Tools Seminar

Week 1 - Basic Configuration

Hongzheng Chen

Nov 15, 2019

1

Text Editors

IDE

IDE (Integrated Development Environment)

- Visual Studio (VS)
- PyCharm
- Intellij, Eclipse
- DevC++
- ...

Text Editor

✗ IDE: Too heavy

✓ Text Editor: Light-weight

- Visual Studio Code: Highly recommended!
- Sublime Text: Extreme light-weight
- Notepad++: Maybe popular years ago
- Vim: Linux
- XCode: Apple
- ...

Text Editor

✗ IDE: Too heavy

✓ Text Editor: Light-weight

- Visual Studio Code: Highly recommended!
- Sublime Text: Extreme light-weight
- Notepad++: Maybe popular years ago
- Vim: Linux
- XCode: Apple
- ...

No definite winner!

The one most suitable to you is the best!

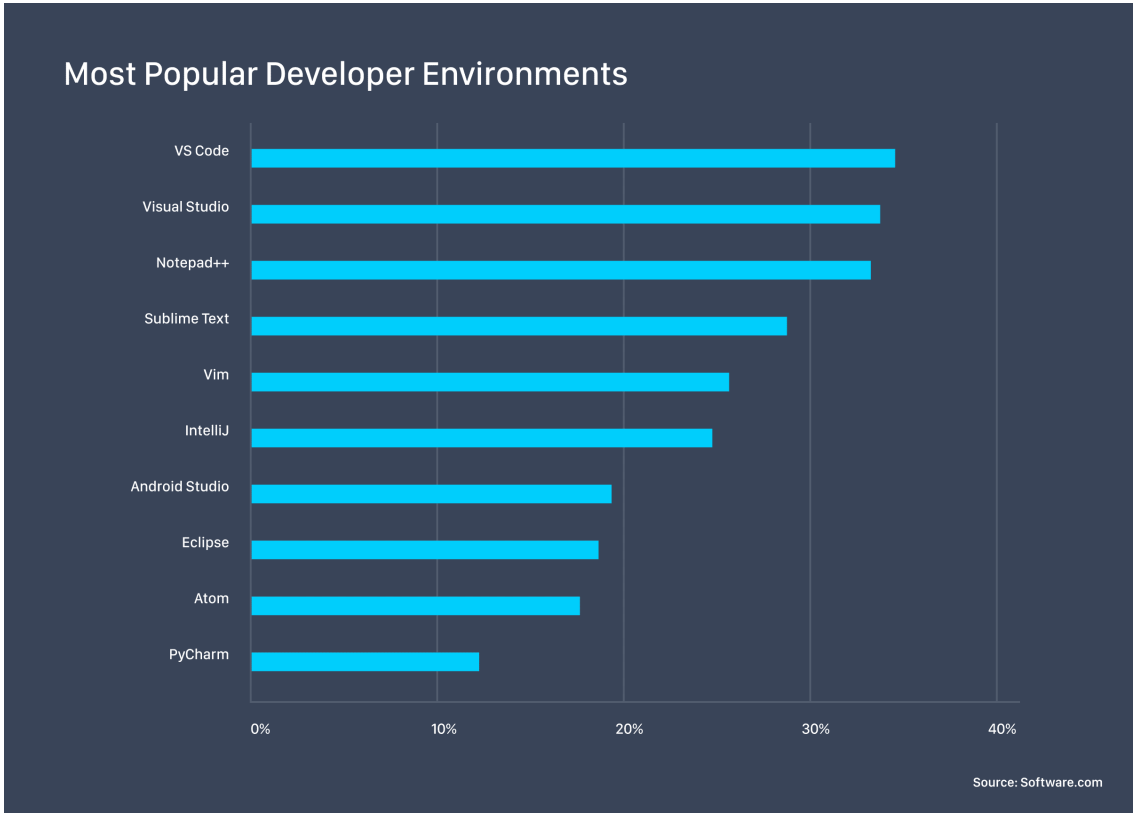
VS Code

Why VS Code? Pro:

- Very popular nowadays (first ver in mid-2015) with fast-growing & active open-source society (Microsoft)
- Integrated terminal, debugger, source control (git)
- Build-in intelligent code-completion tool, IntelliSense
- Various extensions to different languages & cross-platform
- Web version since v1.40 (Oct 2019)

Con: Electron-based app, a bit slower than Sublime Text

VS Code



“Biggest ecosystem means more developers adding to value over time!”

VS Code

- Official site: https://code.visualstudio.com/
- Keyboard shortcut: https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf
- New features in Chinese: https://zhuanlan.zhihu.com/vs-code

Some features need to know

- Just open a folder w/o configuration
- Interface overview
- Interactive editor playground
- Tips and tricks

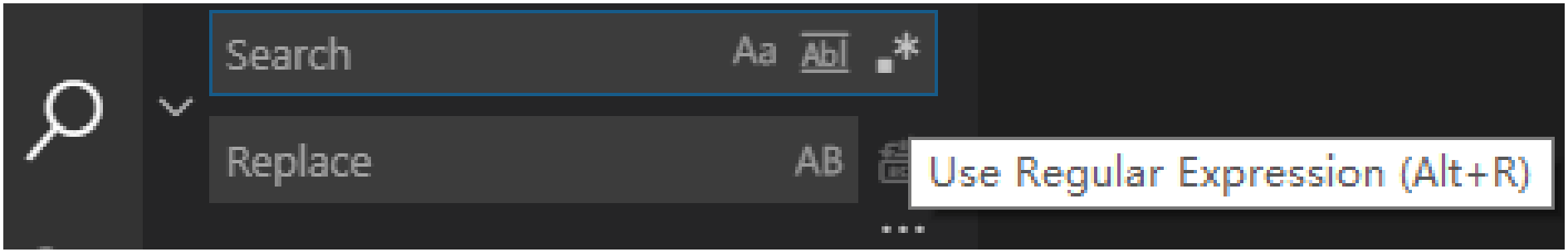
Regular Expression (regex)

What is it?

A sequence of characters that define a search pattern used to find/replace sth. in strings.

- Tutorial: https://github.com/ziishaned/learn-regex
- Online test: https://regex101.com

VS Code × Regex



2

Environment Setting

Use Unix-Like Environment!!!

Why Linux?

- Broken development tool chains on Windows
- Most of the packages are developed on Linux w/ little support to Windows
- All the severs use Linux environments
- * Exam/Competition system

Use Unix-Like Environment!!!

Why Linux?

- Broken development tool chains on Windows
- Most of the packages are developed on Linux w/ little support to Windows
- All the severs use Linux environments
- * Exam/Competition system

Thus,

Get familiar with Linux!

Several Approaches

- Buy an Apple computer with macOS!

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system
  - Time-consuming with backups to be made

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system
  - Time-consuming with backups to be made
- Set up a dual system

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system
  - Time-consuming with backups to be made
- Set up a dual system
  - Unstable

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system
  - Time-consuming with backups to be made
- Set up a dual system
  - Unstable
- Use a terminal emulator

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system
  - Time-consuming with backups to be made
- Set up a dual system
  - Unstable
- Use a terminal emulator
  - MSYS2, Cygwin

Several Approaches

- Buy an Apple computer with macOS!
  - Expensive
- Switch to a Linux system
  - Time-consuming with backups to be made
- Set up a dual system
  - Unstable
- Use a terminal emulator
  - MSYS2, Cygwin
  - Not fully functional

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming
  - ✓, full system support with GUI & portable

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming
  - ✓, full system support with GUI & portable
- Use SSH to login in a remote server

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming
  - ✓, full system support with GUI & portable
- Use SSH to login in a remote server
  - AWS, Azure, Google Cloud, Aliyun, Tencent Cloud

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming
  - ✓, full system support with GUI & portable
- Use SSH to login in a remote server
  - AWS, Azure, Google Cloud, Aliyun, Tencent Cloud
  - ✓, we will provide accounts of Lab servers later

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming
  - ✓, full system support with GUI & portable
- Use SSH to login in a remote server
  - AWS, Azure, Google Cloud, Aliyun, Tencent Cloud
  - ✓, we will provide accounts of Lab servers later
- Windows Subsystem for Linux (WSL)

Several Approaches (Cont.)

- Install a virtual machine and use graphical user interface (GUI)
  - VirtualBox, VMWare
  - Very slow and memory-consuming
  - ✓, full system support with GUI & portable
- Use SSH to login in a remote server
  - AWS, Azure, Google Cloud, Aliyun, Tencent Cloud
  - ✓, we will provide accounts of Lab servers later
- Windows Subsystem for Linux (WSL)
  - ✓✓, easy to configure & act like a real Linux!

No GUI, Use Terminal!

Be familiar with terminal & get rid of IDE!

- Many packages have no GUI!
- Most of industrial server have no GUI
- GUI is very slow
- When you write huge projects, you have to configure all the settings & dependency in Makefile and bash, which also has no GUI support.

A Tradeoff . . .

Code on Windows w/ GUI, run on Linux w/o GUI

Windows Subsystem for Linux (WSL)

Windows Subsystem for Linux (WSL)

Why WSL?

- Very light-weight with most of Linux featues
- Efficient to have Windows & Linux running together
- WSL 2 will be released in 2020 ($>$ Win 10 Build 18917)

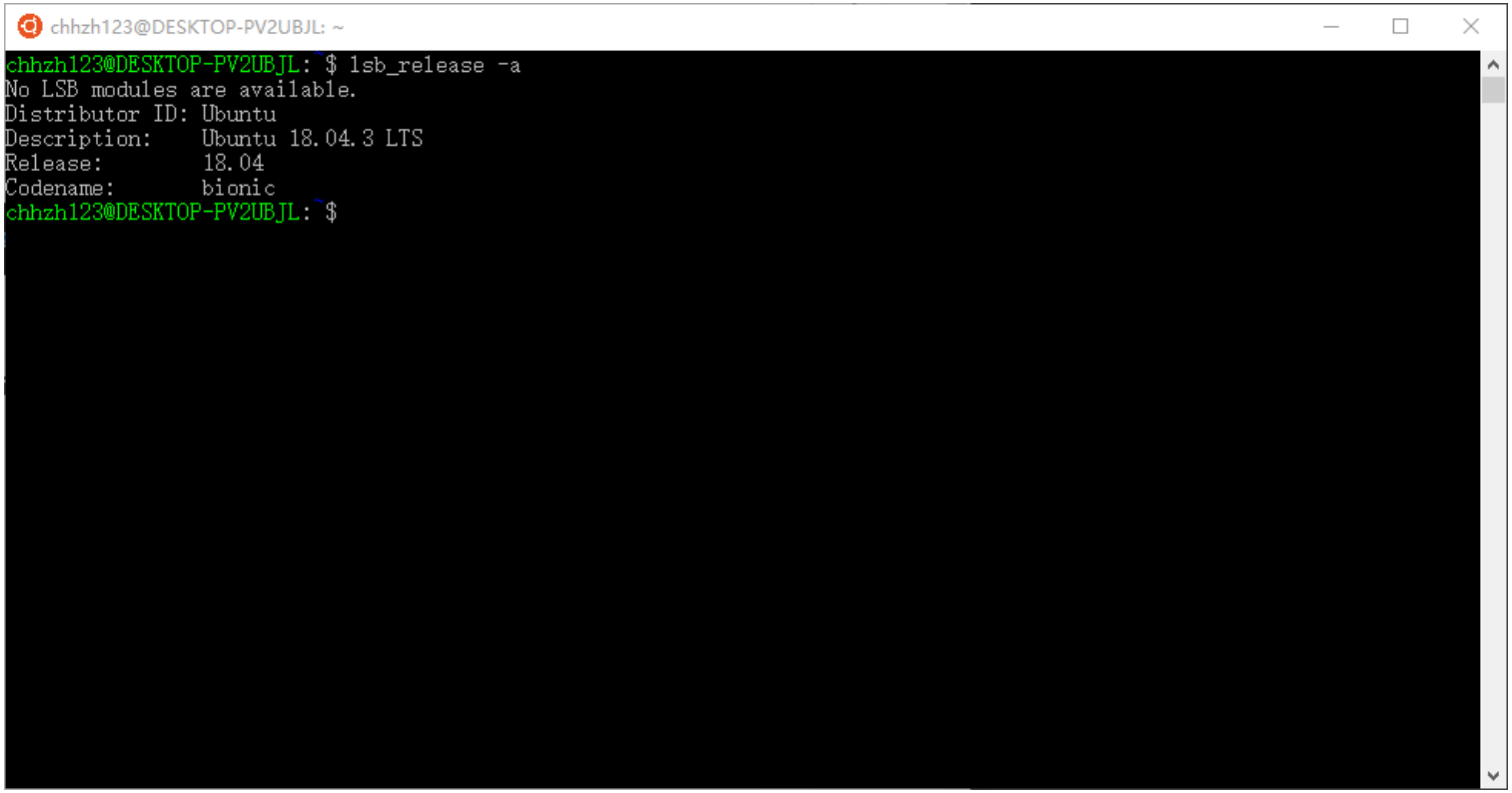
Known issues: no support to perf & GPU

Begin with WSL

Tutorial:
https://docs.microsoft.com/en-us/windows/wsl/install-win10

- Ubuntu LTS 18.04: Most commonly used (recommended!)
- Debian, CentOS: Enterprise

Ubuntu LTS 18.04



Some difference between Unix & Windows:

- Line ending: `\n` in Linux, `\n\r` in Windows
- File path: `/` in Linux, `\\` in Windows
- WSL file system: `/mnt/c`

Linux Command Lines (bash)

Chinese tutorial: https://linux.cn/article-6160-1.html

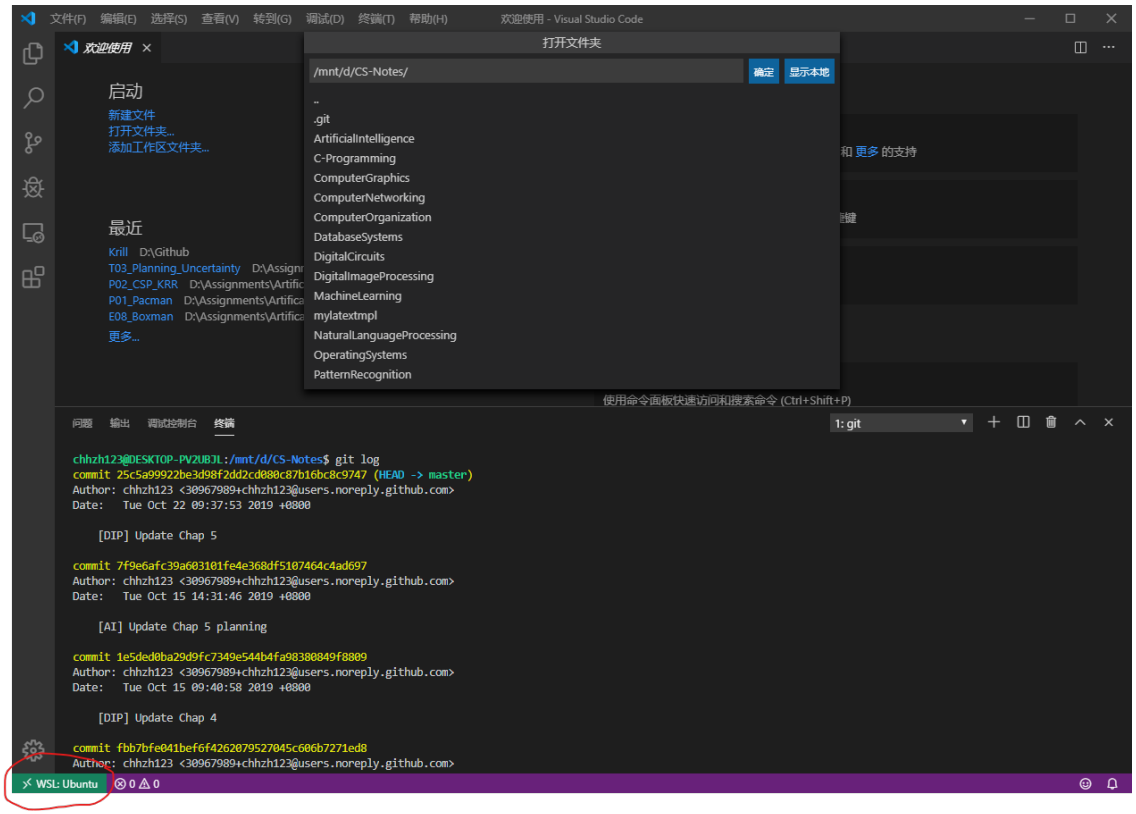
- `cd` (w/ TAB usage)
- `ls`
- `mkdir`
- `rm`, `cp`, `mv`
- `apt-get`
- `wget`
- `vim`

Please get familar & google for help.

VS Code × WSL

Install "Remote - WSL" in VS Code Extension

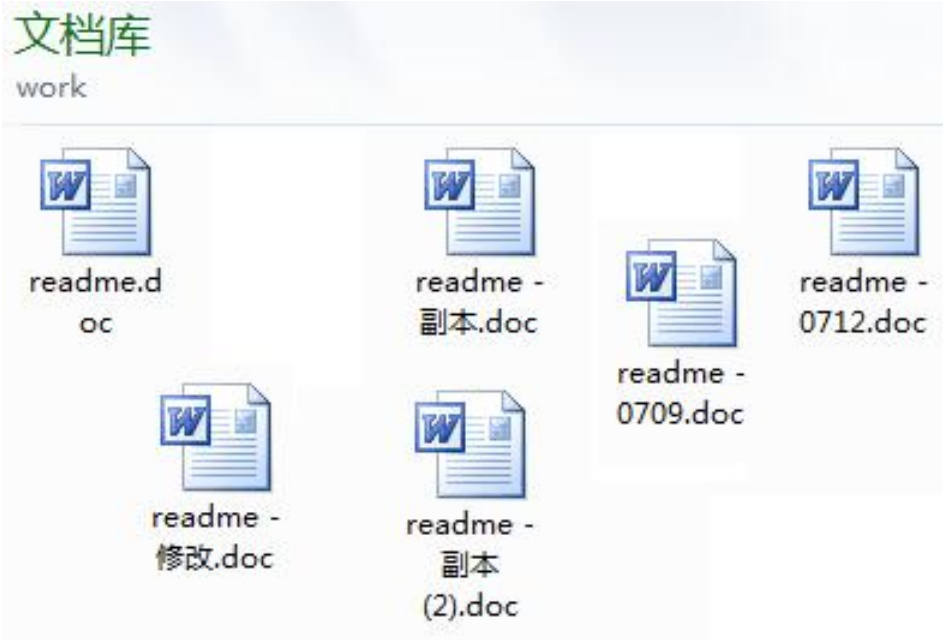
- Workplace & file path will change
- Full WSL support



3

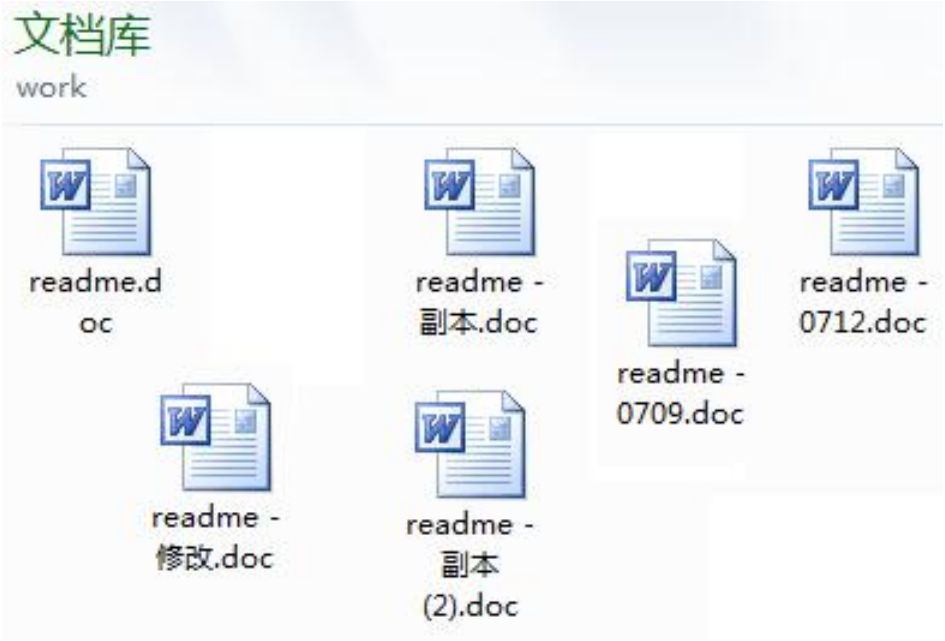
Code Management

Git & Github



Version control (pic via Xuefeng Liao)

Git & Github



Version control (pic via Xuefeng Liao)

NO!!!

Git & Github

Then we have Git!

Git is a distributed version control system

Some history:

- Linus created open-source Linux in 1991

Git & Github

Then we have Git!

Git is a distributed version control system

Some history:

- Linus created open-source Linux in 1991
- Before 2002, Linus merged codes from all over the world

Git & Github

Then we have Git!

Git is a distributed version control system

Some history:

- Linus created open-source Linux in 1991
- Before 2002, Linus merged codes from all over the world
- 2002-2005, used BitKeeper

Git & Github

Then we have Git!

Git is a distributed version control system

Some history:

- Linus created open-source Linux in 1991
- Before 2002, Linus merged codes from all over the world
- 2002-2005, used BitKeeper
- In 2005, Linus coded git in C in one month!

Git & Github

Then we have Git!

Git is a distributed version control system

Some history:

- Linus created open-source Linux in 1991
- Before 2002, Linus merged codes from all over the world
- 2002-2005, used BitKeeper
- In 2005, Linus coded git in C in one month!
- In 2008, Github was created and became the most popular code management platform

Basic concepts of Git

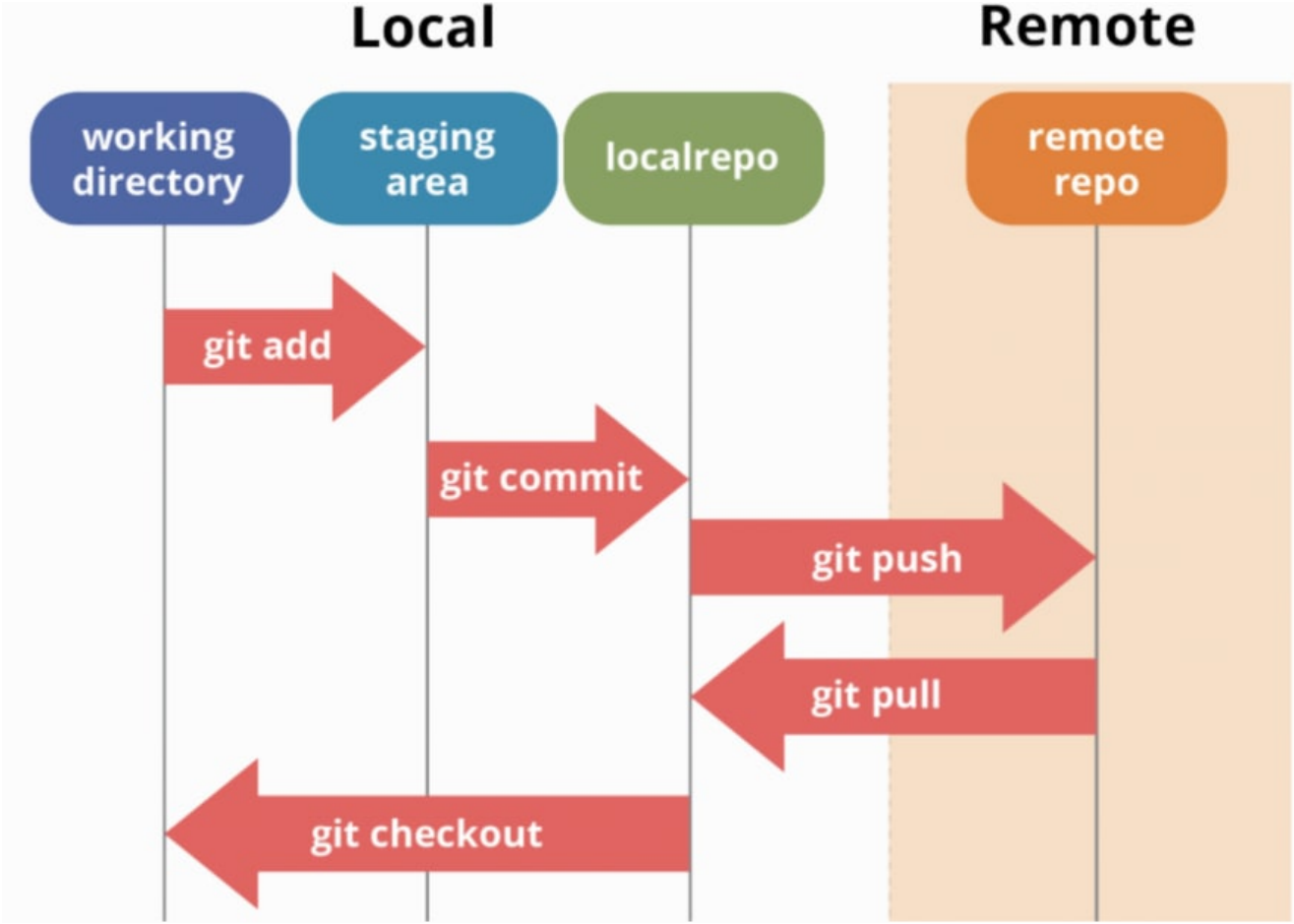
Git is originally provided in Linux!

Tutorial: https://www.liaoxuefeng.com/wiki/896043488029600

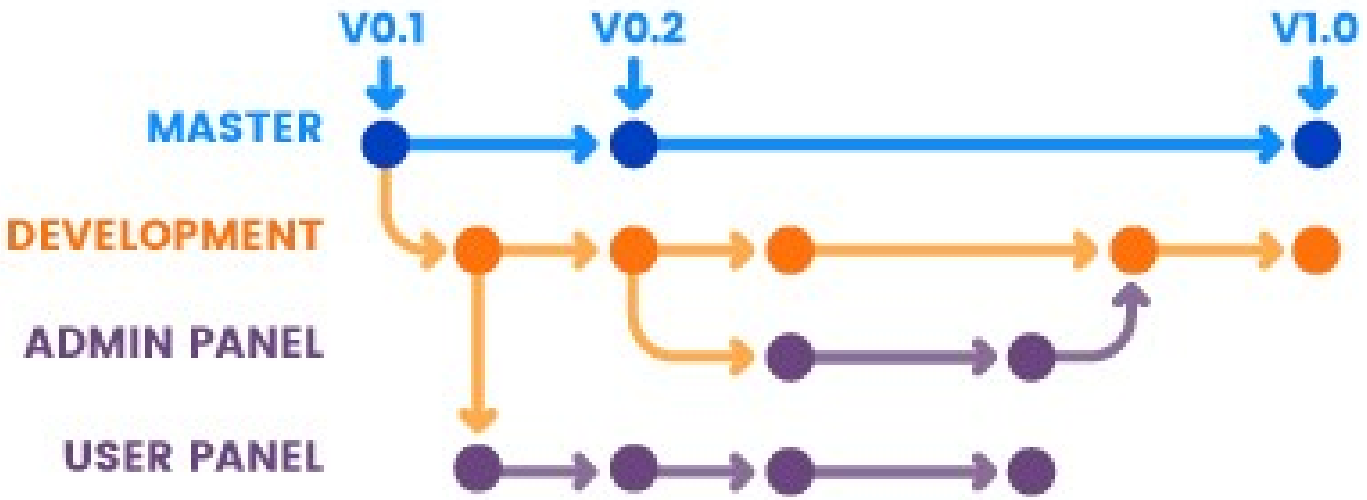
- Your repo

- Others' repo: `git clone`, `git pull`

Basic concepts of Git



Basic concepts of Git



* Pic via https://rubygarage.org/blog/most-basic-git-commands-with-examples

Basic concepts of Git

- `.gitignore`: ignore some files in your repository
  - Generated/Intermediate files: `.exe`
  - Sensitive infomation
- Submodule

Commit message

A template:
https://blog.coding.net/blog/commit_message_change_log

- Be specific about what you have done in this commit
- DONT: `1st commit`, `Commit`, `Message`
- DO: `[DOC] fix typo in README.md`

VS Code × Git

Make sure your computer have Git installed

Github

The world's leading software development platform
(The biggest social networking platform (?))

Start from Github

Individuals

Free

$0

Per month

The basics of GitHub for every developer

- Unlimited public repositories
- Unlimited private repositories
- 3 collaborators for private repositories
- Issues and bug tracking
- Project management

Pro

$7

Per month

Pro tools for developers with advanced requirements

- Unlimited public repositories
- Unlimited private repositories
- Unlimited collaborators
- Issues and bug tracking
- Project management
- Advanced tools and insights

Included free alongside other real-world

Basic concepts of Github

Some demo here ...

- Repositories
- Projects:
  - Watch, Star, Fork
  - Issues, pull requests, merge requests

README Page

Left to next course

Licenses

Commonly used licenses: https://choosealicense.com/licenses/

- When you use others' code, please view licenses
- The most restrictive licenses is GPL
- The most permissive licenses is MIT
- Other popular licenses are Apache License 2.0 and BSD
- Your daily-life project needs not add licenses

4

Code Specification

Style notes

Key: **Style in one way!**

- Variable naming convention
  - Camel case: `newString`
  - Pascal case: `NewString`
  - Underline: `new_string`
- Empty space / newlines are needed
  - `for (i = 0; i < 10; ++i)`
  - Newline bracket or not depends on yourself
  - Use space & empty lines to separate different code sections
- Please be specific (or you will not pass vmatrix)
  - DO: `wordLen`, `drawImage()`
  - DONT: `a`, `aaaaa`, `method1`
- **Add comments** unless what your code does is evident
- Please **use English** (utf-8) all the way!

Google Style Guide

Google Style Guide: http://google.github.io/styleguide/

- Need not be rigorous, but please keep ONE format in one project
- Using formatting in VS Code is enough

5

Summary

Summary of Week 1 - Basic Configuration

- Text Editors: VS Code/Vim, regular expression
- Environment: Linux (WSL), bash
- Code management: git, Github
- Code spec: Google Style Guide

Assignments

Clone this seminar via

```
git clone --recursive https://github.com/chhzh123/ToolsSeminar-CS.git
```

1. Be familiar with Text editors which you may use every day
2. Have a Linux machine & learn basic bash commands
3. Register a Github account and take a look of Github
4. Finish assignment on easy **string matching**
5. Create a repository to store assignments of this seminar
6. Upload your assignments to Github

Details can be found in `Materials-BasicConfiguration.pdf`.