1. Welcome

You should be excited to document your code!

1. What is Documentation?

A cook book is a way to document recipes, technical documentation helps us make sense of code.

1. Quiz: Who is Documentation For?

Good documentation is not boring! It is written for humans, like code is for computers.

Who might be a potential end-user of documentation?

You, your coworkers, and your users will all use the documentation you provide!

1. How Does Nija Consume Documentation?

Nija is a DevOps engineer. Uses third party libraries and tools, and the documentation helps them find examples. It is much nicer than reverse engineering a bunch of code.

1. Why Should Art Have Documented His Code?

Art, horror story, used a dummy phone number in his source code that needed to be replaced before using the actual software, but did not document this. It was shipped, and the customer was really mad. Future Walter said present Walter got lazy, and documentation can really help yourself out.

1. Quiz: Introduction to READMEs

Don’t write a cook book when all you need is a single directory.

Example READMEs

1. [README #1 - factory\_girl](https://github.com/thoughtbot/factory_girl)
2. [README #2 - can.viewify](https://github.com/zkat/can.viewify)
3. [README #3 – create-your-own-adventure](https://github.com/udacity/create-your-own-adventure)
4. Anatomy of a README

All three have essential information necessary to get the respective project running.

Should provide just enough context to get another user up and running with your code.

A few good questions to ask yourself are…

1. What steps need to be taken?
2. What should the user already have installed or configured?
3. What might they have a hard time understanding right away?
4. Quiz: Documenting a Growing Codebase

Licenses are important. <https://choosealicense.com/>

Contributing

Code Status

Don’t make your life more complicated than it needs to be.

As a code base grows, which sections might you add to your README?

Known bugs, Frequently asked questions, and Table of contents.

1. Readable READMEs with Markdown

Can be written in any format, Markdown lets you generate well formatted content easy for web viewing and scanning through visually.

1. Basic Markdown Syntax

# Markdown 101

Markdown is a light markup language often used for READMEs (though you'll find other use cases for it, too!). It is fairly straightforward, and much of the syntax is intuitive.

But as it turns out, there are many different dialects of Markdown, just like in a spoken language. Each of these dialects is known as a flavor of Markdown. Most of these dialects are pretty much the same, with only minor differences.

Since your READMEs will ultimately end up on GitHub, we'll be using **GitHub Flavored Markdown**. I've included a link to the full documentation for it in the instructor notes (and we'll be using that later in this course), but I'll get you started with a quick crash course.

## Feeling Bold?

To make text **bold**, surround it with double asterisks. So this code:

Isn't today a \*\*wonderful\*\* day?

becomes: Isn't today a **wonderful** day?

This reads a bit more nicely than a <strong> tag would in HTML, and takes considerably fewer keystrokes to type out.

## Italics, I tell you!

To italicize text, surround it with underscores. So this code:

And **in** **that** moment I thought **to** myself: \_Did I turn off **the** stove?\_

becomes: And in that moment I thought to myself: Did I turn off the stove?

Much like the previous example, this code reads much more like English, which is great for when you're skimming the original document.

## To code, or not to code?

Inline code is useful for indicating that you're writing code and not a regular word. For this, you'll surround text in backticks (`, not a single quote). So this code:

You should **use** the `strong` tag.

becomes: You should use the strong tag.

...which makes much more sense than "You should use the strong tag."

## The Title Sequence

Headings are even simpler! For h1 through h6 tags, all you'll need is a # before your text. The number of #s you include tells Markdown which header tag you're using. For example:

## This is an h2.

### This is an h3.

becomes...

## This is an h2.

### This is an h3.

Let's practice!

Here's a link to the basic [**Markdown documentation**](https://help.github.com/articles/markdown-basics/). You can also check out the differences in GitHub flavored Markdown [**here**](https://help.github.com/articles/github-flavored-markdown/).

1. Quiz: Basic Markdown Syntax Quiz

#Here is your task

`code`

\*\*here\*\*

\_there\_

1. More Markdown Syntax

# More Markdown

Now it's your turn. Dive into the [**Markdown documentation**](https://help.github.com/articles/markdown-basics/), and explore more ways to write beautiful READMEs.

A few important items you may want to write and should pay extra attention to include:

* Ordered and unordered lists
* Links and images
* Large blocks of code

## HTML Is Still a Thing

Something to keep in mind when using Markdown is that HTML is still valid in Markdown. If there's ever something fancy you can't accomplish with just Markdown, it's okay to fall back to HTML.

The catch here is that you may be overcomplicating your life. If you need to fall back to plain HTML, there's a good chance that you could communicate whatever it is you are trying to say in a simpler format.

## Working with .md Files

Much like how your HTML files should be saved with a .html extension, your Markdown files should be saved with a .mdextension.

Markdown itself can't be opened in the browser like an HTML document. If you want to preview your Markdown files, [**Dillinger**](http://dillinger.io/) is a great online resource for you to do so.

If you are using Sublime Text, there is a [**plugin**](https://packagecontrol.io/packages/GitHub%20Flavored%20Markdown%20Preview) you can download to let you preview Markdown files right on your computer. If you are using Atom text editor, Markdown preview is baked right into the program (in the 'Packages' menu).

Here's a link to the basic [**Markdown documentation**](https://help.github.com/articles/markdown-basics/). You can also check out the differences in GitHub flavored Markdown [**here**](https://help.github.com/articles/github-flavored-markdown/).

To preview Markdown in the browser, try [**Dillinger**](http://dillinger.io/).

1. Markdown Syntax Practice

# My Fabulous Recipe

This recipe for \*\*cereal and milk\*\* has been passed down my family for months.

## Ingredients

\* Cereal (you can find cool cereals [here](www.example.com/coolcereals))

\* Milk

## Directions

If I were writing these out as \_code\_, it might look something like this:

```

if bowl is empty:

add cereal

if bowl only has cereal in it:

add milk

```

1. Document Everything!

Make some documents for previous code!