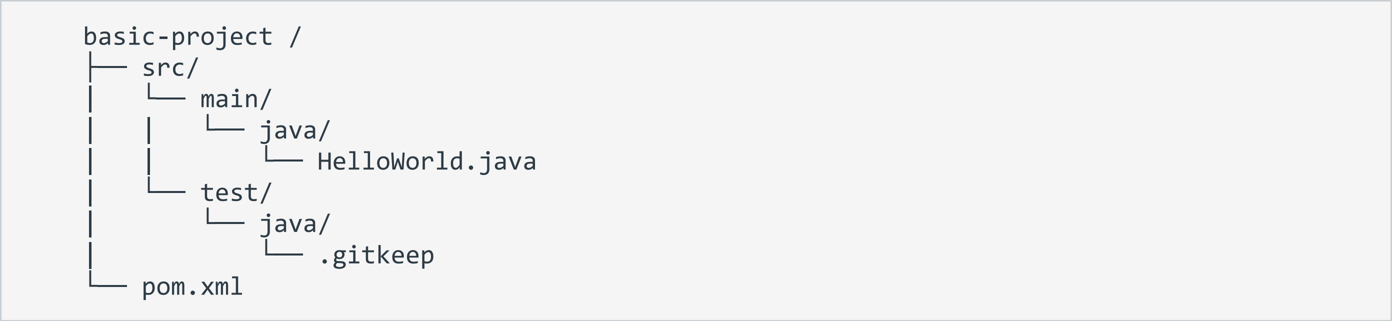
Exercise

As a reminder, all of your work for this exercise should be completed in the `LearnToCode/Workbook1/Mod01` folder.

EXERCISE 1

In this exercise, we will use CLI commands to create the file structure for a Java project. This exercise will demonstrate how to create a new Java project structure from the command line.

Step 1: The following diagram shows what this project directory will look like by the end of this exercise. Take a few minutes to study it.



```
basic-project /
├── src/
│   └── main/
│   │   └── java/
│   │       └── HelloWorld.java
│   └── test/
│       └── java/
│           └── .gitkeep
└── pom.xml
```

Note: Anything in the "tree" view above that ends with a forward-slash (`/`) is a *directory*, and everything else is a *file.*

Step 2: Create your project folder

Open Git Bash. Now, navigate (with `cd`) to your `LearnToCode/Workbook1/Mod01` folder. Confirm you are in the proper directory by running the `pwd` command.

Once you know you are in `Mod01` folder, create a folder named `basic-project`. Finally, `cd` into the project directory.

```
$ cd /c/LearnToCode

$ pwd

$ cd Workbook1

$ pwd

$ cd Mod01

$ pwd

$ mkdir basic-project

$ cd basic-project
```

NOTE: If you see a `$` at the beginning of a line like this in this course, on the Web, or in a book, it refers to your command prompt. You should not *type* that symbol yourself. Just type whatever follows the `$`.

Step 3: **`touch`** (create) your first file!

Now that you are in the `basic-project` directory, create an `index.html` file using the `touch` command. You can use the `pwd` command first to make sure you are in the correct folder.

```
$ pwd

$ touch pom.xml
```

NOTE: It is always good to use `pwd` and `ls` frequently as you change directories, touch files, or make directories, so that you can keep track of where you are and what changes you've made.

```
$ pwd

$ ls
```

Step 4: Make more directories!

Make the rest of the directories described in Step 1.

```
$ mkdir src

$ mkdir src/main

$ mkdir src/main/java

$ mkdir src/test

$ mkdir src/test/java
```

Use `ls` to examine the changes.

The file structure should now look something like this:

```
    basic-project/
    ├── src/
    └── pom.xml
```

Step 6: Add a `HelloWorld.java` file to the `src/main/java` directory.

Use the `cd` command to change into the src/main/`java` folder. Then, create `HelloWorld.java`.

```
$ cd src/main/java

$ pwd

$ touch HelloWorld.java
```

Step 7: Add a file named `.gitkeep` to the `src/test/java` directory.

NOTE: You will learn more about this `.gitkeep` file in the next module

Use the `cd` command to change into the `test/java` folder. You will need to navigate to the `src/test.java` folder first. Then, create `.gitkeep`.

```
$ cd ../..

$ pwd

$ cd test/java

$ pwd

$ touch .gitkeep
```

Again, we will want to use `ls` to examine the changes.

Step 8: Prepare to code by opening IntelliJ IDEA.

You have now created the typical project structure of a Java application.

Start IntelliJ IDEA and select Projects -> Open. In the file browser, select the `basic-project` directory and open it. Your project appears in the IDE, and you can now find your java source file and edit it.