



University of  
**Sheffield**

**COM1001 SPRING SEMESTER**

Professor Phil McMinn

[p.mcminn@sheffield.ac.uk](mailto:p.mcminn@sheffield.ac.uk)

# Views

# What is a “View”?

**Controllers** **should not** contain any **HTML**, or any code for constructing the front-end of a web application.

Instead, the front-end of a Sinatra web application consists of **Views**.

**Views** are templates for constructing the HTML of a web page.

**Views** exist in files that are separate from the rest of the Sinatra application.

**Views** ensure that front-end presentation code is separated out from the rest of the web application. **Why do you think this might be important?**

# A Simple View:

## simple-view-example

```
get "/" do
  erb :index
end
```

week1/simple\_view\_example/  
controllers/controller.rb

```
<html>
<head>
  <title>A Simple View</title>
  <link rel="stylesheet" href="style/style.css">
</head>
<body>
  

  <h1>Simple View Example</h1>

  <p>Showing how to incorporate CSS and images into the
  pages of your web application.</p>
</body>
</html>
```

week1/simple\_view\_example/views/index.erb

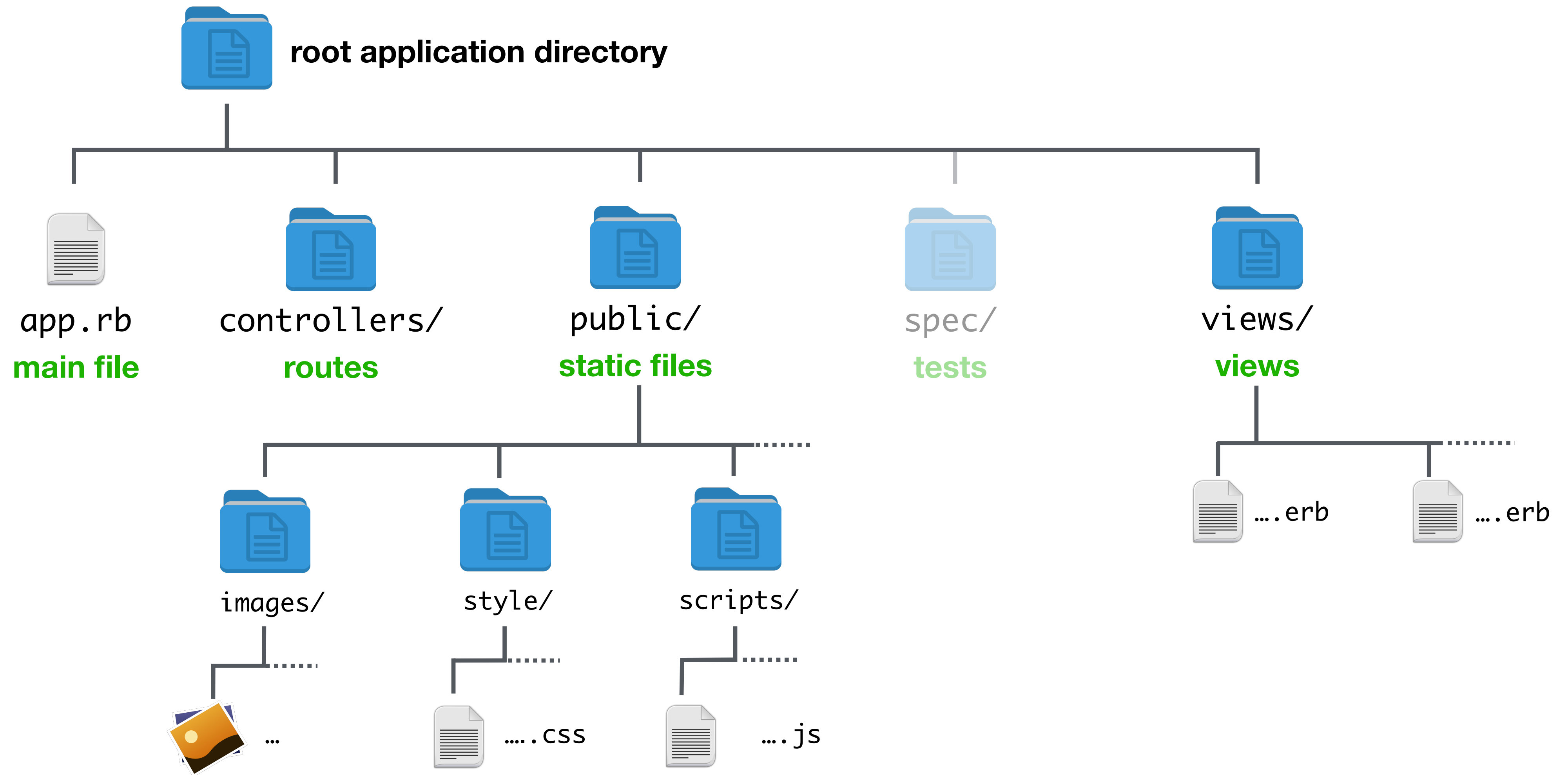
The **erb** method reads in the contents of the view and returns it as a string. →

This is an example of a Ruby **symbol**. A symbol is like a string, except it does not need to be enclosed in quotes and is prefixed with a colon. Symbols help simplify code.

The **erb** method uses a symbol to look up the file containing the view. Here symbol **:index** tells it to find the file **“views/index.erb”**. Rather than writing the whole filename in the code, we just provide the part that it needs (i.e., “index”), in the form of a symbol.

# The File Structure of a Sinatra Application

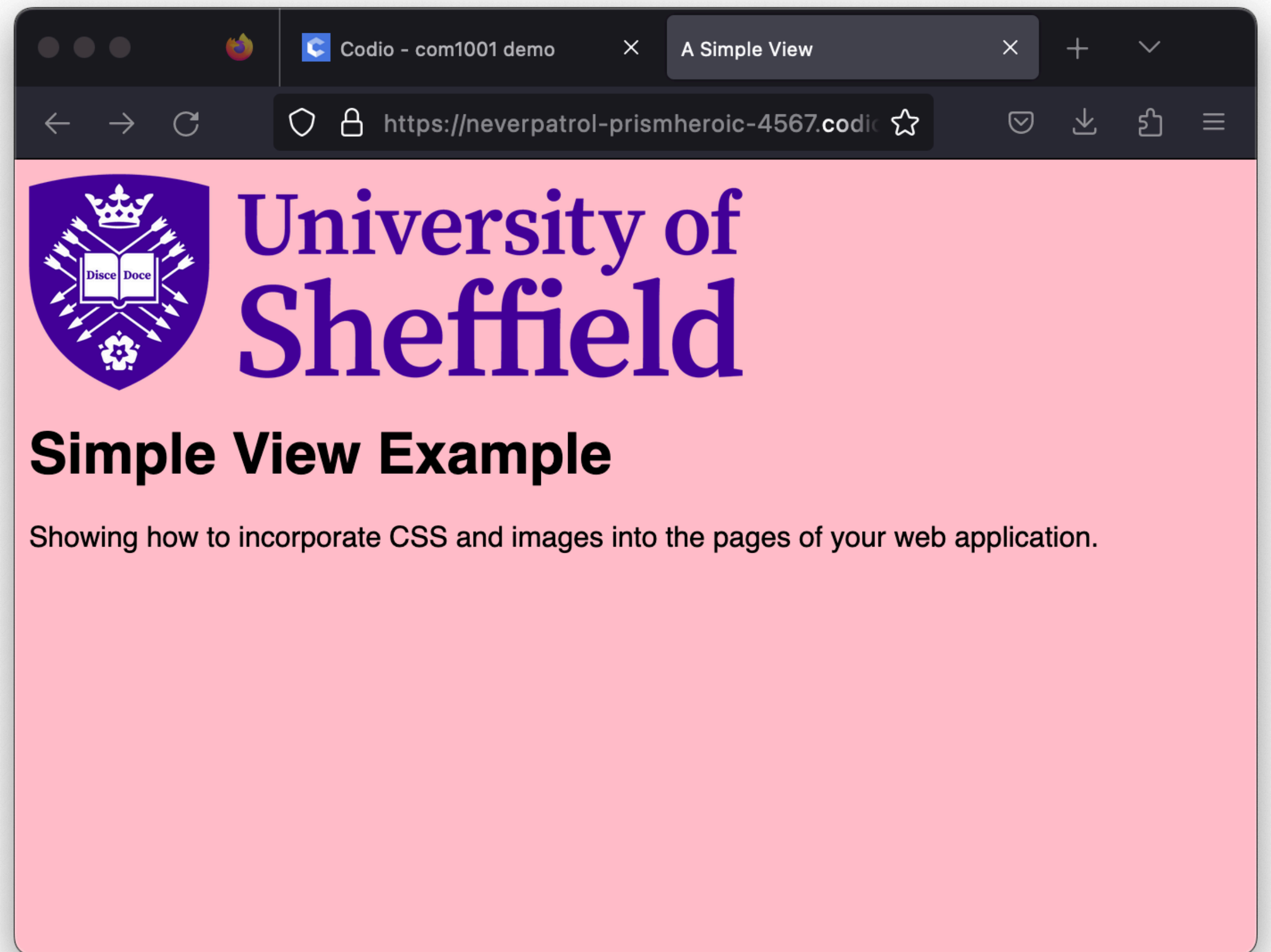
## (The Story So Far...)



```
<html>
<head>
  <title>A Simple View</title>
  <link rel="stylesheet" href="style/style.css">
</head>
<body>
  

  <h1>Simple View Example</h1>

  <p>Showing how to incorporate CSS and images into the
  pages of your web application.</p>
</body>
</html>
```



But this view isn't a "template" at all – *it's more like a regular static HTML file!*

**Views can be much more powerful...**





# Live Demonstration:

`week1/times_table_example`  
(from the COM1001 GitHub repository)

## Featuring:

- How to use variables inside views
- How to use control structures inside views
- The effect of puts

# Views – Summary

- **Views are HTML files that can include Ruby code using special tags:**
  - Anything between `<%` and `%>` can be Ruby code
  - `<%= var %>` injects the value of the variable `var` into the HTML of the page at that location
  - To use a variable from the controller, the variable's name needs to be prefixed with `@` (in the controller *and* the view).
- The inclusion of Ruby code in a view allows us to dynamically generate content.
- Views live in a special subdirectory of the application – `views/`