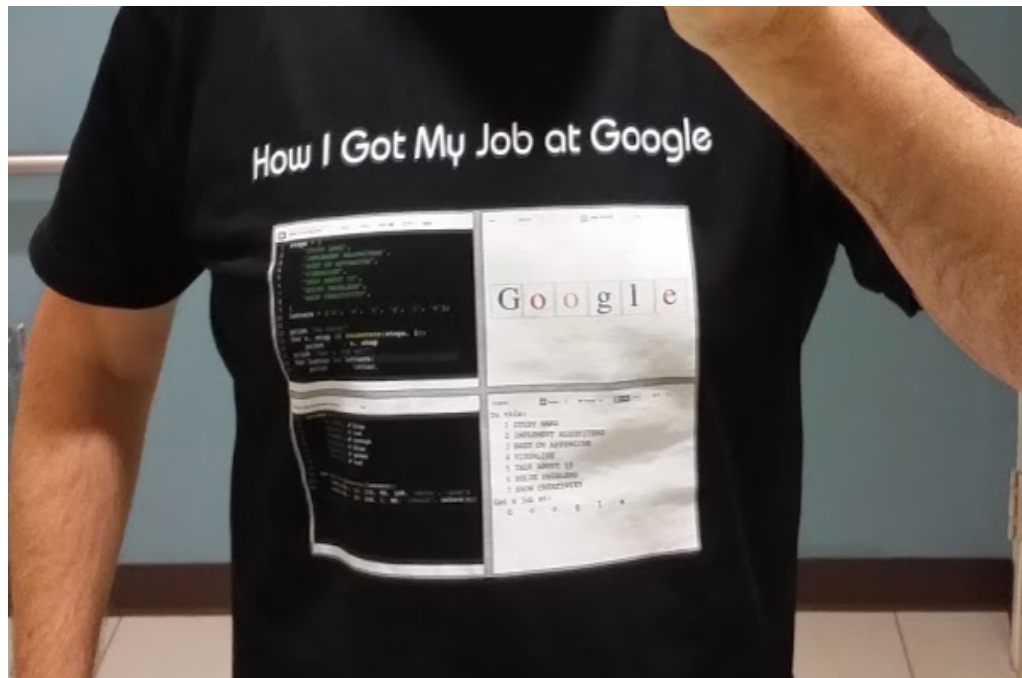


How I got my job at Google

Or: Python Algorithm Visualization in the Browser



Chris Laffra
Software Engineer at Google

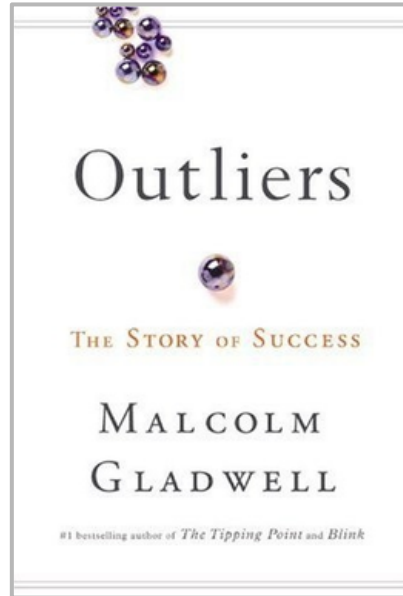
THEORY



You **Tube**



EXPERIENCE



PERSONALITY

EMOTIONAL INTELLIGENCE

Self-Awareness
Self-Management
Social Awareness
Relationship Management

ORGANIZATIONAL JUSTICE

Decision Fairness
Information Sharing
Outcome Concern

CHARACTER

Integrity
Credibility
Values
Differences

DEVELOPMENT

Lifelong Learning
Developing Others

[Travis Bradberry, Leadership 2.0](#)



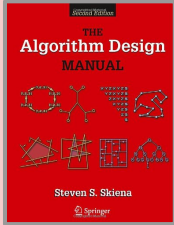
How I Got My Job At Google

Or: Python Algorithm Visualization in the Browser

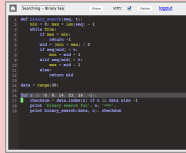


Chris Laffra

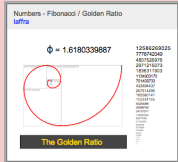
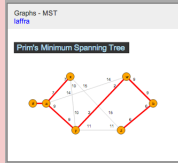
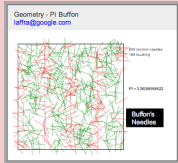
My Approach *



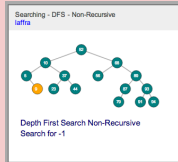
1. Study





2. Practice



3. Visualize



CodeMirror  

Replay

- Source Line
- Animation
- Print Output

HTML5 Canvas D3.js

Edit

- Edit in the Browser
- Use VIM Key Bindings
- Experiment/Play
- Share/Publish

browser

JSON

Render

monkey patch

Jinja

server

\$.ajax

2 — Run sandbox

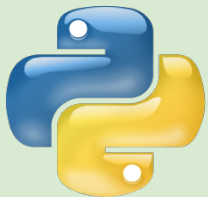
3 — Record sys.settrace

- Python hosted on AppEngine
- Scripts versioned in NDB
- HTML rendered with Jinja
- Scripts executed on server

Try it out yourself. See: <http://chrislaffra.com>

* = Prior results are no guarantee for future success.





CodeMirror



The screenshot shows a web-based code editor with the following content:

```

1 iterations: w, h = 9, 20, 20
2
3 def mandelbrot(x1, x2, y1, y2, iterations):
4     img = [[-1]*w for _ in range(h)]
5     for py in range(h):
6         for px in range(w):
7             z = x1 + px + 1j * (y1 + py)
8             for i in range(iterations):
9                 z = z*z
10                c = complex(real, imag, 0.0)
11                if abs(z) > 2:
12                    break
13                color = i
14            else:
15                color = iterations
16            img[py][px] = color
17    return img
18
19 # See http://nbviewer.ipython.org/570733546af4643c43
20 image = mandelbrot(-2.0, 1.0, -1.0, 1.0, 30)
21
22

```

Below the code, there is a visualization of the Mandelbrot set, a "Show Visualization Errors" section, and an "Output" section that says "Program finished. Hit F9 or Ctrl-Enter to run the script again."

HTML5
Canvas
D3.js



Replay

- Source Line
- Animation
- Print Output

5

Edit

- Edit in the Browser
- VIM Key Bindings
- Experiment/Play
- Share/Publish

1

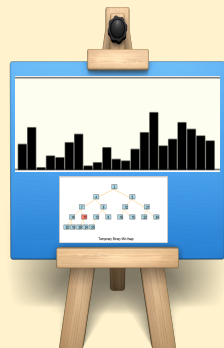
JSON



\$.ajax

JSON

\$.ajax



Render

monkey
patch



4

2

Run

sandbox



3

Record

sys.settrace



- Python hosted on AppEngine
- Scripts versioned in NDB
- HTML rendered with Jinja
- Scripts executed on server

```
3 def runScript(script, viz):
4     context = ...
5     with SandBox(script, viz):
6         with Tracer():
7             exec script in context
8
9 class Sandbox(object):
10    def __enter__(self):
11        monkeyPatchAllBadStuff()
12        staticallyAnalyzeScripts()
13
14    def __exit__(self, *args):
15        unMonkeyPatchAllBadStuff()
16
17 class Tracer(object):
18    def __enter__(self):
19        sys.settrace(self.trace)
20
21    def trace(self, frame, event, args):
22        checkIfRanTooLong()
23        if insideUserScript(frame):
24            exec self.viz in { 'rect': ..., 'line': ... }
25            return self.trace
26
27    def __exit__(self, *args):
28        sys.settrace(None)
29
```

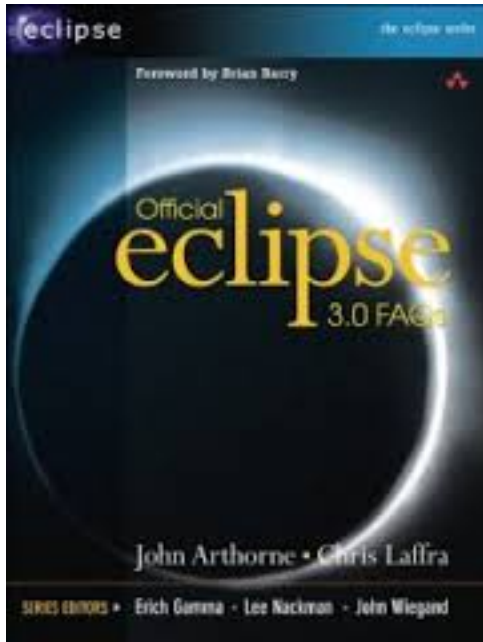
Pseudo-code for sandbox and visualizing tracer, using context managers, monkey patching and sys.settrace.



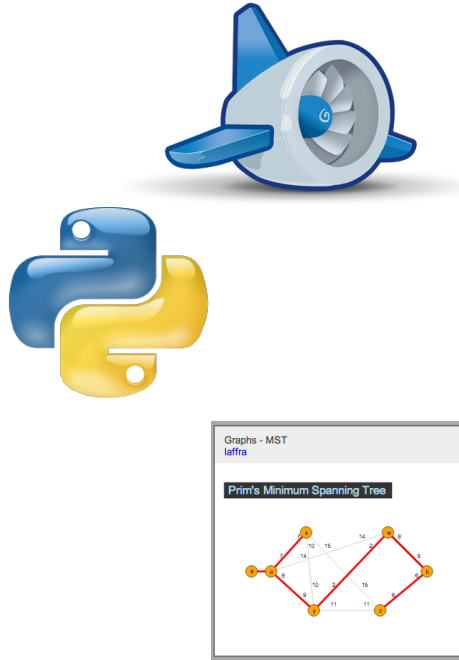
Amateur sandboxing in Python



THEORY



EXPERIENCE



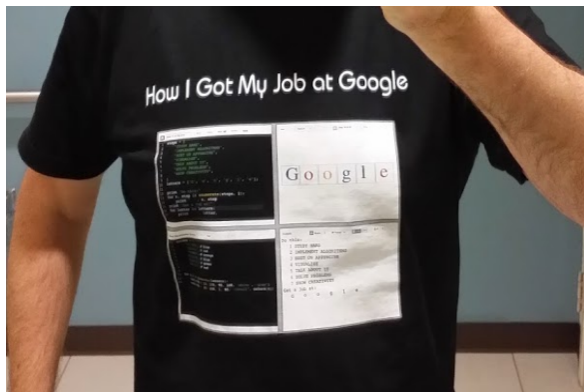
PERSONALITY



Can I try it?

Questions?

Open Source?



IPython
Notebook?

Chris Laffra
Software Engineer at Google

Can you get me
an interview?