Table of Contents

Introduction

The GeoScript Groovy Cookbook contains short recipes on how to use the GeoScript Groovy library.

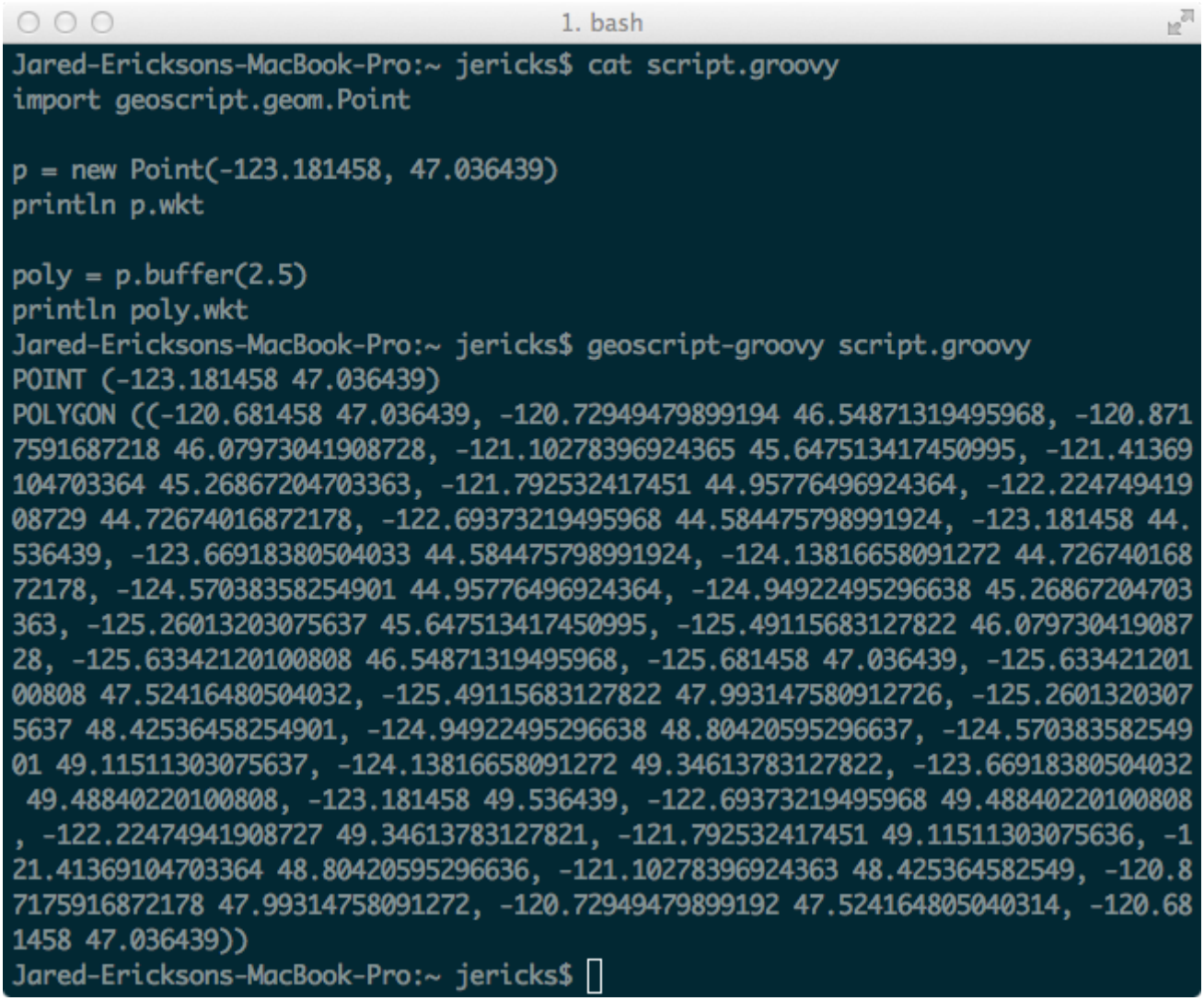
GeoScript is a geospatial library written in Groovy. It provides modules for working with geometries, projections, features, layers, rasters, styles, rendering, and tiles. It is built on top of the Java Topology Suite (JTS) and GeoTools libraries. GeoScript Groovy is open source and licensed under the MIT license.

Using GeoScript

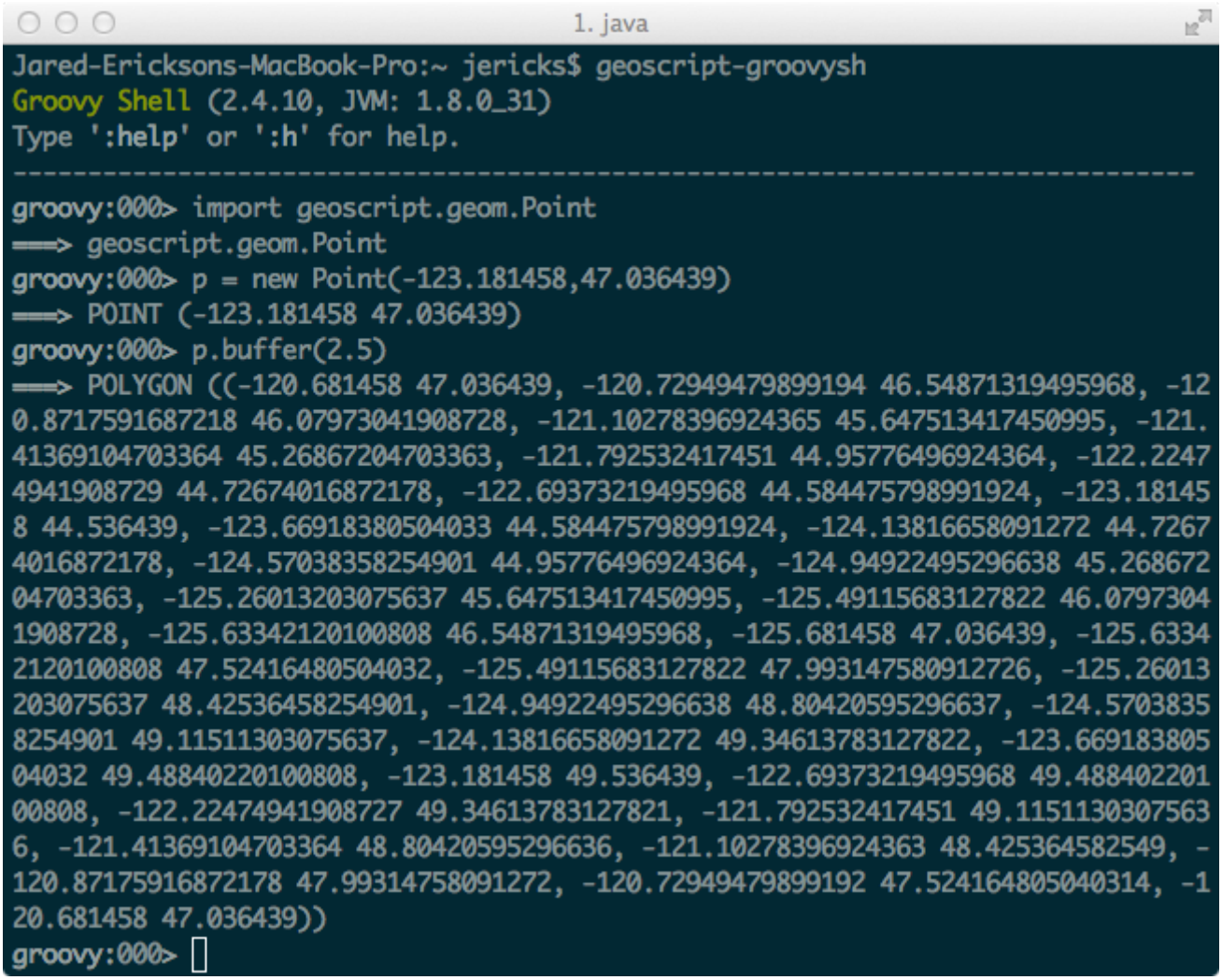
To use GeoScript Groovy you need Java and Groovy installed and on your PATH. Next, download the latest stable release, the latest in development build, or build the code yourself. Then put the GeoScript Groovy bin directory on your PATH. You are now ready to use GeoScript Groovy!

GeoScript Groovy has three commands:

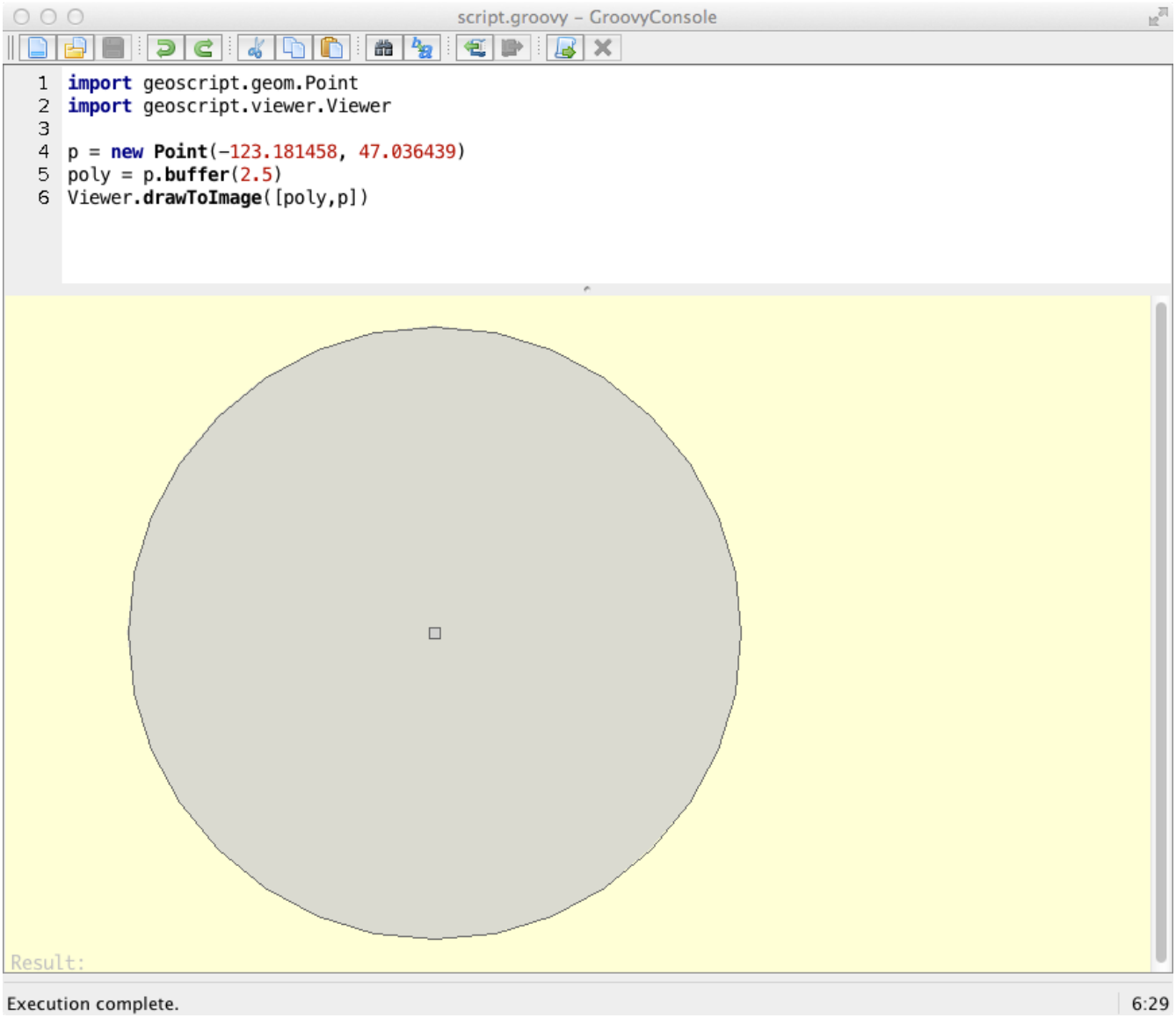
1. geoscript-groovy (which can run Groovy files)



1. geoscript-groovysh (which starts a REPL shell)



1. geoscript-groovyConsole (which starts a graphical editor/mini IDE)



geoscript-groovy

The geoscript-groovy command can run scripts file but it can also run inline scripts.

Convert a shapefile to geojson

```
geoscript-groovy -e "println new
geoscript.layer.Shapefile('states.shp').toJSONString()"
```

Get the Bounds of a Shapefile as a Geometry

```
geoscript-groovy -e "println new
geoscript.layer.Shapefile('states.shp').bounds.geometry"
```

Count the number of Features in a Shapefile

```
geoscript-groovy -e "println new geoscript.layer.Shapefile('states.shp').count"
```

Render a Shapefile to an image

```
geoscript-groovy -e "geoscript.render.Draw.draw(new
geoscript.layer.Shapefile('states.shp'), out: 'image.png')"
```

Pipe a Shapefile's geometry to another command line application that buffers each feature

```
geoscript-groovy -e "new geoscript.layer.Shapefile('states.shp').eachFeature{ println
it.geom.centroid}" | geom combine | geom buffer -d 1.5
```

Pipe the results of buffering a point to convert it to KML

```
echo "POINT (1 1)" | geom buffer -d 10 | geoscript-groovy -e "println
geoscript.geom.Geometry.fromWKT(System.in.text).kml"
```