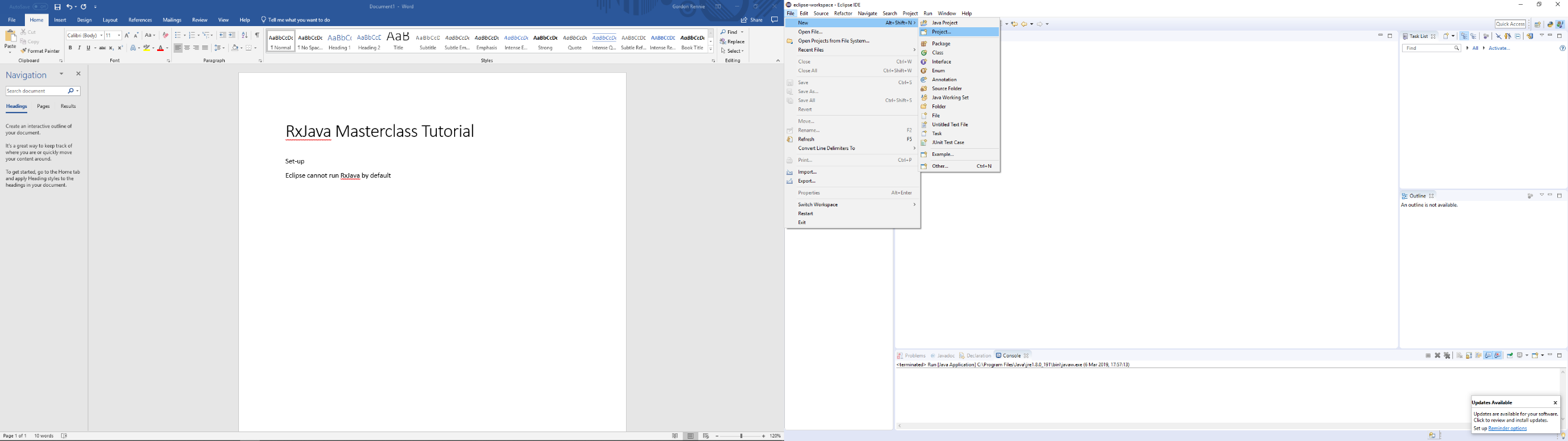
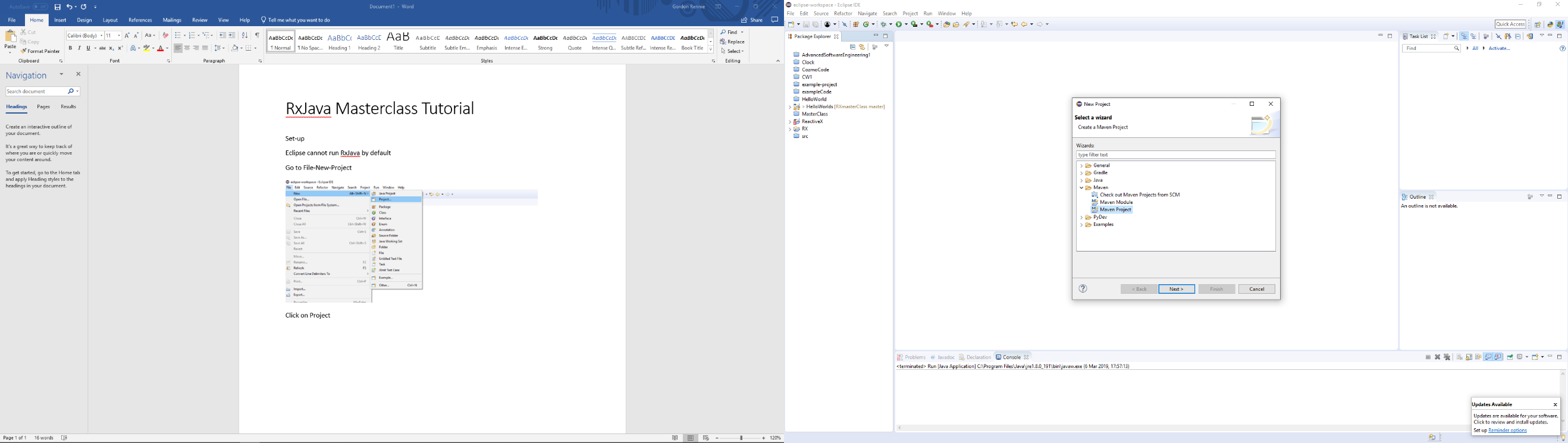
RxJava Masterclass Tutorial

Set-up

Eclipse cannot run RxJava by default

Go to **File-New-Project**

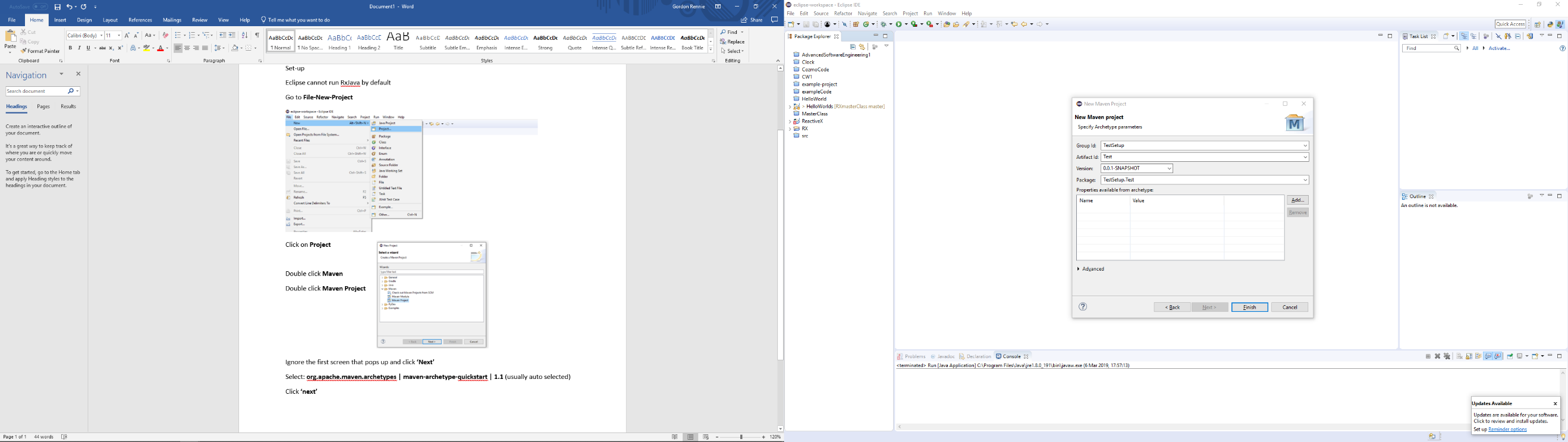
Click on **Project**

Double click **Maven**

Double click **Maven Project**

Ignore the first screen that pops up and click **‘Next’**

Select: **org.apache.maven.archetypes | maven-archetype-quickstart | 1.1** (usually auto selected)

Click **‘next’**

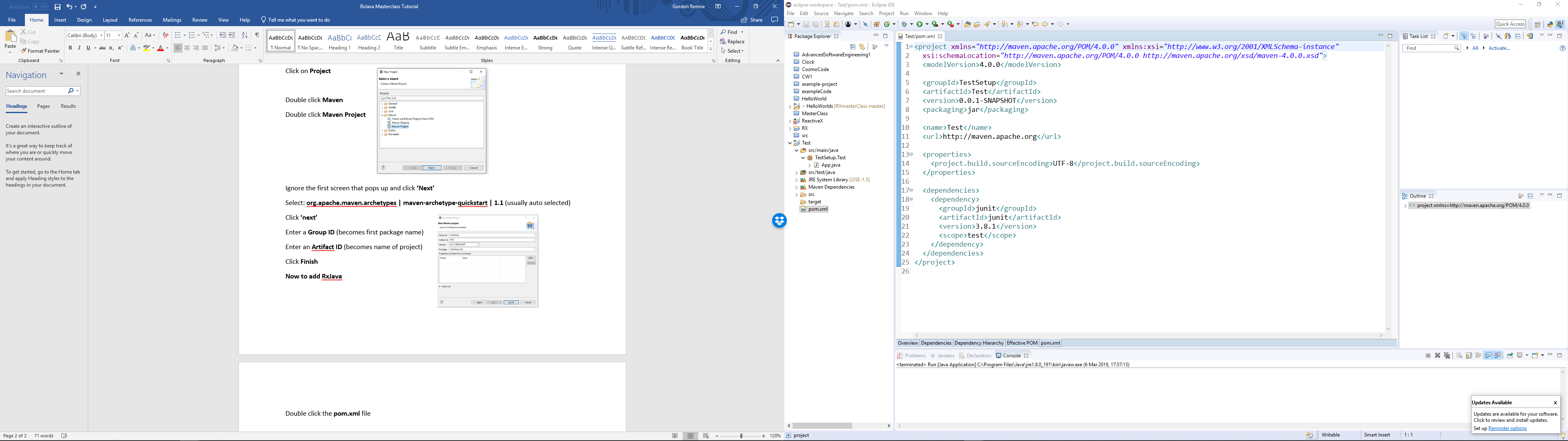
Enter a **Group ID** (becomes first package name)

Enter an **Artifact ID** (becomes name of project)

Click **Finish**

**Now to add RxJava**

Double click the **pom.xml** file. Yours should look something like below.



To make RxJava work we add it to the dependencies. Copy the following code. It comes from the central repository, a big store of loads of dependencies that can be added to a project – the code below is the one you’ll need.

<dependency>

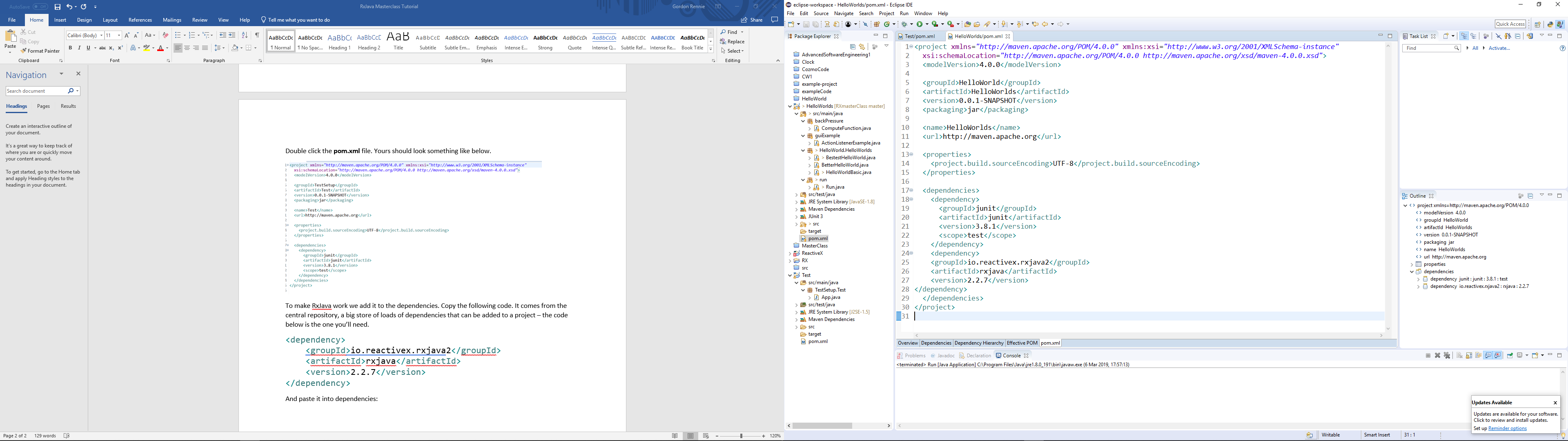
<groupId>io.reactivex.rxjava2</groupId>

<artifactId>rxjava</artifactId>

<version>2.2.7</version>

</dependency>

And paste it into dependencies. Make sure your XML is right, if you’re unsure – ask.



Now you’re ready to get started.

The actual RxJava Tutorial

# Practicing Operators

1. Basics

Set up a basic observable which emits ten numbers.

Add a time delay to your emission

Filter this observable so that only even numbers are printed

1. Tell me the time

Create an observable that emits the current time formatted as HH:MM:SS

Then make it emit the time every second for ten seconds

Create a second observable that emits the current Unix time

Use operators to make an observable that emits both normal and Unix time