

# Set things up...

Always want to know exactly where we are? What do we do? So, in my case

- Product is Linux x64 and the system runs on Ubuntu 20.04
- Precisely, I am looking for java 8
- I need 3 things to perfectly set up the development environment

- Download java development environment-JDK
- Download Eclipse IDE, **choose the appropriate IDE name "Eclipse for java developers"**
- A text editor which might be helpful to write, solve and execute small instance of problems.

- Somehow I learned to use the compression tool "tar". But if I face issue then I would use an interactive GUI to understand file compression and extraction in an interactive way.
- Try to know more about the hardware information before proceeding to install software programs.

## Getting started with java 1.8

### 1 Create Java Virtual Machine folder

```
sudo mkdir /usr/lib/jvm
```

( mkdir->make directory )( /usr/lib/jvm->working directory )

### 2 Add directory to the stack

```
pushd /usr/lib/jvm
```

Popd to remove specified directories from the stack

```
popd
```

"You can also use cd( change directory ) command, that is totally okay"

### 3 Extract the downloaded jdk-xuxxx-linux-x64.tar.gz in the Downloads folder

```
tar -xvzf ~/Downloads/jdk-8u271-linux-x64.tar.gz
```

Get help from **tar --help**, manual page using command **man tar**

### 4 Open the environment file using editors any of your favourite configuration editors nano,vi,etc.,

```
sudo vim /etc/environment
```

Then, set-up a variable environment. It might look this

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/jvm/jdk1.8.0_271/bin"
JAVA_HOME="/usr/lib/jvm/jdk1.8.0_271"
```

Good, keep going! Basic vi command

i -> insert

visual mode-> For cut,copy and paste.

wq! -> to save and exit

w -> to save and continue editing

q! -> quit

escape -> switch from insertion mode to other operations.

Example: Press escape key  
Then shift+semicolon(:) and type wq!

### 5 Set java location

```
sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jdk1.8.0_271/bin/java" 0
sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0_271/bin/javac" 0
sudo update-alternatives --set java /usr/lib/jvm/jdk1.8.0_271/bin/java
sudo update-alternatives --set javac /usr/lib/jvm/jdk1.8.0_271/bin/javac
```

### 6 Verify the configuration

```
update-alternatives --list java
update-alternatives --list javac
```

### 7 Open the bashrc with vi-editor

```
sudo vi ~/.bashrc
And add the following at the bottom of the file
👉 JAVA_HOME="/usr/lib/jvm/jdk1.8.0_271"
MAVEN_HOME="/usr/share/maven"
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin:$MAVEN_HOME/bin
export JAVA_HOME
export MAVEN_HOME
export PATH
```

**"Please, make sure you save the content."**

"I think, maven home can be excluded in case it is not needed at this moment. Well, programmer decision"

### 8 Verify the java version and java\_home path

```
👉 java -version && echo $JAVA_HOME
```

### Resources

<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands>

<https://www.makeuseof.com/tag/windows-run-commands-cheat-sheet/>

<https://www.makeuseof.com/tag/linux-commands-reference-pdf/>

<https://cheatography.com/davechild/cheat-sheets/linux-command-line/>

<https://www.makeuseof.com/tag/mac-terminal-commands-cheat-sheet/>

<https://catswhocode.com/vim-commands/>

<https://www.linux.com/training-tutorials/vim-101-beginners-guide-vim/>

[https://docs.oracle.com/javase/8/docs/technotes/guides/install/install\\_overview.html](https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html) [Windows,

Mac, Linux ]

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

<https://www.eclipse.org/downloads/packages/>

<https://www.eclipse.org/forums/index.php/t/1068754/>

<https://docs.oracle.com/javase/tutorial/>

<https://docs.oracle.com/javase/tutorial/getStarted/index.html>

<https://www.w3schools.com/java/>

<http://stackoverflow.com>