The newbie developer starter kit

IMG: Information Management Group, IIT Roorkee

```
"Hello World!"
```

Watching this message on your output screen is perhaps one of the best feelings this world has to offer. It's the joy of creation, watching something you made out of nothing come to life! At IMG we constantly experience this feeling as we build new apps on new technologies and we would love to share this feeling with you. So we have an interactive workshop plus AMA session coming up **very soon** (*like and subscribe to our Facebook page for updates!*) so that you too can get a head start into the amazing world of programming without hesitation.

But as far as workshops go, we'll be under the constant pressure of limited time and lots of things to cover. Understandably, we would love for you to come with your computers prepared so that on the day of the workshop we can get straight to the good stuff. It's a pretty simple process and we'll walk you through it. Are you ready to dive in? *Let's begin!*

Free and Open Source

First things first, FOSS (short for **Free and Open Source Software**) is movement encouraging developers to create and use free software in order to promote collaboration and unrestricted flow of knowledge. Here free means both unpaid (as in *free beer*) and unrestricted (as in *free speech*). FOSS is the opposite of the traditional approach of software being copyrighted and patented so that others may not derive from it or use it without paying the original developer.

At IMG (and in the developer community in general), we prefer to use open-source software wherever possible, from the operating system on our servers to the languages we use to develop apps.

The actual newbie developer starter kit

Now that you know what FOSS is, let's look deeper into the tools of the trade, the nuts and bolts required to run a web application. Note that these are just introductions and installation guidelines. A detailed description of their functioning and usage would be given in the workshop itself.

1 Operating system

An operating system is one software that every computer needs. It runs the low-level hardware management and provides developers a canvas to build their applications on. Most computers come with a proprietary operating system pre-installed. Very few laptops ship with a FOSS OS.

Most of you have Microsoft Windows installed on your laptop, yes? No? Then Apple OS X or macOS, right? These are great operating systems, no doubt, but both of them are for-profit, not open-source and not free either.

Developers love the FOSS alternative to these operating systems, **GNU/Linux** (which is just a very formal way of referring to Linux), for one major reason: the vast repository of developer tools and applications available for the platform.

Over the course of the article, you'll see how much easier it is to set up a simple web application server on Linux than it is to do on Windows or macOS. Just trust us and install **Ubuntu** (if you are a beginner) or **Fedora** (if you have some experience with Linux) on your PC. You'll be glad you did. And there's no reason for you not to, because they can run right alongside your current OS!

At the time of writing, the latest Long-Term Support release of Ubuntu is 16.04LTS and the latest release of Fedora is 26. Ubuntu LTS releases tend to be stable and Fedora releases tend to feature the latest cutting-edge technologies. You're free to pick.

See also: Ubuntu community guide for installation

(a) Ubuntu (b) Tux (c) Fedora

Figure 1: Tux is the mascot of Linux, the two most popular distributions of which are Ubuntu (recommended for beginners) and Fedora (recommended for more experienced people).

Give yourself some love and get Linux!

2 Web server

A typical website works like this. Your computer, the client, sends a request to a computer far away, the server. That server sends a response (usually in the form of a webpage) to the client. This is one request-response cycle.

To serve a website, what do you need? Yeah you guessed it, a web server! Currently the two amazing web servers on the Internet are **Apache HTTP server** and NGINX. We'll go with Apache because it is so much easier to set up with PHP, the language we are going to use in the workshop.

If you have Ubuntu on your computer. Whip up the terminal (`Ctrl+Alt+T` might work) and enter these commands.

```
sudo apt install -y apache2
sudo systemctl start apache2
sudo systemctl enable apache2
```

There you have it. Your Apache web server is up and running. Visit `http://localhost/` in your browser and you'll be greeted with Apache's default welcome web page.

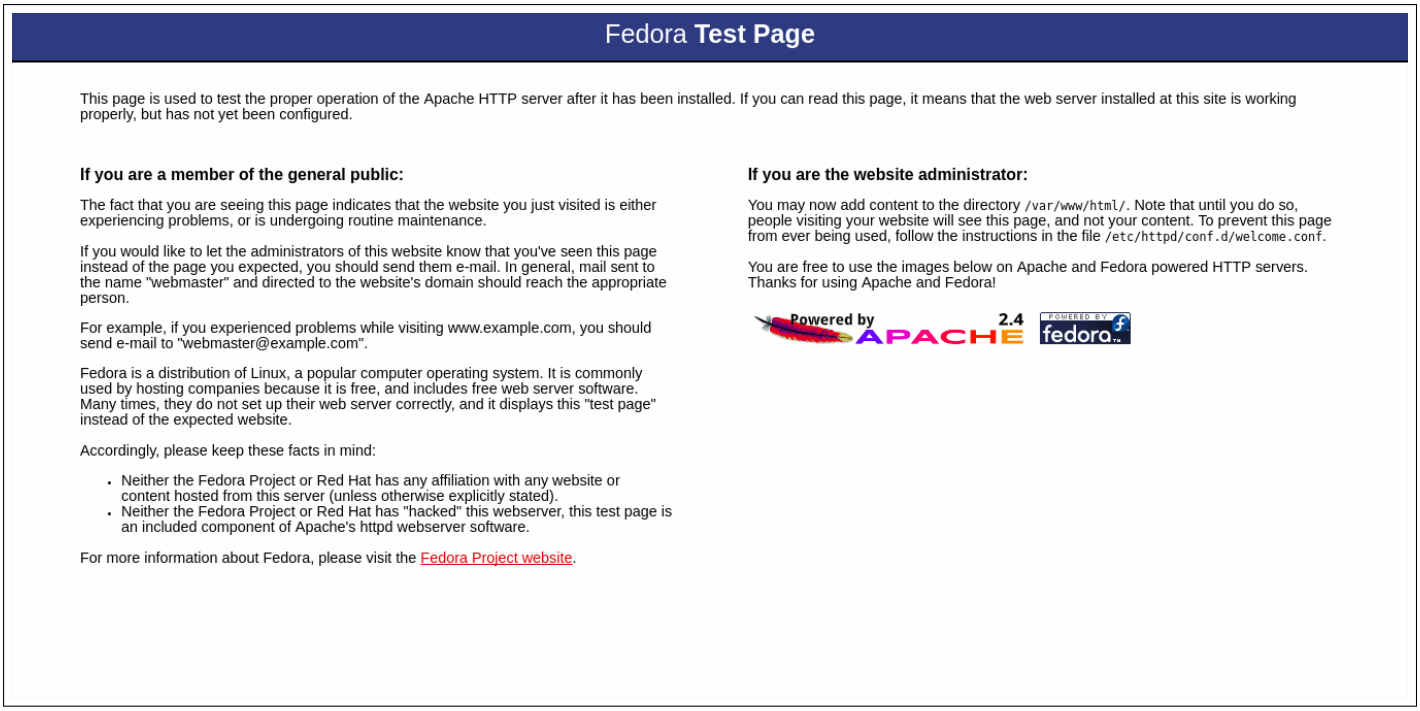

Figure 2: This is what the Fedora version of the Apache HTTP server, *httpd*, renders

Was that so hard?

3 Database provider

Every website has got to have a database to store information. Every web application receives lots and lots of data from the users and it has to be stored in a structured and organized way. Database Management Systems (DBMSes) can be of two types:

- **Relational DBMS**: They store data in tables linked with each other via keys and references
- **Non-relational DBMS**: They store data in unstructured ways and leave the user to decide the structure for their data

We'll just stick to the simplest of the bunch and yet fully functional, **MySQL**. MySQL is (rather *was*) a FOSS implementation of a relational DBMS. To install MySQL on your Ubuntu machine run these commands.

```
sudo apt install -y mysql-server
```

You will be asked to set the password for the user `root` which can be anything. Make it long and complex but memorable.

Don't worry if your PC installs MariaDB instead of MySQL because both of them are equivalent at this point. MariaDB is the community-driven fork of MySQL since MySQL was acquired by Oracle some time ago. Forking is an open-source term that means branching off to take a different approach using the same base code as the original software. At this point MariaDB is a drop-in replacement for MySQL.

Breeze, right?

4 Backend language

You also need a language to write your code in. Server side applications can be written in many many languages and almost every language has a couple of powerful frameworks for app development. For example Python applications can be written in Django, Java apps can be written in Spring and PHP applications can be written in Laravel. JavaScript is also gaining popularity as Node.js with the Express framework.

For the workshop we'll go with the simple and powerful **PHP** (version 7, which is the latest one at the time of writing) and not use any framework. These days you can write web application backends in just about any language like Python, Java and JavaScript and there is no wrong choice (however there is always *a more appropriate* choice).

To get PHP on Ubuntu, run these commands in the terminal.

```
sudo apt install -y php libapache2-mod-php
```

Now you have PHP installed as well. You see why we asked you to go with Linux?

Everything is just one `apt` *command away!*

5 Text editor

To type code, you need an editor. Something as simple as a text editor gets developers all riled up because if you want to be productive, you need to find a decent editor and tweak it to your liking. Once you're set on an editor switching might prove disturbingly painful and so it's better to pick the best one from the start.

Ubuntu and Fedora, both come with Gedit as the default editor but it is **not** a nice choice. If you are experienced in the art of development, you'll probably swear by one of the two gods of text editing, **Vim** and **Emacs**. Installing Vim is as easy as this. If you aren't aware of Vim's existence, please give it a go.

```
sudo apt install vim
```

But for beginners, we recommend **Atom** by GitHub or **Visual Studio Code** by Microsoft or **Sublime Text** simply because all these ship with a graphical user interface and a very gentle learning curve. Except for Sublime Text, they are even free!

Links:

- **Atom** - Atom is a hackable editor for the 21st century. It has a beautiful interface and is extensible with a ton of features.
- **Visual Studio Code** - It is a very powerful editor, extensible and comes with IntelliSense, an enhanced version of autocomplete.
- **Sublime Text** - It is the ancestor to these new editors and is much faster and also extensible via plugins.

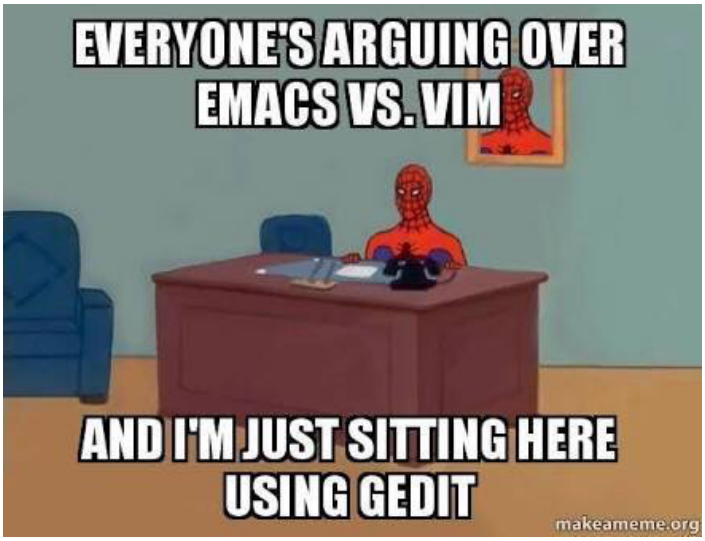

Figure 3: Do yourself a solid and don't use Gedit *ever*!

6 Version control

Developers make mistakes. That's as obvious a fact as saying that the sun rises in the east. When things break, getting the work back on track can be a huge problem if your method of backup has been keeping a copy of your files in date-labelled folders. This is a very primitive method and does not enable you to collaborate with other developers and leads to a huge mess of overwritten code and pain.

Making regular backups of your code is difficult, more so when things are moving fast and many people are working on the same application. Enter **Git**, the distributed version control system, trusted by developers worldwide for it's simplicity to newcomers and it's flexibility to professionals. With Git, everytime you make some significant changes to your code, you *commit* it and then if you go wrong, you can easily revert to the last working commit with one command! *Isn't that great?*

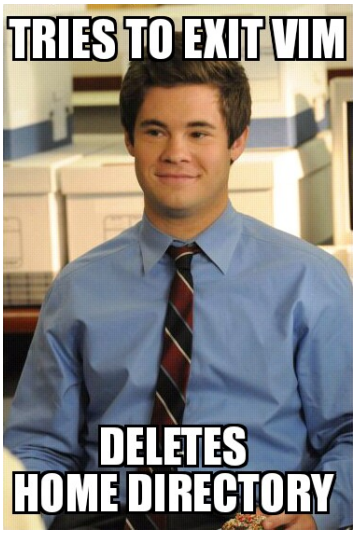

Figure 4: Git can help you recover your code in situations like this. *Also this is the main reason we recommend Vim only to experienced people. If you aren't one, you have these alternatives.*

Installing Git is as easy as anything else on Ubuntu.

```
sudo apt install git
```

Try telling me you didn't see that coming!

7 GitHub account

Everyone likes to show off what they've created. **GitHub** provides a platform for developers like you and us to do just that! A good GitHub profile is essential and if you haven't one already, it's a privilege for us to introduce you to it.

GitHub is the website that fosters collaboration across geographical and political borders on software. Applications that you make under the Git version control system can easily be pushed to GitHub for backup, publicity and reviews. Getting started is easy and all it takes is an account on the GitHub website.

Figure 5: Welcome to GitHub! Octocat says 'Hi!'

Bonus tips:

- **IMG on GitHub** - No surprises here! IMG is present on GitHub and we'd love for you to go through our code, try it out and give us feedback. You could find or fix some bugs and send in a PR as well. More on that in the workshop.
- **GitHub Education** - Since the folks over at GitHub are awesome, they've partnered with a variety of organizations to get students a variety of freebies and resources to get started with software development. It is amazing enough to warrant a visit to their website.

There you have it.

Summary

So in a matter of minutes your Linux computer becomes a full-fledged development machine. You can also do the same for Windows by installing WampServer but we do **not** encourage it.

Recap

```
sudo apt install -y apache2 php libapache2-mod-php mysql-server git vim
sudo systemctl start apache2
sudo systemctl enable apache2
```

Next is what?

You're all set for Sunday, **the 15th of October 2017**, to join us in a code-along where we'll walk you through building a simple web-application. If you have any doubts, questions or suggestions, we'll also be conducting a short introductory talk on Friday, **the 13th of October 2017**. Mark the date and follow us on Facebook for more interesting tips and further updates!

Apart from that, you can reach out to us on our email address: `img [at] iitr[dot]ac[dot]in`, or pay us a visit at **IMG lab**, on the ground floor at the Institute Computer Center, *literally* anytime.

Our door is always open! *No, not really, it's locked and you gotta knock.*

IMG
Information Management Group, IIT Roorkee