

# Table of Contents

Filter Recipes .....	1
Creating Properties .....	1
Evaluating Properties .....	1
Creating Colors .....	2
Getting Color Formats .....	4
Displaying Colors .....	4
Using Color Palettes .....	5

# Filter Recipes

## Creating Properties

*Create a Property from a string*

```
Property property = new Property("name")
println property
```

name

*Create a Property from a Field*

```
Field field = new Field("geom", "Polygon")
Property property = new Property(field)
println property
```

geom

## Evaluating Properties

*Evaluate a Property to get values from a Feature. Get the id*

```
Feature feature = new Feature([
    id: 1,
    name: "Seattle",
    geom: new Point(-122.3204, 47.6024)
], "city.1")
```

```
Property idProperty = new Property("id")
int id = idProperty.evaluate(feature)
println id
```

1

*Get the name*

```
Property nameProperty = new Property("name")
String name = nameProperty.evaluate(feature)
println name
```

Seattle

*Get the geometry*

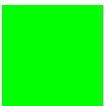
```
Property geomProperty = new Property("geom")
Geometry geometry = geomProperty.evaluate(feature)
println geometry
```

POINT (-122.3204 47.6024)

## Creating Colors

*Create a Color from a RGB color string*

```
Color color = new Color("0,255,0")
```



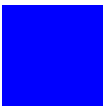
*Create a Color from a CSS color name*

```
Color color = new Color("silver")
```



*Create a Color from a hexadecimal string*

```
Color color = new Color("#0000ff")
```



*Create a Color from a RGB List*

```
Color color = new Color([255,0,0])
```



### Create a Color from a RGB Map

```
Color color = new Color([r: 5, g: 35, b:45])
```



### Create a Color from a HLS Map

```
Color color = new Color([h: 0, s: 1.0, l: 0.5])
```



### Get a Random Color

```
Color color = Color.getRandom()
```



### Get a Random Pastel Color

```
Color color = Color.getRandomPastel()
```



### Get a darker Color

```
Color color = new Color("lightblue")  
Color darkerColor = color.darker()
```



### Get a brighter Color

```
Color color = new Color("purple")  
Color brigtherColor = color.brighter()
```



# Getting Color Formats

## Create a Color

```
Color color = new Color("wheat")
```



## Get Hex

```
String hex = color.hex  
println hex
```

```
#f5deb3
```

## Get RGB

```
List rgb = color.rgb  
println rgb
```

```
[245, 222, 179]
```

## Get HSL

```
List hsl = color.hsl  
println hsl
```

```
[0.10858585256755147, 0.7674419030001307, 0.8313725489999999]
```

## Get the java.awt.Color

```
java.awt.Color awtColor = color.asColor()  
println awtColor
```

```
java.awt.Color[r=245,g=222,b=179]
```

# Displaying Colors

### *Draw a List of Colors to a BufferedImage*

```
Color color = new Color("pink")
BufferedImage image = Color.drawImage(
    [color.brighter(), color, color.darker()],
    "vertical",
    40
)
```



### *Draw a List of Colors to a simple GUI*

```
List<Color> colors = Color.getPaletteColors("YlOrBr")
Color.draw(colors, "horizontal", 50)
```



## Using Color Palettes

### *Get all color palettes*

```
List<String> allPalettes = Color.getPaletteNames("all")
allPalettes.each { String name ->
    println name
}
```

YlOrRd  
PRGn  
PuOr  
RdGy  
Spectral  
Grays  
PuBuGn  
RdPu  
BuPu  
YlOrBr  
Greens  
BuGn  
Accents  
GnBu  
PuRd  
Purples  
RdYlGn  
Paired  
Blues  
RdBu  
Oranges  
RdYlBu  
PuBu  
OrRd  
Set3  
Set2  
Set1  
Reds  
PiYG  
Dark2  
YlGn  
BrBG  
YlGnBu  
Pastel2  
Pastel1  
BlueToOrange  
GreenToOrange  
BlueToRed  
GreenToRedOrange  
Sunset  
Green  
YellowToRedHeatMap  
BlueToYellowToRedHeatMap  
DarkRedToYellowWhiteHeatMap  
LightPurpleToDarkPurpleHeatMap  
BoldLandUse  
MutedTerrain  
BoldLandUse  
MutedTerrain

### *Get diverging color palettes*

```
List<String> divergingPalettes = Color.getPaletteNames("diverging")
divergingPalettes.each { String name ->
    println name
}
```

```
PRGn
PuOr
RdGy
Spectral
RdYlGn
RdBu
RdYlBu
PiYG
BrBG
BlueToOrange
GreenToOrange
BlueToRed
GreenToRedOrange
```

### *Get sequential color palettes*

```
List<String> sequentialPalettes = Color.getPaletteNames("sequential")
sequentialPalettes.each { String name ->
    println name
}
```



YlOrRd  
Grays  
PuBuGn  
RdPu  
BuPu  
YlOrBr  
Greens  
BuGn  
GnBu  
PuRd  
Purples  
Blues  
Oranges  
PuBu  
OrRd  
Reds  
YlGn  
YlGnBu  
Sunset  
Green  
YellowToRedHeatMap  
BlueToYellowToRedHeatMap  
DarkRedToYellowWhiteHeatMap  
LightPurpleToDarkPurpleHeatMap  
BoldLandUse  
MutedTerrain

### *Get qualitative color palettes*

```
List<String> qualitativePalettes = Color.getPaletteNames("qualitative")
qualitativePalettes.each { String name ->
    println name
}
```

Accents  
Paired  
Set3  
Set2  
Set1  
Dark2  
Pastel2  
Pastel1  
BoldLandUse  
MutedTerrain

### Get a Blue Green Color Palette

```
List colors = Color.getPaletteColors("BuGn")
```



### Get a Purple Color Palette with only four colors

```
colors = Color.getPaletteColors("Purples", 4)
```



### Get a Blue Green Color Palette

```
colors = Color.getPaletteColors("MutedTerrain")
```



### Get a Blue Green Color Palette

```
colors = Color.getPaletteColors("BlueToYellowToRedHeatMap")
```



### Create a Color palette by interpolating between two colors

```
Color startColor = new Color("red")  
Color endColor = new Color("green")  
List<Color> colors = startColor.interpolate(endColor, 10)
```



### Create a Color palette by interpolating between two colors

```
Color startColor = new Color("wheat")  
Color endColor = new Color("lightblue")  
List<Color> colors = Color.interpolate(startColor, endColor, 8)
```

