Install-athon

Welcome to the Installathon! This is an opportunity to download tools and apps onto your machine that you may end up using during the hackathon.

By the end of this, you will have a working website that you can customize to build the hackathon project of your dreams!

We'll get you set up with Node and a React app template. If you already know what these things are and you don't want them, no worries, we'll see you at the other hackathon events!

1. Getting started

Before we can start installing stuff, we need to install some more different stuff!!

Windows Users Only

To get a better experience going forward, we're going to install **WSL** - Windows Subsystem for Linux. This will give us a little mini linux computer running inside windows! Cool, huh? Let's get started:

1. Open Powershell
    a. Open start and search for "powershell". You see a little blue icon. Open that!

2. Copy and paste `wsl --install`
    a. This will install wsl!
3. **Reboot** when that finishes, and meet us back here for the next step
4. Open start again and search for “wsl”. You should see a little penguin, run that!
5. A black window should appear and say it’s setting something up. Wait for that to happen, and then when prompted, enter a username and a password!
    a. Make sure the password is something that you can type in easily, since you’ll be doing that a lot soon.

Done! You will be using this window for all the rest of the commands coming up. **If there are no Windows instructions, follow the linux ones!!** If you lose it, just search for “wsl” and open it back up!

Mac Users Only

To install some of the developer-specific things we need to get started, we’ll need a new program - **Homebrew**! This allows us to install things from the command line super easily. To start, open a terminal (just search for it!) and then just run this command (it’s one line, just reallyyyy long):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

VSCode

Highly recommended for all users, but especially if you are a Windows user.

1. Go to https://code.visualstudio.com/download, then download and run the thing from there.
2. Once it’s installed and open, click this link to install an extension we’ll need later:
    a. vscode:extension/ms-vscode-remote.vscode-remote-extensionpack
        i. If that doesn’t work, go to this url in your browser: https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.vscode-remote-extensionpack
    b. That’s right, VS Code can have extensions just like Chrome, Firefox, or Safari! Cool, huh? Might be a fun hackathon project…
3. IF YOU’RE ON MAC: You need to register the Code command manually. Follow the instructions here: https://code.visualstudio.com/docs/setup/mac#_launching-from-the-command-line

On your own, you can launch VSCode from the command line using the `code .` *command*

CLI

The rest of the tools are installed with the Command Line Interface (aka the Terminal). If you are unfamiliar, make your way through `pwd`, `cd`, `mkdir`, `ls` from this tutorial: https://tutorials.codebar.io/command-line/introduction/tutorial.html

If you are not familiar with the Command Line, here are a few basic tips to be successful:

1. **Don’t panic!** This can be a scary interface for the first time. Take a deep breath and recognize that it’s a new way of interacting with a computer. Over time, you’ll get accustomed to the interfaces and conventions.
2. **Read the errors.** When something goes wrong, it can be scary. The truth is that installations can be super hard and cumbersome, because everybody’s machine has a different setup. Versions and compatibility get crazy! Part of your success will depend on you slowly reading through the long errors that get output and figuring out how to understand them. If you get stuck, take a deep breath and read the error out loud. Then try to Google for answers by just pasting in different parts of the error. Chances are lots of other people have had the same problem and someone might already have found the solution!
3. **Feel the power.** With your new command line skills, you will have more power over your computer than you ever had before. “With great power, comes great responsibility.”
    a. Be careful what you paste into your terminal, especially if it says “sudo”. People on the internet are mostly nice, but not always so if you don’t know what something does, check out https://explainshell.com/ which tells you what commands do, or over the hackathon, ask a volunteer!

(vvv keep reading! vvv)

2. Version Control

When you're working on software, it's helpful to be able to go back in history or work with other people! The tools people use for this are called **Version Control**, and the most popular version control today is **git**.

Git

Git is a version control system that makes it easier for teams to work on the same codebase, by automatically merging files, tracking history, and more. If you use it right, it's got your back if you delete something, need to combine your work with someone else's, or just want to see how that weird thing worked 2 years ago!

Install Git

When you see something like the below box, just use the commands for the type of computer you have! Remember, if you're on windows, run the commands in wsl*, the thing we installed earlier, or they won't work!*

Windows	**Mac**

`sudo apt update`
`sudo apt install git`
 	
This will 1) update your computer's list of all things it can install and 2) install the latest version of git it can find!
	`brew install git`
	
	This tells homebrew to just go find and install git!

*If you haven't used Git before, learn Git on https://try.github.io/. Also look through this [Hello World tutorial](). *

Github

Github is the most popular hosted provider of Git, and is free for open source projects.

Your team will likely share its code using a repository on Github, so we recommend getting set up with it beforehand:

1. Sign up for a [free Github account]().
2. If you have a Mac, [download Github for Mac](), which means you won't have to use the command-line interface.

Try making your own repository for a personal project and pushing your code to Github. It can be anything you want! Just remember, if you choose to make it public, anyone will be able to see what you upload, so be careful not to upload those launch codes!!!

If you would like a more in-depth walkthrough of this stuff, join us for the Git & Github for Hackathons session on November 7th: https://girldevelopit.com/events/details/girl-develop-it-virtual-hackathon-2021-presents-git-github-for-hackathons/ (you really should come! It's gonna be fun and you'll learn a lot.)

Generate an SSH Key for Github

Now, we should have git installed and our very own GitHub account set up, great! But now, we need to get the two to trust each other and prove to GitHub that we are in fact the user of this computer. How do we do that? Something called…. **SSH Keys!!**

Let's make our own keys and tell GitHub about them!

1. First, we're going to make a new keypair, which is basically a really long paragraph and a shorter one that are linked in only one way with ✨fancy math✨. *Make sure to switch out the email for the one you used when you made your GitHub!*
    a. `ssh-keygen -t ed25519 -C "your_email@example.com"`
    b. It'll ask you a bunch of questions. Press enter for most of them! When it gets to the password one, you can choose to put one or not. You *probably* should, but (and don't tell anyone!!) I usually don't. Your call!
2. Start up the SSH agent in the background (this is the thing that holds the keys for you!
    a. `eval "$(ssh-agent -s)"`
3. Add your new key to the agent
    a. `ssh-add ~/.ssh/id_ed25519`
4. Copy your public key to the clipboard
    a. `cat ~/.ssh/id_ed25519.pub`
        i. This will print a bunch of weird text out to your clipboard. This is called your public key, and this is safe to share with people. **Never share the other file,** id_ed25519 (without .pub)**, as people can use it to impersonate you!!**
    b. To copy from WSL, just select it with your mouse and right click, or select and do Cmd+C on mac.
5. Follow steps 2-8 from here to add it to github: Adding a new SSH key to your GitHub account
6. To check it worked, run the next command. It should come back saying your username!
    a. `ssh -T git@github.com`
        i. If that didn't work, talk to one of the people here, we'll help out!

With that, you're good to go with Git and GitHub!!

3. NodeJs & React

Now, we're going to install a programming language. There are so many to choose from to do different tasks, you might already have a personal favorite, and that's ok! If you know what you want to do in this department already, you can feel free to head out now, we'll see you at other hackathon events!

If you don't have a specific language choice in mind, stick around! NodeJS is a very versatile language and a great one to get started with, since you can do just about anything with it, from websites to mobile apps to server code.

We're going to get you going with a basic starter website you can customize to your heart's content, and it might even be the starting point for your project! To install the latest version of Node, follow these steps for your platform:

Windows

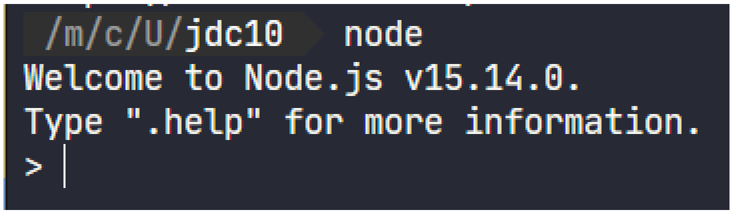
```
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
sudo apt-get install -y nodejs
```

(this should be 2 lines)

Mac

```
brew install node
```

Now type `node` and you should see something like this:



```
 /m/c/U/jdc10   node
Welcome to Node.js v15.14.0.
Type ".help" for more information.
> |
```

If so, congrats! If not, call over a volunteer and we'll help sort you out!

(vvv keep reading! vvv)

Parts of Node

Node is made of a few parts and we'll look at a few now!

1. **Node**
    a. This program allows you to run javascript code on your computer! You'll be using this later on to run what you write!
2. **npm**
    a. The **Node Package Manager**, this allows you to install other people's javascript code onto your computer. Lots of people have written helpful things for you to use, and we'll use a few in just a bit.
3. **npx**
    a. A combo of both Node and npm, it runs something straight from npm without installing it. We're going to do that now!

Project Templates

To get you started with a **React** website, we're going to quickly install a project template that'll get you on your feet fast!

1. First, create a new directory for your projects!
    a. `mkdir ~/Projects`
2. Now, open that folder
    a. `cd ~/Projects`
3. Finally, create a starter react app:
    a. `npx create-react-app installathon-app`
4. Open the directory with the project:
    a. `cd ./installathon-app`
5. Finally launch VS Code here to start looking around at the code you just created:
    a. `code .`
6. *Bonus step - start the website you just created!*
    a. `npm run start`

Great going! We did a lot today, but now you have everything you need ready to make your own webapp! Hope to see you around at the other hackathon events, and best of luck!

If you have any questions, please reach out to a volunteer. We're always happy to help!

4. Building Native Android Mobile Apps (UNSUPPORTED)

Instead of doing a website, you can also choose to build a native android mobile app. It’s pretty cool, but here at the Hackathon we don’t have anyone who can provide support so you’ll be on your own. If you still want to give it a try, be our guest!

If you’d like to build an Android app, you only have a few things to install.
1. Download the Java Development Kit (JDK):

Ref- https://www.oracle.com/java/technologies/downloads/

The Java Development Kit (JDK) is one of three core technology packages used in Java Programming, along with the Java Virtual Machine (JVM) and Java Runtime Environment (JRE). For better understanding of how they’re connected:

* The JVM is the Java platform component that executes programs.
* The JRE is the on-disk part of Java that creates the JVM.
* The JDK allows developers to create Java programs that can be executed and run by the JVM and JRE.

2. Download Android Studio:

Ref- https://developer.android.com/studio/install

That’s it! You are now ready to create mobile apps. If you are not familiar with Kotlin, below are some good resources for you.

Mobile App Development Important Resources
1. Official Kotlin Docs:
Ref- https://kotlinlang.org/docs/home.html
2. Quick & Free Kotlin Course from Google:
Ref- https://developer.android.com/kotlin/first
3. Stackoverflow (create an account if you don’t have one)
Ref- https://stackoverflow.com/