The Joy of Software Development

ABOUT ME

Nemo
@captn3m0
captnemo.in

Work @Razorpay

WHY?

- Data Structures
- Computer Architecture
- Algorithms
- Operating Systems
- Software Eng
- Computer Networks
- Compiler Theory

CSE course-structure

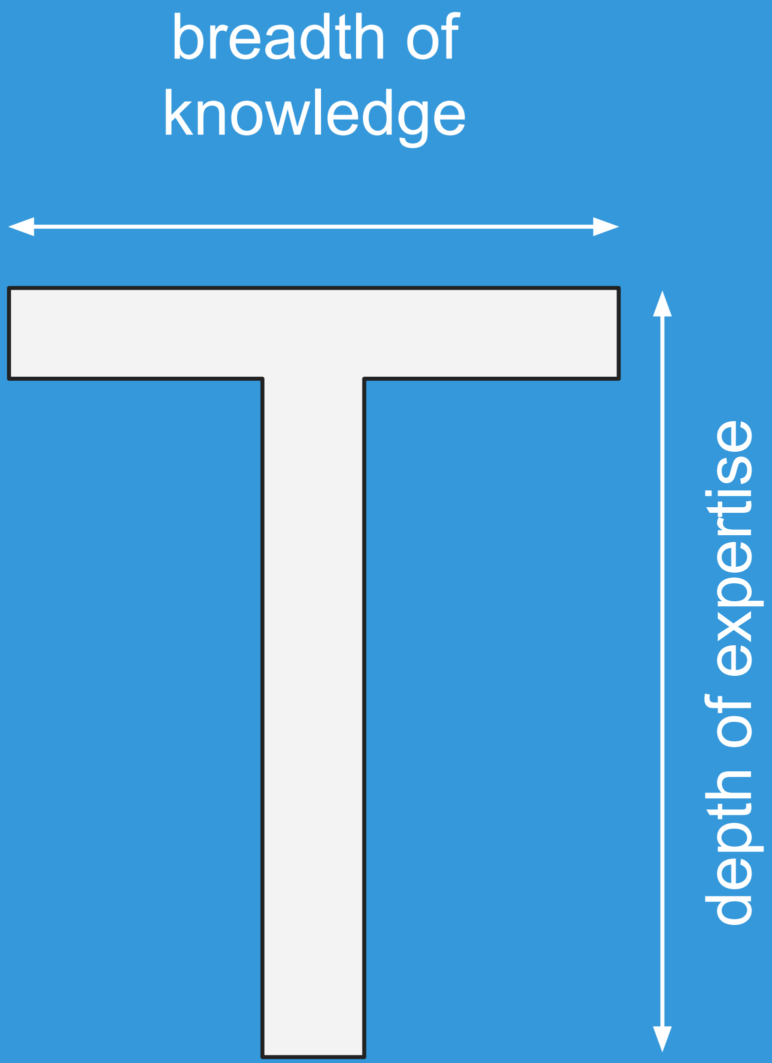
- HTML, JavaScript, CSS and GWT.
- Python, JavaScript, and C++
- Web applications, databases, distributed systems, and machine learning
- UI development, JavaScript, open source development.

Job requirements at Google+Quora

*OVERVIEW**

NON-EXHAUSTIVE
LOTS OF CONCEPTS
VERY LITTLE CODE
SLIDES WILL BE UP SOON
NO NEED TO TAKE NOTES
NEEDS BEFORE SOLUTIONS
Q&A AT END

Breadth-first learning



When I need to know more about something, then I dig into it and learn what I need to know. Breadth first, depth as needed.

- JustaProgrammer

*SOFTWARE DEVELOPMENT**

WATERFALL MODEL
AGILE DEVELOPMENT
RAPID PROTOTYPING
EXTREME PROGRAMMING
SCRUM

WTH?

WHAT IS THE MOST IMPORTANT CHARACTERISTIC OF A SOFTWARE?

Across all viewpoints (Developer, User, …)

*ANSWERS I EXPECTED**

- Correctness (Do what it’s supposed to do)
- Secure (Confidentiality, Integrity, Availability)
- Available (Be in market, not development)
- Usability
- Complexity
- Maintainable

ITERATIVE DEVELOPMENT

DONE

IS BETTER THAN

PERFECT

ITERATIVE DEVELOPMENT

STAY
FOCUSED

&

KEEP
SHIPPING

ITERATIVE DEVELOPMENT

EXPERIMENT

FAIL

LEARN

REPEAT

*PRACTICAL ITERATIVE DEVELOPMENT**

- Launch ASAP
- Take user feedback *regularly*
- Have frequent deploys/Ship regularly
- Have a tight feedback loop

*CONNECT **

- Sony PlayStation Network
- LinkedIn
- Gawker Media
- LastPass
- RSA Security
- Sony Entertainment

SOFTWARE SECURITY IS

COUNTERINTUITIVE

“security is, in most cases, the opposite of obscurity. It's really hard to explain to a non-programmer that the most secure system is the one that everyone understands perfectly.”

- neilk on HN

SOFTWARE SECURITY IS

HARD

Heartbleed remained undetected for almost 3 years in a piece of code used by everyone.

Shellshock vulnerability was introduced in the bash code in 1989. It was identified in 2014.

SOFTWARE SECURITY IS

EASY TO GET WRONG

- Often well meaning security patches bring on new vulnerabilities.
- Encryption is very easy to get wrong:
    - nonce reuse
    - RNG vulnerabilities
    - Padding Attacks

SOFTWARE SECURITY NEEDS

JUST ONE DEDICATED ATTACKER

“Almost everything can be hacked. Its just a matter of time and dedication.”

HOW TO GET STARTED

- Use bcrypt for hashing passwords.
- Run software at least privileges.
- *Never* trust user input
- Read and understand the OWASP Top 10
- Try some beginner CTFs
- Understand vulnerabilities and keep up

AGNOSTIC DEVELOPMENT

“denoting or relating to hardware or software that is compatible with many types of platform or operating system.”

AGNOSTIC DEVELOPMENT

Do system development in C++

Write quick one-time scripts in perl

Machine Learning in Python

Frontend development in Javascript

iOS -> Swift

Android -> Java

CHOOSE THE RIGHT TOOL FOR THE JOB

FREE & OPEN SOURCE DEVELOPMENT

“Name any closed source generalist programming language?”

FREE & OPEN SOURCE DEVELOPMENT

“Name any closed source generalist programming language?”

https://github.com/dotnet/roslyn

The .NET Compiler Platform ("Roslyn") provides open-source C# and Visual Basic compilers with rich code analysis APIs.

FREE & OPEN SOURCE DEVELOPMENT

- PHP
- GCC
- .NET
- Java
- ECMAScript
- Python
- Ruby
- Go

- Firefox
- Chrome(ium)
- Notepad++
- Vim

- W3C
- PNG
- JPG
- IETF
  - HTTP
  - HTTP/2
  - SMTP
- Unicode

FREE & OPEN SOURCE DEVELOPMENT

Open Source Movement:

- allowing users to change and redistribute the software will make it more powerful and reliable.

Free Software Enthusiast:

- Your software may be more powerful and reliable, but it does not *respect my freedom*

https://www.gnu.org/philosophy/open-source-misses-the-point.html

FREE & OPEN SOURCE DEVELOPMENT

1. *Don’t get scared*
2. *Participate in a community that values these principles*
    a. *Linux*
    b. *Hacker News*
    c. *GNU*
3. *Participate any way you can:*
    a. *Help out people*
    b. *Ask questions, file bugs*
    c. *Fix issues*

VERSION CONTROL

DO YOU EVEN GIT?

VERSION CONTROL

USE GIT

Or Mercurial, maybe. I won't judge.

VERSION CONTROL

BENEFITS

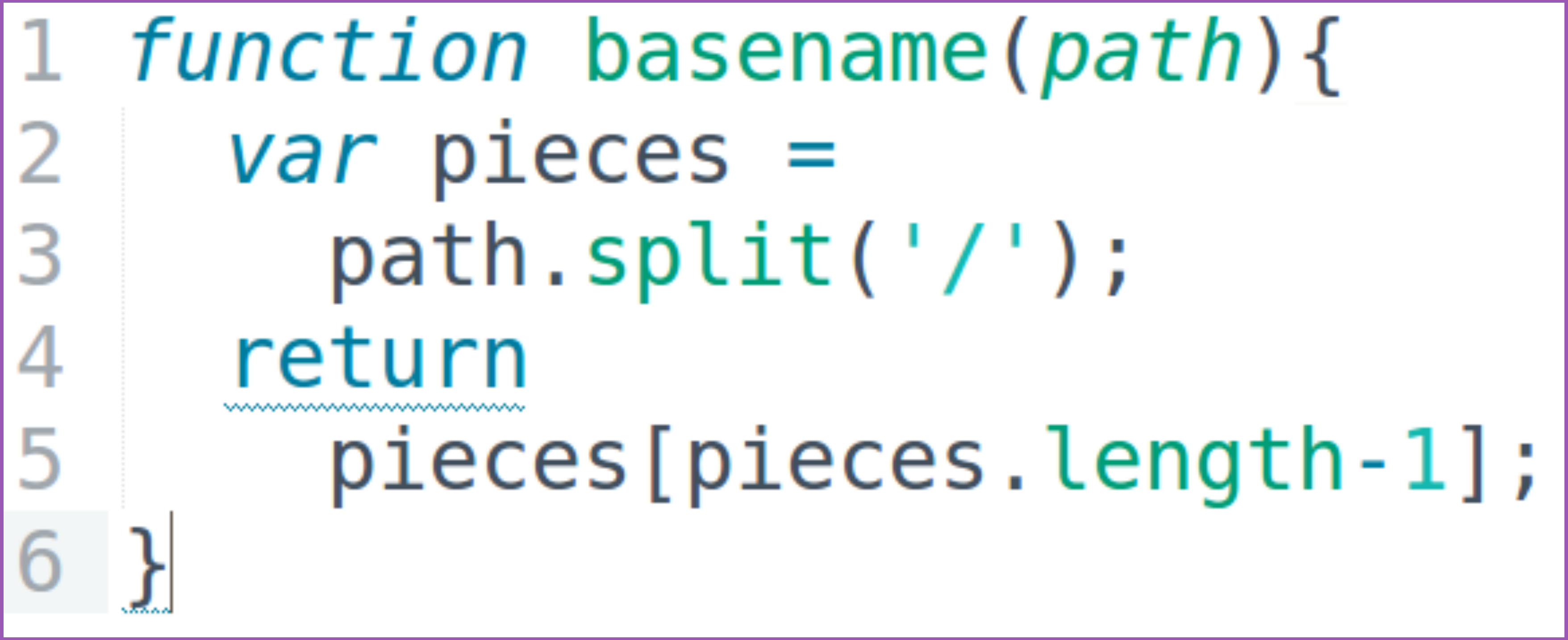
- Never hunt for backups again.
- Know when the bug was introduced
- Track changes easily
- Code reviews become easier
- Far better than emailing zip files

TEST DRIVEN DEVELOPMENT

How many of you:

1. Know about writing tests?
2. Have written tests yourselves?

TEST DRIVEN DEVELOPMENT



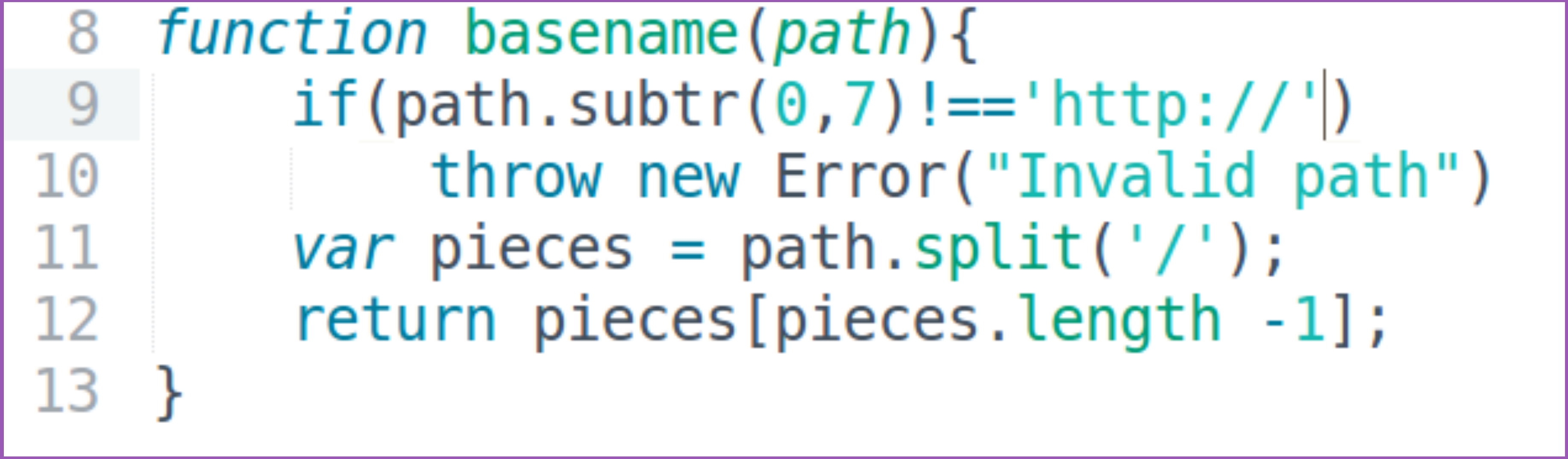
```
function basename(path){
  var pieces =
    path.split('/');
  return
    pieces[pieces.length-1];
}
```

TEST DRIVEN DEVELOPMENT

```
basename('/etc/passwd')
```

```
basename('http://twitter.com/elonmusk');
```

TEST DRIVEN DEVELOPMENT



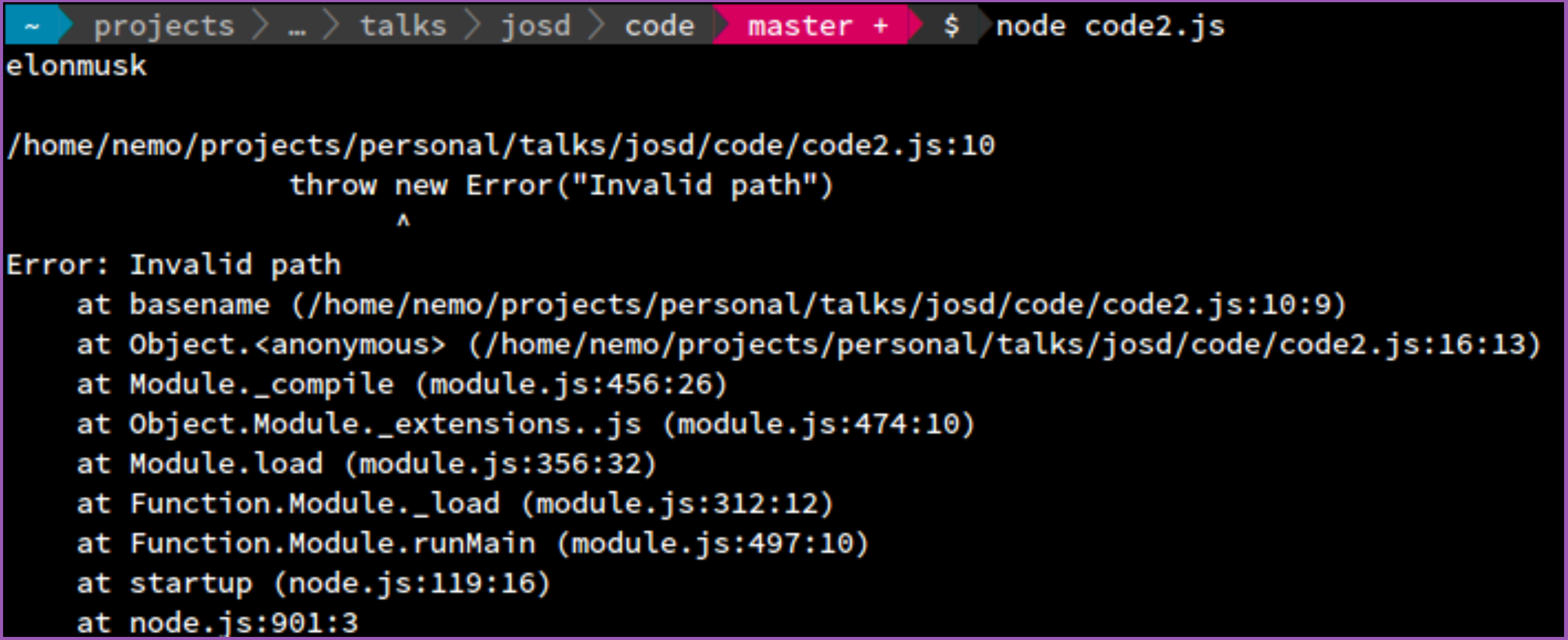
```
function basename(path){
    if(path.subtr(0,7)!=='http://')
        throw new Error("Invalid path")
    var pieces = path.split('/');
    return pieces[pieces.length -1];
}
```

TEST DRIVEN DEVELOPMENT

```
basename('/etc/passwd')
```

?

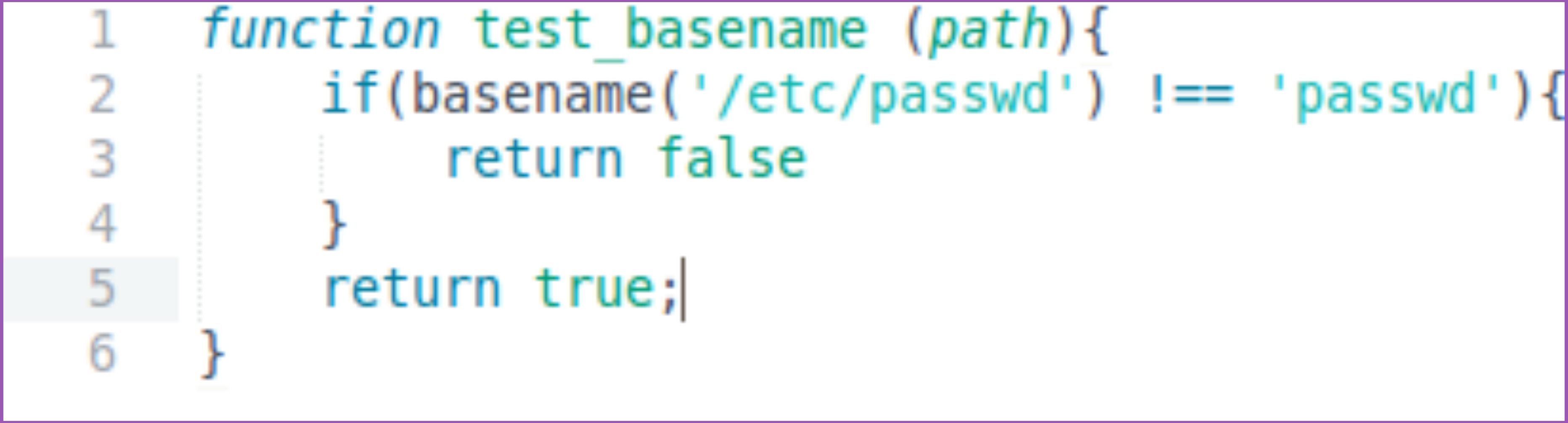
TEST DRIVEN DEVELOPMENT



```
~ 〉 projects 〉 … 〉 talks 〉 josd 〉 code 〉 master + 〉 $ 〉 node code2.js
elonmusk

/home/nemo/projects/personal/talks/josd/code/code2.js:10
                throw new Error("Invalid path")
                      ^
Error: Invalid path
    at basename (/home/nemo/projects/personal/talks/josd/code/code2.js:10:9)
    at Object.<anonymous> (/home/nemo/projects/personal/talks/josd/code/code2.js:16:13)
    at Module._compile (module.js:456:26)
    at Object.Module._extensions..js (module.js:474:10)
    at Module.load (module.js:356:32)
    at Function.Module._load (module.js:312:12)
    at Function.Module.runMain (module.js:497:10)
    at startup (node.js:119:16)
    at node.js:901:3
```

TEST DRIVEN DEVELOPMENT



```
function test_basename (path){
    if(basename('/etc/passwd') !== 'passwd'){
        return false
    }
    return true;
}
```

TEST DRIVEN DEVELOPMENT

- Automated Testing
- TDD
    - Write Tests First
    - Red. Green. Refactor
    - Clean Code
- Regression Testing
- Unit Testing
- Integration Testing

TEST DRIVEN DEVELOPMENT

- Automated Testing (Write tests)
- TDD
  - Write Tests First
  - Red. Green. Refactor
  - Clean Code
- Regression Testing (Catch bugs)
- Unit Testing (Write better code)
- Integration Testing (Test entire code)

REST & APIs

Don’t Reinvent The Wh🚲l

REST & APIs

NIH

REST & APIs

NIH

Not Invented Here

*REST & APIs**

- display a map
- and draw over it
- and drop pins
- and measure distances
- get geolocation data

Use Google Maps

*REST & APIs**

- Real time access to news
- Contextual information for each item
- Figure out patterns in online discussions

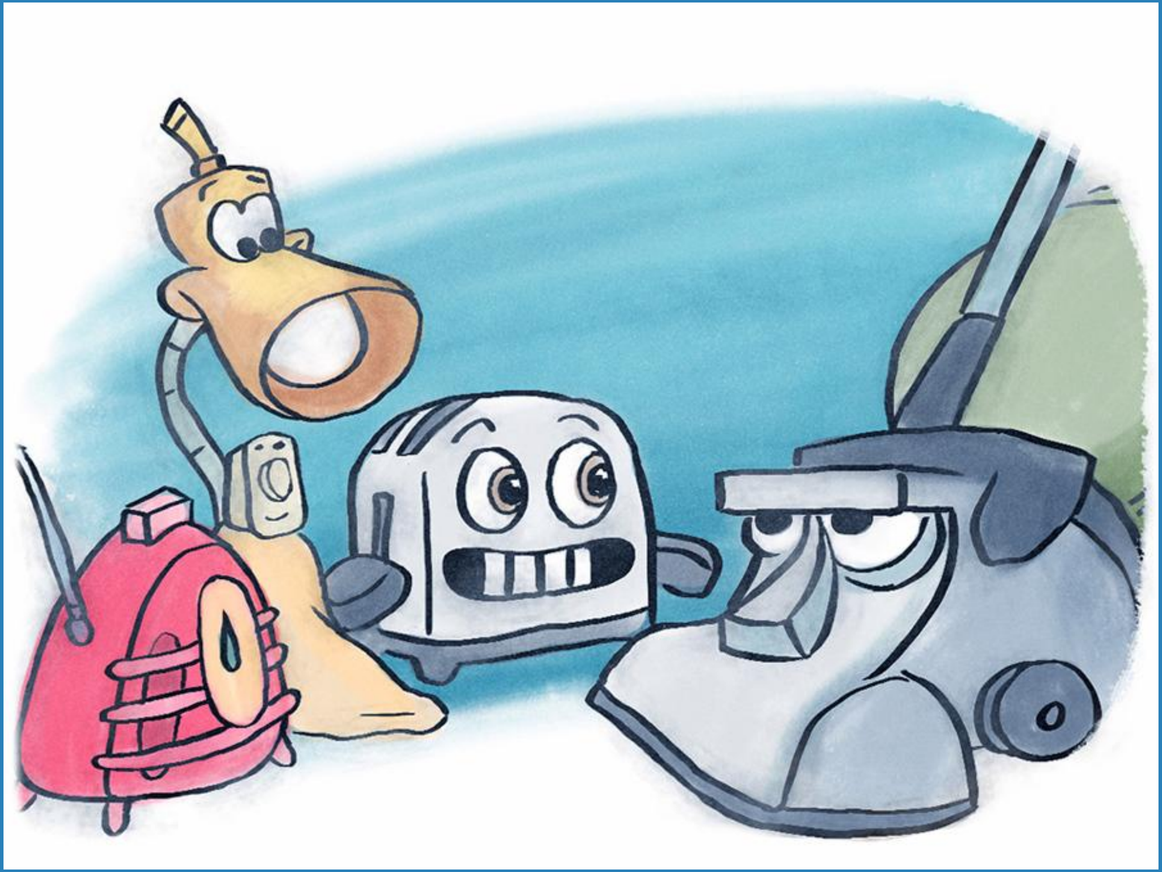
Use Twitter API

*REST & APIs**

- Send mails to thousands of people
- In synchonization
- Within a few minutes
- Without hitting spam or getting flagged

Use SendGrid

QUICK PRIMER ON REST



Inter-machine communication

QUICK PRIMER ON REST

Representational State Transfer

= REST

QUICK PRIMER ON REST

~~Representational State Transfer~~

= REST

QUICK PRIMER ON REST

API over HTTP

(Application-Programming-Interface)

= REST

QUICK PRIMER ON REST

OOP over HTTP

(Object-Oriented-Programming)

= REST

QUICK PRIMER ON REST

REST:

URL = OBJECT

QUICK PRIMER ON REST

/photos/23

This is a PHOTO

*QUICK PRIMER ON REST**

GET	/photos/23	(see the photo)
POST	/photos	(post a photo)
PUT	/photos/23	(edit a photo)
DELETE	/photos/23	(delete a photo)

REST GIVES

- a way for machines to talk
- a technique for building good APIs
- a brilliant solution to a real problem

UNIX PHILOSOPHY

“Read a file of text, determine the n most frequently used words, and print out a sorted list of those words along with their frequencies.”

- Communications of the ACM (1986)

// 5 slides left

UNIX PHILOSOPHY

Donald Knuth

UNIX PHILOSOPHY

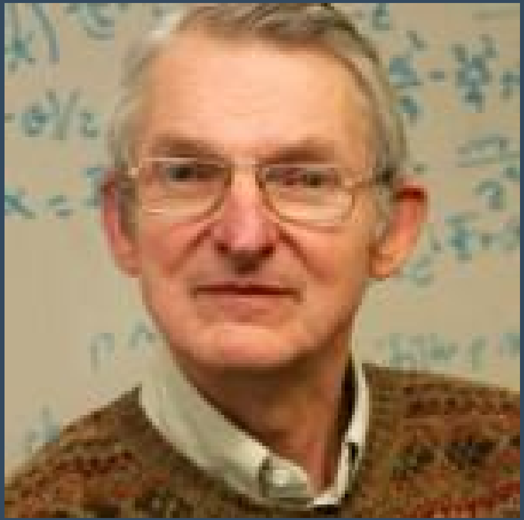
Donald Knuth

- 10 page program
- literate programming
- written in WEB
- based on Pascal
- used custom data structure

In short, as expected from the brilliant scientist

UNIX PHILOSOPHY

Donald Knuth



Doug McIlroy

UNIX PHILOSOPHY

```
tr -cs A-Za-z '\n' |
tr A-Z a-z |
sort |
uniq -c |
sort -rn |
sed ${1}q
```

UNIX PHILOSOPHY

1. **Remove non word characters**
2. **Convert to lower case**
3. **Sort to bring identical words together.**
4. **Remove duplicates and include a count**
5. **Sort in reverse (-r) numeric (-n) order.**
6. **Remember to quit after reading $1 lines**

// last boring slide

UNIX PHILOSOPHY

“Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones which can.”

- Zawinski's Law

UNIX PHILOSOPHY

- Small is beautiful.
- Make each program do one thing well.

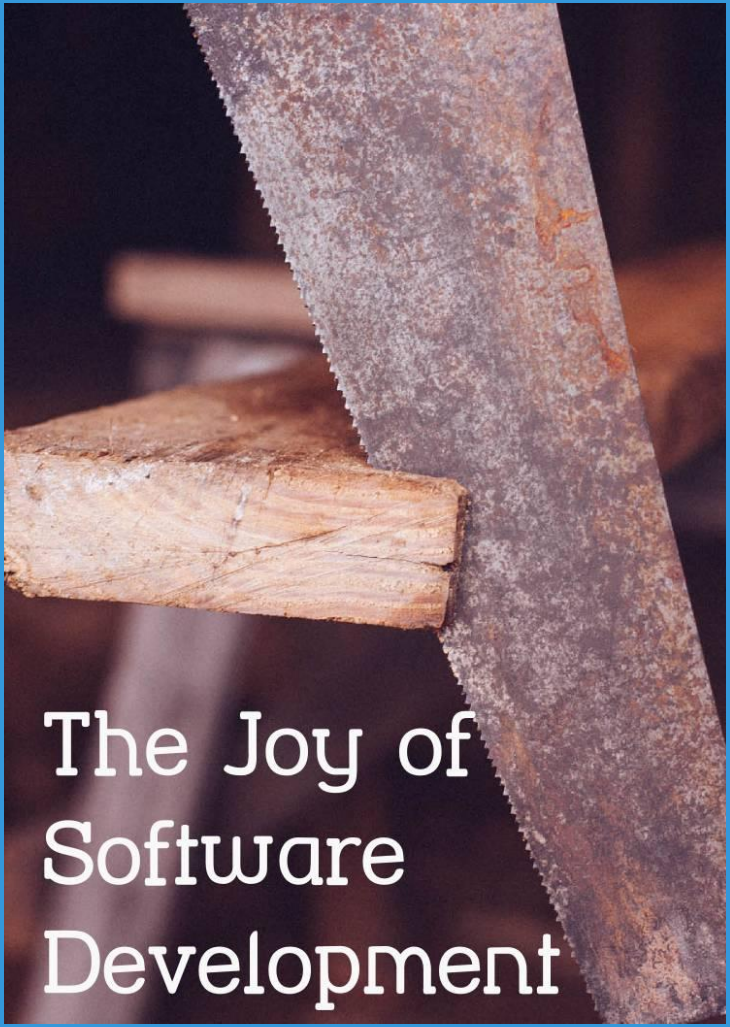
onethingwell.org

How to get better at Software Development?

1. Join a community
2. Contribute to Open Source
3. Write all code publicly
4. Do tech talks
5. Stay updated *
6. Learn more langauges
7. Concepts matter *
8. Ship products
9. Have side projects *
10. Read technical books

THE JOY OF SOFWARE DEVELOPMENT

THINK
BUILD
SHIP



THE JOY OF SOFWARE DEVELOPMENT

Read the book at

josd.captnemo.in

- Creative Commons licensed
- Written on GitHub
- Free to read
- Free to share
- Feedback welcome

Not yet finished.