

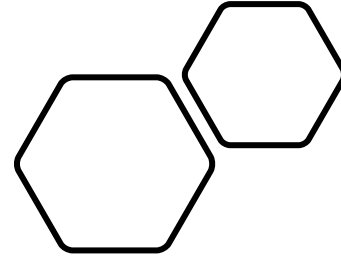


# BridgeLabz

Employability Delivered

Junit with  
IntelliJ Intro

# IntelliJ Get Started

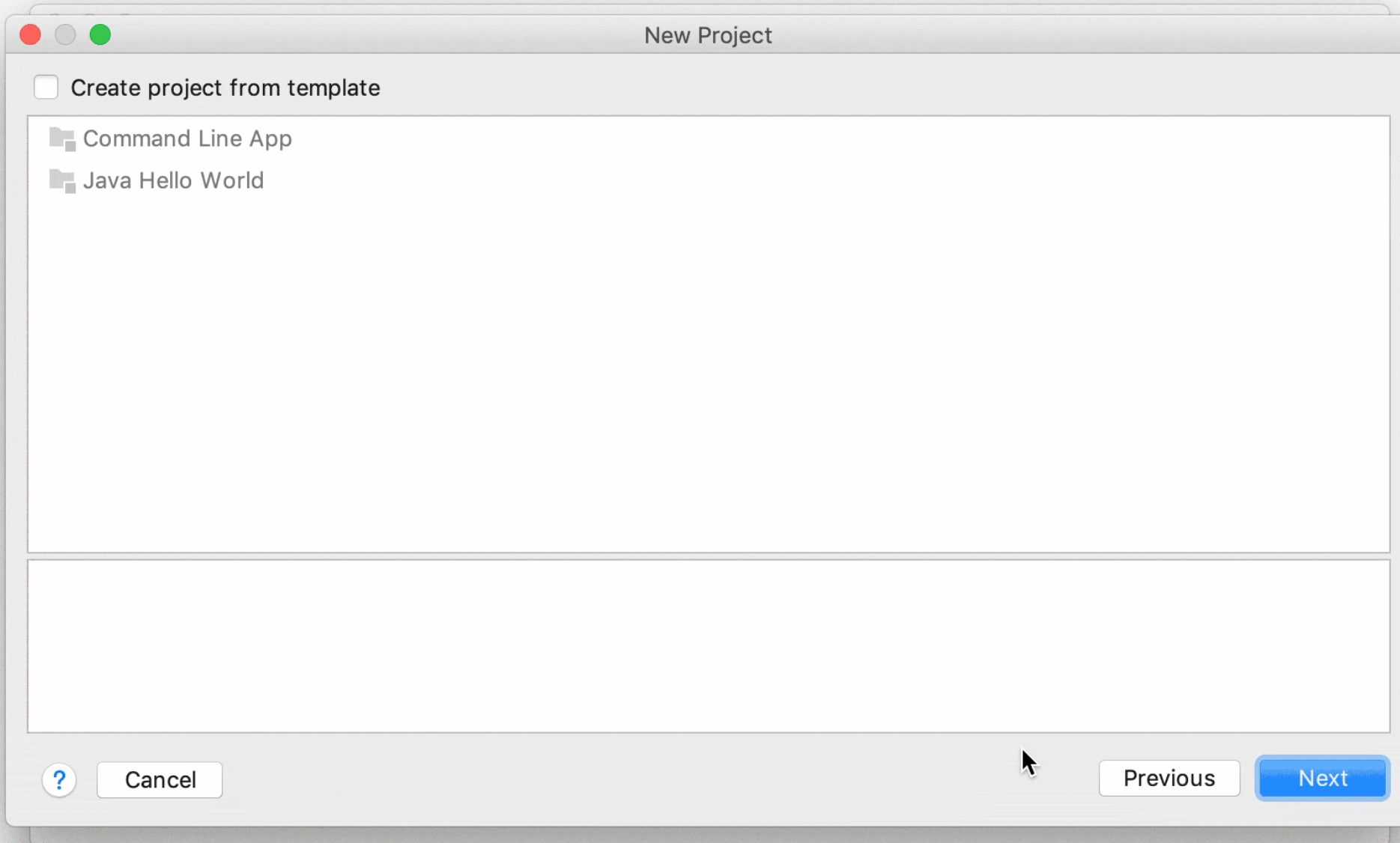


Create your first  
Java HelloWorld  
Application

# Create New Project - Set Java Version



# Create New Project – Set Project Name



New Project

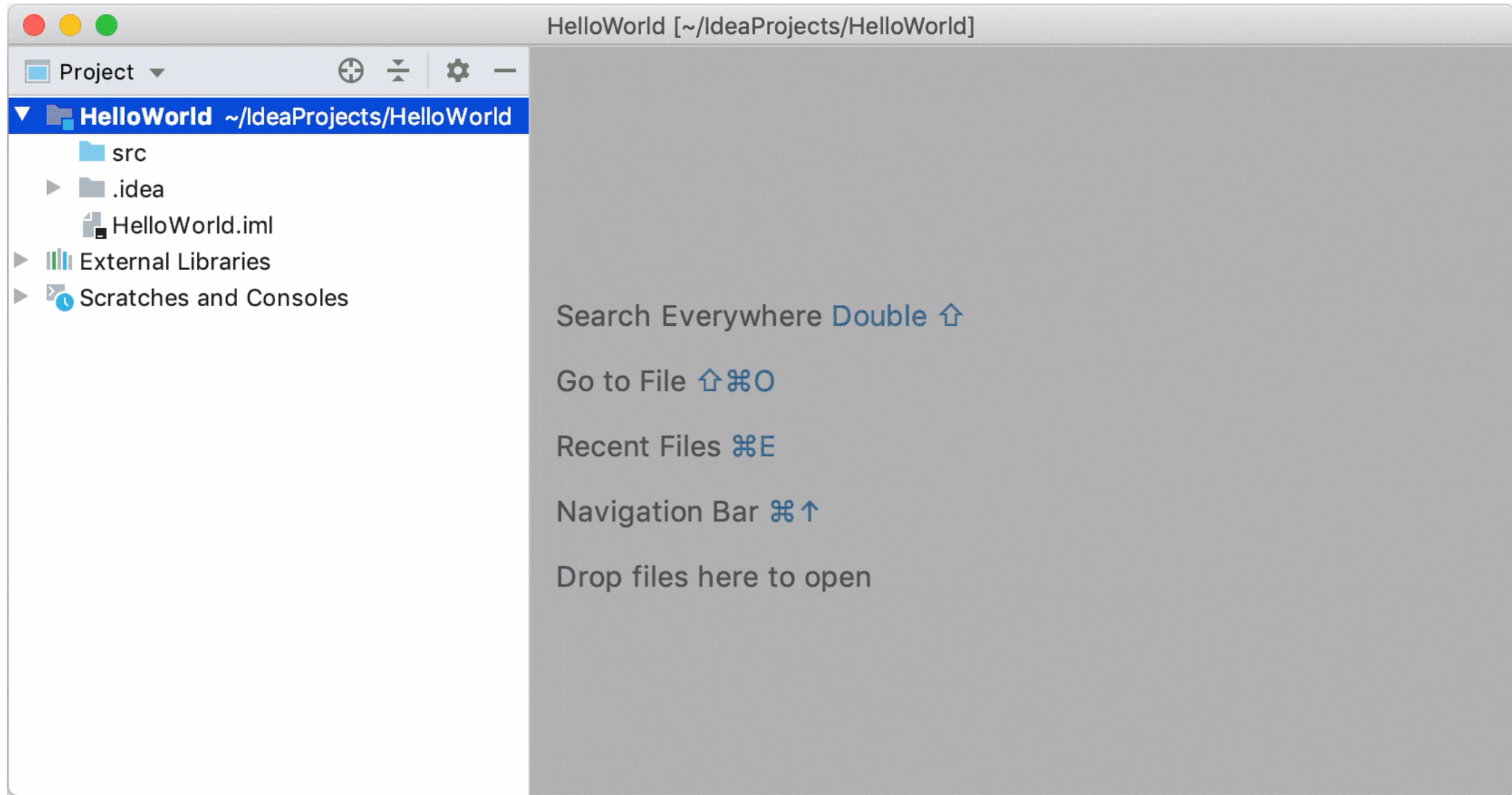
☐ Create project from template

- Command Line App
- Java Hello World

? Cancel Previous Next



# Intro to IntelliJ – Create HelloWorld Java File



# Intro to IntelliJ – Write HelloWorld Program

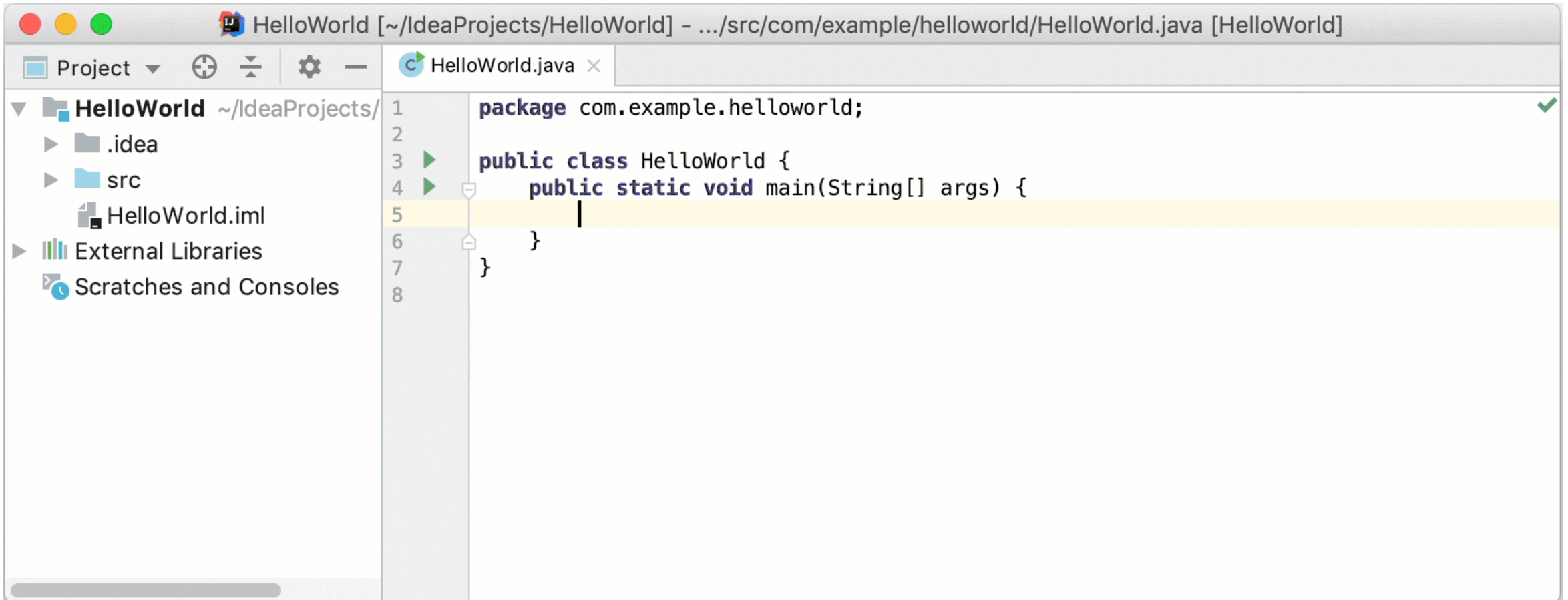


The screenshot shows the IntelliJ IDEA IDE interface. The title bar indicates the project is 'HelloWorld' and the file being edited is 'HelloWorld.java'. The left sidebar shows the project structure with folders '.idea', 'src', and 'com.example.helloworld', and files 'HelloWorld' and 'HelloWorld.iml'. The main editor window displays the following Java code:

```
1 package com.example.helloworld;
2
3 public class HelloWorld {
4 }
```

The status bar at the bottom indicates the current class is 'HelloWorld'.

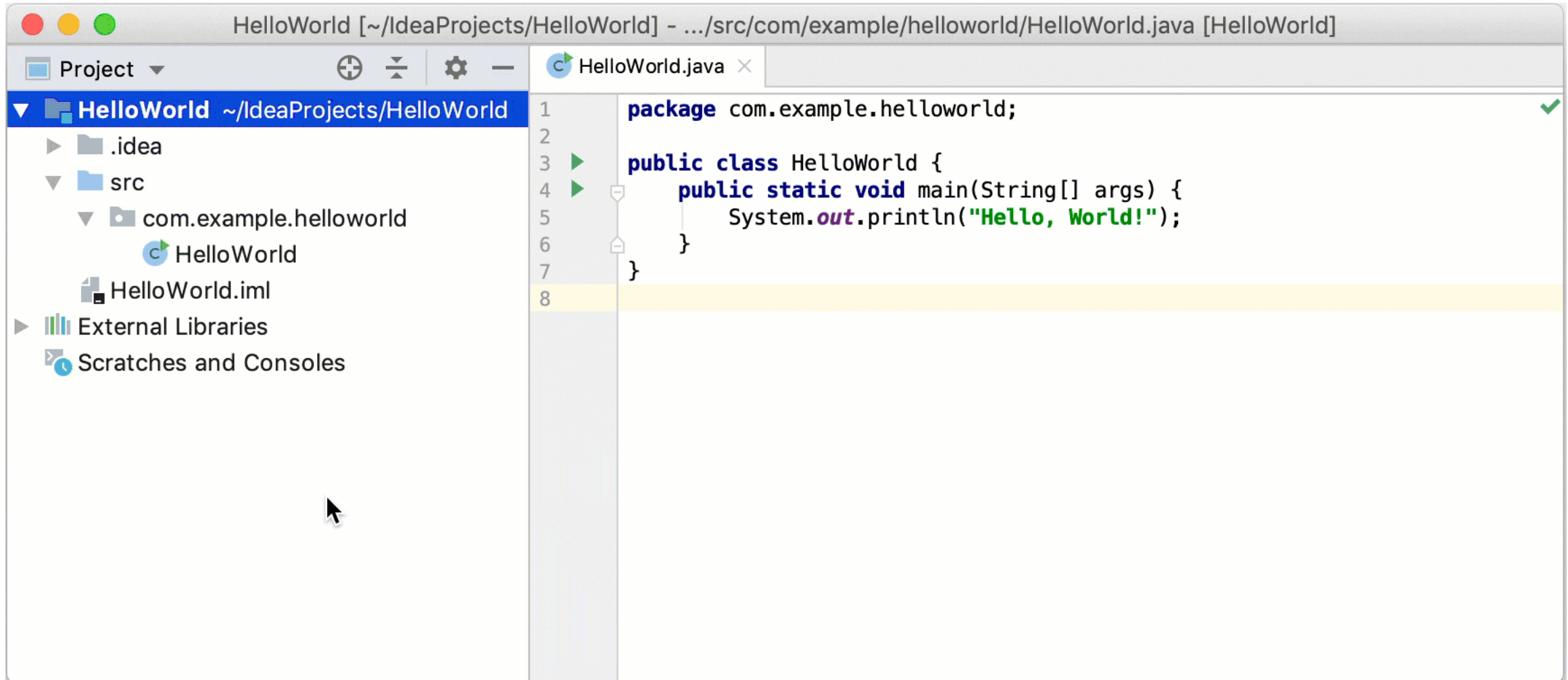
# Intro to IntelliJ – Smart Lookup



```
1 package com.example.helloworld;
2
3 public class HelloWorld {
4     public static void main(String[] args) {
5     }
6 }
7
8
```



# Intro to IntelliJ – Run HelloWorld Program





# Intro to IntelliJ – Create New Gradle Project

- **Gradle** is a build automation tool that allows to cache dependency and download them in parallel.
- Primarily **used** for JVM languages such as Java, Groovy or Scala.
- **Gradle** can be configured to run Tasks which do things like compile jar s, run tests, create documentation and much more.

## Create a new Gradle project

1. Launch the [New Project wizard](#). If no project is currently opened in IntelliJ IDEA, click **Create New Project** on the welcome screen. Otherwise, select **File | New | Project** from the main menu.
2. Select **Gradle** from the options on the left.
3. Specify the project SDK and an additional framework or a library (IntelliJ IDEA adds the appropriate plugin to the `build.gradle` file). Click **Next**.

To see how IntelliJ IDEA selects the installed Gradle JVM version, refer to the [Gradle JVM selection](#) section.

4. On the next page of the wizard, specify the fields which resemble [Maven naming conventions](#) ↗. These settings might be helpful if you decide to deploy your project to a Maven repository. The fields you specify are added to the `build.gradle` file.
  - **GroupId** - `groupId` of the new project. You can omit this field if you plan to deploy your project locally.
  - **ArtifactId** - `artifactId` that is added as a name of your new project.
  - **Version** - `version` of the new project. By default, this field is specified automatically.

If a parent Gradle project is specified, the specified fields can be inherited (click **Inherit**) from the parent.  
Click **Next**.

5. Specify the name and location settings. Click **Finish**.



# BridgeLabz

Employability Delivered

# Thankyou