dais Learnathon cheat sheet: ImageJ1 - ImageJ2 transition

Starting ImageJ

ImageJ1

```
new ij.ImageJ();
```

ImageJ1/ImageJ2 mix

```
ImageJ ij = new net.imagej.ImageJ();
ij.ui().showUI();
```

Show images

ImageJ1

```
imagePlus.show();
```

ImageJ1/ImageJ2 mix

```
ij.ui().show(testImg);
```

```
ImageJFunctions.show(testImg);
```

```
ImageJFunctions.wrap(testImg, "testImg").show();
```

Convert image types

ImageJ1

```
ImagePlus imp = ImageJFunctions.wrap(img, "Title");
```

ImageJ1/ImageJ2 mix

```
Img<T> realImg          = ImageJFunctions.wrapReal(imp);
```

```
Img<FloatType> floatImg = ImageJFunctions.convertFloat(imp);
```

```
Img<FloatType> realImg2 = ImageJFunctions.wrap(imp);
```

Show regions

ImageJ1

```
imagePlus.setRoi(roi);
```

ImageJ1/ImageJ2 mix

```
Img<BitType> mask; // = ...
ImagePlus maskImp = ImageJFunctions.wrap(mask, "mask");
// threshold the mask to get an ROI
ImageProcessor imageProcessor = maskImp.getProcessor();
imageProcessor.setThreshold(128, 258, ImageProcessor.NO_LUT_UPDATE);
Roi roi = new ThresholdToSelection().convert(imageProcessor);
imagePlus.setRoi(roi);
```

Run plugins

ImageJ1

```
IJ.run(imagePlus, "Normalisation", "");
```

ImageJ1/ImageJ2 mix

```
ij.command().run(ImageNormalizerIJ2Plugin.class, false, new
Object[]{"input", img, "ij", ij});
```

```
IJ.run(imagePlus, "Normalisation (IJ2)", "");
// don't forget the IJ legacy dependency!
<dependency>
   <groupId>net.imagej</groupId>
   <artifactId>imagej-legacy</artifactId>
</dependency>
```

Define plugins

ImageJ1

```
public class ImageNormalizerPlugin implements
             PlugInFilter {
```

resources/plugins.config:

```
Plugins>Filtering, "Normalisation", NormalizerPlugin
```

ImageJ1/ImageJ2 mix

```
@Plugin(type = Command.class, menuPath = "Plugins>Normalisation")
public class ImageNormalizerIJ2Plugin implements Command {
```

CBG Max Planck Institute of Molecular Cell Biology and Genetics

Robert Haase, Scientific Computing Facility, MPI CBG Dresden

dais Learnathon cheat sheet: ImageJ1 - ImageJ2 transition

Writing tables

ImageJ1

```
// create table
ResultsTable table = new ResultsTable();



// add content row by row
table.incrementCounter();
table.addValue("Town", "Shanghai");
table.addValue("Population", 24256800.0);

table.incrementCounter();
table.addValue("Town","Karachi");
table.addValue("Population", 23500000.0);



// show the table
table.show("Title");
```

ImageJ2

```
// create table
GenericTable table = new DefaultGenericTable();

// create columns
GenericColumn nameColumn = new GenericColumn("Town");
DoubleColumn populationColumn = new DoubleColumn("Population");

// fill the columns; add row at the end
nameColumn.add("Karachi");
populationColumn.add(23500000.0);

// fill the columns; add row at the beginning
nameColumn.add(0, "Shanghai");
populationColumn.add(0, 24256800.0);

// add the columns to the table
table.add(nameColumn);
table.add(populationColumn);

// show the table
ij.ui().show(table);
```

Reading tables

```
ResultsTable tableIn; // = ...

// get table structure
tableIn.getCounter();
tableIn.columnExists(columnIndex);

// read header of a column
tableIn.getColumnHeading(columnIndex);

// read value of a field (row/column)
String value = tableIn.getStringValue(columnIndex,
               rowIndex);
double value = tableIn.getValueAsDouble(columnIndex,
               rowIndex);
```

```
GenericTable tableIn; // = ...
Column column = tableIn.get(columnIndex);
// get table structure
tableIn.getRowCount();
tableIn.getColumnCount();

// read header of a column
String header = column.getHeader();

// read value of a field (row/column)
Object value = column.getValue(rowIndex);
```