**ImageJ** is an image processing software written entirely in Java, open source, with an architecture designed to be easily extended through plugins. Plugins are the most common method used by developers and users to extend ImageJ's features, such as image analysis and editing, so that they can be easily implemented.

A plugin essentially consists of a .class file, possibly with an associated Java source file. To load a new plugin into ImageJ, simply copy the .class file into the "*ImageJ - Plugins*" directory and restart

the software.

If necessary, this folder can be divided into subgroups, so that plugins can be divided by type. The new plugin will then be automatically available in the menu when the programme is started.
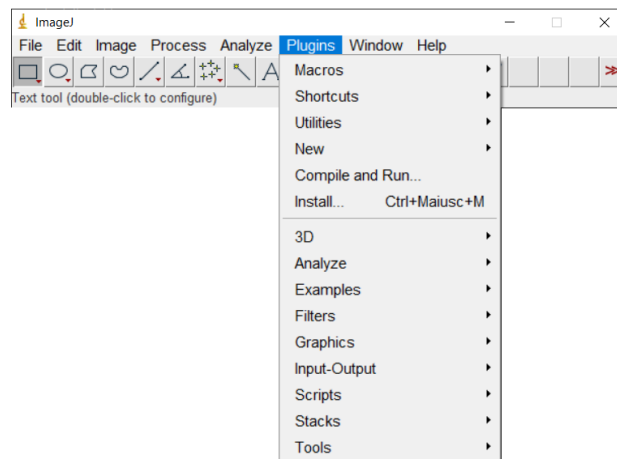
In addition to their application in ImageJ, plugins can be implemented in a web page via Java applets, so pages with iterative functions can be created for any user. But you must be careful to have a browser that is Java-enabled to run ImageJ in this way!

Step 1 – Edit archive "*ij.jar*":

Create a copy of the "*ij.jar*" file in the ImageJ installation folder.

In this archive, which can be opened with any decompression software, insert the .class file of your plugin (or the package containing it). Use the folder of the .jar archive as the working path. Then, within the .jar, you must edit the file "*IJ_Props.txt*", adding the following to the line

<div align="center">

#plug-in08=>"User_Plugins

</div>

a new line with the content

<div align="center">

plug-in10="Nome_Plugin",Nome_FileClass

</div>

and save the file.

Step 2 – Creation file keystore:

First, open the command prompt and go to the bin subfolder of the path where your Java Runtime Environment is installed. If it is not yet installed, you can easily download it from the [website](#).

Next, issue the following command, choosing your own alias instead of <namealias>:

keytool -genkey -validity 600 -alias <nomealias> -keyalg RSA -keystore C:\keystore

This command will first ask the user to enter a new password, which will be re-used later, and then ask various other questions. The most important of all is the first:

<div align="center">

Specify first and last name [Unknown]:

</div>

Here you must enter the name of your domain, e.g., *www.fictitiouswebsite.ch*. Finally, after re-entering the previously chosen password, the keystore file will be created in the folder *C:* .

Step 3 – Creation certified file .jar:

Open the command prompt and go to the folder where the .jar file created in step 1 is located, then run the following command:

jarsigner -keystore C:\keystore -storepass <password> -signedjar signedij.jar ij.jar <nomealias>

The file "signedij.jar" will then be created in the working directory, i.e. our archive signed with our own certificate.

Warning! Each time changes are made to the .jar file, this procedure must be repeated!

Step 4 – Collegamento pagina web:

Insert the following lines of HTML code into your web page:

```
<APPLET code=ij.ImageJApplet.class width=0 height=0 archive="signedij.jar">
<PARAM NAME="code" VALUE="ij.ImageJApplet.class">
<PARAM NAME="width" VALUE="1">
<PARAM NAME="height" VALUE="1">
<PARAM NAME="archive" VALUE="signedij.jar">
</APPLET>
```

Warning! You need to enable Java in your browser to run ImageJ. In addition, it is important to specify the location of the signedij.jar file in relation to the location of the HTML page.