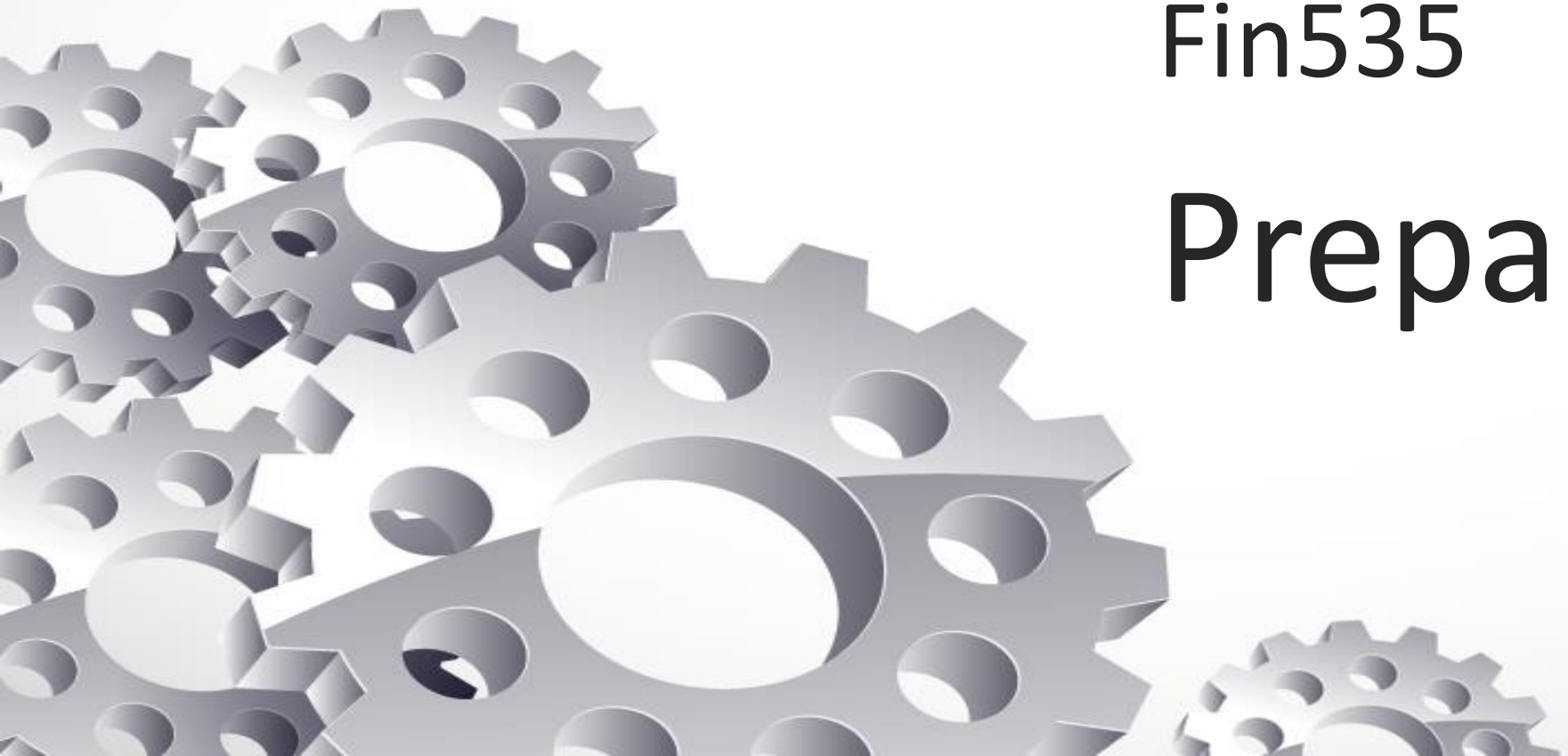

Fin535

Preparation

1. Linux Platform

This course will be conducted using Linux platform, ie. operating systems commands are linux-based and file/directory names are case sensitive. This is to prevent any interference to your scripts due to platform dependent quirks when running lab sessions in this course.

a. Windows Users

Watch this video (https://youtu.be/pkD2S08ra70) to understand how to install **WSL2 with Ubuntu 20.04 LTS**. You can refer to the Microsoft's instructions for more details. (https://docs.microsoft.com/en-us/windows/wsl/install-manual).

b. Mac Users

For macOS Users, there is no need to do anything since macOS is already a linux-based operating systems.

2. Terminal Client Application

The terminal client application is used for sending shell commands to your operating system.

a. Windows Users

After installing WSL for Ubuntu from the previous step, you can login to WSL using a terminal client. Watch this video (https://youtu.be/tMHrU3oupjg) on installing Windows Terminal. You can refer to further details from Microsoft (https://docs.microsoft.com/en-us/windows/terminal/get-started)

b. Mac Users

For macOS Users, you can use macOS Terminal application (https://support.apple.com/en-sg/guide/terminal/apd5265185d-f365-44cb-8b09-71a064a42125/mac).

3. Install VS Code and Extensions

We will be using VS Code for this course. This is currently the defacto code editor for cross-platform development in the industry. And its free.

Reference:

- https://code.visualstudio.com/download

Video:

- Install VS Code: https://youtu.be/wCOwc9u7boY
- Install VS Code Development Extension: https://youtu.be/o09JxB0qbjE
- Install VS Code Prettier Extension: https://youtu.be/hjY_Sbq2ZQA

Instructions:

1. Download here (https://code.visualstudio.com/download)
2. Open Visual Studio Code and click on the Extensions icon.
3. Search for the extension `**Remote Development**` and install it. This extension is needed for editing your source code in WSL.
4. Search for the extension `**Prettier**` and install it. This extension is useful for formatting your source code with you click save.
   - Press **CTRL** + **,**
   - Search for **editor.defaultFormatter** and set to `Prettier`
   - Search for **editor.format** and turn on `Editor: Format On Paste` and `Editor: Format On Save`

4. Install Node.JS (1/2)

NOTE: For Windows Users, please install Node.js in WSL and not the Node.js for Windows version. This is to ensure maximum cross-platform compatibility when using third party packages for this course. However, if you are unable to install WSL for whatever reason, then Node.js for Windows is your only option.

Instructions for Installing Node.JS for macOS

- https://amanhimself.dev/blog/install-nodejs-using-nvm-on-macos-m1/

Instructions for installing Node.JS for WSL/Ubuntu
(Video: https://youtu.be/rOu-XbY9abA)

1. Open Windows Terminal.

2. Prepare packages.
   - The first command is to get the latest list of packages (`apt update`) and upgrade existing packages (`apt upgrade`). The commands has to be executed as a privileged user (`sudo`).
   - The second command is needed to install basic build tools for some node packages to be built during installation.

```
$ sudo apt update && sudo apt upgrade -y
$ sudo apt install build-essential
```

3. Install Node Version Manager
   - This command will download and install the Node Version Manager (NVM). NVM is used to allow the use of different versions of node in case of compatibility issues when running certain packages.

```
$ wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash
```

Note: If encountered the output '* (HEAD detached at FETCH_HEAD) master' just press 'q' to exit

4. Install Node.JS (2/2)

4. Restart or open a new terminal session.

5. Install Node.js and npm

This command will install the latest version of Node.js. It will also install the latest version of Node Package Manager(npm) which is used to download Node packages.

```
$ nvm install node --latest-npm
```

NOTE:

At the time of writing, the latest Node.js version is v17.0.1 and npm is 8.1.0. This version will break certain libraries, such as create-react-app@4.0.3. The following instructions show how to use NVM to switch Node.js to version 16.8.

```
$ nvm install 16.8
Now using node v16.8.0 (npm v7.21.0)
$ node --version v16.8.0
$ npm --version 7.21.0
$ nvm alias default 16.8
default -> 16.8 (-> v16.8.0)
```

6. Install Standard Development Packages

```
$ npm i -g typescript bower grunt gulp less sass yarn webpack
```

The command `npm i -g` is used to install the list of packages following it into the global directory in ~/.nvm.

5. Documentation Convention (1/2)

The documentation will contain code snippets and instructions for terminal commands shown as code boxes. We will adopt the following convention when explaining the instructions.

1. Terminal Input

- The code box for terminal input will begin each line with a $ character denoting the command prompt. When reading the instructions, enter the commands that comes after the **$** but do not include the **$** character in your input.
- We will use **#** to denote the example output that will be returned from running the command. Do not enter anything from the line starting with **#**.

```
$ this is a command
$
# this is the output returned from your command
```

2. Commandline Console

- Sometimes, we will be issuing command into a command-line console application. Similar to terminal commands, we will precede any instructions with the > which denotes the command prompt in a commandline console. Enter the commands that comes after the **>** but do not include the **>** in your input.
- We will use **//** to denote the example output that will be returned from running the command. Do not enter anything from the line starting with **//**.

```
> let account = ethers.getSigner();
> account
// '0x167081A9f679a73ED3984265Ca84b91F8b19Cf15'
```

5. Documentation Convention (2/2)

3. Code Snippets

- Code snippets can contain **...** which represent additional code fragment that can exist but is relevant to the current context.
- Do not enter the **...** as part of your code but replace it with relevant code when required.

```
function hello() {
  ...
}
```

4. Quotation Marks

- Use only straight quotation marks - either ' or " or `.
- Do not use any curly quotation in your lab, eg. ‘ or ’ or “ and ”