geom

version 0.6

Jared Erickson

July 04, 2019

Contents

Geometry Commands

Geometry Commands is a command line application for processing geometry. It is inspired by the wonderful Java Topology Suite (JTS) library and the Unix Philosophy.

It contains one command line application (geom) with numerous subcommands (buffer, centroid, envelope) that generally read WKT geometry from standard input and then write WKT geometry to standard output. This enables several geometry commands to be chained together using pipes.:

```
echo "POINT (1 1)" | geom buffer -d 10 | geom envelope
```

Geometry Commands is open source under the MIT license. I hope you find it useful. The code is available at github. If you find any bugs or would like any enhancements please use the GitHub issue tracker.

Command Documentation

PDF

Install

To install, simple download the latest zip file and place the **bin** directory in your path. You should then be able to run the **geom** command.

You can also download an uber jar that contains all of the required code and dependencies in one file that you can run with the **java -jar geom.X.X.jar** command.

Usage

View a list of all geom commands:

```
geom list
```

Buffer a geometry:

```
echo "POINT (1 1)" | geom buffer -d 10
```

Reproject a geometry:

```
echo "POINT (1 1)" | geom project -s "EPSG:4326" -t "EPSG:2927"
```

Generate random points:

```
echo "POINT (1 1)" | geom buffer -d 100 | geom random -n 200
```

Draw a geometry to an image:

```
echo "POINT (1 1)" | geom buffer -d 100 | geom draw && open image.png
```

Commands

list

Name:

geom list

Description:

List all of the geometry commands.

Arguments:

- -d --description: Include the description
- --help : Print help message
- --web-help : Open help in a web browser

Example:
```
geom list
```

version

Name:

geom version

Description:

Get the version

Arguments:

- --help : Print help message
- --web-help : Open help in a web browser

Example:
```
geom version
```

help

Name:

geom help

Description:

Get help

Arguments:

- --help : Print help message
- --web-help : Open help in a web browser

Example:
```
geom help
```

angle

Name:

geom angle

Description:

Calculate the angle between two Points

Arguments:

- -t --type: The type can be degrees (default) or radians
- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:
```
geom angle -g "POINT (1 1)" -o "POINT (10 10)"
```

arc

Name:

geom arc

Description:

Creates an arc linestring from a start angle and an angle extent.

Arguments:

- -a --startAngle: The start angle (in radians)
- -e --angleExtent: The size of angle (in radians)
- -d -degrees: The flag for whether given angle measures are in degrees (true) or radians (false)
- -w --width: The width
- -h --height: The height
- -p --numberOfPoints: The number of points
- -r --rotation: The rotation
- -c --center: The flag to use center (true) or the base (false)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom arc -g "POINT (100 100)" -d -a 45 -e 90 -w 50 -h 50 -p 10
```
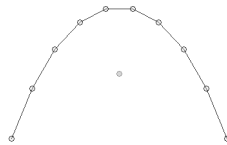

arcpoly
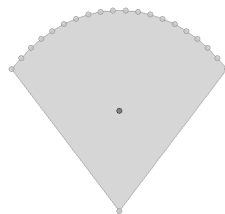
Name:

geom arcpoly

Description:

Creates an arc polygon from a start angle and an angle extent.

Arguments:

- -a --startAngle: The start angle (in radians)
- -e --angleExtent: The size of angle (in radians)
- -d --degrees: The flag for whether given angle measures are in degrees (true) or radians (false)
- -w --width: The width
- -h --height: The height
- -p --numberOfPoints: The number of points
- -r --rotation: The rotation
- -c --center: The flag to use center (true) or the base (false)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom arcpoly -g "POINT (100 100)" -a 45 -e 90 -d -p 20 -w 500 -h 500
```


area

Name:

geom area

Description:

Calculate the area of a Geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom area -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```


asciiart

Name:

geom asciiart

Description:

Get the Geometry as WKT ASCII Art

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom asciiart -g "POINT (10 10)"
```

boundary

Name:

geom boundary

Description:

Calculate the boundary of a Geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom boundary -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```


buffer

Name:

geom buffer

Description:

Buffer a geometry by a distance.

Arguments:

- -d --distance: The buffer distance
- -q --quadrantSegments: The number of quadrant segments
- -c --endCapStyle: The end cap style (round, flat/butt, square)
- -s --singleSided: The flag for whether the buffer should be single sided
- -f --simplifyFactor: The simplify factor
- -m --mitreLimit: The mitre limit
- -j --joinStyle: The join style (round, mitre, bevel)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom buffer -g "POINT (1 1)" -d 10
```
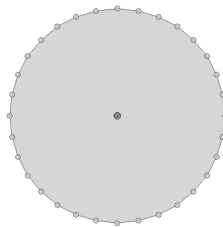
centroid

Name:

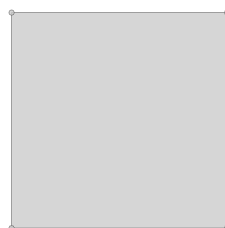geom centroid

Description:

Calculate the centroid of a Geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom centroid -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```
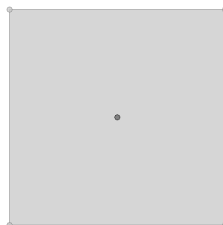
closelinestring

Name:

geom closelinestring

Description:

Close an open LineString.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom closelinestring -g "LINESTRING (0 0, 4 0, 4 4, 0 4)"
```

combine

Name:

geom combine

Description:

Combine geometries on separate lines to create a GeometryCollection.

Arguments:

- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
echo -e "POINT(1 1)
```

POINT(2 2)" | geom combine

contains

Name:

geom contains

Description:

Calculate whether the first geometry contains the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom contains -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))" -o "POINT (2 2)"
```

convexHull

Name:

geom convexHull

Description:

Calculate the convex hull of a Geometry.

Arguments:

- -g --geometry: The input geometry

- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom convexhull -g "POLYGON ((9 52, 9 50, 7 50, 7 48, 10 47, 10 46, 13 46, 11 52, 10 52, 9 5
```

coordinates

Name:

geom coordinates

Description:

Get the coordinates of the geometry.

Arguments:

- -u --unique: The flag to only include unique coordinates
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom coordinates -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```
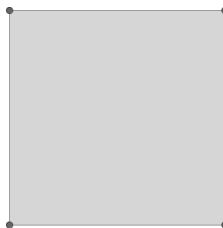
count

Name:

geom count

Description:

Count the number of geometries in the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom count -g "MULTIPOINT (1 1, 2 2, 3 3)"
```

countpoints

Name:

geom countpoints

Description:
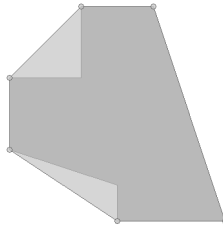
Count the number of Points in the input Geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom countpoints -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```

coveredby

Name:

geom coveredby

Description:

Determine whether the first geometry is covered by the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom coveredby -g "POINT (2 2)" -o "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```

covers

Name:

geom covers

Description:

Determine whether the first geometry covers the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom covers -g "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))" -o "POINT (2 2)"
```

crosses

Name:

geom crosses

Description:

Determine whether the first geometry crosses the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom crosses -g "LINESTRING (1 1, 10 10)" -o "LINESTRING (1 10, 10 1)"
```

delaunay

Name:

geom delaunay

Description:

Generate a delaunay triangulation of the input geometry.

Arguments:

- -c --conforming: The flag for whether to use conforming algorithm
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom buffer -g "POINT (1 1)" -d 100 | geom random -n 100 | geom delaunay
```
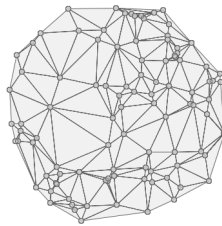

densify

Name:

geom densify

Description:

Densify the coordinates of the input geometry.

Arguments:

- -d --distance: The distance tolerance
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom densify -g "LINESTRING (1 1, 100 100)" -d 20
```

draw

Name:

geom draw

Description:

Draw the input geometry to an image file.

Arguments:

- -f --file: The output File
- -w --width: The image width
- -h --height: The image height
- -b --background: The background color
- -i --backgroundImage: The background image url or file
- -s --stroke: The stroke Color
- -t --strokeOpacity: The stroke opacity
- -r --strokeWidth: The stroke width
- -l --fill: The fill Color
- -o --fillOpacity: The fill opacity
- -m --shape: The marker shape (circle, square, ect..)
- -z --size: The marker size
- -c --drawCoords: The flag for drawing coordinates or not
- -e --envelope: The geographical bounds (minx, miny, maxx, maxy)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom draw -g "POLYGON ((5 5, 5 15, 15 15, 15 5, 5 5))"
```

drawbase64

Name:

geom drawbase64

Description:

Draw the input geometry to a base 64 encoded string.

Arguments:

- -y --type: The image type (png or jpeg)
- -p --prefix: The flag to include the prefix (data:image/png;base64,) or not
- -w --width: The image width
- -h --height: The image height
- -b --background: The background color

- -i --backgroundImage: The background image url or file

- -s --stroke: The stroke Color

- -t --strokeOpacity: The stroke opacity

- -r --strokeWidth: The stroke width

- -l --fill: The fill Color

- -o --fillOpacity: The fill opacity

- -m --shape: The marker shape (circle, square, ect..)

- -z --size: The marker size

- -c --drawCoords: The flag for drawing coordinates or not

- -e --envelope: The geographical bounds (minx, miny, maxx, maxy)

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom drawbase64 -g "POLYGON ((5 5, 5 15, 15 15, 15 5, 5 5))"
```

dump

Name:

geom dump

Description:

Put each geometry from a GeometryCollection on it's own line.

Arguments:

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom dump -g "MULTIPOINT (1 1, 2 2)"
```

ellipse

Name:

geom ellipse

Description:

Create an ellipse of circle around the input geometry.

Arguments:

- -w --width: The width
- -h --height: The height
- -p --numberOfPoints: The number of points
- -r --rotation: The rotation
- -c --center: The flag to use center (true) or the base (false)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom ellipse -g "POINT (100 100)" -p 30 -w 500 -h 500
```
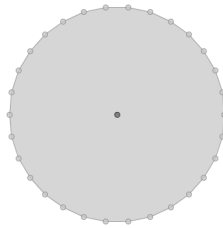

difference

Name:

geom difference

Description:

Calculate the difference between the input geometries.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom difference -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((5 5, 5 15, 15 15,
```

dimension

Name:

geom dimension

Description:

Get the dimension of the Geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom dimension -g "POINT (1 1)"
```

hausdorffdistance

Name:

geom hausdorffdistance

Description:

Calculate the discrete hausdorff distance between the two input geometries.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom hausdorffdistance -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((5 5, 5 15,
```

disjoint

Name:

geom disjoint

Description:

Determine whether the first geometry is disjoint from the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom disjoint -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "LINESTRING (15 15, 20 20)"
```

distance

Name:

geom distance

Description:

Calculate the distance between the two input geometries.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom distance -g "POINT (5 5)" -o "POINT (20 21)"
```

Geometry Commands

distanceline

Name:

geom distanceline

Description:

Generate a LineString representing the shortest distance between two geometries.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom distanceline -g "POINT (5 5)" -o "POINT (20 21)"
```

envelope

Name:

geom envelope

Description:

Calculate the envelope of a Geometry.

Arguments:

- -e --expandBy: The distance to expand the Envelope
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom envelope -g "POLYGON ((9 52, 9 50, 7 50, 7 48, 10 47, 10 46, 13 46, 11 52, 10 52, 9 52)
```

equals

Name:

geom equals

Description:

Determine whether the first geometry equals the second geometry.

Arguments:

- -t --type: The type of equals (exact, norm, topo)
- -l --tolerance: The tolerance when type is exact
- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom equals -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((0 0, 0 10, 10 10, 10
```

Geometry Commands

fromwkb

Name:

geom fromwkb

Description:

Convert a WKB to a WKT Geometry.

Arguments:

- -b --wkb: The WKB text
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom fromwkb -b 00000000013FF00000000000004000000000000000
```

interiorpoint

Name:

geom interiorpoint

Description:

Calculate the interior point of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom interiorpoint -g "POLYGON ((8 54, 8 51, 14 51, 14 47, 7 47, 7 42, 17 45, 16 53, 13 53,
```

interpolatepoint

Name:

geom interpolatepoint

Description:

Interpolate the location of a point on the input linear geometry given a percentage position.

Arguments:

- -p --position: The position between 0 and 1
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom interpolatepoint -g "LINESTRING (0 0, 5 5, 10 10)" -p 0.25
```

intersection

Name:

geom intersection

Description:

Calculate the intersection between two geometries.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom intersection -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((5 5, 5 15, 15 1
```

intersects

Name:

geom intersects

Description:

Determine if the first geometry intersects the second geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom intersects -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((5 5, 5 15, 15 15,
```

isccw

Name:

geom isccw

Description:

Is the geometry's coordinates oriented counter clockwise of not.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom isccw -g "LINEARRING (15 20, 10 20, 10 10, 15 10, 15 20)"
```

isclosed

Name:

geom isclosed

Description:

Determine if the input geometry is closed or not.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom isclosed -g "LINESTRING (1 1, 1 5, 5 5, 5 1, 1 1)"
```

isempty

Name:

geom isempty

Description:

Determine if the input geometry is empty or not.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom isempty -g "POINT EMPTY"
```

isrectangle

Name:

geom isrectangle

Description:

Determine if the input geometry is rectangular or not.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom isrectangle -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))"
```

isring

Name:

geom isring

Description:

Determine if the input geometry is a ring or not.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom isring -g "LINESTRING (1 1, 1 5, 5 5, 5 1, 1 1)"
```

issimple

Name:

geom issimple

Description:

Determine if the input geometry is simple or not.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom issimple -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))"
```

isvalid

Name:

geom isvalid

Description:

Determine if the input geometry is valid or not.

Arguments:

- -t --type: The flag to show the validation error message, the error location, or validity (msg, loc, or val)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom isvalid -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))"
```

iswithindistance

Name:

geom iswithindistance

Description:

Determine if the input geometry is within the given distance of the other geometry.

Arguments:

- -d --distance: The distance
- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom iswithindistance -g "POINT (1 1)" -o "POINT (20 23)" -d 30
```

kochsnowflake

Name:

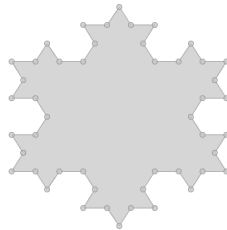geom kochsnowflake

Description:

Generate a koch snow flake.

Arguments:

- -n --number: The number of points.
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom kochsnowflake -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -n 100
```


hilbertcurve

Name:

geom hilbertcurve

Description:

Create a hilbert curve.

Arguments:

- -n --number: The number of points.
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom hilbertcurve -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -n 18
```

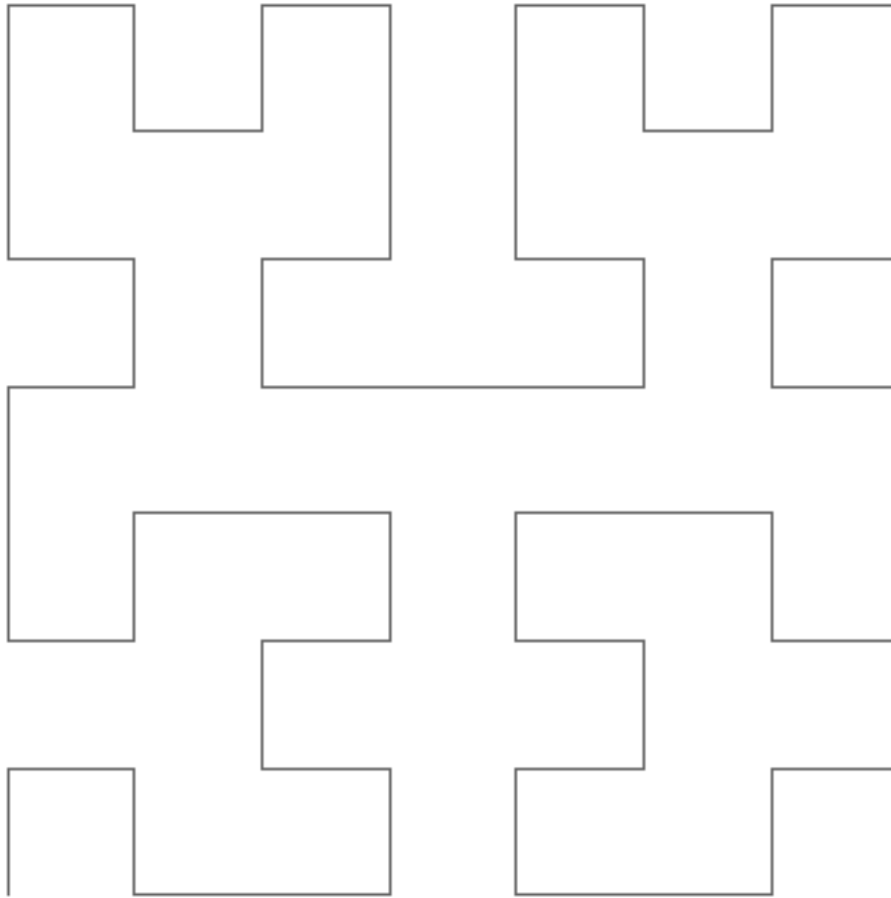## *mortoncurve*

Name:

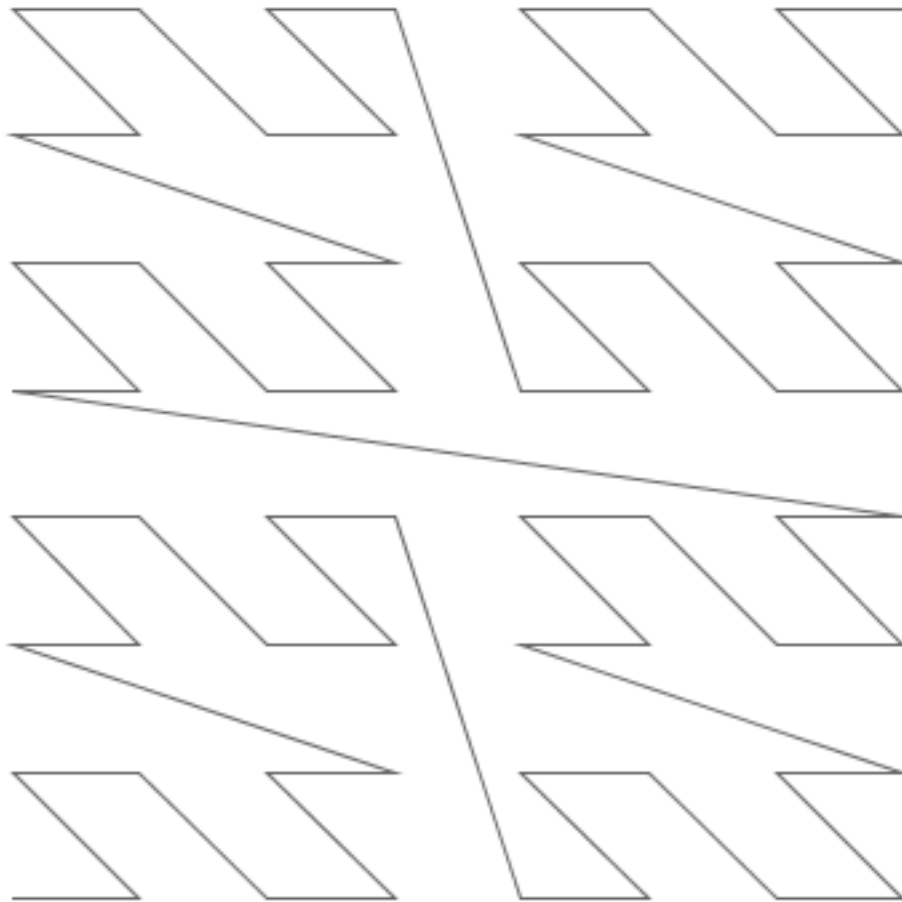geom mortoncurve

Description:

Create a morton curve.

Arguments:

- -n --number: The number of points.
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom mortoncurve -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -n 54
```

linedissolve

Name:

geom linedissolve

Description:

Dissolve LinesStrings

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom linedissolve -g "LINESTRING (1143429.5177049513 646812.5700195221, 1148620.8088546866 6
```

linemerge

Name:

geom linemerge

Description:

Merge lines of the input geoemtry together.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom linemerge -g "MULTILINESTRING((-29 -27,-30 -29.7,-36 -31,-45 -33),(-45 -33,-46 -32))"
```

locatepoint

Name:

geom locatepoint

Description:

Locate the position of a point on the linear geometry as a percentage of the distance.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom locatepoint -g "LINESTRING (0 0, 5 5, 10 10)" -o "POINT (2.5 2.5)"
```

type

Name:

geom type

Description:

Get the type of the geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom type -g "POINT (1 1)"
```

grid

Name:

geom grid

Description:

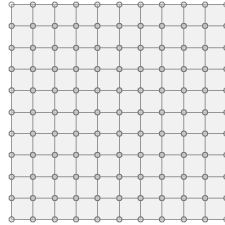Calculate a grid around the input geometry.

Arguments:

- -c --columns: The number of columns
- -r --rows: The number of rows

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom grid -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -c 10 -r 10
```


endpoint

Name:

geom endpoint

Description:

Get the end point of a LineString or a MultiLineString.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom endpoint -g "LINESTRING (1 1, 5 5, 10 10)"
```

get

Name:

geom get

Description:

Get a sub geometry from a geometry collection by index.

Arguments:

- -n --index: The index number of the Geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom get -g "MULTIPOINT (1 1, 5 5, 10 10)" -n 1
```

startpoint

Name:

geom startpoint

Description:

Get the start point of a LineString or MultiLineString.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom startpoint -g "LINESTRING (1 1, 5 5, 10 10)"
```

mincircle

Name:

geom mincircle

Description:

Calculate the minimum bounding circle of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom mincircle -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))"
```


minclearance

Name:

geom minclearance

Description:

Calculate the minimum clearance of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom minclearance -g "POLYGON ((16 51, 7 53, 12 46, 5 41, 12 41, 16 51))"
```

mindiameter

Name:

geom mindiameter

Description:

Calculate the minimum diameter of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom mindiameter -g "POLYGON ((16 51, 7 53, 12 46, 5 41, 12 41, 16 51))"
```

minrect

Name:

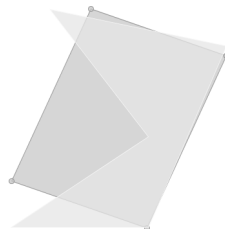geom minrect

Description:

Calculate the minimum rectangle of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom minrect -g "POLYGON ((16 51, 7 53, 12 46, 5 41, 12 41, 16 51))"
```


narrow

Name:

geom narrow

Description:

Convert a geometry collection to it's most specific type.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom narrow -g "GEOMETRYCOLLECTION (POINT (1 1), POINT(10 10))"
```

nearestpoints

Name:

geom nearestpoints

Description:

Get the nearest points between two geometries

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
null
```

node

Name:

geom node

Description:

Node the linestrings from the input geometry.

Arguments:

- -n --number: The number of decimal places
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom node -g "LINESTRING (5 51, 8 54, 12 50, 8 47, 10 55, 7 52, 5 53, 6 51)" -n 5
```

normalize

Name:

geom normalize

Description:

Calculate the normalized version of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom normalize -g "MULTIPOINT ((12.5 12.5), (13.5 7.5), (7 9), (9.5 13), (8 7.5))"
```

octagonalenvelope

Name:

geom octagonalenvelope

Description:

Calculate the octagonal envelope of the input geometry.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom octagonalenvelope -g "MULTIPOINT ((12.5 12.5), (13.5 7.5), (7 9), (9.5 13), (8 7.5))"
```

overlaps

Name:

geom overlaps

Description:

Determine whether the first geometry overlaps with the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom overlaps -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((2 2, 2 14, 14 14, 1
```

placepoint

Name:

geom placepoint

Description:

Place a point on the input linear geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom placepoint -g "LINESTRING (0 0, 5 5, 10 10)" -o "POINT (3 4.5)"
```

pointatangle

Name:

geom pointatangle

Description:

Calculate a point at a given angle distance from the input point.

Arguments:

- -d --distance: The distance
- -a --angle: The angle (in degrees)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom pointatangle -g "POINT (10 10)" -a 90 -d 10
```

polygonize

Name:

geom polygonize

Description:

Creates polygons from lines.

Arguments:

- -f --full: Whether to include a full report (polygons, cutEdges, dangles, and invalidRingLines)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom polygonize -g "MULTILINESTRING ((-5.5 45, -4.5 46.5), (-4.5 46.5, 1 52), (1 52, 2.5 54)
```

reduceprecision

Name:

geom reduceprecision

Description:

Reduce the precision of the input geometry.

Arguments:

- -t --type: The precision model type (FIXED, FLOATING, FLOATING_SINGLE)
- -s --scale: The precision model scale when type is FLOATING
- -p --pointWise: Whether the precision reducer operates pointwise
- -r --removeCollapsed: Whether the precision reducer should remove collapsed geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom reduceprecision -g "POINT (-5.70068359375 45.1416015625)" -s 2 -t fixed
```

project

Name:

geom project

Description:

Project the input geometry from one coordinate system to another.

Arguments:

- -s --source: The source projection
- -t --target: The target projection
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom project -g "POINT (1179931.55 645310.31)" -s EPSG:2927 -t EPSG:4326
```

random

Name:

geom random

Description:

Generate random points inside the input geometry.

Arguments:

- -n --number: The number of points
- -r --gridded: The flag for whether the random points should be gridded.
- -c --constrained: The flag for whether the random points should be constrained to a circle when gridded.
- -f --gutterFraction: The gutter distance or padding for random points when gridded.
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom random -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -n 100
```

randomwalk

Name:

geom randomwalk

Description:

Generate a random walk as a linestring.

Arguments:

- -n --number: The number of walks
- -d --distance: The distance between Coordinates
- -p --probability: The probability of changing direction
- -a --angle: The angle increment (in degrees) when changing direction
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom randomwalk -n 100 -a 45 -d 10 -g "POINT (1 1)"
```

rectangle

Name:

geom rectangle

Description:

Create a rectangle from the input geometry.

Arguments:

- -w --width: The width
- -h --height: The height
- -p --numberOfPoints: The number of points
- -r --rotation: The rotation
- -c --center: The flag to use center (true) or the base (false)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom rectangle -g "POINT (100 100)" -p 30 -w 500 -h 500
```
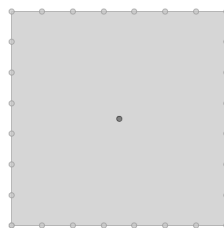

reflect

Name:

geom reflect

Description:

Create a new geometry by applying the reflect affine transformation on the input geometry.

Arguments:

- -0 --x0: The x-ordinate of a point on the reflection line
- -1 --y0: The y-ordinate of a point on the reflection line
- -2 --x1: The x-ordinate of a another point on the reflection line
- -3 --y1: The y-ordinate of a another point on the reflection line
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom reflect -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -0 5 -1 2
```

relate

Name:

geom relate

Description:

Determine if the input Geometry and the other Geometry are related according to the DE-9IM intersection matrix or calculate the DE-9IM.

Arguments:

- -m --matrix: The DE-9IM intersection matrix
- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom relate -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POINT (5 5)"
```

reverse

Name:

geom reverse

Description:

Reverse the coordinates of the input geometry

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom reverse -g "LINESTRING (0 0, 5 5, 10 10)"
```

rotate

Name:

geom rotate

Description:

Create a new geometry by rotating the input geometry

Arguments:

- -t --theta: The rotation angle, in radians

- -x --xCoordinate: The x-ordinate of the rotation point

- -y --yCoordinate: The y-ordinate of the rotation point

- -s --sine: The sine of the rotation angle

- -c --cosine: The cosine of the rotation angle

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom rotate -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -t 0.785398163
```

scale

Name:

geom scale

Description:

Create a new geometry by scaling the input geometry

Arguments:

- -s --xscale: The value to scale by in the x direction

- -t --yscale: The value to scale by in the y direction

- -x --xcoordinate: The x-ordinate of the point to scale around

- -y --ycoordinate: The y-ordinate of the point to scale around

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom scale -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -s 2 -t 5
```

shear

Name:

geom shear

Description:

Create a new geometry by apply a shear affine transformation to the input geometry

Arguments:

- -x --xDistance: The value to translate by in the x direction

- -y -yDistance: The value to translate by in the y direction

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom shear -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -x 4 -y 2
```

sierpinskicarpet

Name:

geom sierpinskicarpet
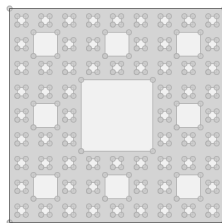
Description:

Create a sierpinski carpet.

Arguments:

- -n --number: The number of points.
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom sierpinskicarpet -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -n 100
```


similarity

Name:

geom similarity

Description:

Calculate the degree of similarity between two geometries.

Arguments:

- -a --algorithm: The algorithm (area/a or hausdorff/h)
- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom similarity -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((2 2, 2 14, 14 14,
```

simplify

Name:

geom simplify

Description:

Simplify the input geometry.

Arguments:

- -a --algorithm: The distance tolerance (douglaspeucker/dp or topologypreserving/tp or visvalingamwhyat/vw)

- -d --distance: The distance tolerance

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom simplify -g "LINESTRING (1 1, 2.5 2.5, 3.5 3.5, 5 5, 6.5 6.5, 8 8, 9 9, 10.5 10.5, 12 1
```

sinestar

Name:

geom sinestar

Description:

Create a sine star.

Arguments:

- -n --numberOfArms: The number of arms

- -l --armLengthRatio: The arm length ratio

- -w --width: The width

- -h --height: The height

- -p --numberOfPoints: The number of points

- -r --rotation: The rotation

- -c --center: The flag to use center (true) or the base (false)

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom sinestar -l 0.9 -n 12 -p 1000 -w 500 -h 500 -g "POINT (100 100)"
```
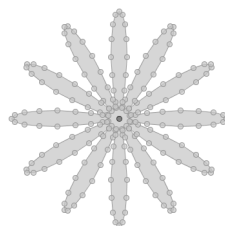

slice

Name:

geom slice

Description:

Get a subset of geometries using a start and end index.

Arguments:

- -s --start: The start index number

- -e --end: The end index number

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
null
```

split

Name:

geom split

Description:

Split a Geometry by another Geometry

Arguments:

- -o --otherGeometry: The other geometry

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom split -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "LINESTRING (0 0, 10 10)"
```

spoke

Name:

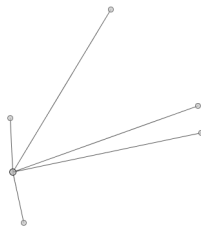geom spoke

Description:

Create a spoke diagram with lines between a single Geometry to other Geometries

Arguments:

- -o --otherGeometry: The other geometry

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom spoke -g "MULTIPOINT ((5.875473869469681 1.0101660098606535), (19.64273518313129 8.0328
```


snap

Name:

geom snap

Description:

Snap the input geometry to the other geometry.

Arguments:

- -d --distance: The distance/tolerance
- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom snap -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((11 11, 11 20, 20 20, 20
```

squircle
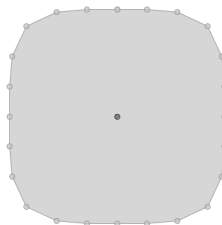
Name:

geom squircle

Description:

Create a squircle.

Arguments:

- -w --width: The width
- -h --height: The height
- -p --numberOfPoints: The number of points
- -r --rotation: The rotation
- -c --center: The flag to use center (true) or the base (false)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom squircle -g "POINT (100 100)" -p 30 -w 500 -h 500
```


subline

Name:

geom subline

Description:

Extract a sub line from a linear geometry.

Arguments:

- -s startPosition: The start position between 0 and 1

- -e endPosition: The end position between 0 and 1

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom subline -g "LINESTRING (0 0, 10 10, 20 20)" -s 0.25 -e 0.75
```

supercircle
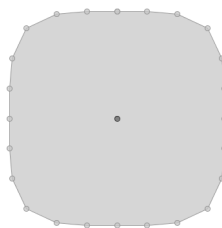
Name:

geom supercircle

Description:

Create a super circle.

Arguments:

- -o --power: The positive power

- -w --width: The width

- -h --height: The height

- -p --numberOfPoints: The number of points

- -r --rotation: The rotation

- -c --center: The flag to use center (true) or the base (false)

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom supercircle -g "POINT (100 100)" -p 30 -w 500 -h 500
```


symdifference

Name:

geom symdifference

Description:

Calculate the symetric difference between two geometries

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom symdifference -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((5 5, 5 20, 20
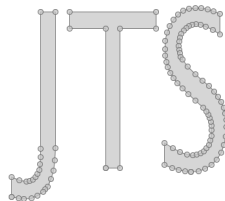```

## *text*

**Name**:

geom text

**Description**:

Create a geometry from a string.

**Arguments**:

- -t --text: The text
- -f --fontName: The font name
- -s --pointSize: The font size
- --help : Print help message
- --web-help : Open help in a web browser

**Example**:

```
geom text -t "JTS"
```


touches

Name:

geom touches

Description:

Determine if the input geometry touches another geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom touches -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((10 10, 10 14, 14 14,
```

towkb

Geometry Commands

Name:

geom towkb

Description:

Write a Geometry to WKB.

Arguments:

- -d --dimension: The output dimension (2 or 3)
- -b --byte-order: The byte order (1 = big endian, 2 = little endian)
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom towkb -g "POINT (10 10)"
```

translate

Name:

geom translate

Description:

Create a new geometry by applying the translate affine transformation on the input geometry.

Arguments:

- -x --xDistance: The value to translate by in the x direction
- -y --yDistance: The value to translate by in the y direction
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom translate -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -x 4 -y 2
```

union

Name:

geom union

Description:

Calculate the union between two geometries.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom union -g "POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0))" -o "POLYGON ((10 10, 10 14, 14 14, 1
```

voronoi

Name:
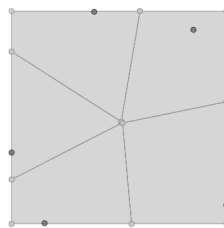
geom voronoi

Description:

Generate a voronoi diagram.

Arguments:

- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom voronoi -g "MULTIPOINT ((12.5 12.5), (13.5 7.5), (7 9), (9.5 13), (8 7))"
```


within

Name:

geom within

Description:

Determine if the input geometry is within the other geometry.

Arguments:

- -o --otherGeometry: The other geometry
- -g --geometry: The input geometry
- --help : Print help message
- --web-help : Open help in a web browser

Example:

```
geom within -g "POINT (2 2)" -o "POLYGON ((1 1, 1 10, 10 10, 10 1, 1 1))"
```

pointsalong

Name:

geom pointsalong

Description:

Place points along a line

Arguments:

- -d --distance: The distance between points

- -g --geometry: The input geometry

- --help : Print help message

- --web-help : Open help in a web browser

Example:

```
geom pointsalong -g "LINESTRING (0 0, 10 10)" -d 1
```

Indices and tables

- *genindex*

- *modindex*

- *search*