General Instructions for Running a JAR file

Tutorial

All instructions are for a Windows machine.

Save the JAR file (I'll call it `myfile.jar`) to your Desktop

Open a Terminal and Navigate to the Desktop

1. From the `Start` button (lower left), in `Search Programs and Files` type 'cmd' (without the single quotes) and click on the first item (it will be either `cmd` or `cmd.exe`). This should open up a window that has some text ending in your name followed by a >. It looks something like this:

   <div align="center">

   `C:\Users\aritz>`

   </div>

   This is called a *command prompt* - you can use this to type various commands to the computer that can open, move, and edit files, as well as run programs (among other things).

2. You are in your *home directory*. Type 'dir' (for 'directories') and hit Enter. What do you see?

3. 'Desktop' should be a directory in this list. We want to move to that directory. You can do that by using the 'cd' (for 'change directory') command. Type

   <div align="center">

   `cd Desktop`

   </div>

   and hit Enter. The line should now look something like

   <div align="center">

   `C:\Users\aritz\Desktop>`

   </div>

   You are now in the 'Desktop' directory, which contains all the files and folders you see on your actual Desktop. By typing 'dir', you should see the JAR file `myfile.jar`.

Run the JAR File with No Arguments

1. From the 'Desktop' directory, you can now run `myfile.jar`. At the command prompt, type:

   `java -jar myfile.jar`

   but replace `myfile.jar` with the name of the JAR file you want to run. You should get some text that explains what the JAR file does and what the JAR file expects. For example, the XML-to-CSV converted expects TWO things: (1) an existing XML file and (2) a file that will contain the converted CSV text.

Copy or Move Necessary Files to your Desktop (i.e. an XML file to convert)

Run the JAR file with Arguments

1. From the 'Desktop' directory, you can now run `myfile.jar` with the arguments it expects. At the command prompt, type:

   `java -jar myfile.jar <arg1> <arg2> ...`

   where `<arg1>` is the first thing the program expects, `<arg2>` is the second thing the program expects, etc.