Wolfram Language™

Gallery of Tweetable Programs

Designed for the new generation of programmers, the Wolfram Language has a vast depth of built-in algorithms and knowledge, all automatically accessible through its elegant unified symbolic language. Scalable for programs from tiny to huge, with immediate deployment locally and in the cloud, the Wolfram Language builds on clear principles—and 25+ years of development—to create what promises to be the world's most productive programming language.

This gallery shows a few examples of tiny programs in the Wolfram Language—and big things they can do...

Core Language & Structure

Data Manipulation & Analysis

Visualization & Graphics

Symbolic & Numeric Computation

Strings & Text

Graphs & Networks

Images

Geometry

Sound

Time-Related Computation

Geographic Data & Computation

Scientific and Medical Data & Computation

Engineering Data & Computation

Financial Data & Computation

Social, Cultural, & Linguistic Data

Higher Mathematical Computation

Documents & Presentation

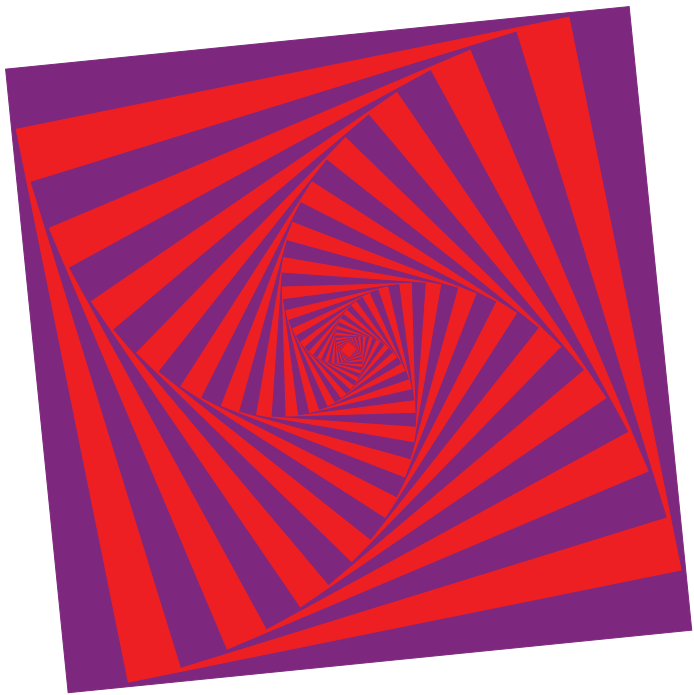
User Interface Construction

System Operation & Setup

External Interfaces & Connections

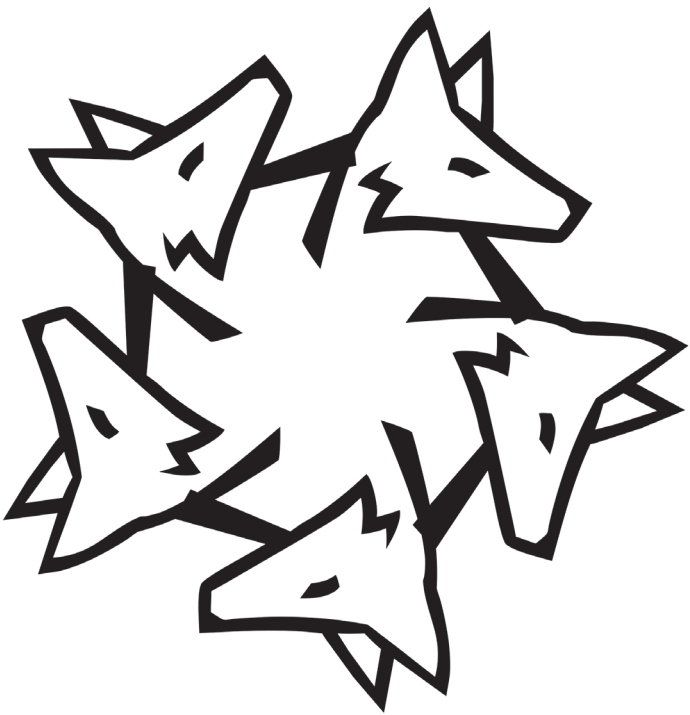
Cloud & Deployment

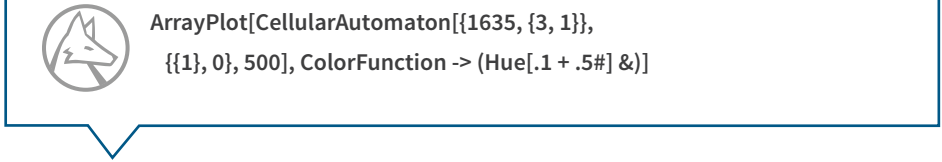
```
Graphics[{White, Riffle[NestList[Scale[Rotate[#, 0.1], 0.9] &,
   Rectangle[], 40], {Purple, Red}]}]
```



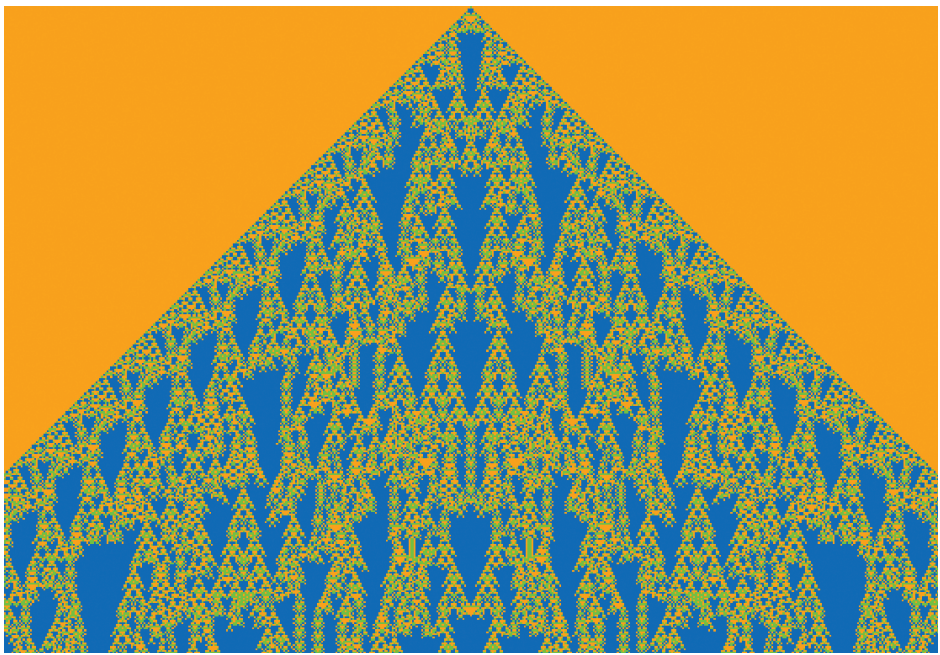
```
c = EntityValue[CityData[{Large, "France"}], "Coordinates"];
GeoGraphics[{Red, Thick, Line@c[[Last[FindShortestTour[c]]]]}]
```

```
Graphics[Table[Rotate[Text[Style[\[WolframLanguageLogo], 160,
  FontFamily -> "Times"], {1, 1.5}], i 2 Pi/5, {0, 0}], {i, 1, 5}]]
```

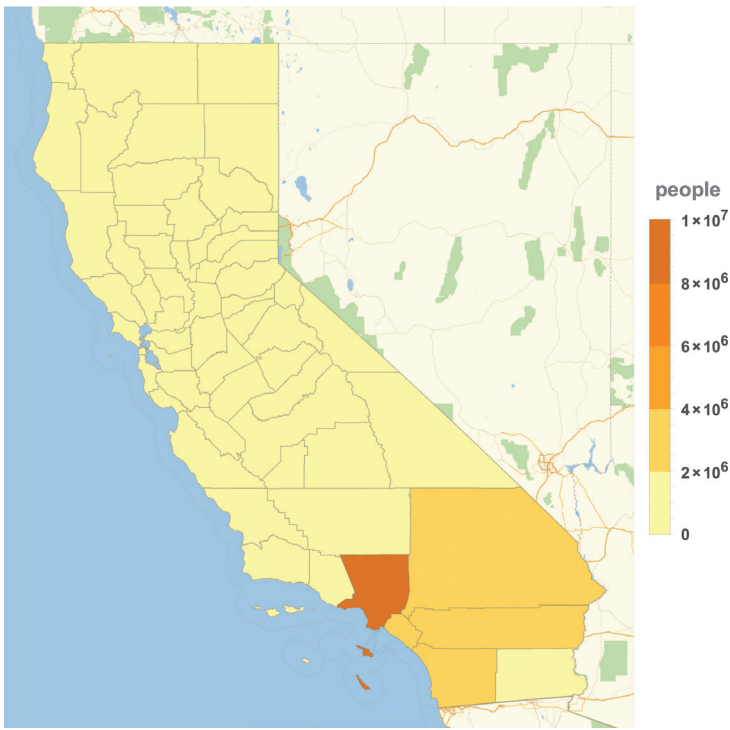




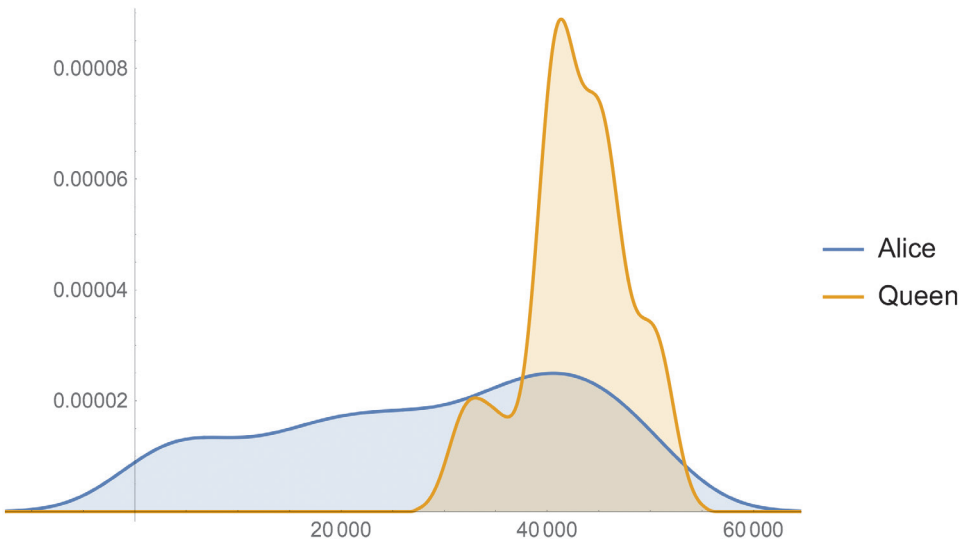
```
ArrayPlot[CellularAutomaton[{1635, {3, 1}},
  {{1}, 0}, 500], ColorFunction -> (Hue[.1 + .5#] &)]
```



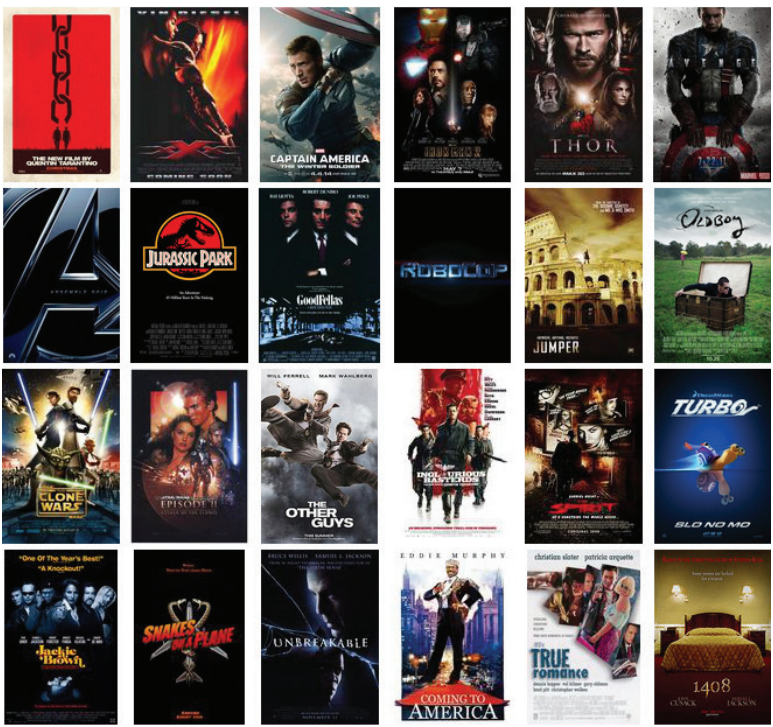
```
GeoRegionValuePlot[Entity["AdministrativeDivision", {_, "California",
  "UnitedStates"}] -> "Population"]
```



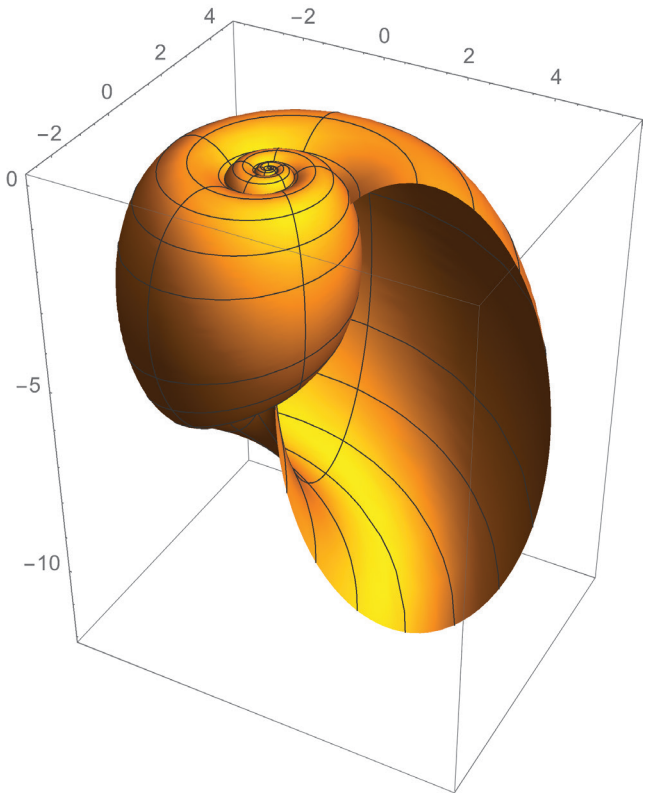
```
SmoothHistogram[Legended[First /@ StringPosition[
  ExampleData@{"Text", "AliceInWonderland"}, #], #] & /@
  {"Alice", "Queen"}, Filling -> Axis]
```



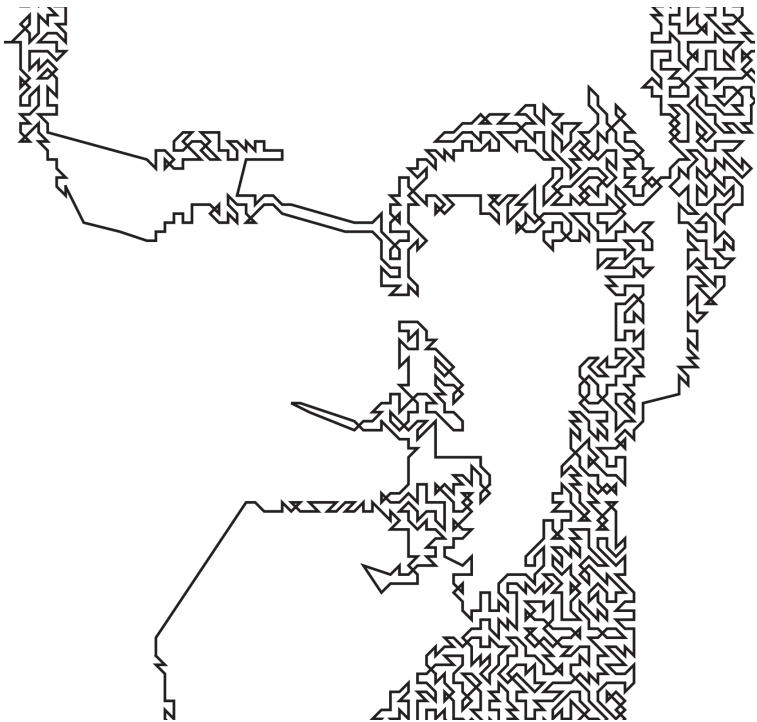
```
Grid@Partition[DeleteMissing[#["Image"] & /@
  Take[samuel jackson["MovieAppearances"], 24]], 6]
```



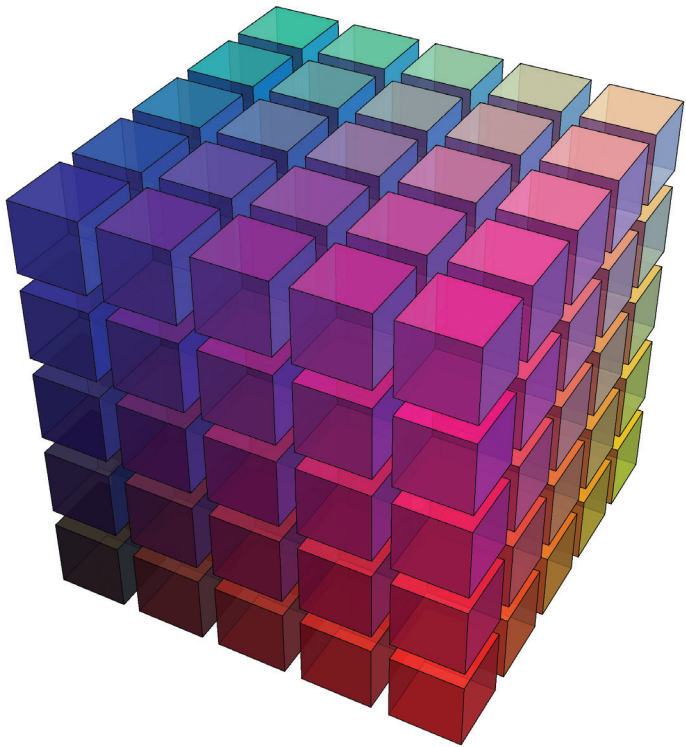
```
ParametricPlot3D[1.2^v {Cos[v] (1 + Cos[u]),
  - Sin[v] (1 + Cos[u]), -2 (1 + Sin[u])}, {u, 0, 2 Pi}, {v, -15, 6},
  PlotRange -> All, PlotPoints -> 40]
```



```
p = PixelValuePositions[Binarize[ stephen wolfram image , .4], 0];
Graphics[Line[p[[FindShortestTour[p][[2]]]]]]
```

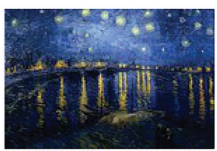
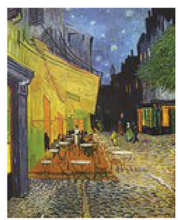


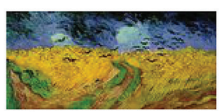
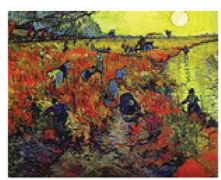
```
Graphics3D[{RGBColor[#/5], Opacity[.8], Cuboid[#, # + .8]} & /@
  Tuples@Table[Range[5], {3}]]
```



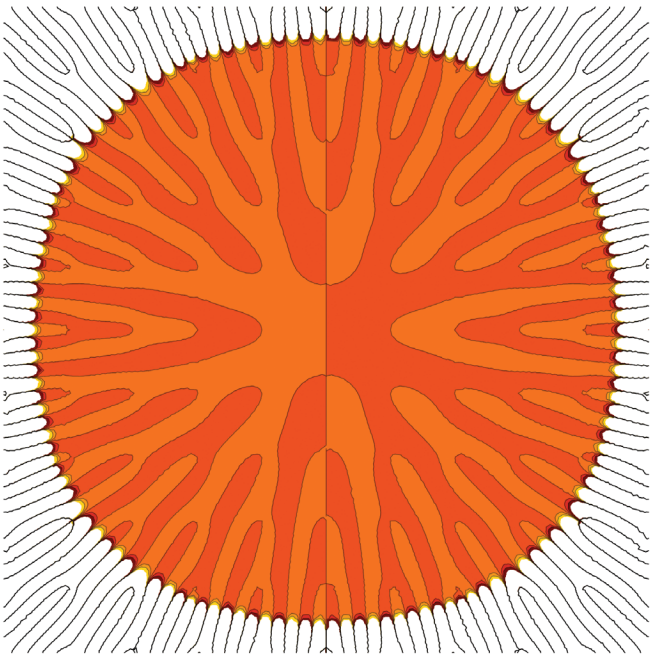
```
Partition[Column[{#, DominantColors[#]}, Alignment -> Center] & /@
(#["Image"] & /@ Take[van gogh artwork, 9]), 3]
```



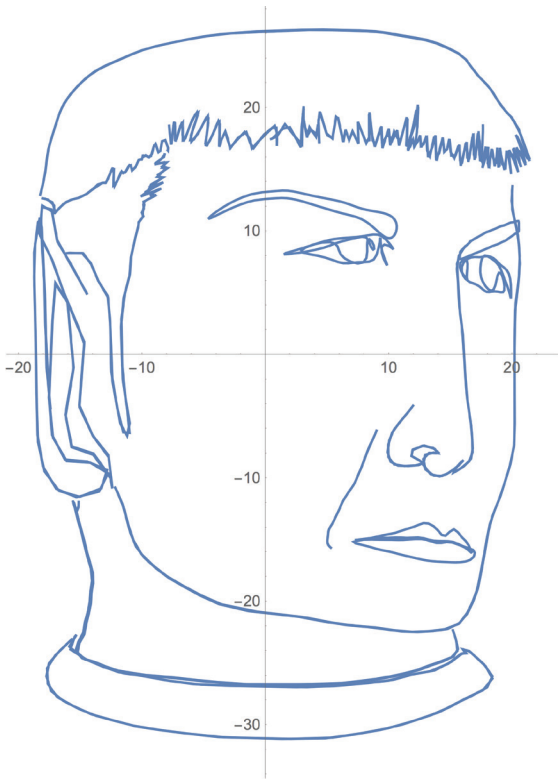




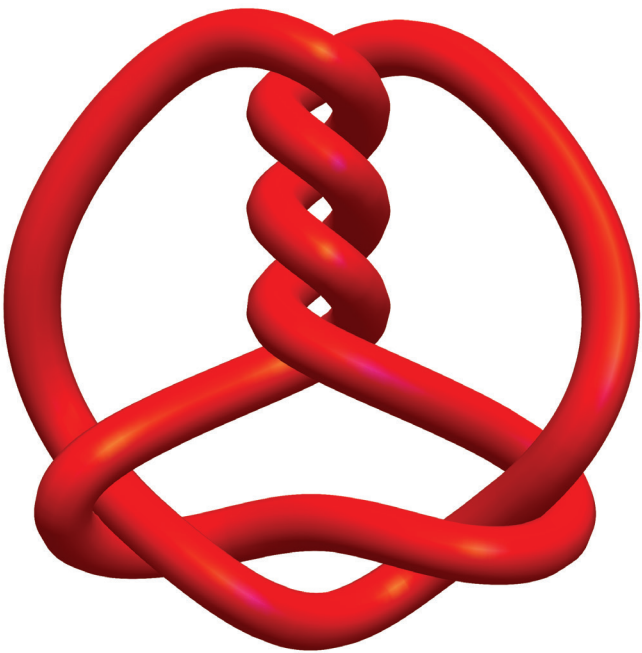
```
ContourPlot[Evaluate[Re[Product[x + I y - (a + I b), {a, -5, 5}, {b, -5, 5}]]],
  {x, -5, 5}, {y, -5, 5}, ColorFunction -> "SolarColors"]
```



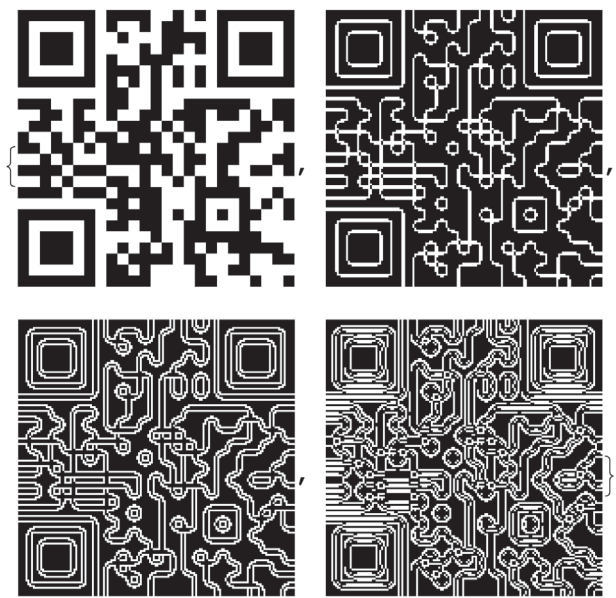
```
c = spock curve equation [t]; ParametricPlot[ Evaluate[Sign[#]
  Abs[#]^(2/3) &@c], {t, 0, 72 Pi}, MaxRecursion -> 6, PlotRange -> All]
```



```
Graphics3D[{Red, Specularity[White, 70], KnotData[{8, 3}, "ImageData"]},
  Boxed -> False, ViewPoint -> {0, 0.1, 5}]
```

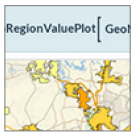


```
NestList[EdgeDetect, BarcodeImage[
  "http://wolframtap.tumblr.com", "QR"], 3]
```



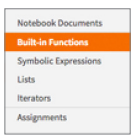
```
GeoGraphics[{EdgeForm[Black], {GeoStyling[{"Image", #2}],
  Polygon[#1]} & @@@ african countries [{"Entity", "Flag"}]},
  GeoBackground -> White]
```

Wolfram Home Page
wolfram.com



Gallery of Tweetable Programs
wolframtap.tumblr.com

Wolfram Language Home Page
wolfram.com/language



Fast Introduction for Programmers
wolfram.com/language/
fast-introduction-for-programmers

Wolfram Language Reference
reference.wolfram.com/language