



**SIT 725 Prac 1**

**Basics of Programming and GIT**



# Contents

- Basics of Git
- Creating Git Repo
- Cloning a Git Repo
- Javascript
- Creating a Basic javascript application
- Live Server Extension
- Questions

# Basics of Git

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



# Basics of Git Cont ...



## Installation on Mac :

- Open Terminal
- Install **HomeBrew** using the command :
- `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
- `brew install git`

## Installation on Linux(Ubuntu):

- Open Terminal
- `sudo apt-get update`
- `sudo apt-get install git-all`

**Homebrew** is a package manager for macOS (and Linux, too). It is one of the first tools you'll need to set up a local development environment for programming on a Mac. Use it to install (and remove) software programs for the terminal, or command line.

## YouTube video :

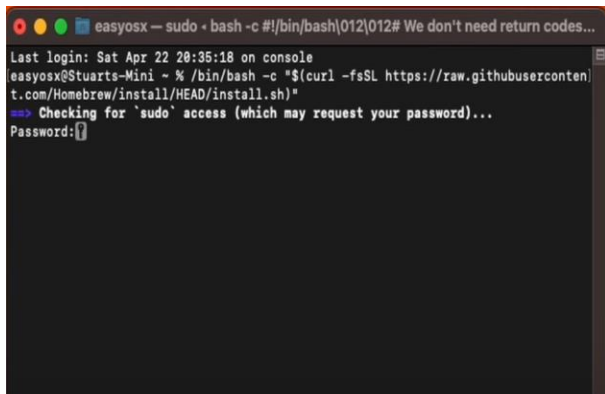
installation on **MacOS**

<https://www.youtube.com/watch?v=IWJKRmFLn-g>

Installation on **Ubuntu**

<https://www.youtube.com/watch?v=pRsHimqix24>

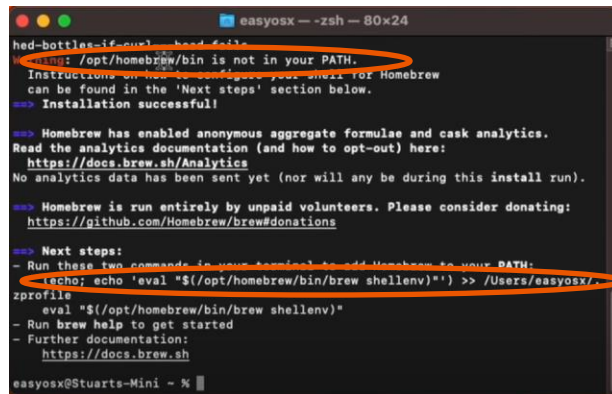
# Installation –macOS steps (video start from 01:32-03:45)



```
easyosx — sudo - bash -c #!/bin/bash[012]012# We don't need return codes...
Last login: Sat Apr 22 20:35:18 on console
easyosx@Stuarts-Mini ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
=> Checking for 'sudo' access (which may request your password)...
Password: [ ]
```

## Step 1

1. Copy the code paste into Terminal
2. Type in your admin password when prompted



```
easyosx -- -zsh -- 80x24
had-bottles-if-curl---head-fails
Warning: /opt/homebrew/bin is not in your PATH.
Instructions on how to configure your shell for Homebrew
can be found in the 'Next steps' section below.
=> Installation successful!

=> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

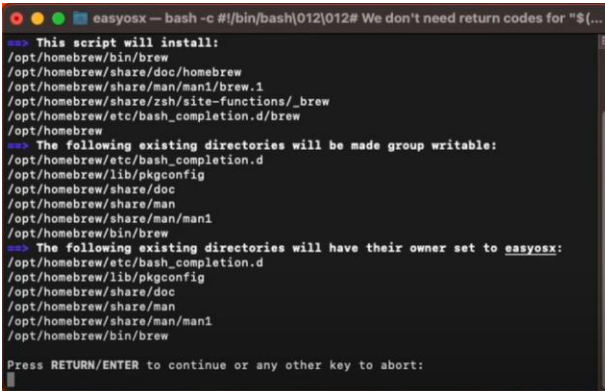
=> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations

=> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
(echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> /Users/easyosx/.zprofile
eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
https://docs.brew.sh

easyosx@Stuarts-Mini ~ %
```

## Step 3

1. Once it's done, you have Brew installed, but you're not done yet. (Brew is not in your path)
2. Set the path where HomeBrew stores its packages and where you will be modify it.
3. Look at "Next steps"

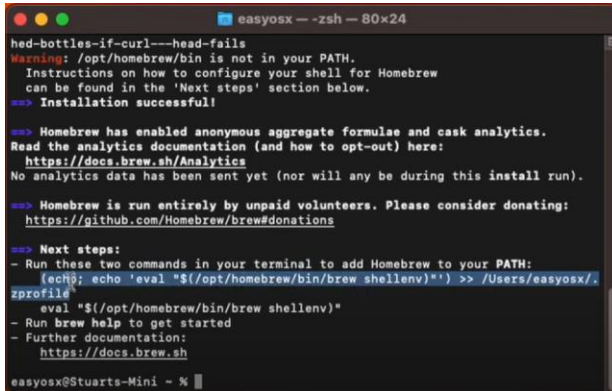


```
easyosx -- bash -c #!/bin/bash[012]012# We don't need return codes for "$(...
=> This script will install:
/opt/homebrew/bin/brew
/opt/homebrew/share/doc/homebrew
/opt/homebrew/share/man/man1/brew.1
/opt/homebrew/share/zsh/site-functions/_brew
/opt/homebrew/etc/bash_completion.d/brew
/opt/homebrew
=> The following existing directories will be made group writable:
/opt/homebrew/etc/bash_completion.d
/opt/homebrew/lib/pkgconfig
/opt/homebrew/share/doc
/opt/homebrew/share/man
/opt/homebrew/share/man/man1
/opt/homebrew/bin/brew
=> The following existing directories will have their owner set to easyosx:
/opt/homebrew/etc/bash_completion.d
/opt/homebrew/lib/pkgconfig
/opt/homebrew/share/doc
/opt/homebrew/share/man
/opt/homebrew/share/man/man1
/opt/homebrew/bin/brew

Press RETURN/ENTER to continue or any other key to abort:
```

## Step 2

1. You will be presented with a list of what is going to be downloaded and installed and from where.
2. Hit Enter
3. Use 1-2 mins to download and install



```
easyosx -- -zsh -- 80x24
had-bottles-if-curl---head-fails
Warning: /opt/homebrew/bin is not in your PATH.
Instructions on how to configure your shell for Homebrew
can be found in the 'Next steps' section below.
=> Installation successful!

=> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

=> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations

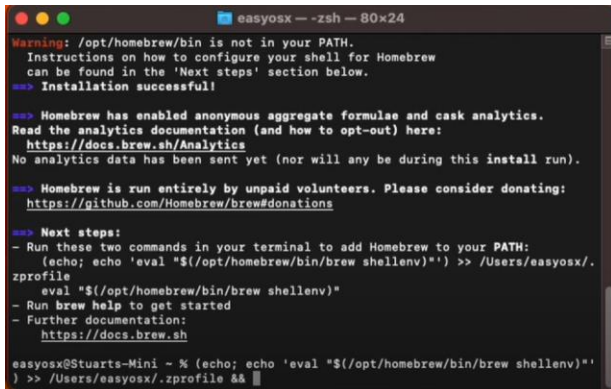
=> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
(ech; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> /Users/easyosx/.zprofile
eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
https://docs.brew.sh

easyosx@Stuarts-Mini ~ %
```

## Step 4

1. Highlight and copy the first command, paste it into the next line.

# Installation – macOS steps Cont



```
easyosx -- -zsh -- 80x24
Warning: /opt/homebrew/bin is not in your PATH.
Instructions on how to configure your shell for Homebrew
can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

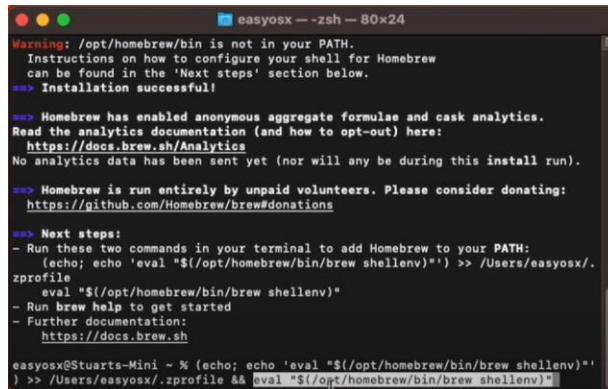
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
  (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/easyosx/.
zprofile
  eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

easyosx@Stuarts-Mini ~ % (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"'
) >> /Users/easyosx/.zprofile &&
```

## Step 5

1. Add **two ampersands** and **an additional space** after the pasted code. (This will tell the Terminal to run each command in order rather than you having to run each command separately.)



```
easyosx -- -zsh -- 80x24
Warning: /opt/homebrew/bin is not in your PATH.
Instructions on how to configure your shell for Homebrew
can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

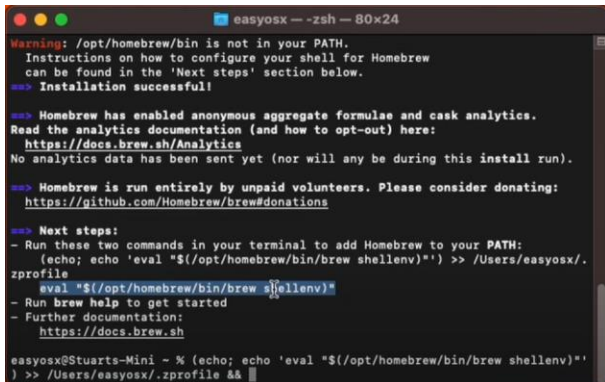
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
  (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/easyosx/.
zprofile
  eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

easyosx@Stuarts-Mini ~ % (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"'
) >> /Users/easyosx/.zprofile && eval "$(/opt/homebrew/bin/brew shellenv)"
```

## Step 7

1. Hit Enter
2. Successfully added HomeBrew to our path and have it ready to go



```
easyosx -- -zsh -- 80x24
Warning: /opt/homebrew/bin is not in your PATH.
Instructions on how to configure your shell for Homebrew
can be found in the 'Next steps' section below.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

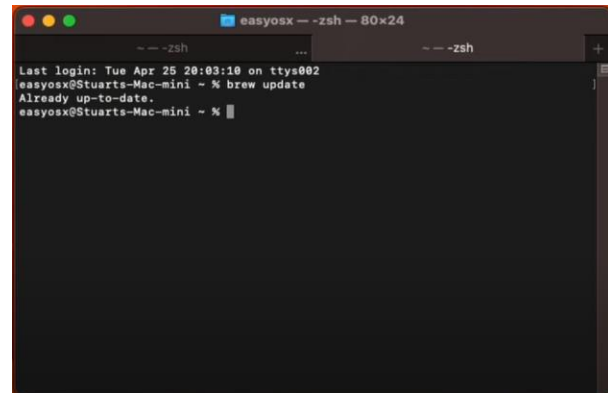
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
  (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/easyosx/.
zprofile
  eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
  https://docs.brew.sh

easyosx@Stuarts-Mini ~ % (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"'
) >> /Users/easyosx/.zprofile &&
```

## Step 6

1. Copy and paste the second command



```
easyosx -- -zsh -- 80x24
-- -zsh
Last login: Tue Apr 25 20:03:10 on ttys002
easyosx@Stuarts-Mac-mini ~ % brew update
Already up-to-date.
easyosx@Stuarts-Mac-mini ~ %
```

## Step 8

1. Open a new tab and test it
2. Type **brew update**
3. Hit Enter

# Installation on Windows:

```
network: ~$ curl -s https://github.com/git-for-windows/git
Cloning into 'git'...
remote: Enumerating objects: 506917, done.
remote: Counting objects: 1806 (1486/1486), done.
remote: Compressing objects: 100% (1413/1413), done.
remote: total 506917 (delta 3444), reused 181 (delta 181), pack-reused 497411
Receiving objects: 1806 (506917/506917), 223.14 MiB | 1.88 MiB/s, done.
Resolving deltas: 100% (181274/181274), done.
Updating files: 100% (4011/4011), done.

network: ~$ cd git
network: ~$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
network: ~$ git
$
```



**Git for Windows** focuses on offering a lightweight, native set of tools that bring the full feature set of the Git SCM to Windows while providing appropriate user interfaces for experienced Git users and novices alike.

## Git BASH

Git for Windows provides a BASH emulation used to run Git from the command line. \*NIX users should feel right at home, as the BASH emulation behaves just like the "git" command in LINUX and UNIX environments.

# Basics of Git Cont ...

## Setup Git local configuration

- Open the command prompt
- Run this command to configure your username and email:

```
git config --global user.name "Your Username"
```

```
git config --global user.email "youremail@email.com"
```

In case you want to generate an SSH Key and add to your git you can do so by following the steps [here](#)

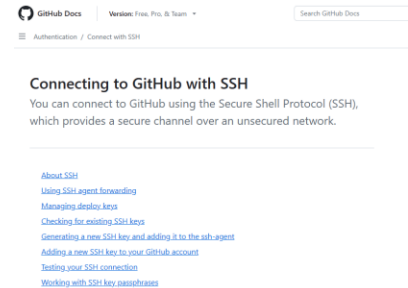
```
Welcome to Git (version 1.9.2-preview20140411)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Bucky@BUCKY-PC ~
$ git config --global user.name "Bucky Roberts"

Bucky@BUCKY-PC ~
$ git config --global user.email "bucky@pornhub.gov"
```

Example1-1

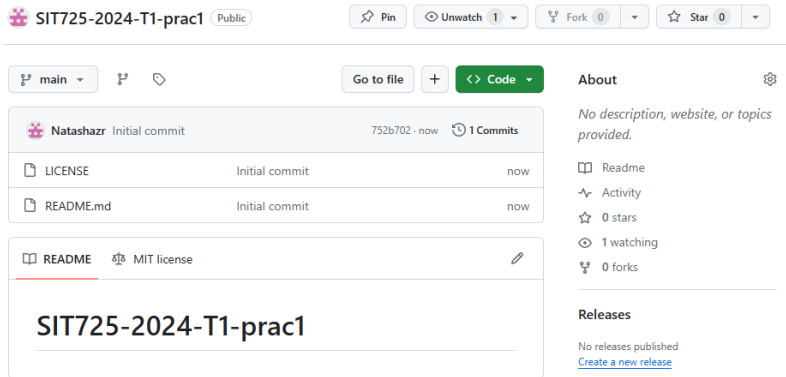




# Creating a Git Repo

We need to create a repo online before we start on the system

- Navigate to <https://github.com/new> and create a repo



Example1-3

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \* Natashazr / Repository name \* SIT725-2024-T1-prac1  
✔ SIT725-2024-T1-prac1 is available.

Great repository names are short and memorable. Need inspiration? How about [shiny-dollop](#) ?

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: **None**

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **MIT License**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set **main** as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

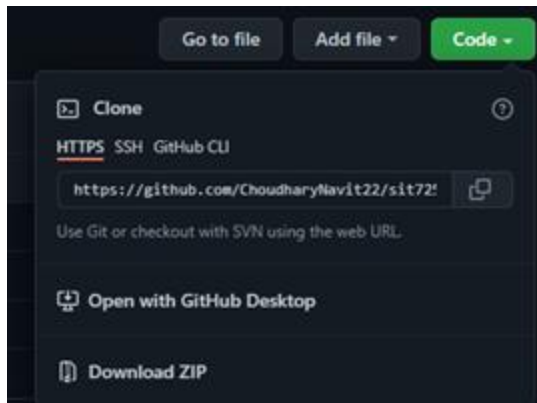
Example1-2

# Cloning a Git Repo-MacOS

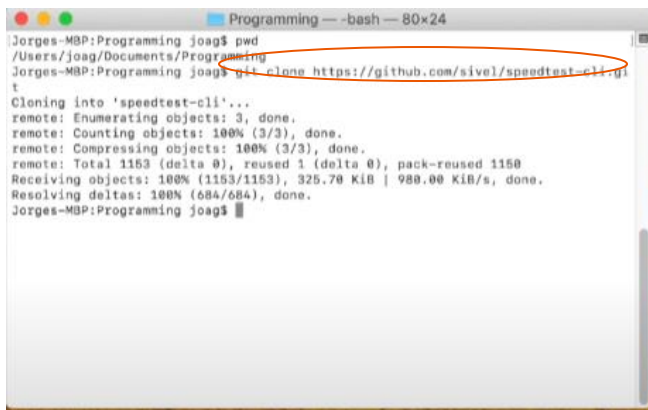
In order to clone a git repo, you simply got to the terminal and use the command

```
$ git clone <link to repo>
```

Link to your repo is present here



Example1-5



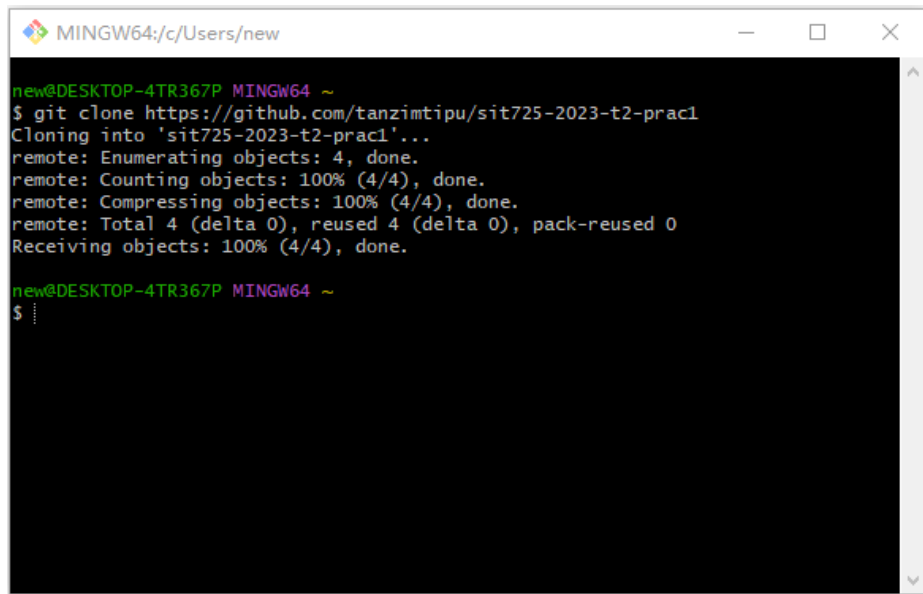
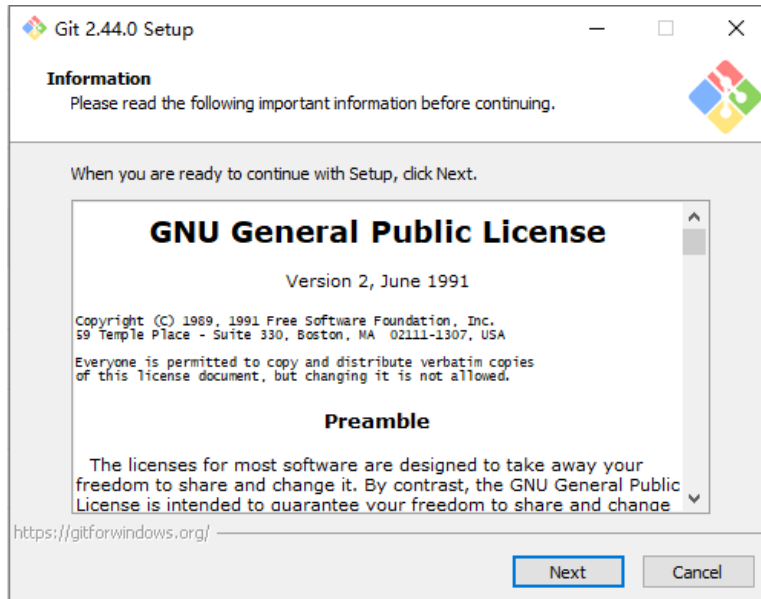
Example1-6

Your password: personal access token

- In the **upper-right corner** of any page, click your profile photo, then click Settings.
- In the **left sidebar**, click **Developer settings**.
- In the left sidebar, you see **Personal access tokens**

You can also connect using gh:  
**brew install gh**  
**gh auth login**

# Cloning a Git Repo-Windows (Git for Windows)



# Javascript

---

JavaScript is a programming language that was first created in 1994 as a way to add functionality and user interaction to a website. JavaScript is the third of the major building blocks of a web page. Without it, we wouldn't have the dynamic content and usability we expect from modern websites.

JavaScript was created with the intention of being utilised solely on the front end as a mechanism for web developers to add functionality to their pages, and it did so in its early days. The "V8 engine," introduced by Google recently, has improved the speed and usefulness of JS. As a result, interesting new front end JavaScript frameworks were developed and released, as well as Node.js, a mechanism to run JavaScript on a server.



# W3schools-Javascript Tutorial

https://www.w3schools.com/js/default.asp

Jobs

Spac

W3schools

Tutorials

Exercises

Certificates

Services

Search...

HTML

CSS

JAVASCRIPT

SQL

PYTHON

JAVA

PHP

HOW TO

W3.CSS

C

C++

C#

BOOTSTRAP

REACT

MYSQL

JQUERY

EXCEL

XML

DJANGO

NUMPY

PANDAS

NODEJS

JS Tutorial

JS HOME

JS Introduction

JS Where To

JS Output

JS Statements

JS Syntax

JS Comments

JS Variables

JS Let

JS Const

JS Operators

JS Arithmetic

JS Assignment

JS Data Types

JS Functions

JS Objects

JS Events

JS Strings

JS String Methods

Sites Designed to Sell

Look professional from the start when you build your website with Squarespace. Squarespace

Open

JavaScript Tutorial

Home

Next

JavaScript is the world's most popular programming language.

JavaScript is the programming language of the Web.

JavaScript is easy to learn.

This tutorial will teach you JavaScript from basic to advanced.

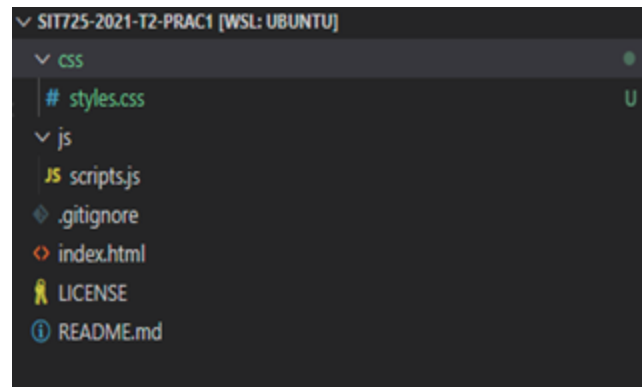
Start learning JavaScript now »

# Creating a Basic Javascript application

A simple javascript project has a few files that help us control our javascript application. These files include:

- **Index.html**: This file holds all the **static code** that needs to be displayed on the web browser
- **styles.css**: This file is responsible for **styling our web page** created by index.html
- **scripts.js**: This file is responsible to **provide the dynamic changes** in our static web page being created by the index.html

The folder structure of a simple javascript file looks something like this.



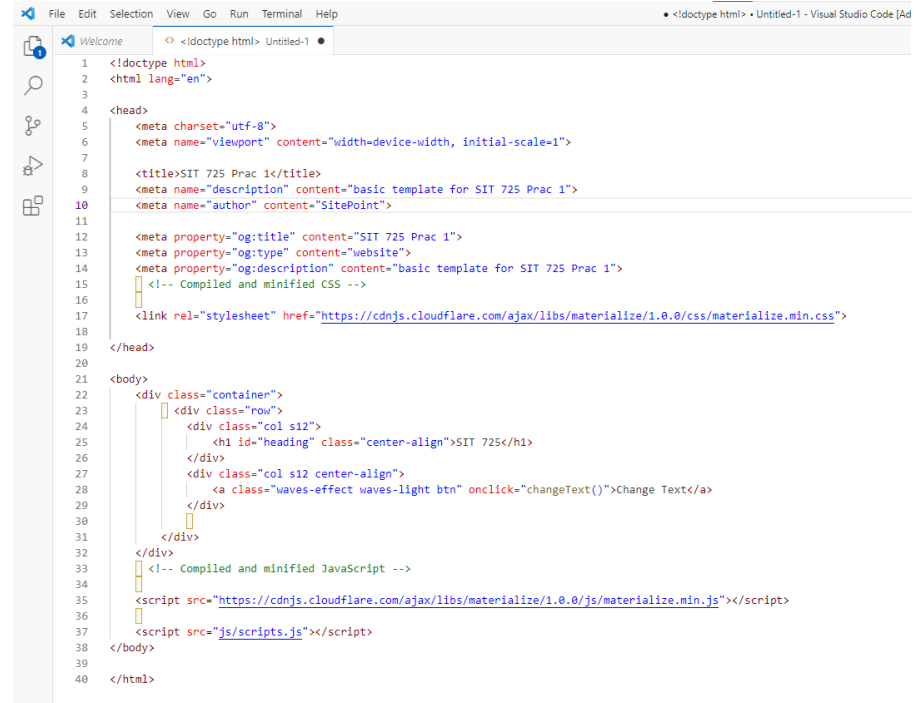
# Creating a Basic Javascript application Cont ...

Now let's add some code to our files.

First we start with **index.html**

In index.html :


- `<!Doctype html>`
- `<head>, </head>;`
- `<body>, </body>;`
- `</html>`




```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7
8   <title>SIT 725 Prac 1</title>
9   <meta name="description" content="basic template for SIT 725 Prac 1">
10  <meta name="author" content="SitePoint">
11
12  <meta property="og:title" content="SIT 725 Prac 1">
13  <meta property="og:type" content="website">
14  <meta property="og:description" content="basic template for SIT 725 Prac 1">
15  <!-- Compiled and minified CSS -->
16
17  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
18
19 </head>
20
21 <body>
22   <div class="container">
23     <div class="row">
24       <div class="col s12">
25         <h1 id="heading" class="center-align">SIT 725</h1>
26       </div>
27       <div class="col s12 center-align">
28         <a class="waves-effect waves-light btn" onclick="changeText()">Change Text</a>
29       </div>
30     </div>
31   </div>
32
33   <!-- Compiled and minified JavaScript -->
34
35   <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
36
37   <script src="js/scripts.js"></script>
38 </body>
39
40 </html>
```

# Creating a Basic Javascript application Cont ...

Then, we create **Script.js** file  
The code can be seen below:

 index.html

 script.js

```
function changeText () {  
    var textsArray = ["Text 1", "Text 2", "Text 3", "Text 4", "Text 5"]  
    var number = getRandomNumberBetween(0, textsArray.length - 1)  
    console.log("Index: ", number)  
    document.getElementById("heading").innerHTML = textsArray[number];  
}  
  
function getRandomNumberBetween (min, max) {  
    return Math.floor(Math.random() * (max-min+1)+min);  
}
```



# Live Server Extension

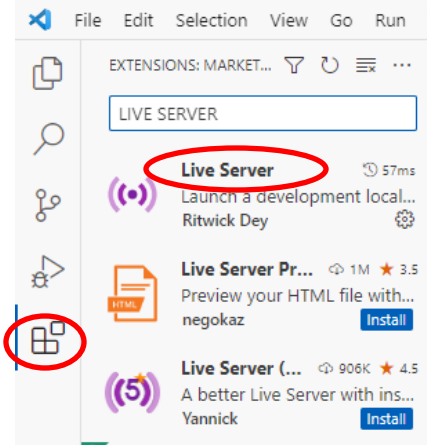
Now that we have created our dynamic web page we want to have a look how it looks right, for doing that lets add an extension to our visual studio code that helps us run a live server of our project. You can find the extension [here](#)

Once the extension has been installed it will look like something this on your visual studio code .



You can simply press go live and it would open the browser with your current project running. Give it a try!

Install extension- Live server in VSC



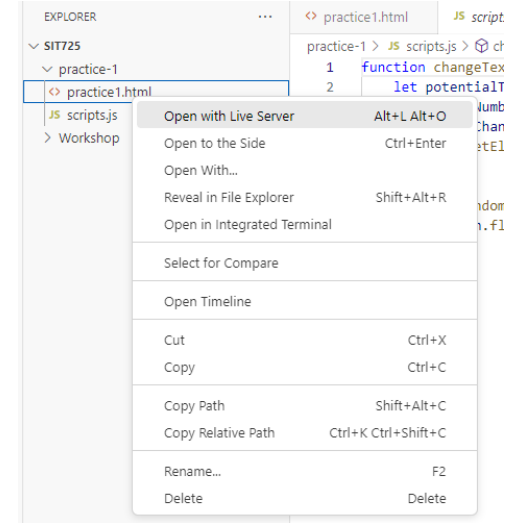
# Live server example

```
practice-1 > practice1.html > html
1 <html>
2   <head>
3     <title>SIT725</title>
4     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
5   </head>
6   <body>
7     <div class="container">
8       <div class="row">
9         <div class="col s12">
10           <h1 id="header" class="center-align">SIT725 - Applied Software Engineering</h1>
11         </div>
12         <div class="col s12 center-align">
13           <a class="waves-effect waves-light btn" onclick="changeText()">Change Text</a>
14         </div>
15       </div>
16     </div>
17     <script src="scripts.js"></script>
18   </body>
19 </html>
```

```
practice-1 > JS scripts.js practice-1 > JS scripts.js \
```

practice-1 > JS scripts.js > changeText

```
1 function changeText() {
2   let potentialTexts = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday'];
3   let randomNumber = getRandomNumber(0, potentialTexts.length - 1);
4   let textToChangeTo = potentialTexts[randomNumber];
5   document.getElementById('header').innerHTML = textToChangeTo;
6 }
7
8 function getRandomNumber(min,max){
9   return Math.floor(Math.random()*(max-min+1)+min);
10 }
```



127.0.0.1:5500/practice-1/practice1.html

SIT725 - Applied Software Engineering

CHANGE TEXT

Thursday

Friday

Monday

Tuesday

CHANGE TEXT

CHANGE TEXT

CHANGE TEXT

CHANGE TEXT

# Ontrack Tasks


SPT725 / Task List

RZ

Applied Software Engineering	
Search tasks...	
Read the welcome package and complete it	
1.1P - Pass Task	
Complete the JS course	
1.2P - Pass Task	
Design your services page	
1.3P - Pass Task	
Intro to programming	
1.4P - Pass Task	
Write a SRS Document	
2.1P - Pass Task	
Rest Api and Express	
2.2P - Pass Task	
Design Aspects	
3.1P - Pass Task	
Bootstrapping Express App	
3.2P - Pass Task	
Project Ideation	
3.3P - Pass Task	
Software Engineering Ethics	
4.1P - Pass Task	
Add a database	
4.2P - Pass Task	
Project Pitch	
4.3P - Pass Task	
Project Management	
5.1P - Pass Task	
MVC model	
5.2C - Credit Task	
Test your code	

Read the welcome package and complete it

≡

↶

↷

📄

⋮

—

+

🗨️

1

of 3


🔍

🖨️

💾

⋯

SIT725  
TASK 1.1P



WELCOME PACKAGE

**WELCOME TO SIT725!**

Dear students,

This unit has been carefully designed to guide you towards becoming professional Software Engineers. The focus will be on analysing, designing, and developing applications. It is crucial to understand that this unit revolve around software development, and to successfully pass them, you must be capable of writing executable code.

Now that we have addressed the obvious, let me extend a warm welcome to SIT725! My name is Muneeb Uj Hassan, and I will be serving as the unit chair in T3. I have worked for several years in the industry as a software engineer, and now, I am here to help you in your journey to become the next generation of Software Engineers, ready to solve world problems.

This unit will be intense but rest assured that the teaching team will be here to support you every step of the way. By investing your efforts and dedication, you will reap the rewards at the end of the unit.

Claus commences in Week 1, and it's important to note that there will be an assessment due in Week 2. Therefore, I strongly advise you to start working on the material from day one to stay on track.

**DELIVERY MODE TEAMS**


The unit will be mainly delivered on Teams, you will find the lectures and the workshops on Team.

**Task :** Join the team by following the Lecture Channel, you'll be added automatically to channel upon enrolment.

Details for Read the welcome package and complete it

### Task Details

- **Pass Task**
- **Start Date** – Aim to start this task by **Sun 11 Feb.**
- **Due Date** – Aim to complete by **Mon 11 Mar.**

 Task Sheet

# SIT725 TASK 1.1P

## TECH STACK

The technology stack is based on NodeJS, React, HTML and Bootstrap/Materialize. You are expected to have learnt programming prior to joining this unit, if you have never written a line of code before, make sure to start learning now. By Week 3 you should have completed the following, it is a 24hours course that should include most things we are going to be doing this trimester. You need to be proficient from day one. The course is mandatory.

**Task1:** Complete the following by Week 3 and start from Week 0 if you are not confident in your programming skills, the course is free, log in to LinkedIn learning using your Deakin account. The whole course needs to be completed by the end of week 4, but some tasks will be weekly.

[https://www.linkedin.com/learning/collections/6579539778742681602?trk=share\\_collection\\_url&accountId=2104084&u=2104084&success=true&authUID=YInIwdrbTrqN19%2FPh2FCig%3D%3D](https://www.linkedin.com/learning/collections/6579539778742681602?trk=share_collection_url&accountId=2104084&u=2104084&success=true&authUID=YInIwdrbTrqN19%2FPh2FCig%3D%3D)

### Task2:

To successfully submit your practical-related tasks during the trimester, it is essential to create a GitHub account for this unit. Follow the simple instructions provided below to create your account:

1. Visit the GitHub website at [www.github.com](https://www.github.com)
2. Click on the "Sign Up" button to begin the account creation process.
3. Enter your preferred username, email address, and a strong password.
4. Select the "Free" plan option,
5. Complete the security verification process, if prompted.
6. Review and accept the terms of service and privacy policy.
7. Click on the "Create account" button to finalize the account creation.

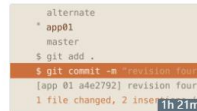
Once you have successfully created your GitHub account, you will be ready to submit your practical-related tasks throughout the trimester.

## Full Stack Developer

Created by Alessio Bonti

Good complete course

Share Copy



COURSE  
Learning Git and GitHub (2015)

By: Ray Villalobos



COURSE  
HTML Essential Training (2017)

By: James Williamson



COURSE  
CSS Essential Training 1 (2017)

By: Christina Truong



COURSE  
Bootstrap 4 Essential Training

By: Ray Villalobos



COURSE  
Web Programming Foundations

By: Morten Rand-Hendriksen



COURSE  
JavaScript Essential Training (2017)

By: Morten Rand-Hendriksen



COURSE  
Node.js Essential Training (2019)

By: Alex Banks

## SUBMISSION DETAILS

Once you finished reading this task sheet, leave a comment describing your programming skills, your familiarity with Java script, and your target grade for this unit, then mark this task as 'ready to feedback'.

# SIT725 TASK 1.2P-COMplete JS COURSE

## Intro

In this unit we will focus mostly on Javascript in the NodeJS flavor to develop our applications. Using online learning resources is very important, also, certificates look really cool on your LinkedIn profile.

## Instructions

Please ensure you log in to LinkedIn Learning using your DEAKIN account to avail of the free access to a wide range of courses. It is essential that you complete the course within the allocated time frame. The course duration is approximately 5 hours. I strongly advise against trying to cram the entire course at the last moment, as well as skipping through the content.

Towards the end of the trimester, you will be presented with coding questions that require you to provide on-the-spot answers. Even if you have achieved a High Distinction (HD) in the unit, failing to answer a basic question such as "Can you please explain how the pop function of an array works?" could result in a failing grade. It is crucial to thoroughly understand the concepts and not rely solely on your academic performance.

<https://www.linkedin.com/learning/collections/6579539778742681602?u=2104084>

## Submission details and Delivery

Once completed, download the certificate and upload it on OnTrack



COURSE  
JavaScript Essential Training  
(2017)

By: Morten Rand-Hendriksen



COURSE  
Node.js Essential Training (2019)

By: Alex Banks

# SIT725 TASK 1.3P-DESIGN YOUR SERVICE



## Intro

In this unit you will learn the basics of developing a complex web application. This can become as complex as you want, but ultimately, it will demonstrate your understanding of the technologies being used.

## Instructions

As a person in IT, you want to be able to showcase your abilities. Start thinking about "who am I " or "what do I want to be known for" ? And from this, start thinking about what services, ultimately, you want to sell.

If you are going to be a web developer, perhaps you want to build your own portfolio website, if you are a data engineer, you want to show how good you are with data, perhaps store it, analyse it, and show it. If you are in IT security, maybe you want a service that allows people to upload their code to be analysed.

In the coming weeks, you are going to be building your own website, week by week during the workshop sessions. Remember, you are to learn the technology, implementing it is up to you.

## Submission details

In 200 words or less, describe your grand plan to become the best IT person your field! Don't be shy, use images or sketches, but keep it within one page.

## Upload a pdf document to Ontrack

## Example

*"As a Product manager and Software engineer, I am at selling my expertise as a consultant for prototyping and blockchain. My website will show the projects I have worked on and will also allow users to create an account on my NFT market. Every new user will receive a wallet and will own an automatically uniquely generated NFT art piece."*

# SIT725 TASK 1.4P-INTRODUCTION TO PROGRAMMING

## Intro

Workshops are compulsory, and you must complete them each week. In the Welcome Package (Task 1.1P) you would have provided your Github account. Make sure you have done so, and that the repositories that you make are public from now on. Every Wednesday morning a script is automatically executed and it will download your repository.

It is a machine, so it will not make mistakes, the only reasons it may not be able to download your code is if it is unreachable. Also, if you submit after the due date (Sunday) it will be automatically marked as late.

## Instructions

In the first workshop you will learn about the following:

- Basic programming
- Git

## Submission details and Delivery

- Once you are done, push your code into your repo, giving the repository the following name `sit725-2023-t2-prac1` , ultimately this should read as <https://github.com/username/sit725-2023-t2-prac1>.
- Share your repo link and provide screenshots of the running code as evidence, convert it to .pdf and upload it to the OnTrack.

If you want to be sure, simply clone your repo. If you can clone it, I can clone it 😊

1. You need to follow the instruction of:  
- slides 14,15,16 and 18 to complete Ontrack 1.1p task.

2. In your prac-1, you need to have `index.html`, `Scripts.js`, `LICENSE`, and `README.md` files.  
( you can write your own `index.html`, use the provided `Script.js` code in slide 16 to show the output. The output should be similar to the example on slide 18)

3. Test the code, run the page, and do screenshots, convert them in PDF format, and submit evidences into Ontrack system.



Thanks



QUESTIONS