

Learn xLearn xCode xCode xLearn xLearn xLearn xLearn x

codecademy.com/courses/learn-git/articles/using-a-gitignore-file-in-your-github-repository

Appsติดตั้ง JDK (Ja...Backtesting A...MATLAB for P...Tableau ServerCoursera-Stan...Reading list

My HomeCourse MenuGet UnstuckTools

# Using a .gitignore File in Your GitHub Repository

In this article, we will learn how and why we should use a .gitignore file to make cleaner and more secure code changes!

## What is a .gitignore file?

What happens when our Git repository contains certain files we never want to commit to a shared or public codebase? We want to be careful that `git add` doesn't accidentally move them to the staging area. That's where a .gitignore file comes in. .gitignore is a plain text file that tells Git to intentionally ignore changes in certain files. This also ensures that no other contributor in the repository accidentally commits those files.

## Why use a .gitignore file?


Each line in .gitignore corresponds to a file, directory, or pattern we would like to ignore when staging. Using a .gitignore file results in cleaner staging areas and prevent files containing sensitive information from being committed. Some of the files or folders we should ignore include:

- Configuration files with API or secret keys such as `.env`
- Compiled binary files or production directories such as `build` or `dist`
- Log files
- Dependencies downloaded from a package manager such as `node_modules`
- System files such as `thumbs.db` on Windows or `.DS_Store` on macOS

Fill in the Code

Complete the sentence below.

A .gitignore file lets Git know which  , folders, and patterns to ignore, and not  then.

 You got it!

Back

Next

Learn x

Learn x

Code x

Code x

Learn x

Learn x

Learn x

Learn x

+

⌵

⌵

⌵

⌵

⌵

⌵

⌵

codecademy.com/courses/learn-git/articles/using-a-gitignore-file-in-your-github-repository

Appsติดตั้ง JDK (Ja...Backtesting A...MATLAB for P...Tableau ServerCoursera-Stan...Reading list

c

My Home

Course Menu

Get UnstuckTools🕒👤

## .gitignore in action

Let's say we run `git status` and see that operating system files, such as `thumbs.db` and `.DS_Store`, are staged and will be committed!

```
~/codecademy master + git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .DS_Store
    new file:   .gitignore
    new file:   node_modules/package1/main.js
    new file:   node_modules/package2/main.js
    new file:   public/index.css
    new file:   public/index.html
    new file:   src/main.js
    new file:   src/old/main.js
    new file:   thumbs.db
```

Let's ignore those files using their exact names. For example, if we add these lines to our `.gitignore` file:

```
# Windows OS file
thumbs.db

# macOS OS file
.DS_Store
```

Git will ignore the special operating system files for Windows and macOS. These files will never be committed for this particular repository regardless of their location in this project. Note that in the file, blank lines are ignored and lines starting with `#` are treated as comments.

After unstaging and running `git add .` again, the output of `git status` shows that the `thumbs.db` and `.DS_Store` files have been ignored!

```
~/codecademy master + git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
    new file:   node_modules/package1/main.js
    new file:   node_modules/package2/main.js
    new file:   public/index.css
    new file:   public/index.html
    new file:   src/main.js
    new file:   src/old/main.js
```

Let's go over how to write a file yourself!

## Creating a .gitignore File

We can create a `.gitignore` file easily using a terminal editor like `nano` or `emacs`, or just using a File explorer. Don't forget the `.` before the filename!

...

BackNext

Learn xLearn xCode xCode xLearn xLearn xLearn xLearn x

codecademy.com/courses/learn-git/articles/using-a-gitignore-file-in-your-github-repository

Appsติดตั้ง JDK (Ja...Backtesting A...MATLAB for P...Tableau ServerCoursera-Stan...Reading list

My HomeCourse MenuGet UnstuckTools

Let's go over how to write a file yourself!

## Creating a .gitignore File

We can create a **.gitignore** file easily using a terminal editor like **nano** or **emacs**, or just using a File explorer. Don't forget the **.** before the filename!

**.gitignore** is usually placed in the root directory of the repository. The filenames inside a **.gitignore** file can be written relative to the location of the **.gitignore** file. For example, we could add the line

```
src/main.js
```

to ignore the file **main.js** under the **src/** directory.

```
~/codecademy } master touch .gitignore
~/codecademy } master ls
node_modules public src thumbs.db
~/codecademy } master ls -a
... .DS_Store .git .gitignore node_modules public src thumbs.db
~/codecademy } master |
```

**Note:** since **.gitignore** is a hidden file, we will need to add the **-a** flag to **ls** to see it.

## Ignore a directory with .gitignore

Sometimes we want to ignore entire directories or specify certain files in a directory. Common directories to leave out of a Git repository are **node\_modules** or **logs** folder. We can ignore an entire directory by simply adding its name to **.gitignore**:

```
node_modules/
```


This will ignore the **node\_modules** directory, and all subdirectories and files inside them. The forward slash **/** specifies that we are ignoring the directory.

Multiple Choice Question

A **.gitignore** file containing **logs/** will ignore all files AND folders named **logs**.

True

False

 Correct! **logs/** will NOT ignore files names **logs** due to the forward slash at the end. This tells Git to only ignore directories named **logs**.

Back

Next

Learn xLearn xCode xCode xLearn xLearn xLearn xLearn x

codecademy.com/courses/learn-git/articles/using-a-gitignore-file-in-your-github-repository

Appsติดตั้ง JDK (Ja...Backtesting A...MATLAB for P...Tableau ServerCoursera-Stan...Reading list

My HomeCourse MenuGet UnstuckTools

## .gitignore Patterns

We can take advantage of **patterns** to match multiple filenames. These help us handle special cases such as ignoring specific file types or ignoring all but one file inside a directory. Some examples of things that make up patterns are:

- Wildcard **\*** to match 0 or more characters except for **/**. For example, adding **\*.html** to **.gitignore** would ignore all files ending with the **.html** extension. **example\*** would match any file starting with **example** such as **example.txt** or **example.htmlFile.html**.
- Negation **!** as a prefix to negate any file that would otherwise be ignored. For example,

```
index*
!public/index.css
```

will ignore all files starting with **index** except for **src/index.css**. But, we cannot negate a file inside an ignored directory.
- Square brackets **[]** can be used to match a single character from a set of characters or a range of characters. Note that the range can be alphabetical: **[a-z]** or **[A-Z]**, numeric **[0-9]**, or a set of characters. If we added **index.[a-i]\*** with both the square bracket and wildcard to **.gitignore**, we would ignore **index.css** and **index.html** but not **index.js**, since "j" is outside of the **[a-i]** range.
- Double asterisk **\*\*** is used to match 0 or more directories. If we had a **temp** folder inside all of the folders in the root directory and we only wanted to match files with the **.log** extension, we could use the pattern **\*\*/temp/\*.log**.

Fill in the Code

Complete the pattern for a **.gitignore** file below to match and ignore ALL files with the **log** extension inside the directories **v1**, **v2**, and **v3**? Be sure to match log files in all of their subdirectories as well.

These folders are located in the **src** directory.

src/  /  /

You got it!

## GitHub Provided Templates

When we create a new repository on GitHub, we have the option to add a

Back

Next

Learn x

Learn x

Code x

Code x

Learn x

Learn x

Learn x

Learn x

+

▼

—

□

×

codecademy.com/courses/learn-git/articles/using-a-gitignore-file-in-your-github-repository

🔍

🌟

⚙️

📖

K

⋮

Apps

ติดตั้ง JDK (Ja...

Backtesting A...


MATLAB for P...


Tableau Server

Coursera-Stan...

»

Reading list

 My Home Course Menu

Get Unstuck Tools 🕒 


Complete the pattern for a **.gitignore** file below to match and ignore ALL files with the **log** extension inside the directories **v1**, **v2**, and **v3**? Be sure to match log files in all of their subdirectories as well.

These folders are located in the **src** directory.

src/

/

/

 You got it!

## GitHub Provided Templates

When we create a new repository on GitHub, we have the option to add a **.gitignore** file from a list of templates. These templates are pulled from [GitHub's gitignore repository](#). For example, below is the template for Java projects.

```
# Compiled class file
*.class

# Log file
*.log

# BlueJ files
*.ctxt

# Mobile Tools for Java (J2ME)
*.mtj.tmp/

# Package Files #
*.jar
*.war
*.nar
*.ear
*.zip
*.tar.gz
*.rar
```

## Conclusion

Good job on completing this article! **.gitignore** files are incredibly useful to make sure our repositories are only tracking changes on relevant files. They also ensure that we can keep files with sensitive data out of the staging area. You'll come across **.gitignore** files in almost every project on GitHub.

This file is shared by the repository, so make sure your patterns are correct and take your time with creating your first **.gitignore** file!

Back

Next