

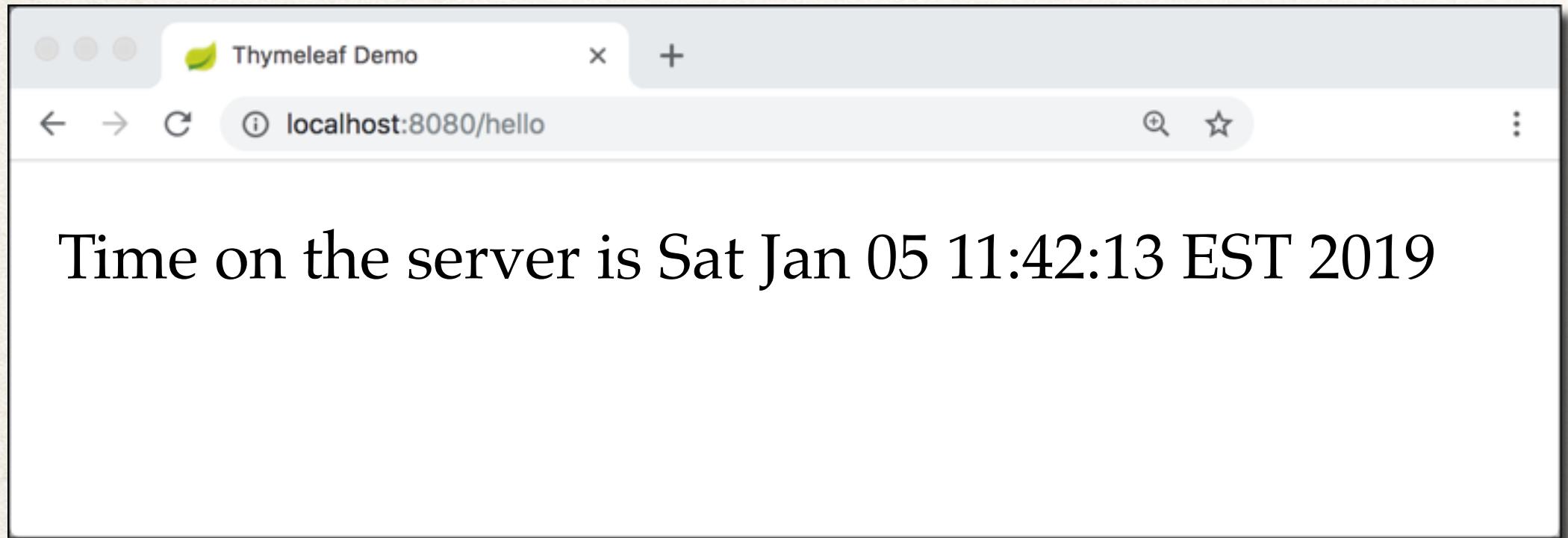
CSS and Thymeleaf



Let's Apply CSS Styles to our Page

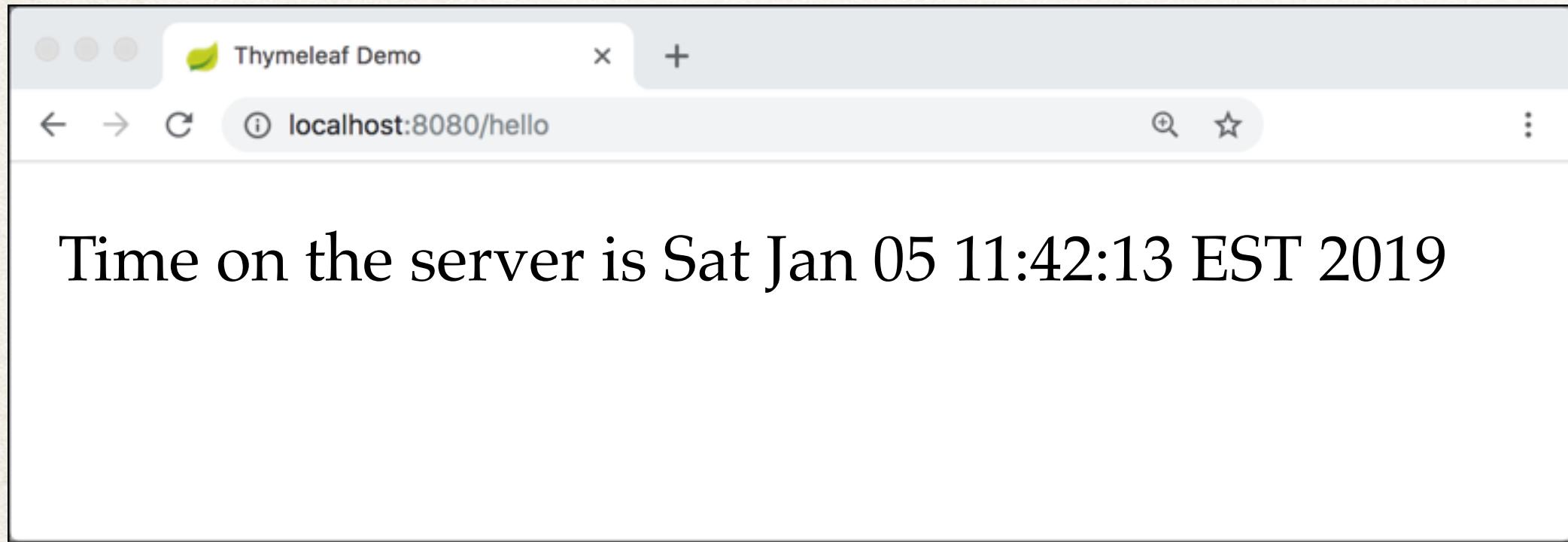
Let's Apply CSS Styles to our Page

Before

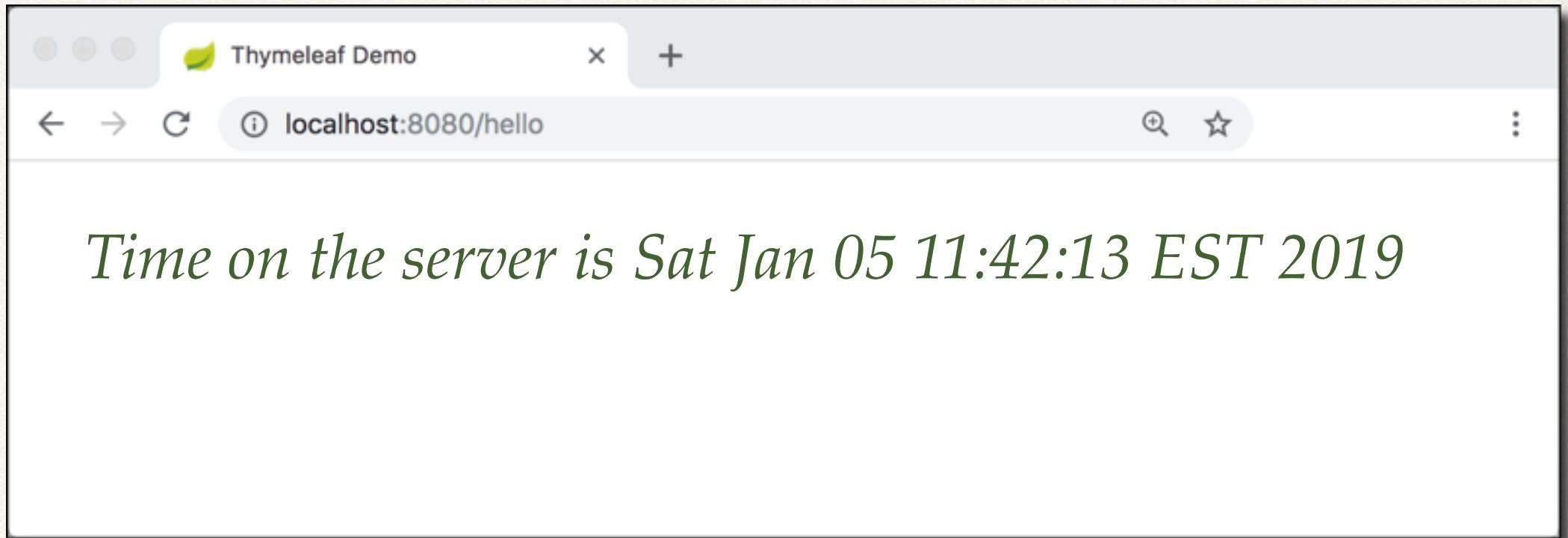


Let's Apply CSS Styles to our Page

Before

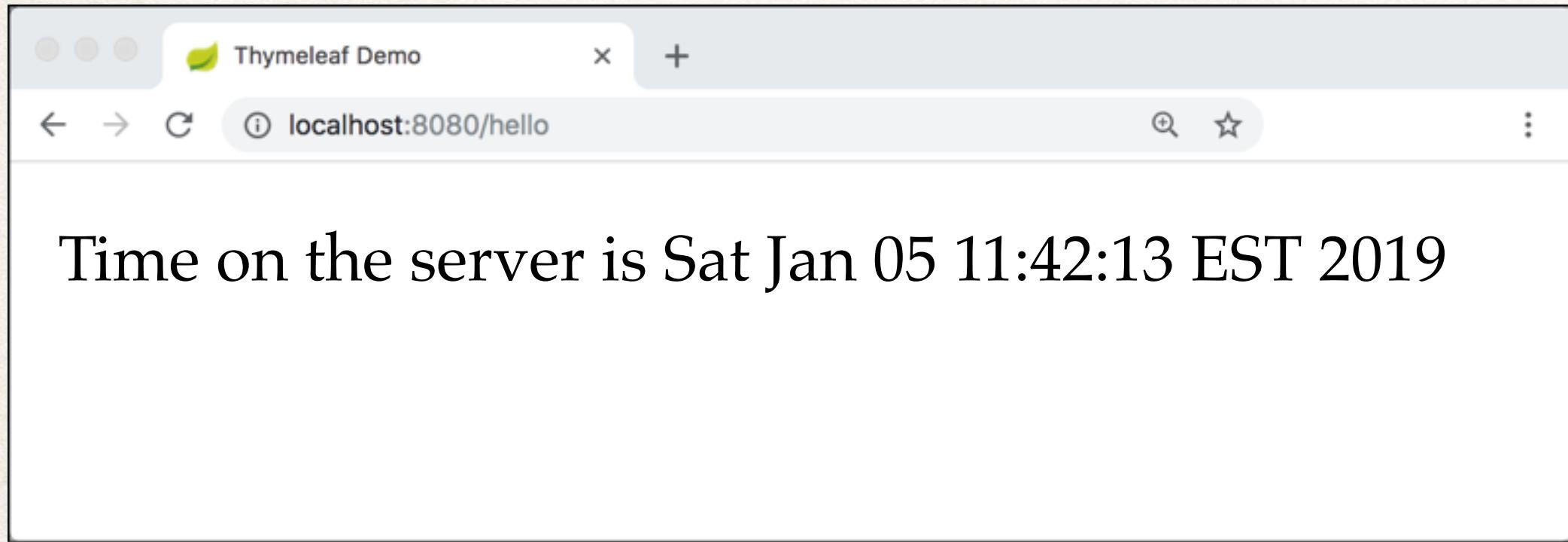


After

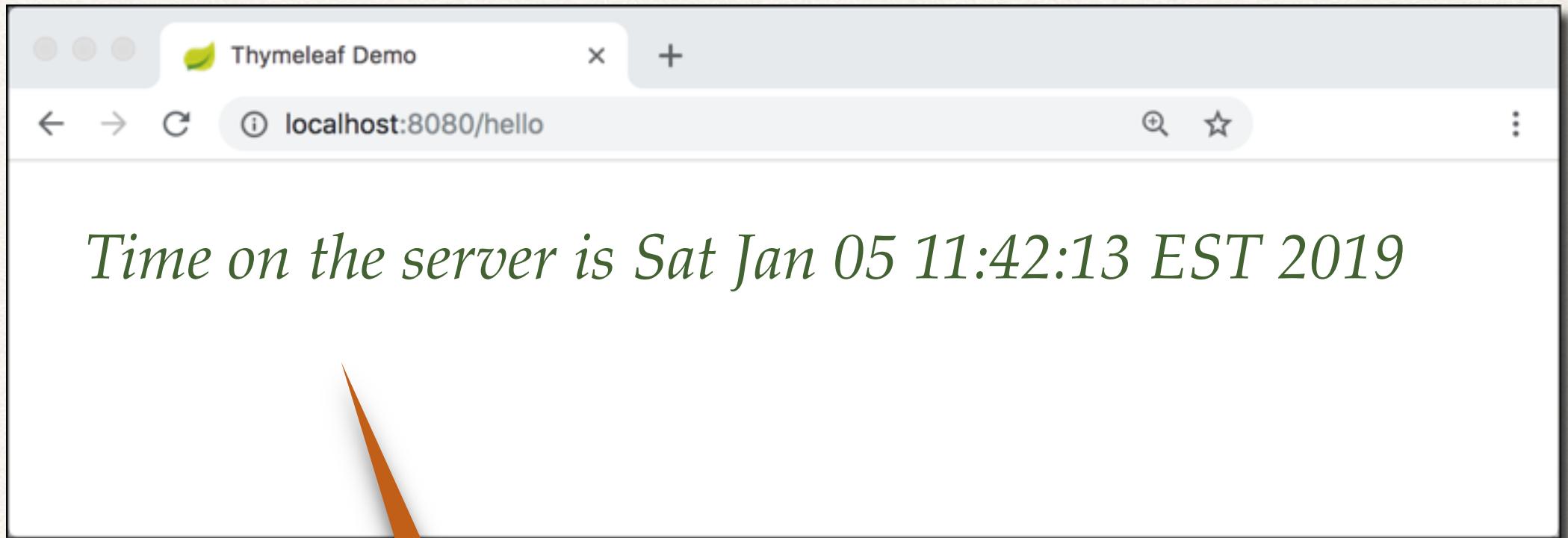


Let's Apply CSS Styles to our Page

Before



After



```
font-style: italic;  
color: green;
```

Using CSS with Thymleaf Templates

Using CSS with Thymleaf Templates

- You have the option of using

Using CSS with Thymleaf Templates

- You have the option of using
 - Local CSS files as part of your project

Using CSS with Thymleaf Templates

- You have the option of using
 - Local CSS files as part of your project
 - Referencing remote CSS files

Using CSS with Thymleaf Templates

- You have the option of using
 - Local CSS files as part of your project
 - Referencing remote CSS files
- We'll cover both options in this video

Development Process

Step-By-Step

Development Process

Step-By-Step

1. Create CSS file

Development Process

Step-By-Step

1. Create CSS file
2. Reference CSS in Thymeleaf template

Development Process

Step-By-Step

1. Create CSS file
2. Reference CSS in Thymeleaf template
3. Apply CSS style

Step 1: Create CSS file

Step 1: Create CSS file

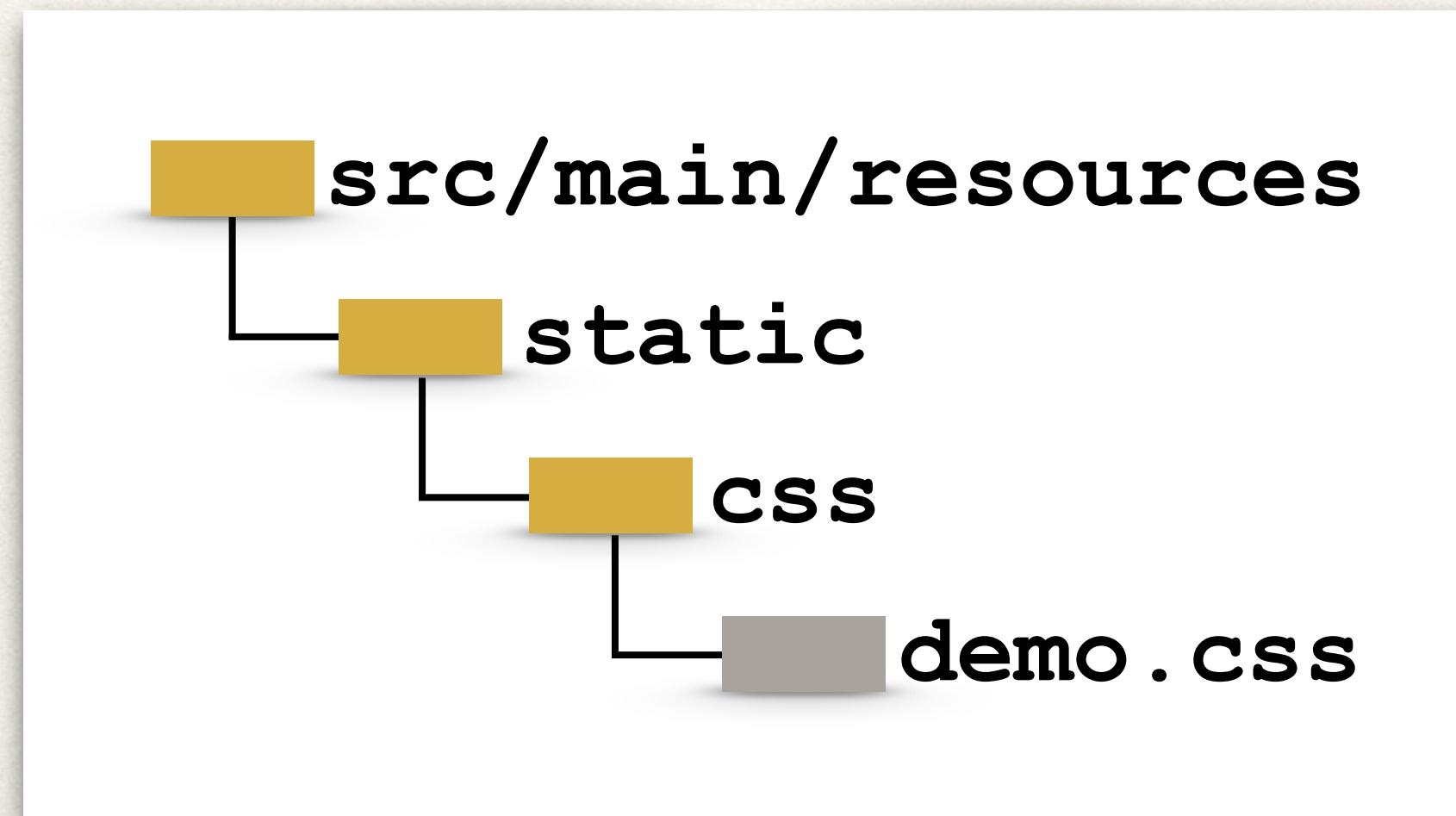
- Spring Boot will look for static resources in the directory

Step 1: Create CSS file

- Spring Boot will look for static resources in the directory
 - **src/main/resources/static**

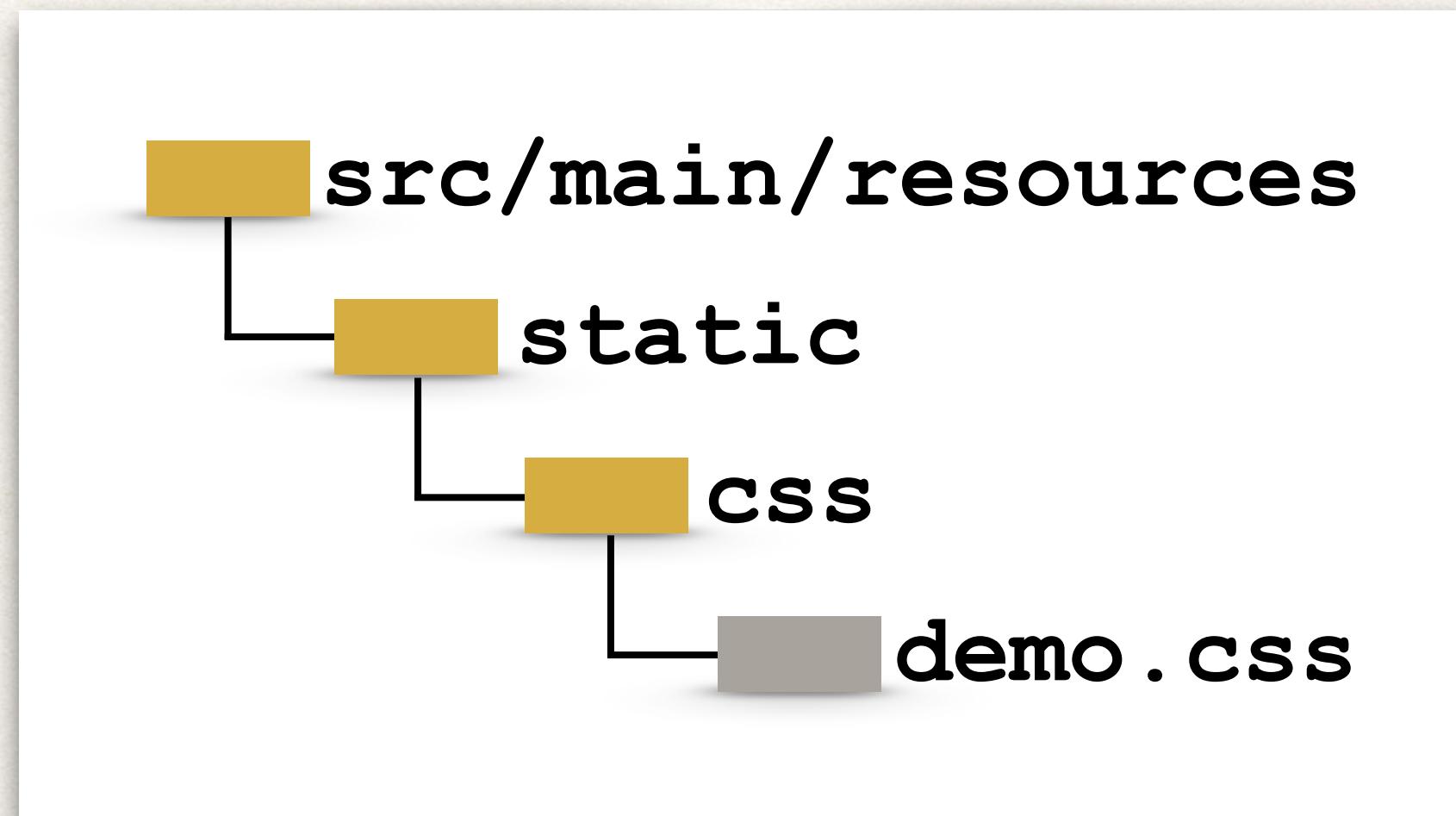
Step 1: Create CSS file

- Spring Boot will look for static resources in the directory
 - **src/main/resources/static**



Step 1: Create CSS file

- Spring Boot will look for static resources in the directory
 - **src/main/resources/static**

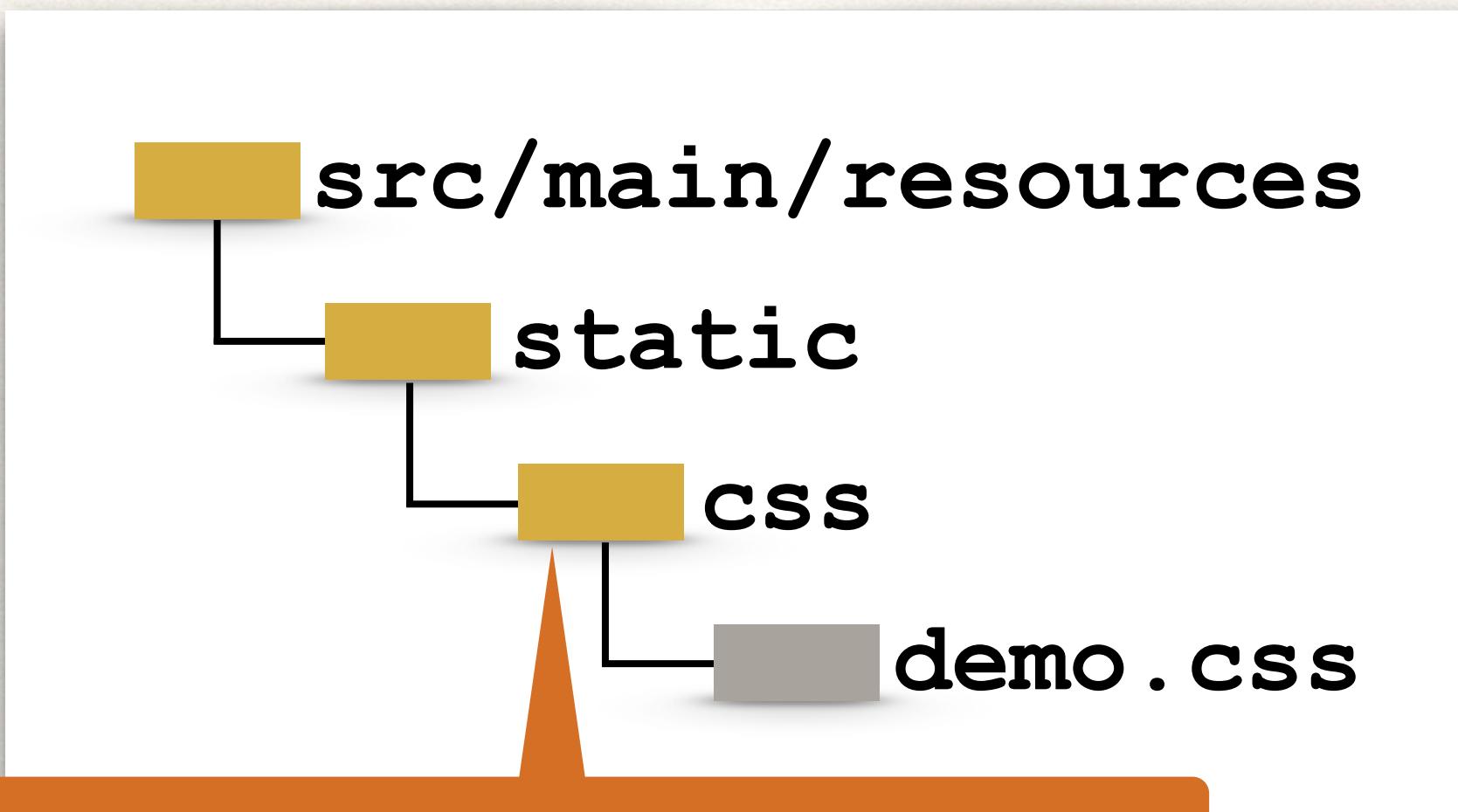


File: demo.css

```
.funny {  
    font-style: italic;  
    color: green;  
}
```

Step 1: Create CSS file

- Spring Boot will look for static resources in the directory
 - **src/main/resources/static**



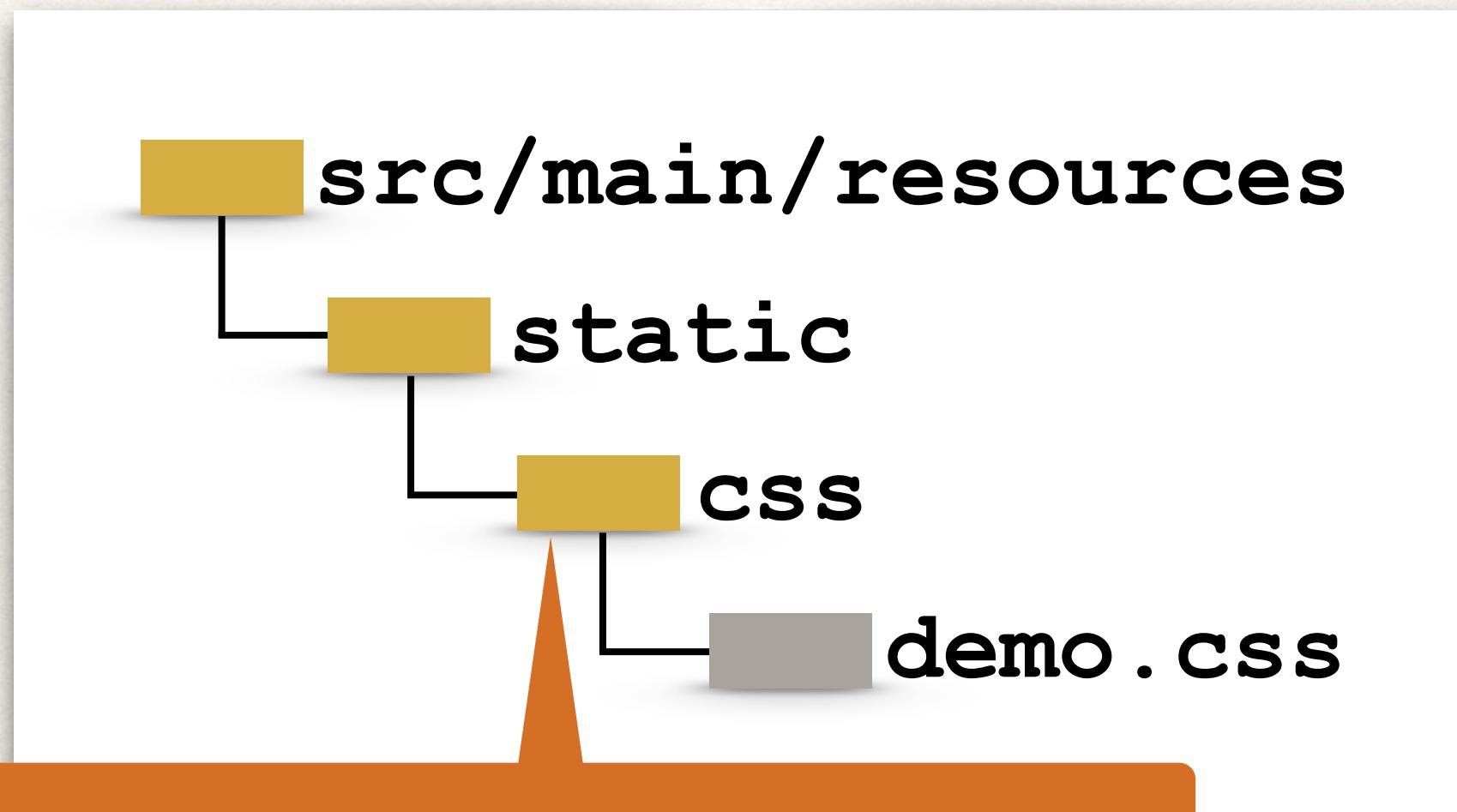
Can be any sub-directory name

File: demo.css

```
.funny {  
    font-style: italic;  
    color: green;  
}
```

Step 1: Create CSS file

- Spring Boot will look for static resources in
 - **src/main/resources/static**



Can be any sub-directory name

You can create your own custom sub-directories
static/css
static/images
static/js
etc ...

File: demo.css

```
.funny {  
    font-style: italic;  
    color: green;  
}
```

Step 2: Reference CSS in Thymeleaf template

Step 2: Reference CSS in Thymeleaf template

File: helloworld.html

```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />

</head>
```

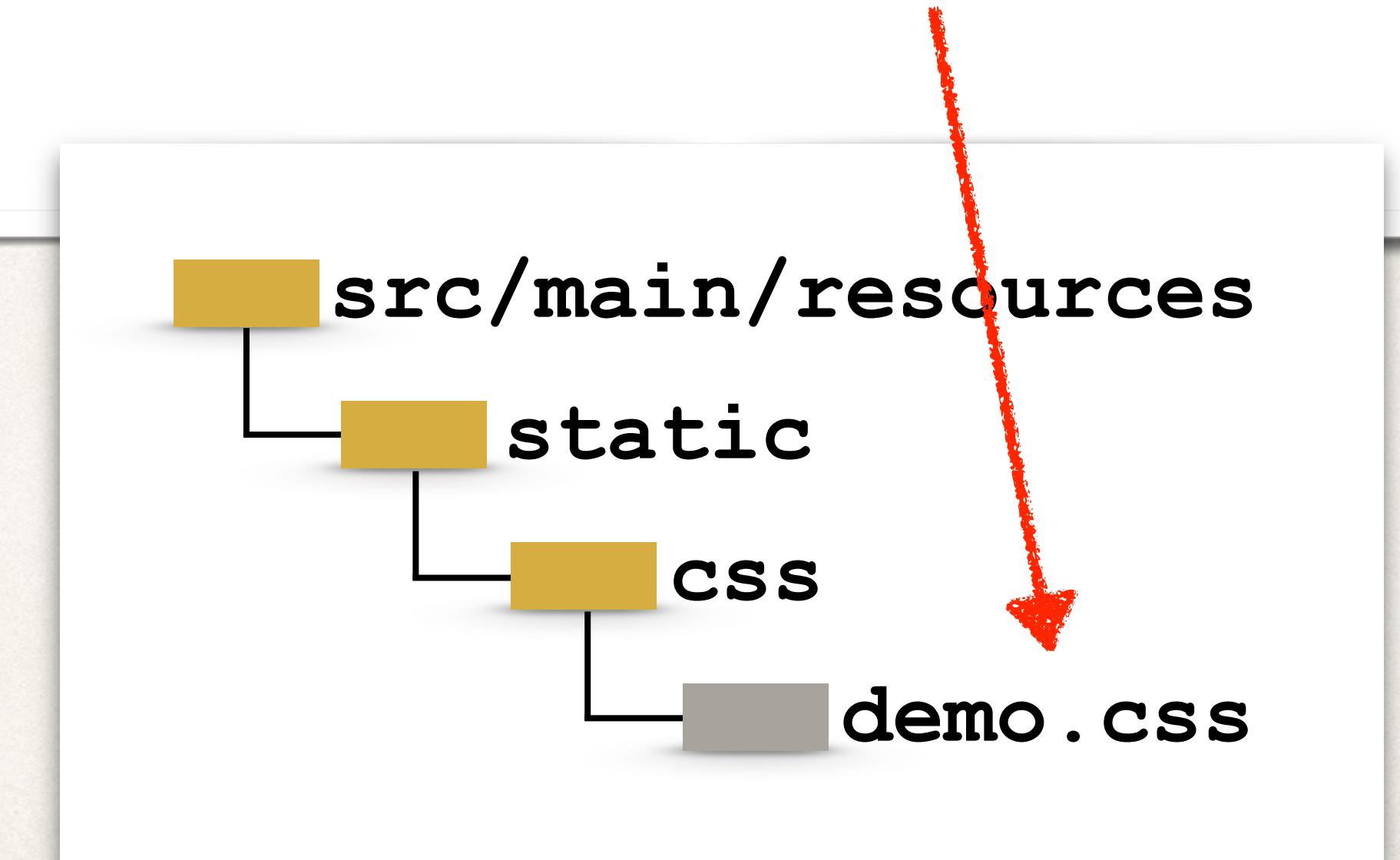
Step 2: Reference CSS in Thymeleaf template

File: helloworld.html

```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />

</head>
```



Step 2: Reference CSS in Thymeleaf template

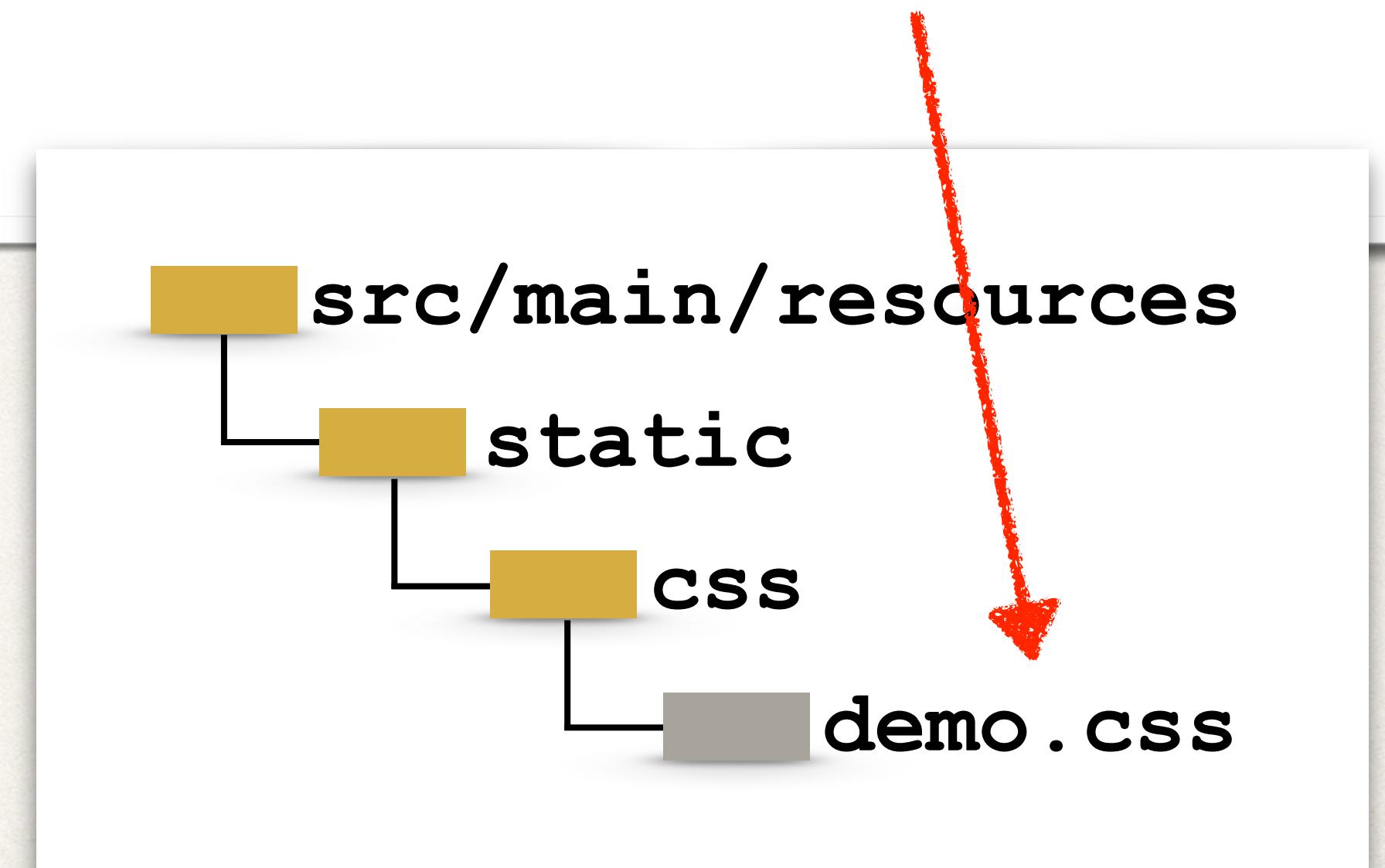
File: helloworld.html

```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />

</head>
```

@ symbol
Reference context path of your application
(app root)



Step 3: Apply CSS

Step 3: Apply CSS

File: helloworld.html

```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />
</head>

<body>
    <p th:text="Time on the server is ' + ${theDate}" class="funny" />
</body>
```

Step 3: Apply CSS

File: helloworld.html

```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />
</head>

<body>
    <p th:text="Time on the server is ' + ${theDate}" class="funny" />
</body>
```

File: demo.css

```
.funny {
    font-style: italic;
    color: green;
}
```

Step 3: Apply CSS

File: helloworld.html

```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />
</head>

<body>
    <p th:text="Time on the server is ' + ${theDate}" class="funny" />
</body>
```

File: demo.css

```
.funny {
    font-style: italic;
    color: green;
}
```

Step 3: Apply CSS

File: helloworld.html

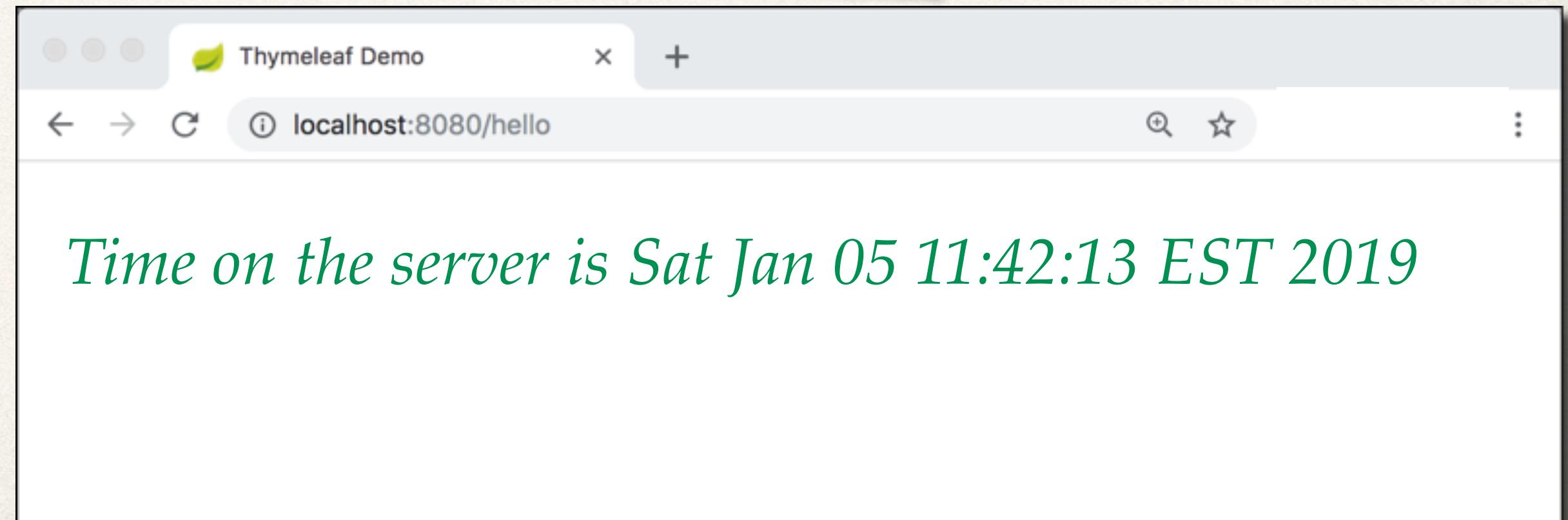
```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />
</head>

<body>
    <p th:text="Time on the server is ' + ${theDate}" class="funny" />
</body>
```

File: demo.css

```
.funny {
    font-style: italic;
    color: green;
}
```



Step 3: Apply CSS

File: helloworld.html

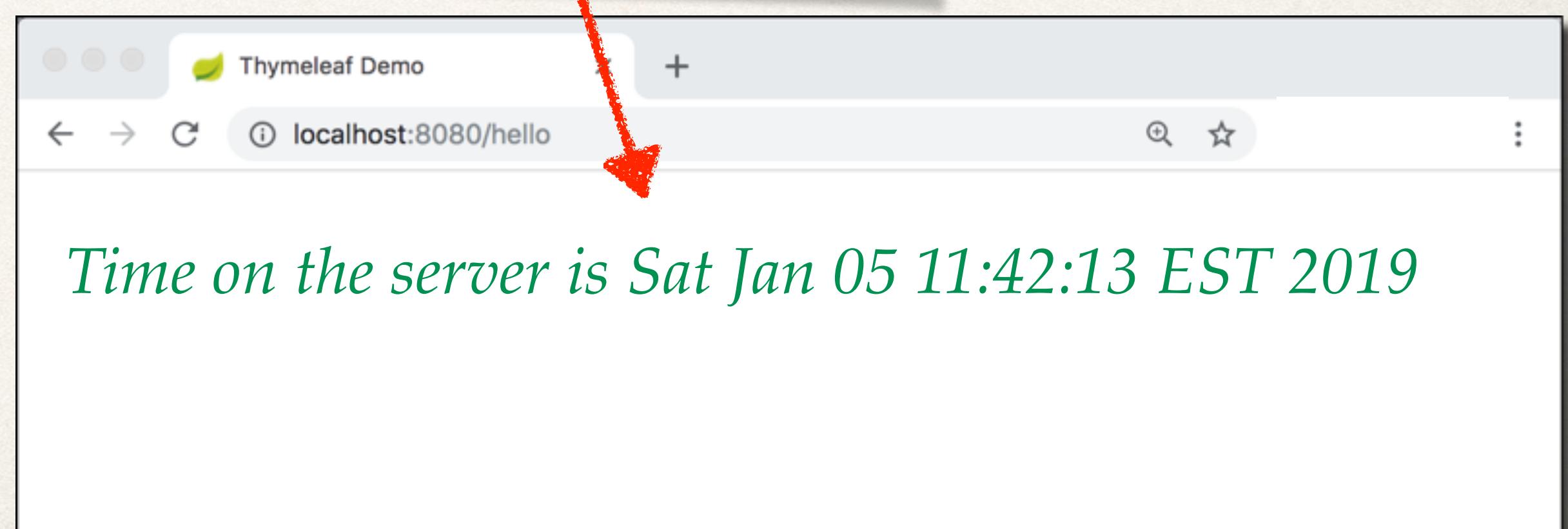
```
<head>
    <title>Thymeleaf Demo</title>

    <!-- reference CSS file -->
    <link rel="stylesheet" th:href="@{/css/demo.css}" />
</head>

<body>
    <p th:text="Time on the server is ' + ${theDate}" class="funny" />
</body>
```

File: demo.css

```
.funny {
    font-style: italic;
    color: green;
}
```



Other search directories

Other search directories

Spring Boot will search following directories for static resources:

/src/main/resources

Other search directories

Spring Boot will search following directories for static resources:

/src/main/resources

1. **/META-INF/resources**

Other search directories

Spring Boot will search following directories for static resources:

/src/main/resources

1. /**META-INF/resources**
2. /**resources**

Other search directories

Spring Boot will search following directories for static resources:

/src/main/resources

1. /**META-INF/resources**
2. /**resources**
3. /**static**

Other search directories

Spring Boot will search following directories for static resources:

/src/main/resources

1. **/META-INF/resources**
2. **/resources**
3. **/static**
4. **/public**

Other search directories

Spring Boot will search following directories for static resources:

/src/main/resources

1. /**META-INF/resources**
2. /**resources**
3. /**static**
4. /**public**



Search order: top-down

3rd Party CSS Libraries - Bootstrap

3rd Party CSS Libraries - Bootstrap

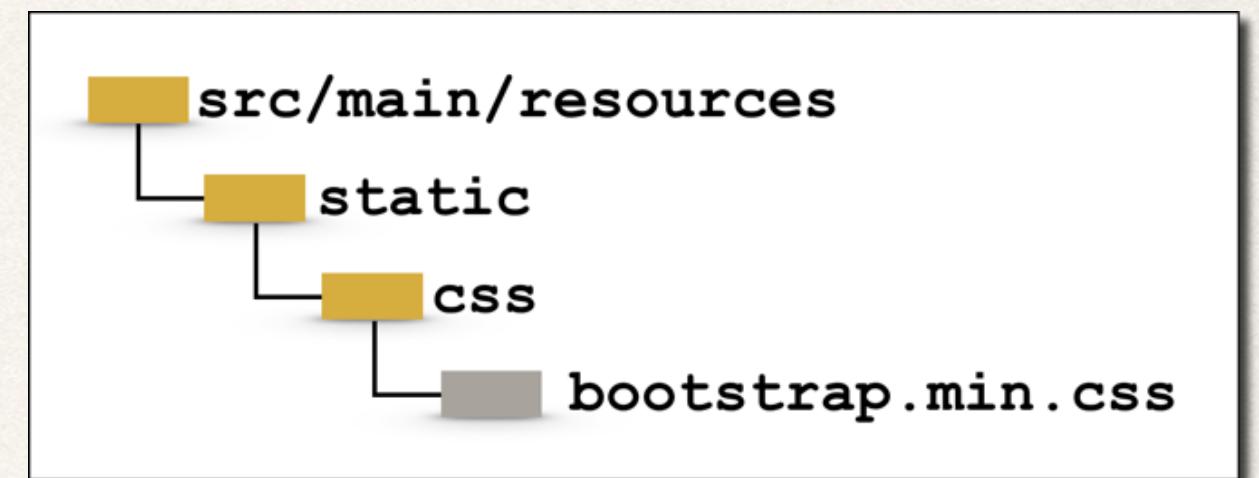
- Local Installation

3rd Party CSS Libraries - Bootstrap

- Local Installation
- Download Bootstrap file(s) and add to **/static/css** directory

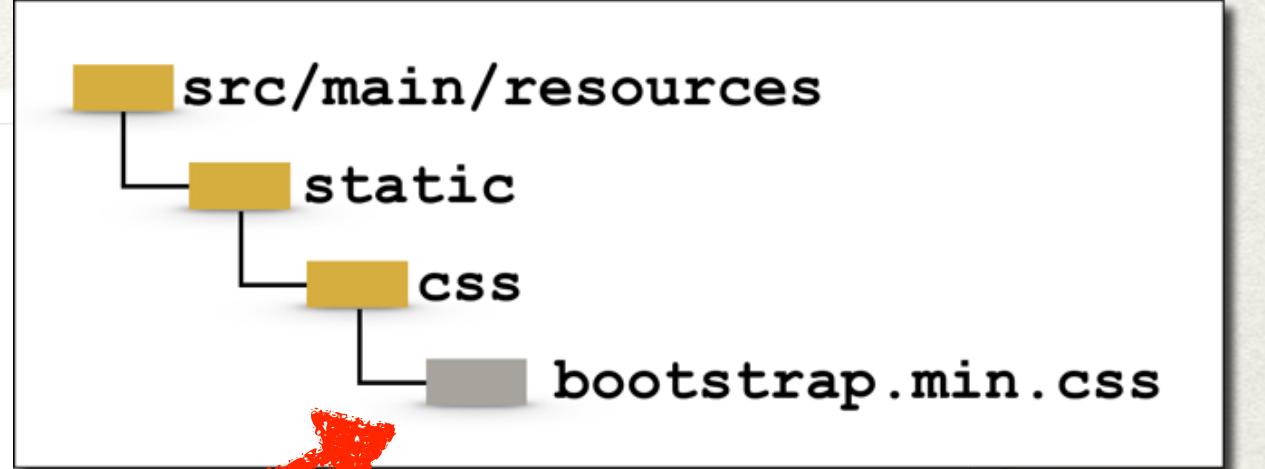
3rd Party CSS Libraries - Bootstrap

- Local Installation
- Download Bootstrap file(s) and add to **/static/css** directory



3rd Party CSS Libraries - Bootstrap

- Local Installation
- Download Bootstrap file(s) and add to **/static/css** directory



The diagram illustrates the directory structure for Bootstrap CSS files. It shows a tree with 'src/main/resources' at the root, which contains a 'static' folder. Inside 'static' is a 'css' folder, which contains a file named 'bootstrap.min.css'. A red arrow points from the 'bootstrap.min.css' file in the diagram to the corresponding line in the code snippet below.

```
<head>
...
<!-- reference CSS file --&gt;
&lt;link rel="stylesheet" th:href="@{/css/bootstrap.min.css}" /&gt;

&lt;/head&gt;</pre>
```

3rd Party CSS Libraries - Bootstrap

3rd Party CSS Libraries - Bootstrap

- Remote Files

3rd Party CSS Libraries - Bootstrap

- Remote Files

```
<head>
...
<!-- reference CSS file --&gt;
&lt;link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" /&gt;
...
&lt;/head&gt;</pre>
```