-behavior-logic

patterns

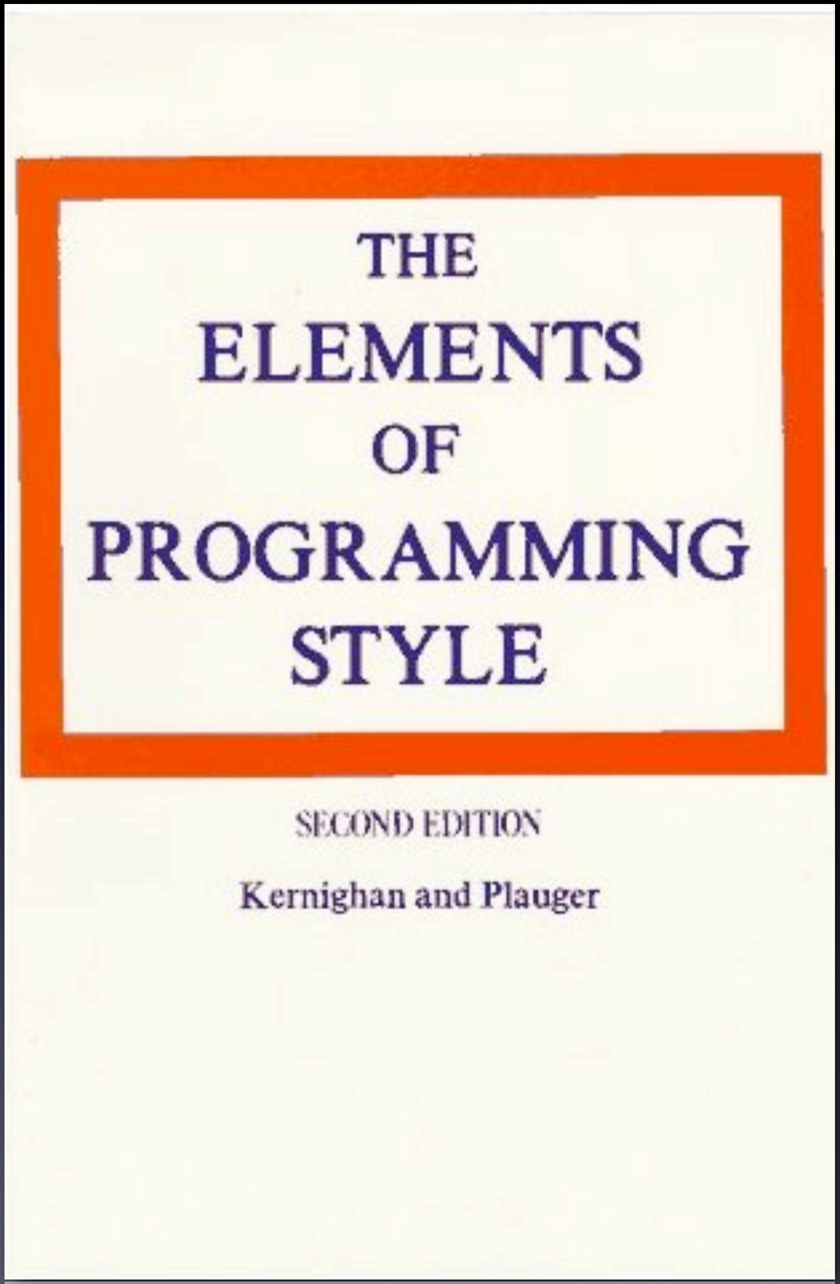

/Theory/In/Practice
Beautiful Code
Leading Programmers Explain How They Think
O'REILLY
Edited by Andy Oram & Greg Wilson

Make it right before you make it fast. Make it clear before you make it faster. Keep it right when you make it faster.

Where there are two bugs, there is likely to be a third.

Make sure your code 'does nothing' gracefully.

Premature optimization is the root of all evil.

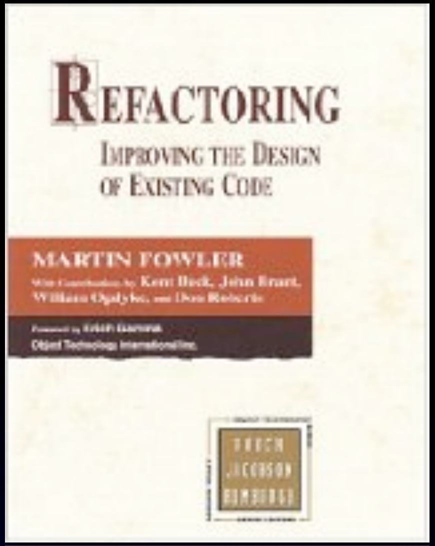

THE ELEMENTS OF PROGRAMMING STYLE

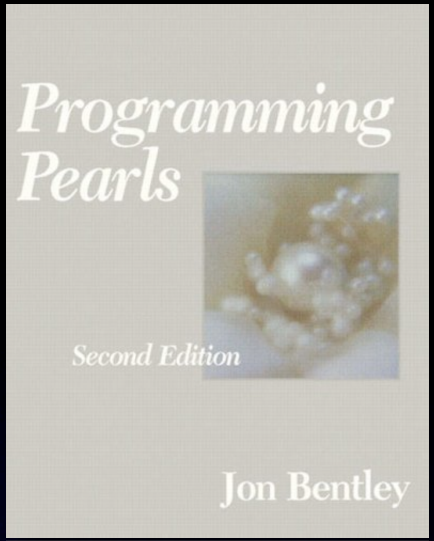
SECOND EDITION

Kernighan and Plauger

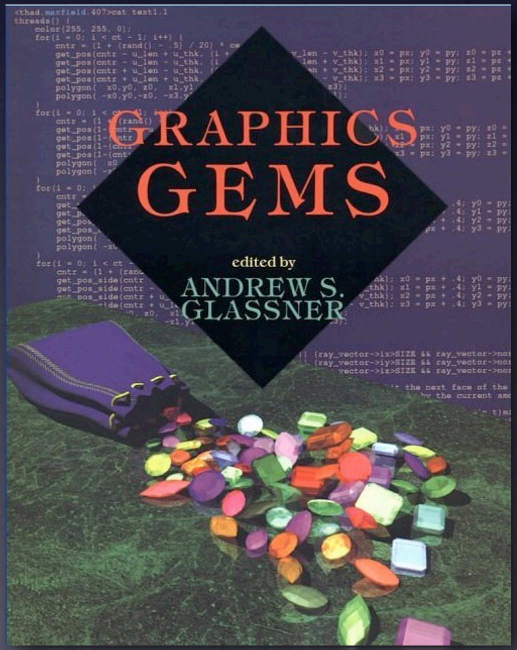
1978

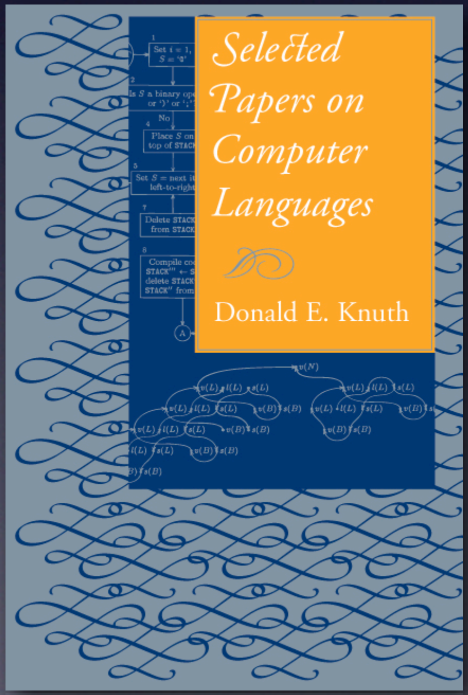
Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch


REFACTORING
IMPROVING THE DESIGN OF EXISTING CODE
MARTIN FOWLER
With Contributions by Kent Beck, John Brant, William Opdyke, and Don Roberts
Foreword by Erich Gamma
Object Technology International Inc.


Programming Pearls
Second Edition
Jon Bentley

Unearthing the excellence in JavaScript
JavaScript: The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford


GRAPHICS GEMS
edited by
ANDREW S. GLASSNER


Selected Papers on Computer Languages
Donald E. Knuth

Lessons from the Past

Importance of Tools

Proficiency in Debugging

Understanding the Leaky Abstraction

Being Pixel Retentive

Inspired by Problems

Students of Beautiful Code

Credits

http://flickr.com/photos/rogersmith/53912456/

http://flickr.com/photos/threesixes/12169049/

http://flickr.com/photos/playstar_rocker/2626983033/

http://flickr.com/photos/jakecaptive/49915119/

http://www.oldcalculatormuseum.com/wang600.html

http://flickr.com/photos/trainor/451799414/

http://flickr.com/photos/tom-b/2441980046/

http://www.1000bit.it/support/manuali/apple/lisa/LisaPatentQuickDraw.pdf

http://en.wikipedia.org/wiki/Image:TI-59.jpg