Lecture 4: Code Archaeology

17-313: Foundations of Software Engineering
Rohan Padhye, Michael Hilton, Chris Timperley, and Daye Nam

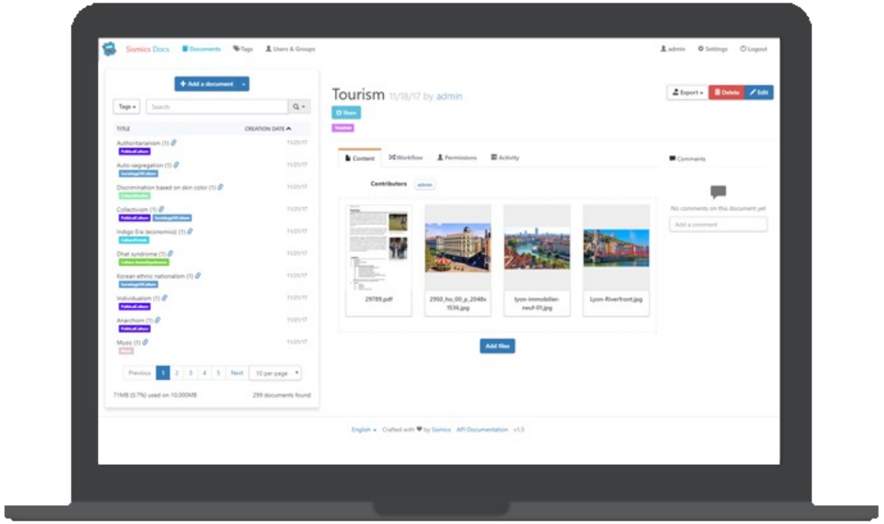
Administrivia

- HW1 is due tonight at 11:59pm
  - don't expect a reply on Slack outside of working hours
- HW2 will be released tomorrow
- Update on Team Formation
- …

Learning Goals

- Understand and scope the task of taking on and understanding a new and complex piece of existing software
- Appreciate the importance of configuring an effective IDE
- Contrast different types of code execution environments including local, remote, application, and libraries
- Enumerate both static and dynamic strategies for understanding and modifying a new codebase

Context: big ole pile of code

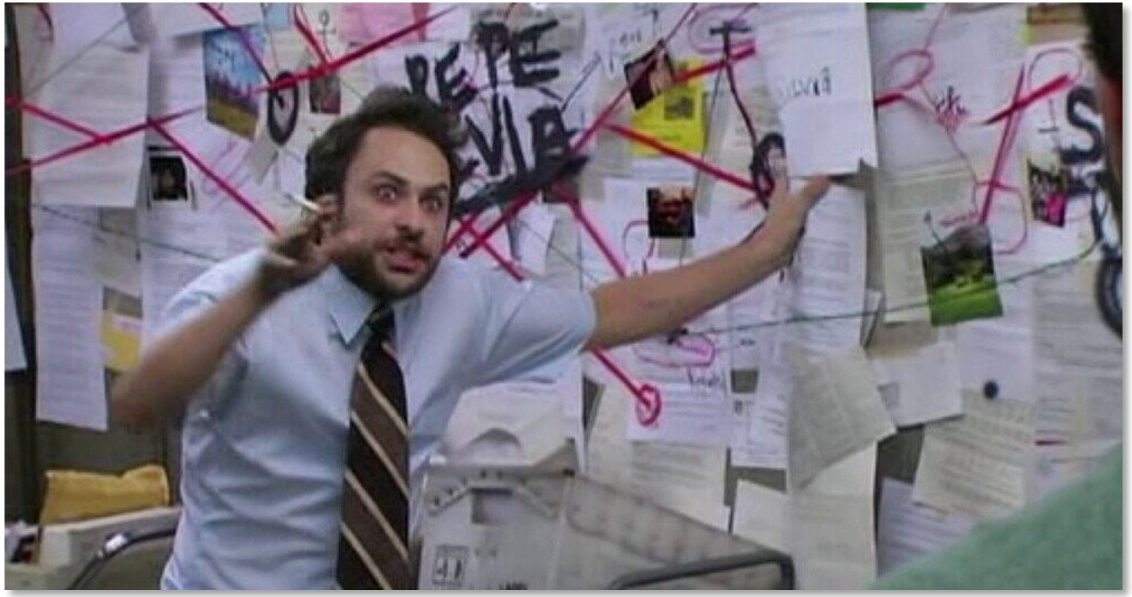


teedy

… do something with it!

You cannot understand the entire system!

Challenge: How do I tackle this codebase?

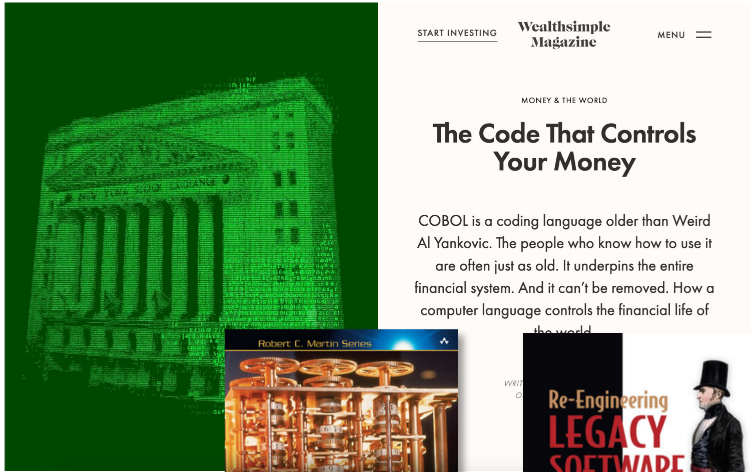


Challenge: How do I tackle this codebase?

- Leverage your previous experiences (languages, technologies, patterns)
- Consult documentation, whitepapers, experts, code owners
- Follow best practices to build a working model of the system

Bad news: There are few helpful resources!

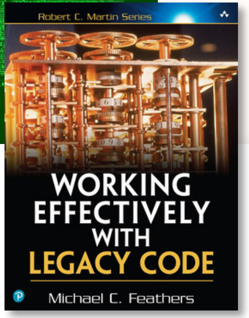
- Working Effectively with Legacy Code. Michael C. Feathers. 2004.
- Re-Engineering Legacy Software. Chris Birchall. 2016.

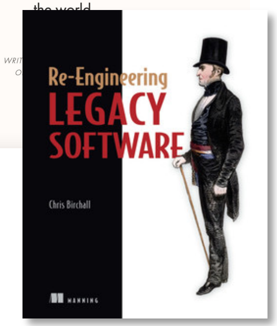

START INVESTING Wealthsimple Magazine MENU

MONEY & THE WORLD

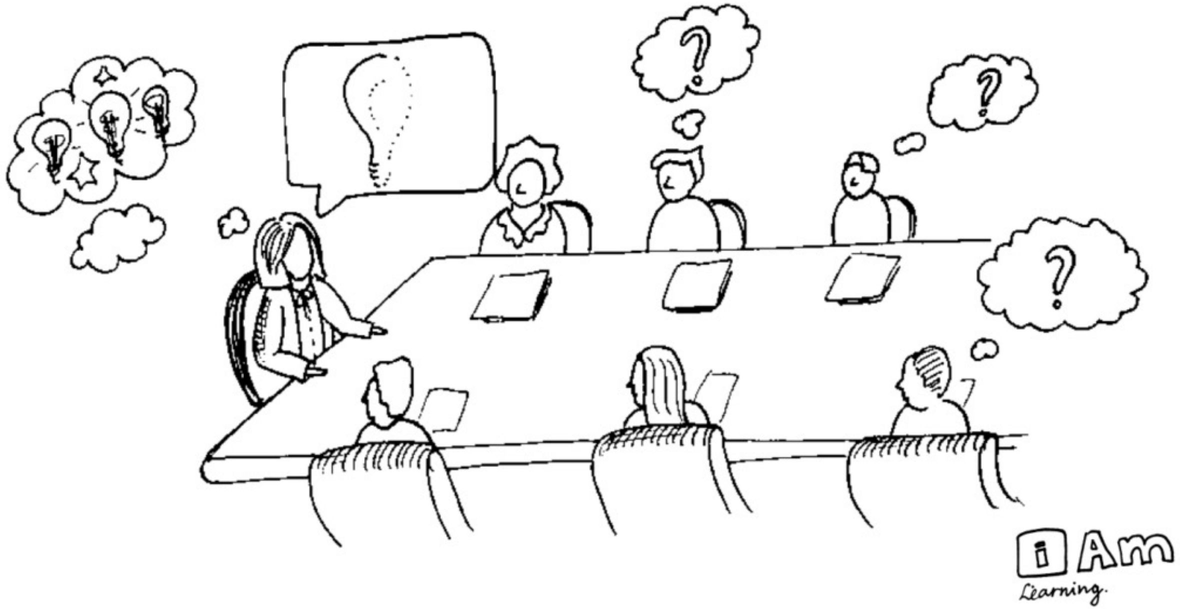
The Code That Controls Your Money

COBOL is a coding language older than Weird Al Yankovic. The people who know how to use it are often just as old. It underpins the entire financial system. And it can't be removed. How a computer language controls the financial life of the world.



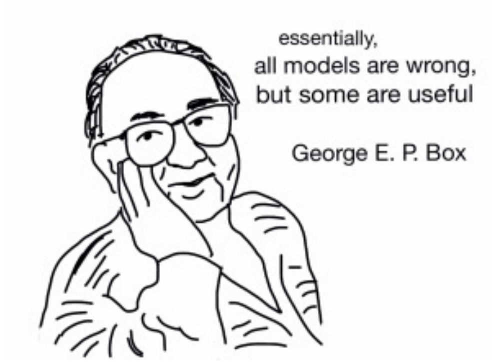


Why? Because of the Curse of Knowledge

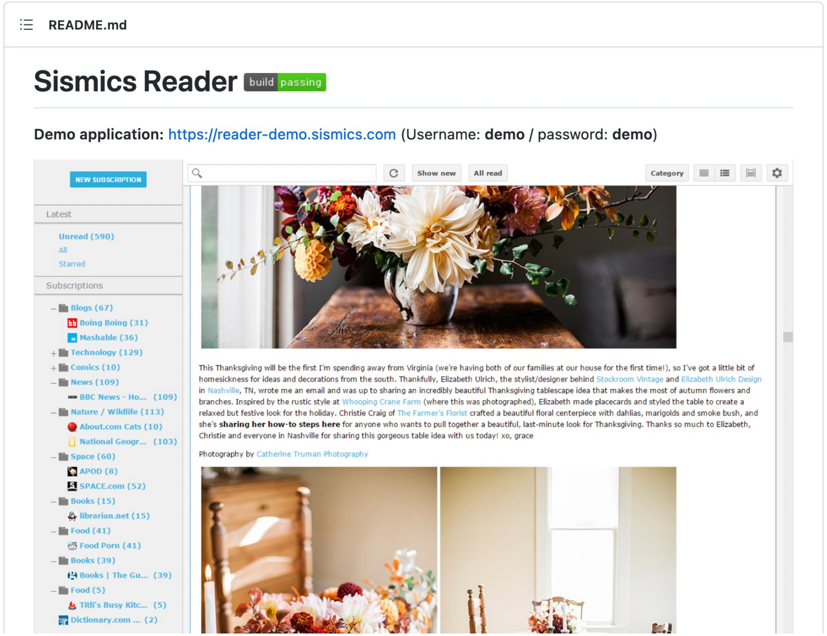


Today: How to tackle codebases

- Goal: develop and test a working model or set of working hypotheses about how (some part of) a system works
- Working model: an understanding of the pieces of the system (components), and the way they interact (connections)
- Focus: Observation, probes, and hypothesis testing
  - helpful tools and techniques!



Live Demonstration: sismics/Reader



README.md

Sismics Reader build passing

Demo application: https://reader-demo.sismics.com (Username: **demo** / password: **demo**)

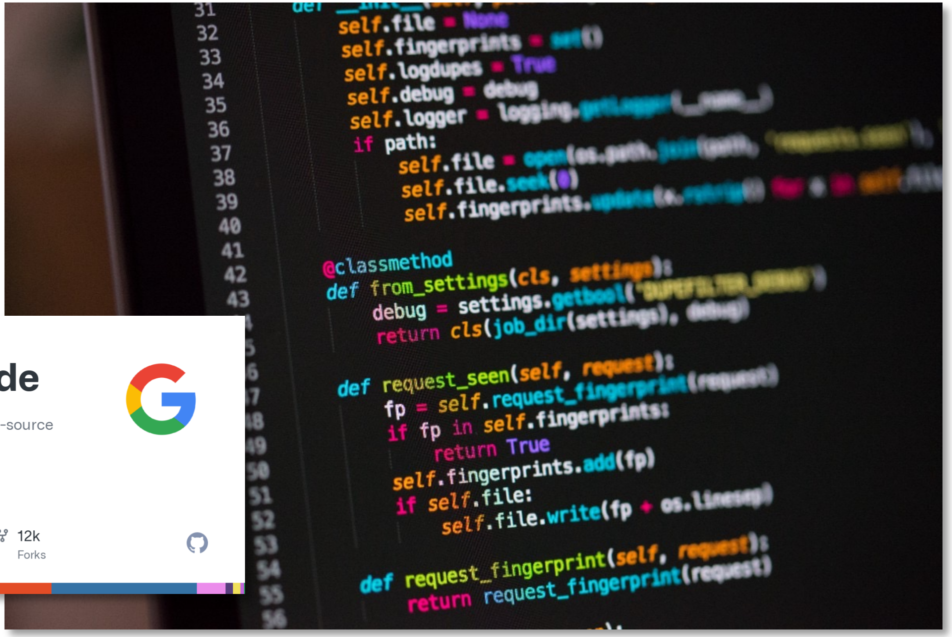
This Thanksgiving will be the first I'm spending away from Virginia (we're having both of our families at our house for the first time!), so I've got a little bit of homesickness for ideas and decorations from the south. Thankfully, Elizabeth Ulrich, the stylist/designer behind Stockroom Vintage and Elizabeth Ulrich Design in Nashville, TN, wrote me an email and was up to sharing an incredibly beautiful Thanksgiving tablescape idea that makes the most of autumn flowers and branches. Inspired by the rustic style at Whooping Crane Farm (where this was photographed), Elizabeth made placecards and styled the table to create a relaxed but festive look for the holiday. Christie Craig of The Farmer's Florist crafted a beautiful floral centerpiece with dahlias, marigolds and smoke bush, and she's **sharing her how-to steps here** for anyone who wants to pull together a beautiful, last-minute look for Thanksgiving. Thanks so much to Elizabeth, Christie and everyone in Nashville for sharing this gorgeous table idea with us today! xo, grace

Photography by Catherine Truman Photography

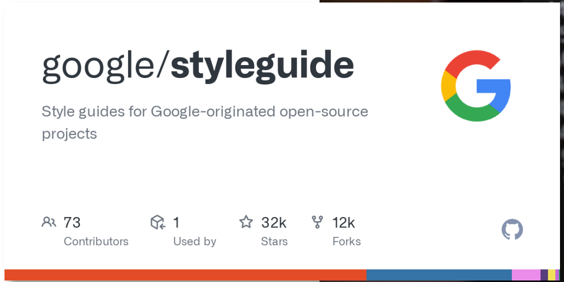
https://github.com/CMU-313/reader

Observation: Software is full of patterns

- File structure
- System architecture
- Code structure
- Names
- …



```
    def __init__(self, path=None, debug=False):
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'), 'a+')
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in self.file)

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DUPEFILTER_DEBUG')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```



google/**styleguide**

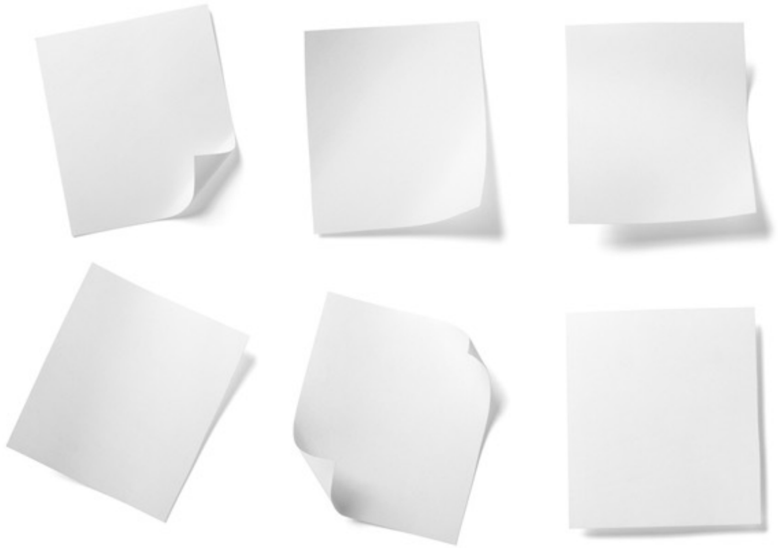
Style guides for Google-originated open-source projects

73 Contributors · 1 Used by · 32k Stars · 12k Forks

Observation: Software is massively redundant

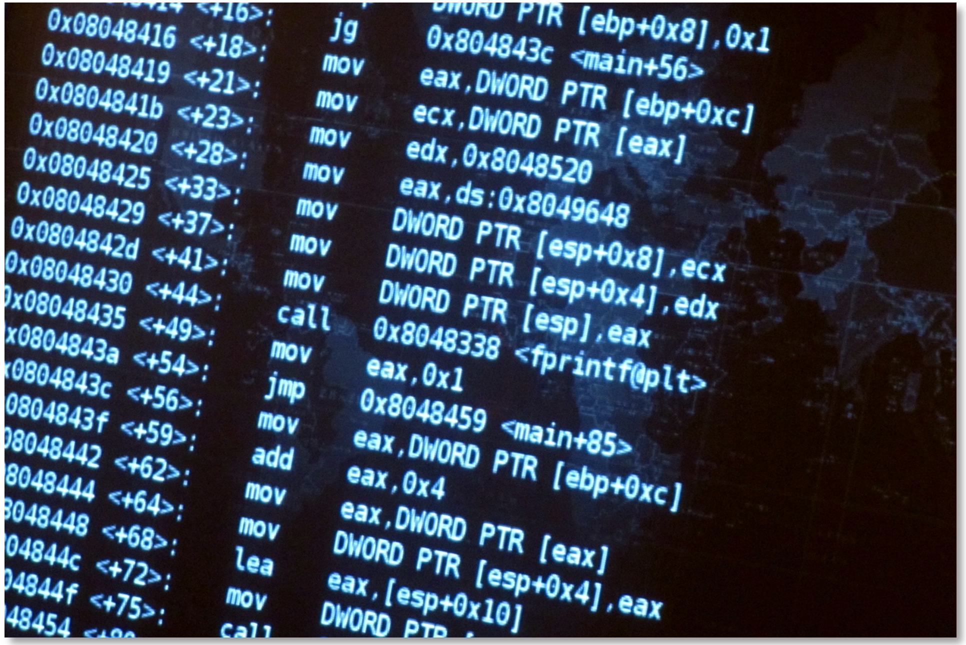
- There's always something to copy/use as a starting point!

Observation: Code must run to do stuff!



Observation: If code runs, it must have a beginning…

Observation: If code runs, it must exist...



```
         <+16>:          DWORD PTR [ebp+0x8],0x1
0x08048416 <+18>:   jg     0x804843c <main+56>
0x08048419 <+21>:   mov    eax,DWORD PTR [ebp+0xc]
0x0804841b <+23>:   mov    ecx,DWORD PTR [eax]
0x08048420 <+28>:   mov    edx,0x8048520
0x08048425 <+33>:   mov    eax,ds:0x8049648
0x08048429 <+37>:   mov    DWORD PTR [esp+0x8],ecx
0x0804842d <+41>:   mov    DWORD PTR [esp+0x4],edx
0x08048430 <+44>:   mov    DWORD PTR [esp],eax
x08048435 <+49>:    call   0x8048338 <fprintf@plt>
x0804843a <+54>:    mov    eax,0x1
x0804843c <+56>:    jmp    0x8048459 <main+85>
0804843f <+59>:     mov    eax,DWORD PTR [ebp+0xc]
08048442 <+62>:     add    eax,0x4
8048444 <+64>:      mov    eax,DWORD PTR [eax]
048448 <+68>:       mov    DWORD PTR [esp+0x4],eax
04844c <+72>:       lea    eax,[esp+0x10]
04844f <+75>:       mov    DWORD PTR
48454
```

The Beginning: Entry Points

- **Locally installed programs**: run cmd, OS launch, I/O events, etc.
- **Local applications in dev**: build + run, test, deploy (e.g., docker)
- **Web apps server-side**: Browser sends HTTP request (GET/POST)
- **Web apps client-side**: Browser runs JavaScript

Code must exist. But where?

- **Locally installed programs**: run cmd, OS launch, I/O events, etc.
  - Binaries (machine code) on your computer
- **Local applications in dev**: build + run, test, deploy (e.g., docker)
  - Source code in repository (+ dependencies)
- **Web apps server-side**: Browser sends HTTP request (e.g., GET, POST)
  - Code runs remotely (you can only observe outputs)
- **Web apps client-side**: Browser runs JavaScript
  - Source code is downloaded and run locally (see: browser dev tools!)

Can running code be **P**robed/**U**nderstood/**E**dited?

Transparent	Translucent		Opaque	
☐	🟨		🟧	
Source code built locally	Binaries running locally		Server-side apps running remotely	
	Open source	Closed source	Open source	Closed source
(P+U+E)	(P+U)	(P)	(U)	-

Creating a model of unfamiliar code

Source code built locally

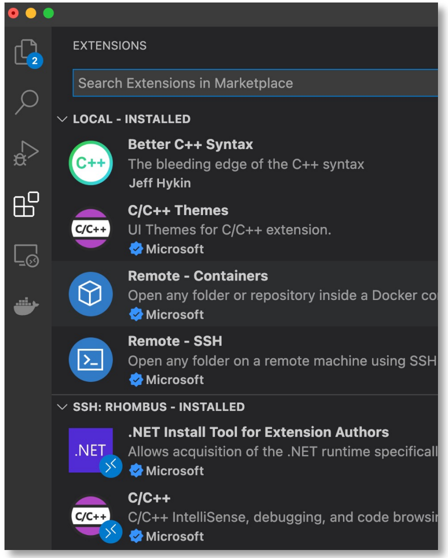
Information Gathering

- Basic needs:
  - Code/file search and navigation
  - Code editing (probes)
  - Execution of code, tests
  - Observation of output (observation)
- Many choices here on tools! Depends on circumstance.
  - grep/find/etc.   Having a command on Unix tools is invaluable
  - A decent IDE
  - Debugger
  - Test frameworks + coverage reports
  - Google (or your favorite web search engine)

At the command line: **grep** and **find**!
(Do a web search for tutorials)

Static Information Gathering: Use an IDE!

Real software is too complex to keep in your head



EXTENSIONS

Search Extensions in Marketplace

LOCAL - INSTALLED

Better C++ Syntax
The bleeding edge of the C++ syntax
Jeff Hykin

C/C++ Themes
UI Themes for C/C++ extension.
Microsoft

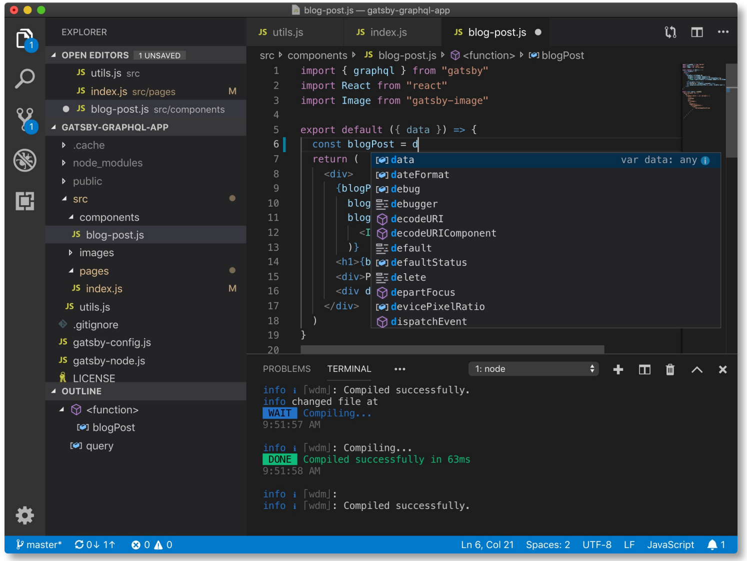
Remote - Containers
Open any folder or repository inside a Docker co
Microsoft

Remote - SSH
Open any folder on a remote machine using SSH
Microsoft

SSH: RHOMBUS - INSTALLED

.NET Install Tool for Extension Authors
Allows acquisition of the .NET runtime specificall
Microsoft

C/C++
C/C++ IntelliSense, debugging, and code browsi
Microsoft



blog-post.js — gatsby-graphql-app

EXPLORER

OPEN EDITORS 1 UNSAVED
- utils.js src
- index.js src/pages M
- blog-post.js src/components

GATSBY-GRAPHQL-APP
- .cache
- node_modules
- public
- src
  - components
    - blog-post.js
  - images
  - pages
    - index.js M
  - utils.js
- .gitignore
- gatsby-config.js
- gatsby-node.js
- LICENSE

OUTLINE
- <function>
  - blogPost
- query

utils.js | index.js | blog-post.js

src › components › blog-post.js › <function> › blogPost

```
import { graphql } from "gatsby"
import React from "react"
import Image from "gatsby-image"

export default ({ data }) => {
  const blogPost = d
  return (
    <div>
      {blogP
        blog
        blog
          <I
      )}
      <h1>{b
      <div>P
      <div d
    </div>
  )
}
```

```
data                var data: any
dateFormat
debug
debugger
decodeURI
decodeURIComponent
default
defaultStatus
delete
departFocus
devicePixelRatio
dispatchEvent
```

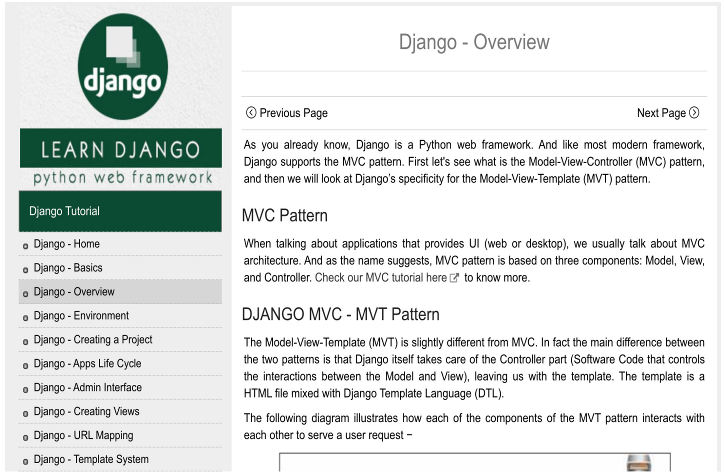
PROBLEMS TERMINAL 1: node

```
info ℹ ｢wdm｣: Compiled successfully.
info changed file at
 WAIT  Compiling...
9:51:57 AM

info ℹ ｢wdm｣: Compiling...
 DONE  Compiled successfully in 63ms
9:51:58 AM

info ℹ ｢wdm｣:
info ℹ ｢wdm｣: Compiled successfully.
```

master* 0↓ 1↑ ⊗ 0 ⚠ 0 Ln 6, Col 21 Spaces: 2 UTF-8 LF JavaScript 🔔 1

Consider documentation and tutorials judiciously

- Great for discovering entry points!
- Can teach you about general structure, architecture (more on this later in the semester)
- As you gain experience, you will recognize more of these, and you will immediately know something about how the program works
- Also: discussion boards; issue trackers



Django - Overview

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django's specificity for the Model-View-Template (MVT) pattern.

MVC Pattern

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. Check our MVC tutorial here to know more.

DJANGO MVC - MVT Pattern

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

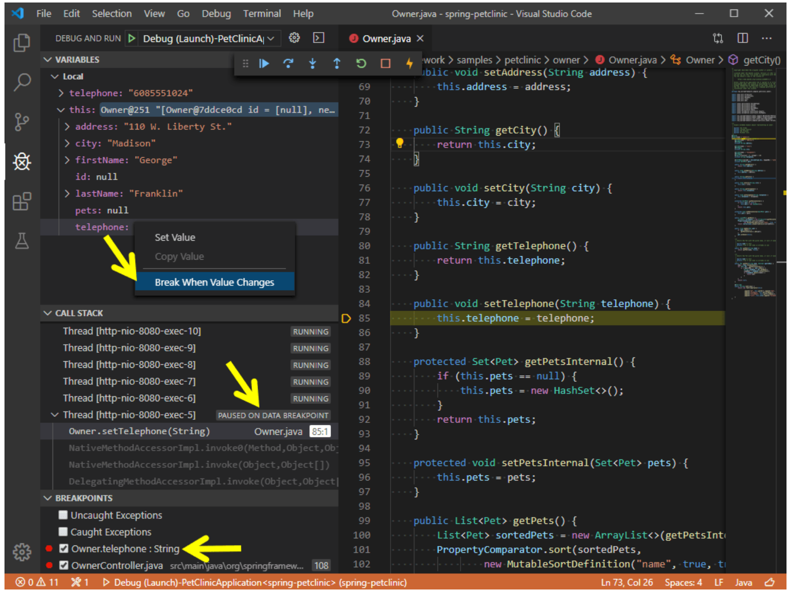
The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request −

Dynamic Information Gathering
Change helps to inform and refine mental models

1. Build it.
2. Run it.
3. Change it.
4. Run it again.
5. How did the behavior change?

Probes: Observe, control or “lightly” manipulate execution

- print(“this code is running!”)
- Structured logging
- Debuggers
  - Breakpoint, eval, step through / step over
  - (Some tools even support remote debugging)
- Delete debugging
- Firefox Developer Tools


File Edit Selection View Go Debug Terminal Help
Owner.java - spring-petclinic - Visual Studio Code
DEBUG AND RUN
Debug (Launch)-PetClinicA|
Owner.java
VARIABLES
Local
Set Value
Copy Value
Break When Value Changes
CALL STACK
BREAKPOINTS
...work > samples > petclinic > owner > Owner.java > Owner > getCity()

```
telephone: "6085551024"
this: Owner@251 "[Owner@7ddce0cd id = [null], ne…
  address: "110 W. Liberty St."
  city: "Madison"
  firstName: "George"
  id: null
  lastName: "Franklin"
  pets: null
  telephone:
```

```
Thread [http-nio-8080-exec-10]   RUNNING
Thread [http-nio-8080-exec-9]    RUNNING
Thread [http-nio-8080-exec-8]    RUNNING
Thread [http-nio-8080-exec-7]    RUNNING
Thread [http-nio-8080-exec-6]    RUNNING
Thread [http-nio-8080-exec-5]    PAUSED ON DATA BREAKPOINT
  Owner.setTelephone(String)     Owner.java 85:1
  NativeMethodAccessorImpl.invoke0(Method,Object,Ob
  NativeMethodAccessorImpl.invoke(Object,Object[])
  DelegatingMethodAccessorImpl.invoke(Object,Object|
```

```
☐ Uncaught Exceptions
☐ Caught Exceptions
● ☑ Owner.telephone : String
● ☑ OwnerController.java  src\main\java\org\springframew...  108
```

```java
    public void setAddress(String address) {
        this.address = address;
    }

    public String getCity() {
        return this.city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public String getTelephone() {
        return this.telephone;
    }

    public void setTelephone(String telephone) {
        this.telephone = telephone;
    }

    protected Set<Pet> getPetsInternal() {
        if (this.pets == null) {
            this.pets = new HashSet<>();
        }
        return this.pets;
    }

    protected void setPetsInternal(Set<Pet> pets) {
        this.pets = pets;
    }

    public List<Pet> getPets() {
        List<Pet> sortedPets = new ArrayList<>(getPetsInt
        PropertyComparator.sort(sortedPets,
                new MutableSortDefinition("name", true, t
```

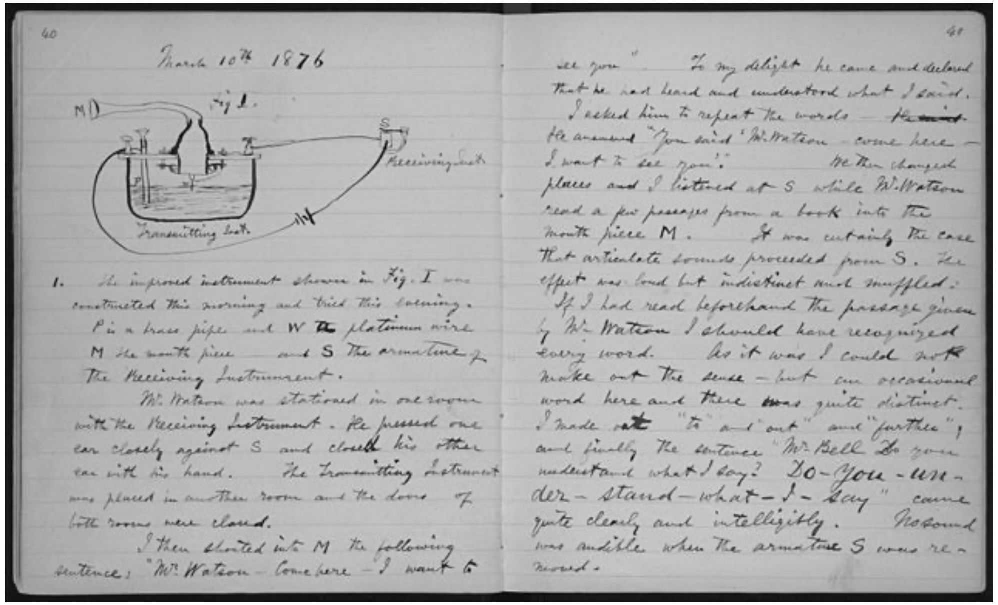
```
⊗ 0 ⚠ 11   1   Debug (Launch)-PetClinicApplication<spring-petclinic> (spring-petclinic)   Ln 73, Col 26   Spaces: 4   LF   Java
```

Step 0: sanity check basic model + hypotheses

- Confirm that you can build and run the code.
  - Ideally *both* using the tests provided, *and* by hand.
- Confirm that the code you are running *is the code you built*
- Confirm that you can make *an externally visible change*
- How? Where? Starting points:
  - Run an existing test, change it
  - Write a new test
  - Change the code, write or rerun a test that should notice the change

Document and share your findings!

- Update README and docs
  - or, better: use a Developer Wiki
  - use Mermaid for diagrams
- Collaborate with others
- Include negative results, too!



M
Fig 1.
S
Receiving Inst.
Transmitting Inst.

40

March 10th 1876

1. The improved instrument shown in Fig. I was constructed this morning and tried this evening. P is a brass pipe and W ~~the~~ platinum wire M the mouth piece and S the armature of The Receiving Instrument.

Mr. Watson was stationed in one room with the Receiving Instrument. He pressed one ear closely against S and closed his other ear with his hand. The Transmitting Instrument was placed in another room and the doors of both rooms were closed.

I then shouted into M the following sentence: "Mr. Watson — Come here — I want to

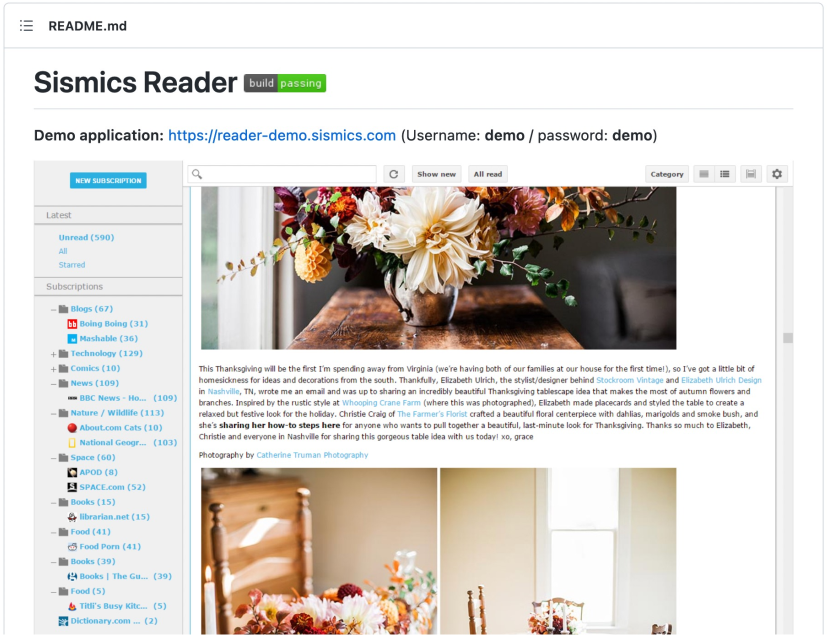
41

see you." To my delight he came and declared that he had heard and understood what I said.

I asked him to repeat the words — ~~He said~~ He answered "You said 'Mr. Watson — come here — I want to see you.'" We then changed places and I listened at S while Mr. Watson read a few passages from a book into the mouth piece M. It was certainly the case that articulate sounds proceeded from S. The effect was loud but indistinct and muffled.

If I had read beforehand the passage given by Mr. Watson I should have recognized every word. As it was I could not make out the sense — but an occasional word here and there ~~was~~ quite distinct. I made ~~out~~ "to" and "out" and "further"; and finally the sentence "Mr. Bell Do you understand what I say? Do-you-un-der-stand-what-I-say" came quite clearly and intelligibly. No sound was audible when the armature S was removed.

Let’s try some of these techniques again…


README.md

Sismics Reader build passing

Demo application: https://reader-demo.sismics.com (Username: **demo** / password: **demo**)

This Thanksgiving will be the first I’m spending away from Virginia (we’re having both of our families at our house for the first time!), so I’ve got a little bit of homesickness for ideas and decorations from the south. Thankfully, Elizabeth Ulrich, the stylist/designer behind Stockroom Vintage and Elizabeth Ulrich Design in Nashville, TN, wrote me an email and was up to sharing an incredibly beautiful Thanksgiving tablescape idea that makes the most of autumn flowers and branches. Inspired by the rustic style at Whooping Crane Farm (where this was photographed), Elizabeth made placecards and styled the table to create a relaxed but festive look for the holiday. Christie Craig of The Farmer’s Florist crafted a beautiful floral centerpiece with dahlias, marigolds and smoke bush, and she’s **sharing her how-to steps here** for anyone who wants to pull together a beautiful, last-minute look for Thanksgiving. Thanks so much to Elizabeth, Christie and everyone in Nashville for sharing this gorgeous table idea with us today! xo, grace

Photography by Catherine Truman Photography

https://github.com/CMU-313/reader