Pygal: Python Data Playbook

GETTING DATA INTO PYGAL

Kishan Iyer
LOONYCORN

www.loonycorn.com

Overview

SVG (Scalable Vector Graphics) is a popular vector format

SVG images are defined as markup that is rendered, usually in a browser

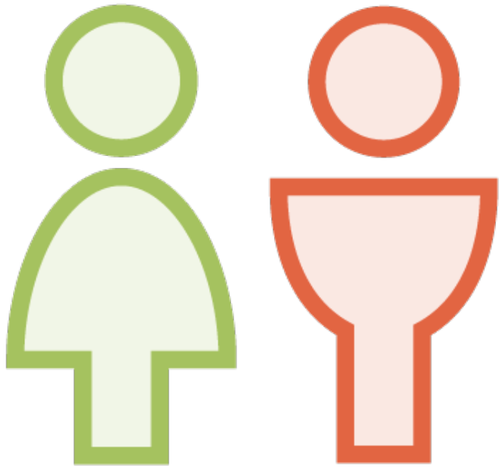
Great for interactivity, indexing, and searching

Retain sharpness regardless of resizing or screen resolution

Pygal is a great Python library for building SVG images

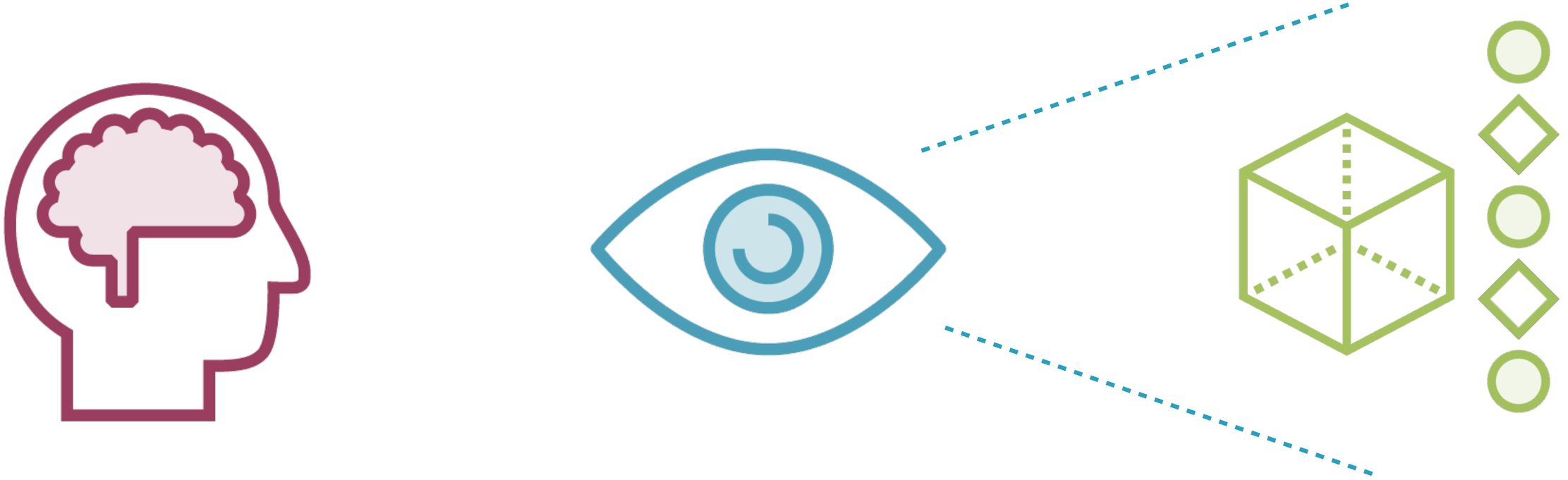
Introducing Pygal

Visualization



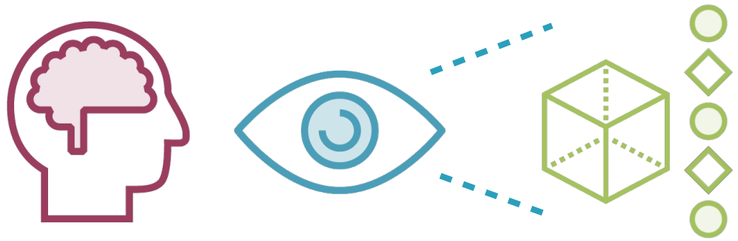
Visualizations are very efficient at conveying information

Visualization



Our brains are wired to understand visualizations - cognitively efficient

Visualization in Exploratory Data Analysis



Important step in data exploration

Helps develop an intuition for relationships in data

Precursor to higher-level data analysis using Machine Learning techniques

Interactivity

Interactivity helps with exploration and experimentation

Interactivity in Visualization

Easy to underestimate importance of interactivity

Lacking from many visualization tools

Enables exploration

Dramatically increases understanding

Pygal

Python visualization library optimized for creating and working with SVG images, with rich support for interactivity and animation

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

Visualization Libraries in Python

Matplotlib

Seaborn

Bokeh

Plotly.py

Many Libraries, Many Niches

Matplotlib is powerful

Seaborn is easy-to-use

Bokeh for interactivity

Plotly.py for collaboration

Pygal's niche is working with SVG images

Understanding SVG

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

Vector and Raster Image Formats

Raster Image Formats

PNG, JPEG

Image files contain grid of pixels representing color

Inherently binary data

Larger image files

Hard to convert to vector format

Simple to render

Vector Image Formats

SVG, PDF

Image files contain points and movements

Can be specified in markup

Smaller image files

Easy to convert to raster format

Complex, specialized programs needed to render

Vector and Raster Image Formats

Raster Image Formats

Harder to search and index

Harder to resize

Tend to render best on high-resolution devices

Inherently non-interactive

Lose quality on zooming, scaling, moving, and resizing

Vector Image Formats

Easier to search and index

Easier to resize

Can render sharply even on poor-resolution devices

Support animation and interactivity

Maintain quality on zooming, scaling, moving, and resizing

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

SVG is an image format like PNG, JPEG, or TIFF

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

SVG image files are actually XML files,
i.e. they contain markup

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

That markup needs to be interpreted and rendered by an external program (e.g. a browser)

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

SVG is a vector format, as opposed to PNG or JPEG, which are raster formats

SVG (Scalable Vector Graphics)

An XML-based vector image format for two-dimensional graphics with support for interactivity and animation

Pygal makes it really simple to build and work with SVG images

SVG

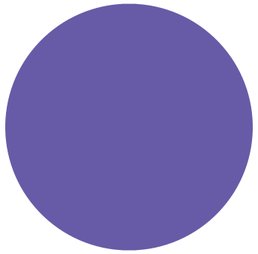
Used to draw shapes using a vector representation

SVG

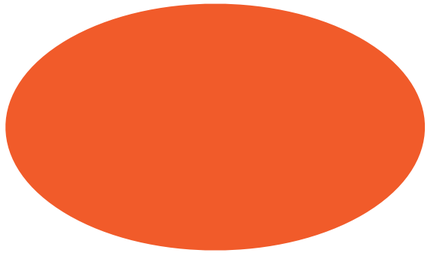
XML format for shape specification

SVG Shapes

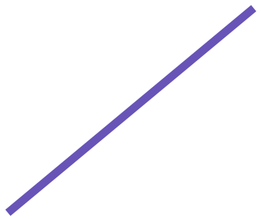
Rectangle



Circle

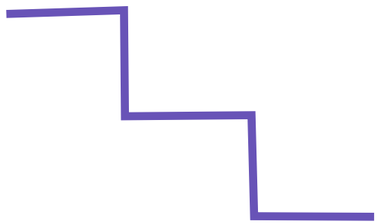


Ellipse



Line

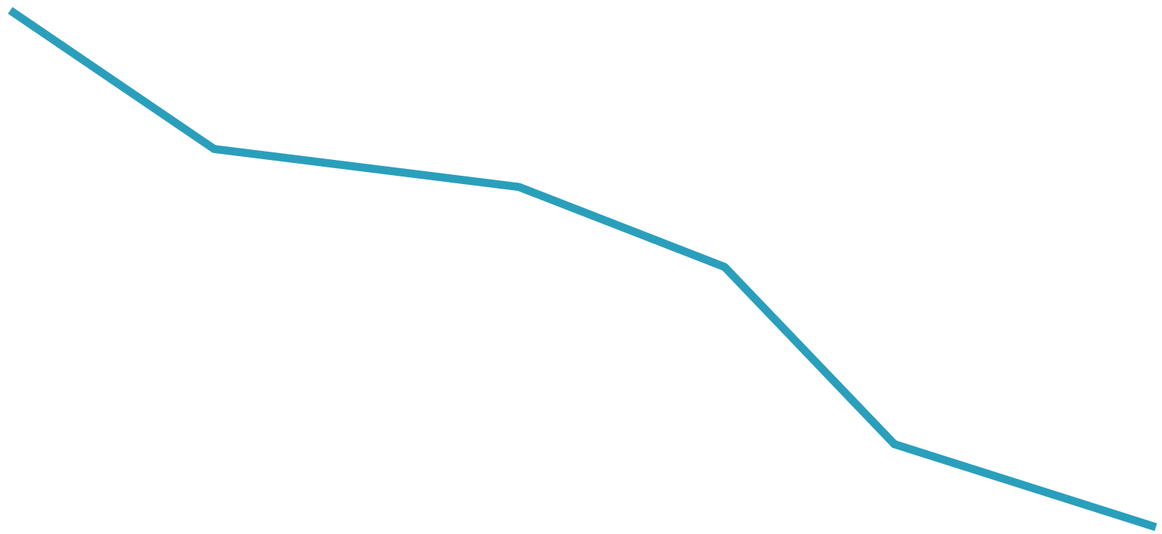
Polygon



Polyline

Predefined shape elements

SVG Path



<path> element is used to define a path

SVG Path

M 1,1 L 1,4 L 4,1 Z

Path is defined with a series of commands

SVG Path

Commands

SVG Path

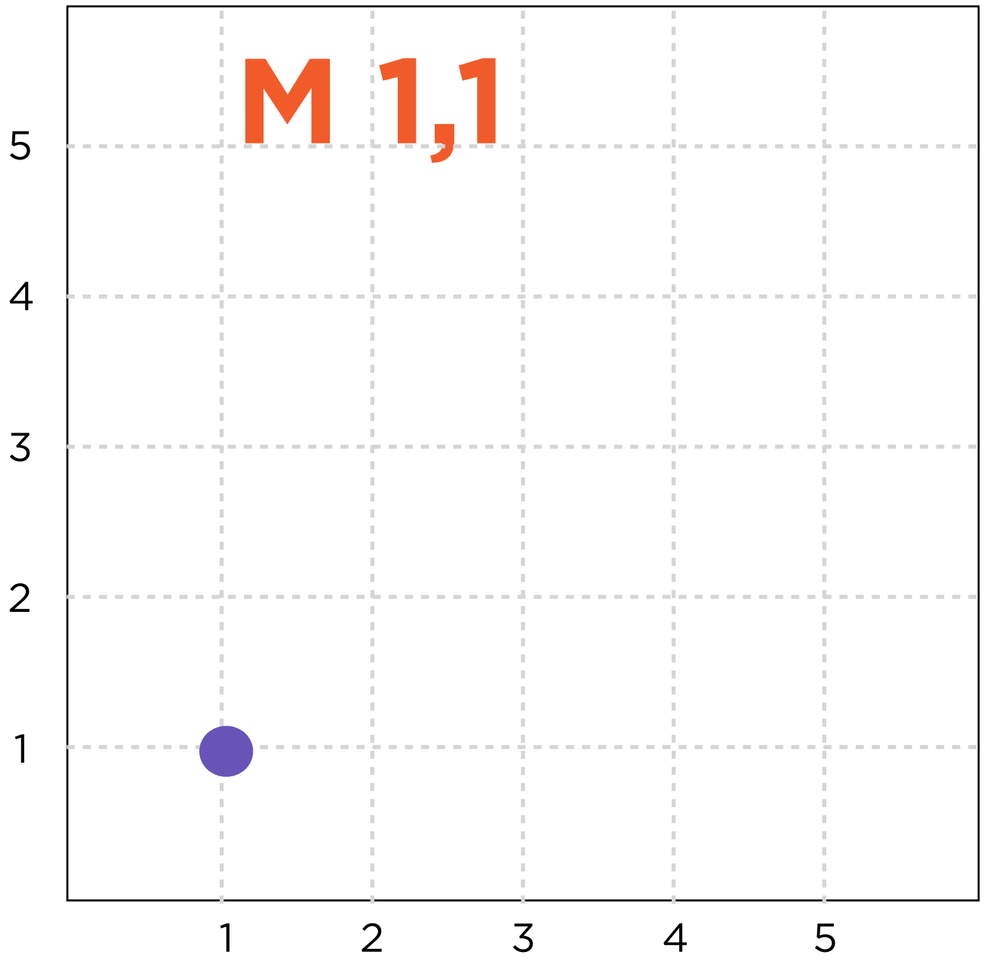
Points on the axes

SVG Path

M	move to	Move from one point to another
L	line to	Create a line
C	curve to	Create a curve
Q	quadratic bezier curve	Create a quadratic bezier curve
Z	close path	Close the path

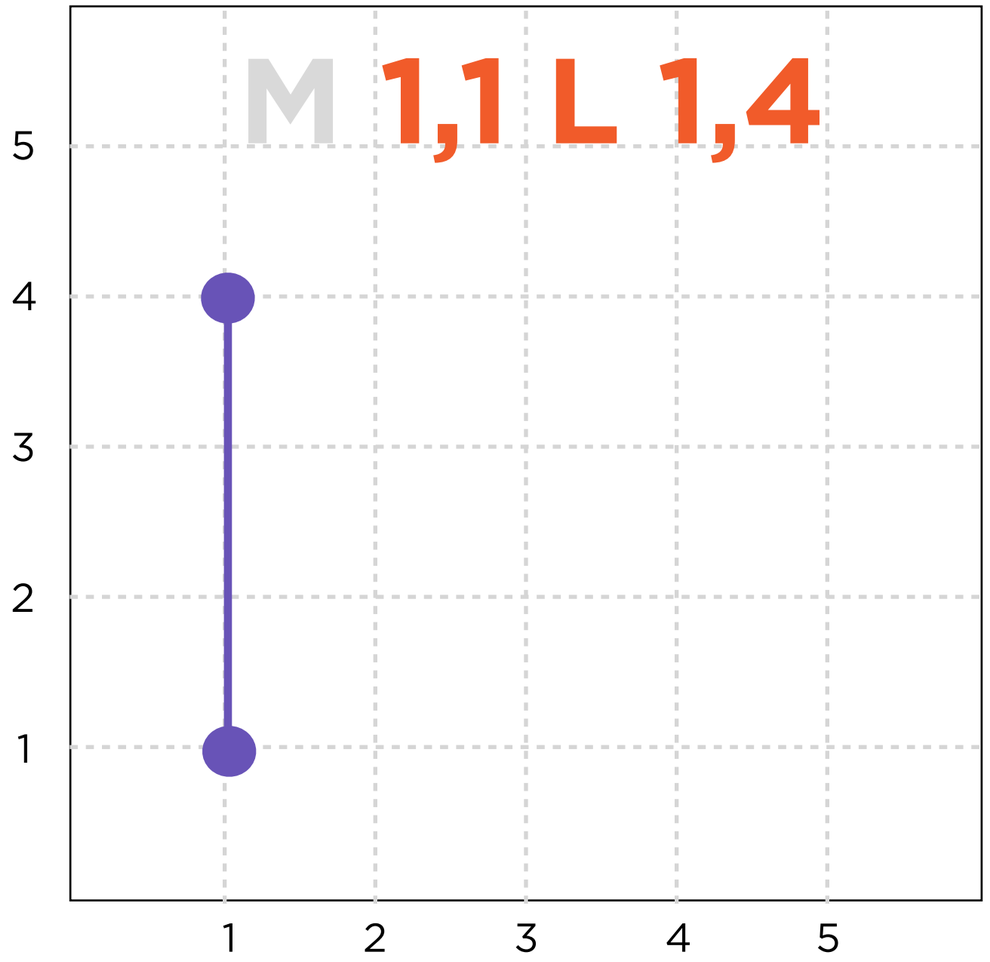
Basic path commands

SVG Path



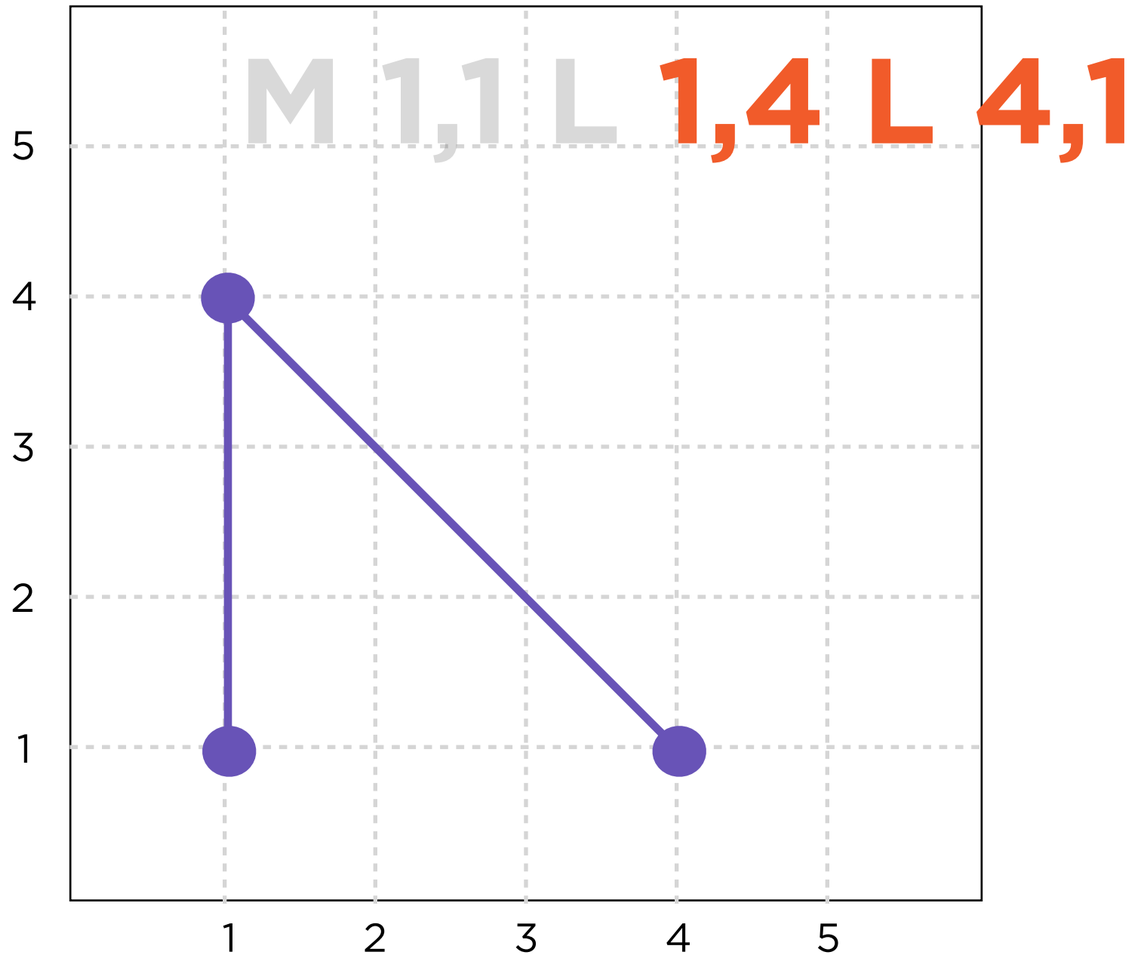
Move cursor to point 1,1

SVG Path



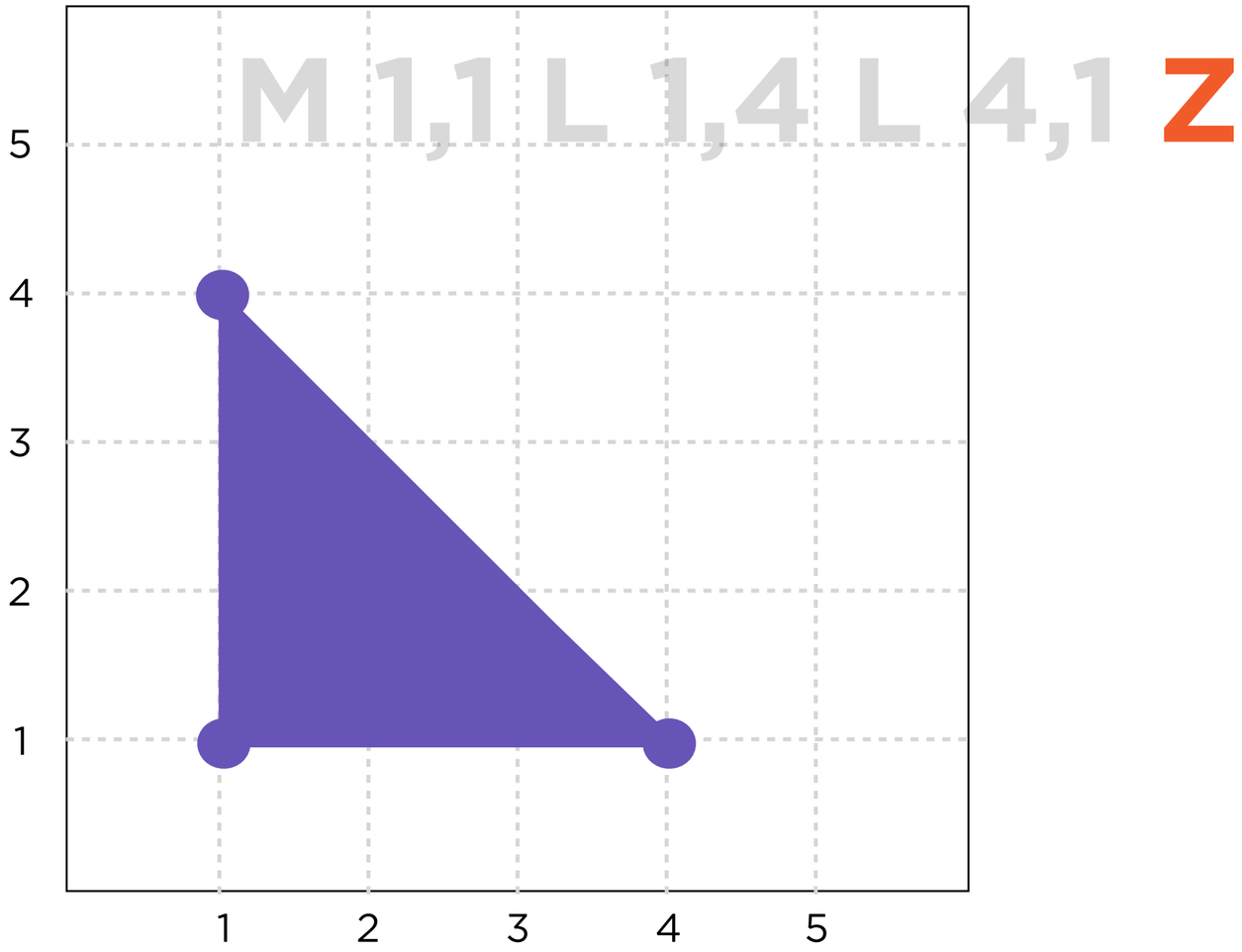
Draw line from 1,1 to 1,4

SVG Path



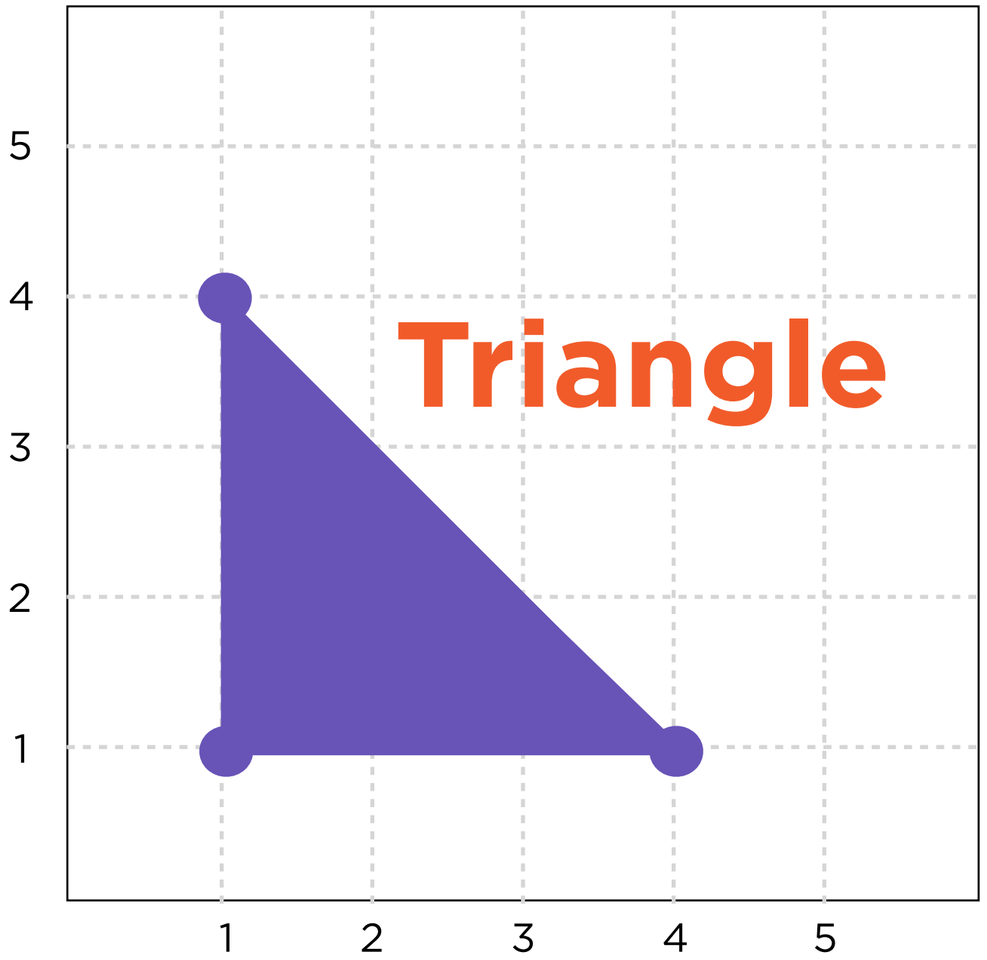
Draw line from 1,4 to 4,1

SVG Path



Close the path

SVG Path



M 1,1 L 1,4 L 4,1 Z

SVG

Can specify several complex features

Can include:

- Nested SVG images
- Embedded raster images

SVG

Paths

Basic shapes

Text

Painting

Color

Fonts

SVG

Animation

Scripting

Links

Filters

Metadata

Demo

Install Pygal

Demo

Plot data contained in a CSV file

Summary

- Used Jupyter and pip to work with Pygal
- Loaded CSV data and visualized using Pygal
- Visualized both in-memory and file data
- Rendered to SVG file format