

# Table of Contents

- Raster Recipes ..... 1
  - Raster Properties ..... 1
  - Raster Values ..... 4
  - Raster Processing..... 7

# Raster Recipes

The Raster classes are in the [geoscript.layer](#) package.

## Raster Properties

Read a Raster from a File

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
```



Get the Raster's Bounds.

```
Bounds bounds = raster.bounds
println "Bounds: ${bounds}"
```

```
Bounds: (-179.99999999999997,-89.9999999998205,179.9999999996405,90.0,EPsg:4326)
```

Get the Raster's Projection.

```
Projection projection = raster.proj
println "Projection: ${projection}"
```

```
Projection: EPSG:4326
```

Get the Raster's Size.

```
List size = raster.size  
println "Size: ${size[0]}x${size[1]}"
```

Size: 800x400

Get the Raster's number of columns and rows.

```
int cols = raster.cols  
int rows = raster.rows  
println "Columns: ${cols} Rows: ${rows}"
```

Columns: 800 Rows: 400

Get the Raster's Bands.

```
List<Band> bands = raster.bands  
println "Bands:"  
bands.each { Band band ->  
    println "  ${band}"  
}
```

Band:  
RED\_BAND  
GREEN\_BAND  
BLUE\_BAND

Get the Raster's block size.

```
List blockSize = raster.blockSize  
println "Block size: ${blockSize[0]}x${blockSize[1]}"
```

Block size: 800x8

Get the Raster's pixel size.

```
List pixelSize = raster.pixelSize  
println "Pixel size: ${pixelSize[0]}x${pixelSize[1]}"
```

Pixel size: 0.4499999999995505x0.449999999999551

Get more information about a Raster's Bounds.

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
List<Band> bands = raster.bands
bands.each { Band band ->
    println "${band}"
    println "  Min = ${band.min}"
    println "  Max = ${band.max}"
    println "  No Data = ${band.noData}"
    println "  Is No Data = ${band.isNoData(12.45)}"
    println "  Unit = ${band.unit}"
    println "  Scale = ${band.scale}"
    println "  Offset = ${band.offset}"
    println "  Type = ${band.type}"
}
```

```
RED_BAND
  Min = 0.0
  Max = 255.0
  No Data = [0.0]
  Is No Data = false
  Unit = null
  Scale = 1.0
  Offset = 0.0
  Type = byte
```

```
GREEN_BAND
  Min = 0.0
  Max = 255.0
  No Data = [0.0]
  Is No Data = false
  Unit = null
  Scale = 1.0
  Offset = 0.0
  Type = byte
```

```
BLUE_BAND
  Min = 0.0
  Max = 255.0
  No Data = [0.0]
  Is No Data = false
  Unit = null
  Scale = 1.0
  Offset = 0.0
  Type = byte
```

Get the minimum and maximum values from a Raster for each band

```
Map extrema = raster.extrema
println "Min values: ${extrema.min} Max values: ${extrema.max}"
```

```
Min value: [56.0, 84.0, 91.0] Max value: [255.0, 255.0, 255.0]
```

## Raster Values

Get values from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
```



Get values from a Raster with a Point.

```
double elevation = raster.getValue(new Point(-121.799927,46.867703))  
println elevation
```

```
3069.0
```

Get values from a Raster with a Pixel Location.

```
List pixel = [100,200]  
elevation = raster.getValue(pixel)  
println elevation
```

```
288.0
```

Set values on a Raster

```

File file = new File("src/main/resources/earth.tif")
GeoTIFF geotiff = new GeoTIFF(file)
Raster raster = geotiff.read("earth")

File arcGridFile = new File("target/earth.asc")
ArcGrid arcGrid = new ArcGrid(arcGridFile)
arcGrid.write(raster)
Raster arcGridRaster = arcGrid.read("earth")

arcGridRaster.eachCell {double value, double x, double y ->
    double newValue = value + 100
    arcGridRaster.setValue([x as int, y as int], newValue)
}

File arcGridAddFile = new File("target/earth_100.asc")
ArcGrid arcGridAdd = new ArcGrid(arcGridAddFile)
arcGridAdd.write(arcGridRaster)
Raster arcGridRasterAdd = arcGridAdd.read("earth_100")

List pixels = [
    [92, 298],
    [393.0, 343.0],
    [795.0, 399.0]
]
pixels.each { List pixel ->
    println "Original: ${raster.getValue(pixel)} New:
    ${arcGridRasterAdd.getValue(pixel)}"
}

```

```

Original: 97.0 New: 197.0
Original: 96.0 New: 196.0
Original: 237.0 New: 337.0

```





## Raster Processing

### Crop

Crop a Raster with a Bounds

```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
Raster croppedRaster = raster.crop(new Bounds(-160.927734, 6.751896, -34.716797
, 57.279043, "EPSG:4326"))
```



### Project

Reproject a Raster to another Projection



```
File file = new File("src/main/resources/earth.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("earth")
Projection projection = new Projection("EPSG:3857")
Raster projectedRaster = raster.crop(projection.geoBounds()).reproject(projection)
```



## Contours

Create vector contours from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
int band = 0
int interval = 300
boolean simplify = true
boolean smooth = true
Layer contours = raster.contours(band, interval, simplify, smooth)
```



## Stylize

Stylize a Raster by baking in a style to create a new Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster stylizedRaster = raster.stylize(new ColorMap([
    [color: "#9fd182", quantity:25],
    [color: "#3e7f3c", quantity:470],
    [color: "#133912", quantity:920],
    [color: "#08306b", quantity:1370],
    [color: "#ffffff5", quantity:1820],
]))
```



## Reclassify

Reclassify a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster reclassifiedRaster = raster.reclassify([
    [min:0,    max:0,    value: 1],
    [min:0,    max:50,   value: 2],
    [min:50,   max:200,  value: 3],
    [min:200,  max:1000, value: 4],
    [min:1000, max:1500, value: 5],
    [min:1500, max:4000, value: 6]
])
```



## Scale

Scale a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
println "Original Raster Size = ${raster.size[0]}x${raster.size[1]}"

Raster scaledRaster = raster.scale(0.5, 0.5)
println "Scaled Raster Size = ${scaledRaster.size[0]}x${scaledRaster.size[1]}"
```

```
Original Raster Size = 800x400
Scaled Raster Size = 400x200
```



## Invert

Invert the values of a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster invertedRaster = raster.invert()
```

## Exponent

Calculate the exponent of the values of a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster expRaster = raster.exp()
```



## Log

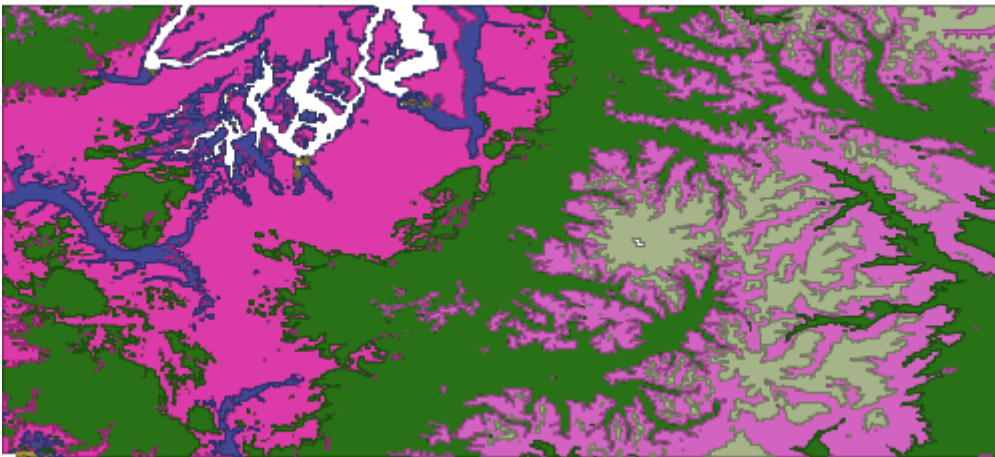
Calculate the log of the values of a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster logRaster = raster.log()
```

## Vectorize

Create a Polygon Layer from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
Raster reclassifiedRaster = raster.reclassify([
    [min:0,    max:0,    value: 1],
    [min:0,    max:50,   value: 2],
    [min:50,   max:200,  value: 3],
    [min:200,  max:1000, value: 4],
    [min:1000, max:1500, value: 5],
    [min:1500, max:4000, value: 6]
])
Layer layer = reclassifiedRaster.polygonLayer
```



Create a Point Layer from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc").crop(new Bounds(-121.878548, 46.808402, -121.636505, 46.896097, "EPSG:4326"))
Layer layer = raster.pointLayer
```



## Raster Algebra

### Add

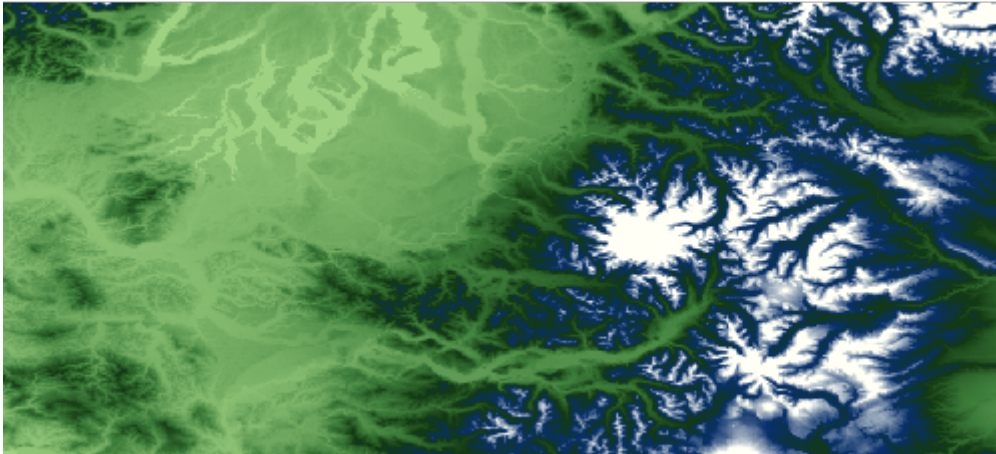
Add a constant value to a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster higherRaster = raster.add(100.00)
double elevation2 = higherRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
3169.0
```





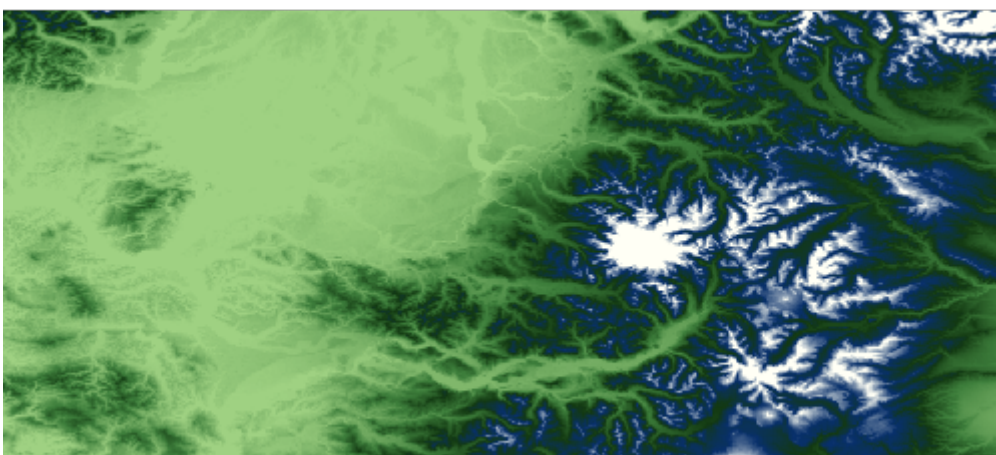
## Subtract

Subtract a constant value from a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster lowerRaster = raster.minus(50.00)
double elevation2 = lowerRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
3019.0
```





## Subtract the Raster from a constant value

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster lowerRaster = raster.minusFrom(2000.0)
double elevation2 = lowerRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

```
3069.0
-1069.0
```



## Multiply

Multiply a constant value against a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster higherRaster = raster.multiply(2.0)
double elevation2 = higherRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

3069.0  
6138.0



## Divide

Divide a constant value against a Raster

```
File file = new File("src/main/resources/pc.tif")
Format format = Format.getFormat(file)
Raster raster = format.read("pc")
double elevation1 = raster.getValue(new Point(-121.799927,46.867703))
println elevation1

Raster lowerRaster = raster.divide(2.0)
double elevation2 = lowerRaster.getValue(new Point(-121.799927,46.867703))
println elevation2
```

3069.0  
1534.5

