Hello, Turing!

today we will discuss:

Setup, Git, Tooling
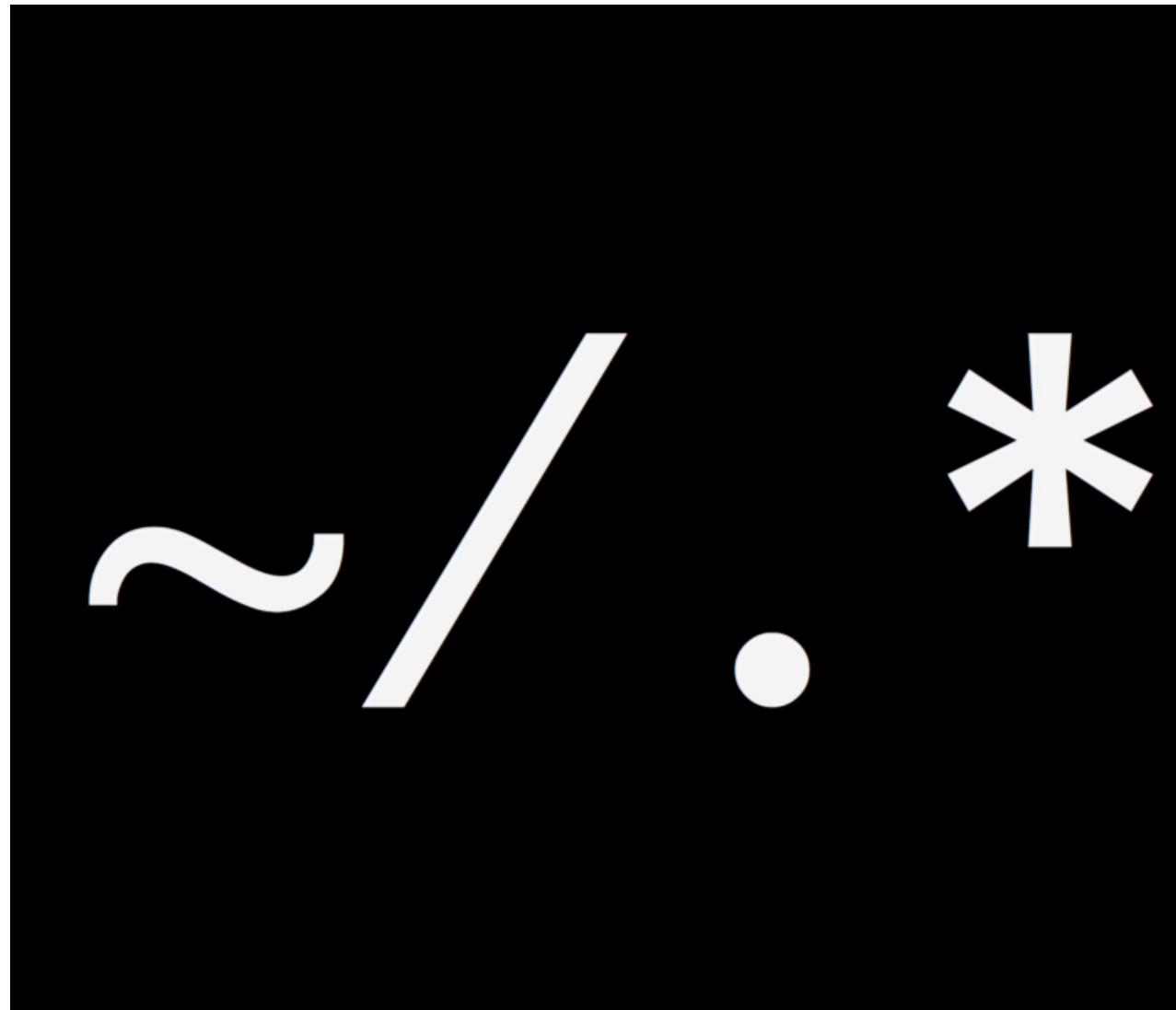
Speakers:

@Aidar Nugmanov

Requirements

- *nix-based OS
- Basic computer literacy

Environment is the key

Shell

wiki: In a Linux system, the _shell_ is a command-line interface that interprets a user's commands and script files, and tells the server's operating system what to do with them

ksh

csh

zsh

bash

sh

Basic shell commands

Linux Bash Shell Cheat Sheet
Basic Commands

Basic Terminal Shortcuts

```
CTRL L = Clear the terminal
CTRL D = Logout
SHIFT Page Up/Down = Go up/down the terminal
CTRL A = Cursor to start of line
CTRL E = Cursor the end of line
CTRL U = Delete left of the cursor
CTRL K = Delete right of the cursor
CTRL W = Delete word on the left
CTRL Y = Paste (after CTRL U,K or W)
TAB = auto completion of file or command
CTRL R = reverse search history
!! = repeat last command
CTRL Z = stops the current command (resume with fg in foreground or bg in background)
```

Basic Terminal Navigation

```
ls -a = list all files and folders
ls <folderName> = list files in folder
ls -lh = Detailed list, Human readable
ls -l *.jpg = list jpeg files only
ls -lh <fileName> = Result for file only

cd <folderName> = change directory
      if folder name has spaces use " "
cd / = go to root
cd .. = go up one folder, tip: ../../../

du -h: Disk usage of folders, human readable
du -ah: "      "   " files & folders, Human readable
du -sh: only show disc usage of folders

pwd = print working directory

man <command> = shows manual (RTFM)
```

Basic file manipulation

```
cat <fileName> = show content of file
               (less, more)
head = from the top
       -n <#oflines> <fileName>

tail = from the bottom
       -n <#oflines> <fileName>

mkdir = create new folder
mkdir myStuff ..
mkdir myStuff/pictures/ ..

cp image.jpg newimage.jpg = copy and rename a file
cp image.jpg <folderName>/ = copy to folder
cp image.jpg folder/sameImageNewName.jpg
cp -R stuff otherStuff = copy and rename a folder
cp *.txt stuff/ = copy all of *<file type> to folder

mv file.txt Documents/ = move file to a folder
mv <folderName> <folderName2> = move folder in folder
mv filename.txt filename2.txt = rename file
mv <fileName> stuff/newfileName
mv <folderName>/ .. = move folder up in hierarchy

rm <fileName> .. = delete file (s)
rm -i <fileName> .. = ask for confirmation each file
rm -f <fileName> = force deletion of a file
rm -r <foldername>/ = delete folder

touch <fileName> = create or update a file

ln file1 file2 = physical link
ln -s file1 file2 = symbolic link
```
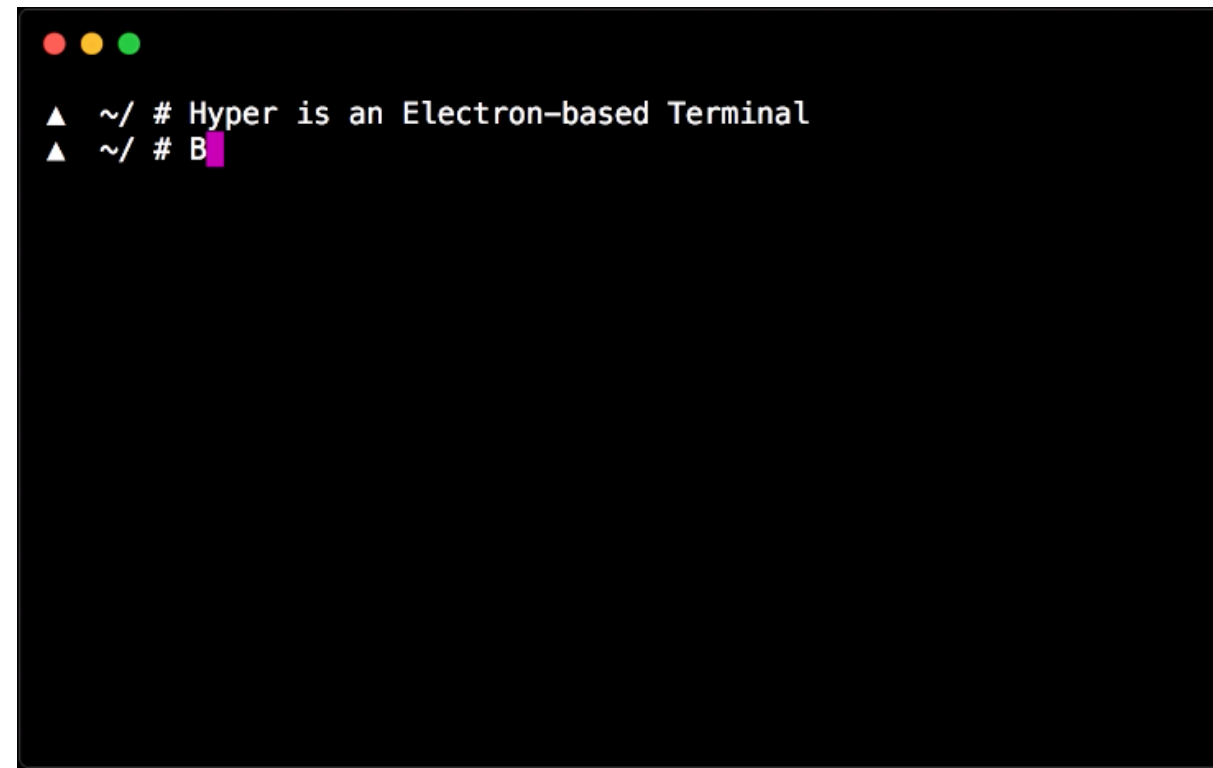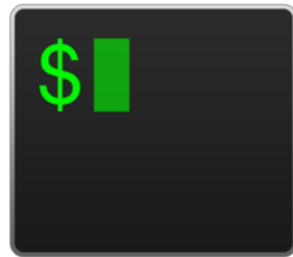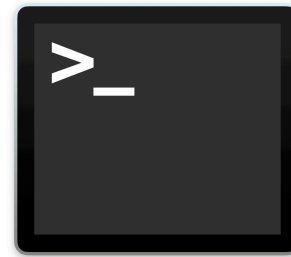
Terminal

wiki: A terminal emulator is a program that allows the use of the terminal in a graphical environment.

Types of terminals

Mac OS

iTerm 2

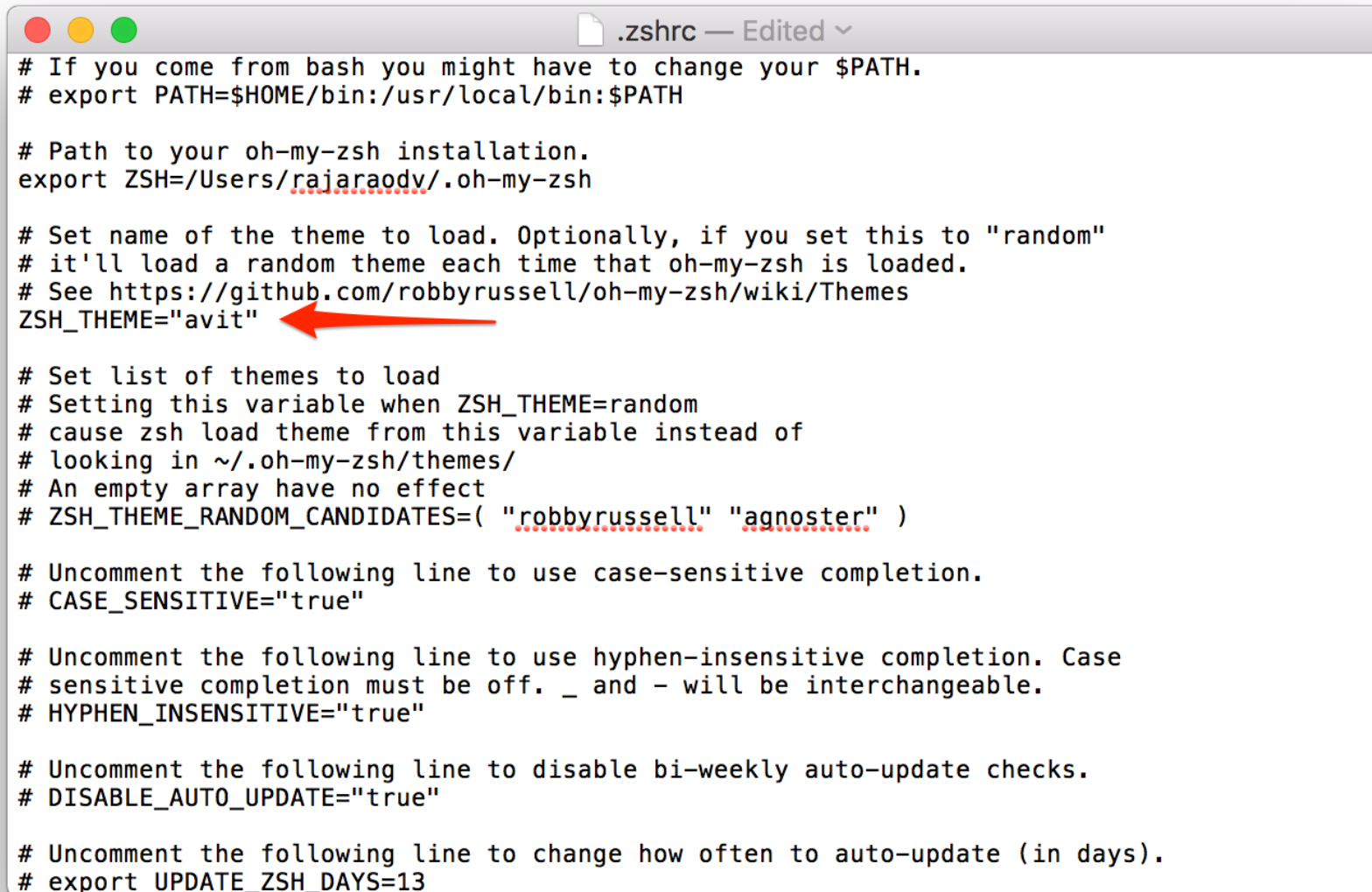**Terminal**

Linux

Guake

Terminator

```
                        📄 .zshrc — Edited ⌄

# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:/usr/local/bin:$PATH

# Path to your oh-my-zsh installation.
export ZSH=/Users/rajaraodv/.oh-my-zsh

# Set name of the theme to load. Optionally, if you set this to "random"
# it'll load a random theme each time that oh-my-zsh is loaded.
# See https://github.com/robbyrussell/oh-my-zsh/wiki/Themes
ZSH_THEME="avit"    ⬅

# Set list of themes to load
# Setting this variable when ZSH_THEME=random
# cause zsh load theme from this variable instead of
# looking in ~/.oh-my-zsh/themes/
# An empty array have no effect
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.
# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion. Case
# sensitive completion must be off. _ and - will be interchangeable.
# HYPHEN_INSENSITIVE="true"

# Uncomment the following line to disable bi-weekly auto-update checks.
# DISABLE_AUTO_UPDATE="true"

# Uncomment the following line to change how often to auto-update (in days).
# export UPDATE_ZSH_DAYS=13
```

```
Last login: Thu Jan 12 14:41:11 on ttys001
~      nocmd
zsh: command not found: nocmd
✗  ~      mkdir test
~      cd test
~/test     git init
Initialized empty Git repository in /Users/kevin/test/.git/
~/test    master    touch .dirty
~/test    master    git checkout -b feature/doink
Switched to a new branch 'feature/doink'
~/test    feature/doink    git checkout -b feature/doink
```

JavaScript Python Ruby ⑂ master

✔ colorls

📄 Brewfile   CODE_OF_CONDUCT.md   CONTRIBUTING.md
💎 Gemfile   💎 Gemfile.lock   📄 MIT-LICENSE
📄 RAILS_VERSION   README.md   RELEASING_RAILS.md
💎 Rakefile   📂 actioncable/   📂 actionmailer/
📂 actionpack/   📂 actionview/   📂 activejob/
📂 activemodel/   📂 activerecord/   📂 activestorage/
📂 activesupport/   📂 ci/   📂 guides/
💎 rails.gemspec   📂 railties/   📂 tasks/
📂 tools/   💎 version.rb

JavaScript Python Ruby ⑂ master

✔

Dotfiles

wiki: In UNIX file systems, *dotfiles* are text-based configurations files that stores settings of almost every application, service and tool running on your system.

Editors/IDEs

A *text editor* is just for writing/modifying text/code

vs.

With an *IDE*, you should do a lot more within that single program: running, debugging, version control, etc.

Vim Emacs Atom Sublime Text

JetBrains products VSCode Eclipse NetBeans

Package managers

Homebrew

Ruby Gems

NPM

wiki: A *package manager* is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer's operating system in a consistent manner.

Tooling Recap

Before you start coding (Information, Opinion, Guidelines)

Draw on paper
- Paper
- Pencil
- Eraser
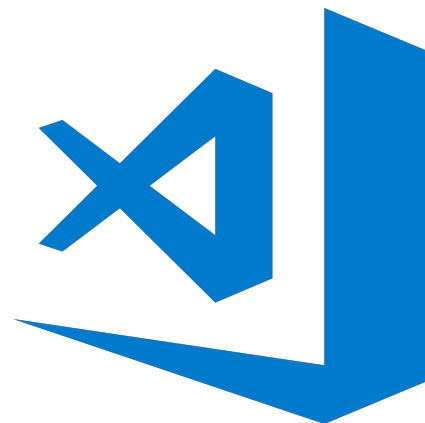- Color markers
- Highlighters

Reuse and open source components

Licenses:	Package management systems:		
- Apache	- RPM	- CPAN	- CTAN
- BSD	- Debian packages	- MacPorts	- MiKTeX
- GPL	- Maven	- Pkgsrc	- Cygwin
- LGPL	- Portage	- CRAN	
- MIT	- FreeBSD Ports	- NuGet	

Easy parallelism

Language-based:	Unix tools:	Libraries:
- OpenMP - Clojure	- Shell pipelines	- ATLAS
- Scala - R/parallel	- GNU parallel	- Frameware
- Haskell - Erlang	- xargs -P	- Intel IPP
- F#	- sgsh	- NAG Library for SMP
	- make -j	- QtConcurrent

XML
- XMLStarlet
- xgawk
- xsltproc

Declarative drawing

Graphviz	- pic	- netpbm	- R Project:
. dot	- gnuplot	- PGF/TikZ	. ggplot2
. neato	- GMT	- ImageMagick	. graphics
. circo	- UMLGraph	- D3.js	
. twopi	- Inkscape	- processing.org	

Tools you want to be aware of

Unix tools

- ar	- grep	- strings
- awk	- head	- tail
- comm	- jar	- tar
- cut	- javap	- uniq
- dd	- jot	- wc
- diff	- less	- wget
- dumpbin	- more	- xargs
- fgrep	- nm	- yes
- find	- sed	- Cygwin
- fmt	- sort	- Outwit

Cloud-based tools (one from each category)

Version control:	Localization:	Email:	Issue tracking, project management, & collaboration:	Collaborative editing:	Payment processing:	Continuous integration:	Bulk email campaigns:
- Bitbucket	- Pootle	- Gmail	-Asana	- Etherpad	- Braintree	- Bamboo	- CampaignMonitor
- GitHub	- Transifex	- Outlook.com	-Basecamp	- Google Docs	- Chargify	- Cloudbees	- Mailchimp
Remote application	- WebTranslateIt	- Yahoo! Mail	-FogBugz	- Stypi	- PayPal	- Codeship	- mailgun
monitoring:	**Real-time communication:**	**Wiki work:**	-GitHub	- Microsoft Office 365	- Stripe	**Deployment servers:**	- Sendgrid
-BugSense	- Google Hangouts	- Confluence	-JIRA	**File sharing:**	**Store fronts:**	- AWS	**A/B testing:**
-Crittercism	- Grove.io	- Bitbucket	-Pivotal Tracker	- Box	- Google Play Store	- Google App Engine	- Optimizely
-Exceptional	- HipChat	- GitHub	-Redmine	- DropBox	- Mac App Store	- Heroku	**IT support:**
-New Relic	- Skype	**Programming:**	-Trello	- Google Drive	- Windows Store	- Microsoft Azure	- Desk.com
-Sentry	- sqwiggle	- Cloud9	-YouTrack	- SkyDrive		- Nodejitsu	- Helpscout
		- VS Cloud				- Rackspace	- Zendesk

Writing code and building it

Effective code editing
- vim
- Emacs
- Your IDE

Efficient use of the Web
- Stack Exchange (Stack Overflow)
- Google
- Delicious
- Wikipedia

How to understand code

- Eclipse Java Browsing Perspective	- Source Insight	- Tracing tools
- Understand C++	- UMLGraph	- Profilers
- etags	- Debugger	- Printouts
- Rational Rose	- Enterprise Architect	

How to write systems software
- select
- poll
- readv / writev
- mmap
- Logging

Scripting languages
- JavaScript
- Perl
- Python
- Ruby

Build tools
- Ant
- make
- Maven

Continuous integration
- Jenkins
- TeamCity
- Bamboo
- CloudBees
- Travis CI

Debugging & improving your code

Add logging statements
- syslog
- util.logging
- log4j
- Boost.Log
- print

Perform static analysis

- CheckStyle	- FindBugs	- ITS4	- Pylint
- Clang	- Flawfinder	- JLint	- RATS
- Coverity Code Advisor	- Frama-C	- Lint4J	- Rational AppScan
- devAdvantage	- FxCop	- PC-Lint	- Splint
- ESC/Java2	- HP Fortify	- PMD	
- Eclipse Codan	- Intel Inspector	- PVS-Studio	

Perform code and design reviews
- Crucible
- Differential (Phabricator)
- Gerrit
- Kiln
- Review Board
- Rietveld

Use tracing tools

- APIS32	- ktrace	- JConsole
- Application Verifier	- ltrace	- truss
- DTrace	- NProf	- valgrind
- EJP	- strace	- Wireshark
- gcov	- SystemTap	
- gprof	- tcpdump	
	- TracePlus	

Compare differences
- md5sum
- dumpbin
- nm
- md5sum
- ldd
- diff
- git bisect

Refactoring on the cheap
- Edit - replace
- sed -i
- find | sed | sh

Operations

Virtualization

- Vagrant	- Xen	- AWS EC2
- Docker	- KVM	
- VMware	- QEMU	
- VirtualBox	- SUSE Studio	

System configuration management tools

- Ansible	- Puppet
- Boxen	- Salt
- Chef	
- CFEngine	

Service orchestration

- Capistrano	- MCollective
- Fabric	- Rundeck
- Func	
- Knife	

Working with others

Version control systems

- BitKeeper	- Darcs	- Subversion
- ClearCase	- Mercurial	- Vault
- Git	- Monotone	- Visual SourceSafe
- Bazaar	- Perforce	

GitHub Student Pack

Open-source philosophy

Approaching all aspects of life **"the open source way"** means expressing a willingness to share, collaborating with others in ways that are transparent (so that others can watch and join too), embracing failure as a means of improving, and expecting—even encouraging—everyone else to do the same.
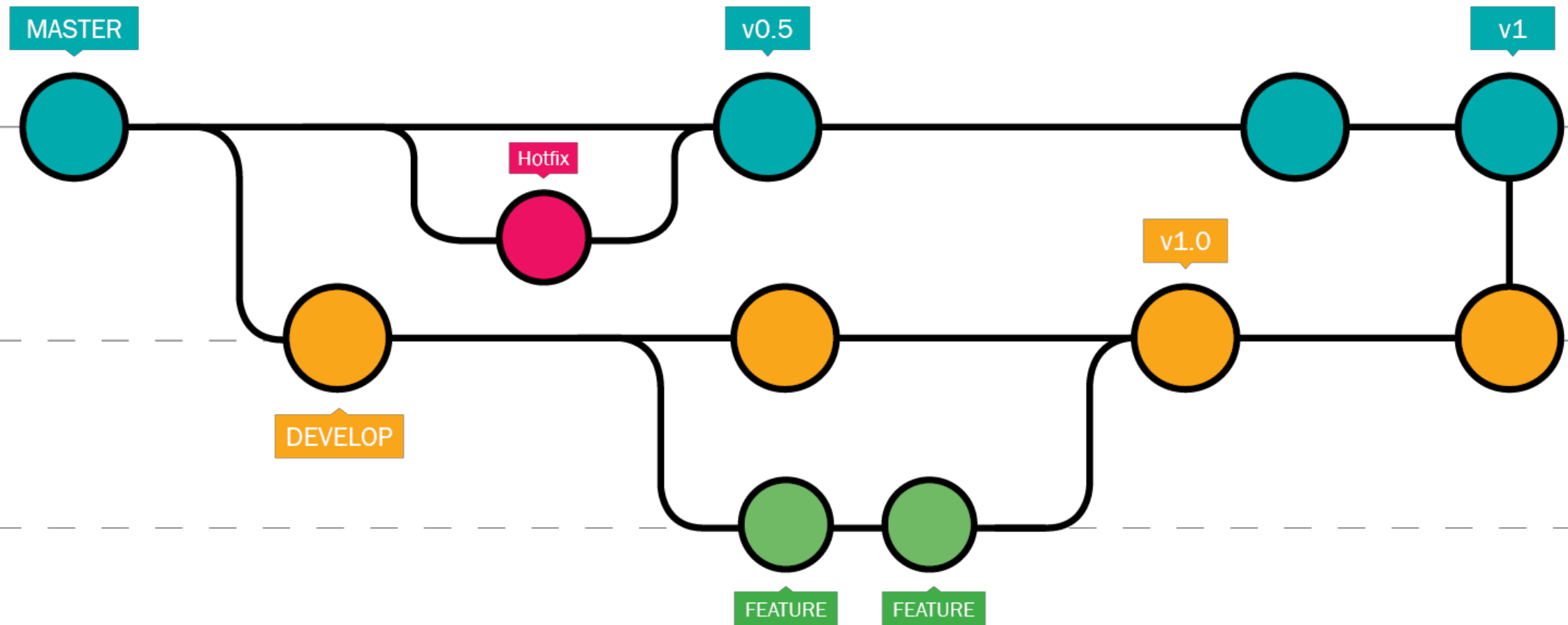
Git (Version Control System)

GitHub is not a **git**!

Git (Version Control System)

The core function of **git**: To save checkpoints (revisions) and share them with other people. Everything revolves around this concept.

Git (Version Control System)

Git commits conventions

|	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

Git (Version Control System)

Demo

Resources

Link!
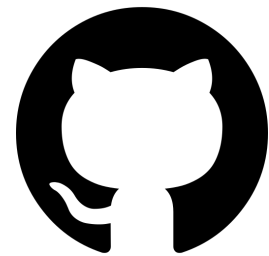
/nugmanoff

/nugmanoff

Thank you, Turing! 👋