[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

PROCESSING **GEOMETRY** [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

A geometry is is an EE variable object that represents a set of points, lines, or polygons. Geometries can be processed by using operations of the types listed below,

which vary according to the nature of that processing. Each operation name is linked to a separate page describing that operation.

**CREATING** GEOMETRIES [ee.Geometry.Point](#Point) [ee.Geometry.MultiPoint](#MultiPoint) [Onscreen Drawing](#Drawing)

[ee.Geometry.LineString](#LineString) [ee.Geometry.MultiLineString](#MultiLineString)

[ee.Geometry.LinearRing](#LinearRing) [ee.Geometry.Rectangle](#Rectangle)

[ee.Geometry.Polygon](#Polygon) [ee.Geometry.MultiPolygon](#MultiPolygon) [ee.Geometry](#Geometry)

**EDITING** GEOMETRIES

BY **REPROJECTING** [geometry.transform](#transform)

**TRANSFORMING** GEOMETRIES [geometry.centroid](#centroid) [geometry.simplify](#simplify)

[geometry.bounds](#bounds) [geometry.convexHull](#convexHull) [geometry.buffer](#buffer)

[geometry.union](#union_) [geometry.intersection](#union_) [geometry.difference](union_#union_)

[geometry.symmetricDifference](#union_)

**REPRODUCING** GEOMETRIES

AS **LISTS** [geometry.geometries](#geometries) [geometry.coordinates](#coordinates)

AS **FEATURES** [ee. Algorithms.Feature](#Feature) [ee.Feature(geometry)](#Feature)

AS **FEATURE COLLECTIONS** [ee.FeatureCollection(geometry)](#FeatureCollection)

**QUERYING** GEOMETRIES

FOR **PROJECTIONS** [geometry.projection](#projection)

**COMPARING** GEOMETRIES [ee.Algorithms.IsEqual(geometry)](#IsEqual)

**MEASURING** GEOMETRIES [geometry.length](#length) [geometry.perimeter](#perimeter)  [geometry.area](#area) [geometry.distance](#distance)

**DESCRIBING** GEOMETRIES [geometry.type](#type) [geometry.isUnbounded](#isUnbounded) [geometry.edgesAreGeodesics](#edgesAreGeo)

**DOCUMENTING** GEOMETRIES [geometry.getInfo](#Describe_) [geometry.toGeoJSON](#Describe_) [ee.Algorithms.Describe(geometry)](#Describe_)

[geometry.toString](#toString_) [geometry.serialize](#toString_) [geometry.toGeoJSONString](#toString_)

**PRESENTING** GEOMETRIES

IN **PRINT** [print(geometry)](#print_console) [console.log(geometry)](#print_console) [alert(geometry)](#alert_confirm) [confirm(geometry)](#alert_confirm)

IN **MAPS** [Map.addLayer(geometry)](#addLayer)

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **ONSCREEN DRAWING**

DESCRIBE THE ONSCREEN DRAWING TOOLS, FOCUSING ONLY ON GEOMETRIES

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.Point creates a new point geometry from specified longitude and latitude coordinates.

newGeometry = ee.Geometry.Point ( longitudeCoordinate, latitudeCoordinate )

The specified latitude, given as a number representing degrees

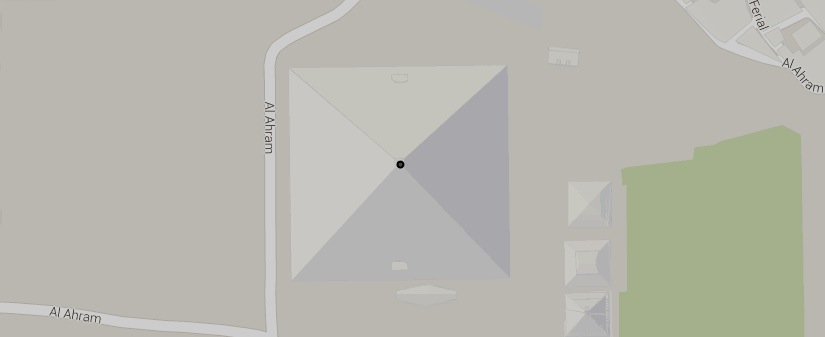
The specified longitude, given as a number representing degrees

The new point geometry

var TheGEOMETRY = ee.Geometry.Point( 31.134204, 29.979241 );

Map.centerObject( TheGEOMETRY,17 );

Map.addLayer( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.LineString creates a new lineString geometry from a specified sequence of longitude-latitude pairs of coordinates.

newGeometry = ee.Geometry.LineString ( coordinatePairs )

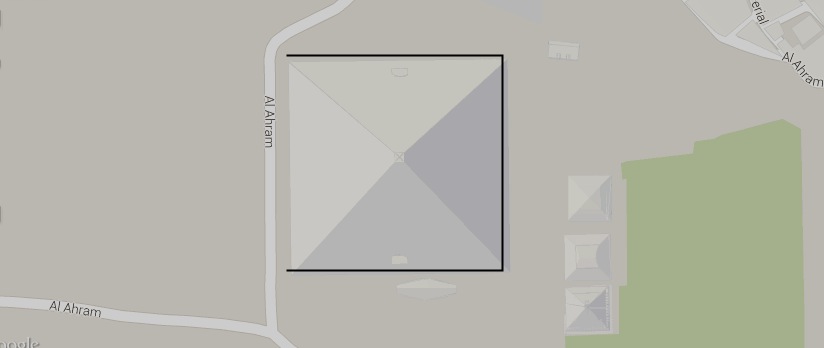
The specified set of coordinate pairs, given as a list of [lon,lat] lists or as a comma-separated sequence of (lon,lat,lon,lat…) numbers.

The new lineString geometry

var TheGEOMETRY = ee.Geometry.LineString( [ [31.1330,29.9802], [31.1353,29.9802], [31.1353,29.9782], [31.1330,29.9782] ] );

Map.centerObject( TheGEOMETRY,17 );

Map.addLayer( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.LinearRing creates a new linearRing geometry from a specified sequence of longitude-latitude pairs of coordinates.

newGeometry = ee.Geometry.LinearRing ( coordinatePairs )

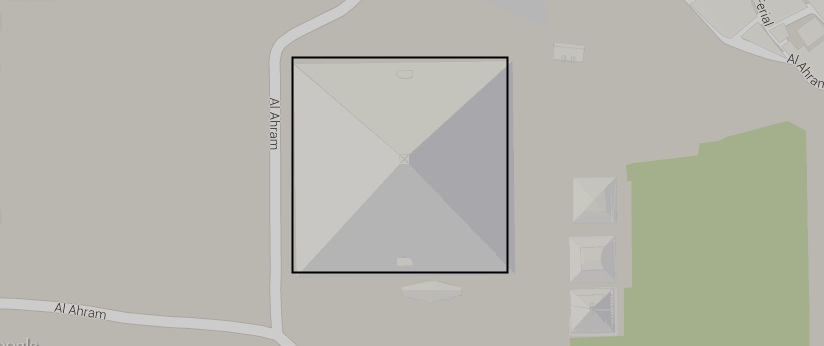
The specified set of coordinate pairs, given as a list of [lon,lat] lists or as a comma-separated sequence of (lon,lat,lon,lat…) numbers.

The new linearRing geometry

var TheGEOMETRY = ee.Geometry.LinearRing( 31.1330,29.9802, 31.1353,29.9802, 31.1353,29.9782, 31.1330,29.9782 );

Map.centerObject( TheGEOMETRY,17 );

Map.addLayer( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.Rectangle creates a new rectangle geometry from a specified set of minimum and maximum longitude and latitude coordinates.

newGeometry = ee.Geometry.Rectangle ( minimumLon, minimumLat, maximumLon, maximumLat )

The specified set of westerly, southerly, easterly, and northerly coordinates, each given as a number

The new rectangle geometry

var TheGEOMETRY = ee.Geometry.Rectangle( 31.1330, 29.9782, 31.1353, 29.9802 );

Map.centerObject( TheGEOMETRY,17 );

Map.addLayer( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.Polygon creates a new polygon geometry from a specified sequence of longitude-latitude pairs of coordinates. Though coordinates may be

specified in either clockwise or counterclockwise order, the latter is usually preferable for reasons described [here](#area).

newGeometry = ee.Geometry.Polygon ( coordinatePairs )

The specified set of coordinate pairs, given as a list of one or more sub-lists. The first sub-list defines the polygon’s exterior perimeter by listing its vertices, each given as a [lon,lat] sub-sublist of coordinates. Any subsequent sub-list defines one of the polygon’s holes in the same manner.

The new polygon

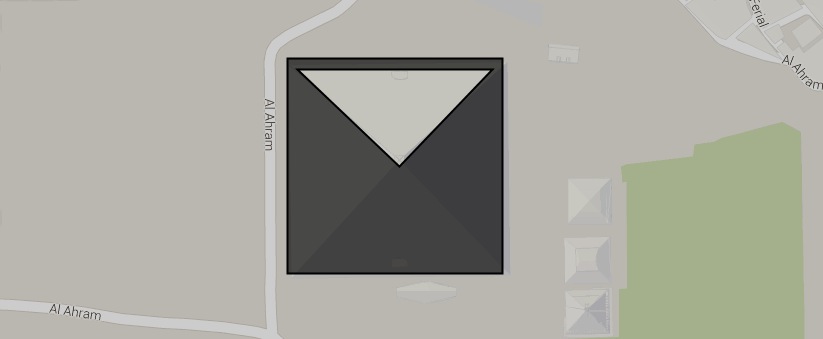
geometry

var ThePOLYGON = ee.Geometry.Polygon( [ [ [31.1330,29.9802], [31.1353,29.9802], [31.1353,29.9782], [31.1330,29.9782] ],

[ [31.1331,29.9801], [31.1352,29.9801], [31.1342,29.9792] ] ] );

Map.setCenter( 31.1342, 29.9792, 17 );

Map.addLayer( ThePOLYGON );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.MultiPoint creates a new multiPoint geometry from a specified set of longitude-latitude pairs of coordinates.

newGeometry = ee.Geometry.Point ( coordinatePairs )

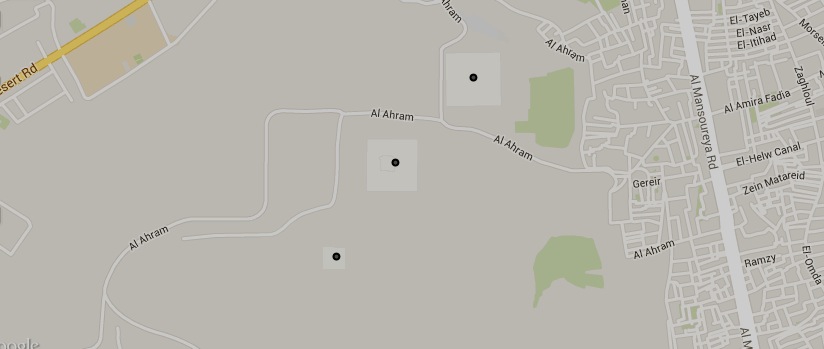
The specified set of coordinate pairs, given as a list of [lon,lat] lists or as a comma-separated sequence of (lon,lat,lon,lat…) numbers.

The new multiPoint geometry

var TheGEOMETRY = ee.Geometry.MultiPoint( 31.134204,29.979241, 31.130855,29.976089, 31.128323,29.972594 );

Map.centerObject( TheGEOMETRY,15 );

Map.addLayer( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.MultiLineString creates a new multiLineString geometry from a specified sequence of longitude-latitude pairs of coordinates.

newGeometry = ee.Geometry.LineString ( coordinatePairs )

The specified set of coordinate pairs, given as a list of linestrings (or **ee.Geometry.LineString** inputs)

The new multiLineString geometry

var TheGEOMETRY = ee.Geometry.MultiLineString( [ [ [31.1330,29.9802], [31.1353,29.9802], [31.1353,29.9782], [31.1330,29.9782] ],

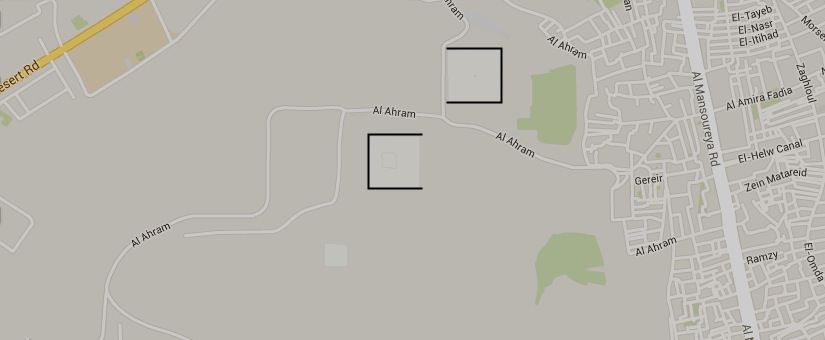
[ [31.1319,29.9770], [31.1296,29.9770], [31.1296,29.9750], [31.1319,29.9750] ]

]

);

Map.centerObject( TheGEOMETRY,15 );

Map.addLayer( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry.MultiPolygon creates a new multiPolygon geometry from a specified sequence of longitude-latitude pairs of coordinates.

newGeometry = ee.Geometry.LineString ( coordinatePairs )

The specified set of coordinate pairs, given as a list of linestrings (or **ee.Geometry.LineString** inputs)

The new multiPolygon geometry

var MultiPOLYGON = ee.Geometry.MultiPolygon( [

ee.Geometry.Polygon([[31.1330,29.9802],[31.1353,29.9802],[31.1353,29.9782],[31.1330,29.9782]]),

ee.Geometry.Polygon([[31.1319,29.9770],[31.1296,29.9770],[31.1296,29.9750],[31.1319,29.9750]])

]

);

Map.setCenter( 31.1342, 29.9792, 15 );

Map.addLayer( MultiPOLYGON );

THE SECOND POLYGON IS NOT BEING RECOGNIZED

****

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**CREATING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Geometry creates a new geometry from a specified GeoJSON object as described [here](http://en.wikipedia.org/wiki/GeoJSON).

newGeometry = ee.Geometry ( oldObject, *coordinateSystem*, *geodesic?* )

A Boolean set to True (only) if line segments in the specified coordinate system are to be regarded as drawn on a sphere rather than a plane. Default: True for spheroidal coordinate systems and False for planar coordinate systems

The specified GeoJSON object

The new geometry

A coordinate system that will override whatever one may have been part of the specified GeoJSON object, given as a named (rather than "linked") CRS code or as a WKT string. Default: "EPSG:4326" (spheroidal)

var MultiPOLYGON = ee.Geometry(

{ "type": "MultiPolygon",

"coordinates": [ [ [ [31.1319,29.9770], [31.1296,29.9770], [31.1296,29.9750], [31.1319,29.9750] ]

],

[ [ [31.1330,29.9782], [31.1353,29.9782], [31.1353,29.9802], [31.1330,29.9802] ],

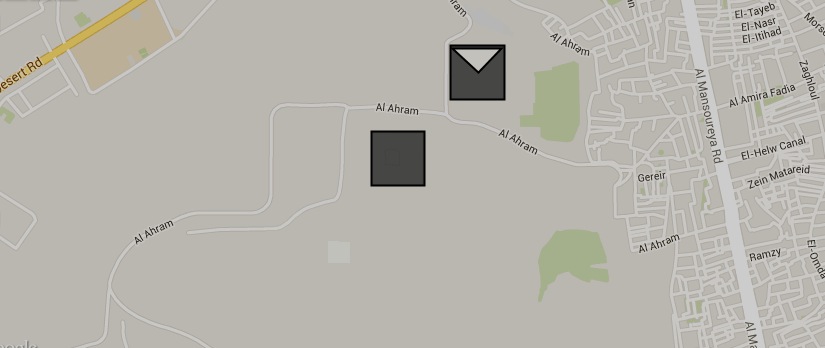
[ [31.1342,29.9792], [31.1352,29.9801], [31.1331,29.9801] ]

]

]

} );

Map.setCenter( 31.1342, 29.9792, 15 );

Map.addLayer( MultiPOLYGON );

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**EDITING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **REPROJECTING**

geometry.transform creates a new geometry by reprojecting a specified geometry to a specified coordinate system, with lines interpreted as either planar

or spheroidal according to the planar or spheroidal nature of that coordinate system.

newGeometry = oldGeometry.transform ( *coordinateSystem*, *errorMargin* )

An ErrorMargin object indicating the maximum

allowable reprojection error in meters

The specified coordinate system, given as an

EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: WGS84

The specified

geometry

The new

geometry

var OldGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

var NewGEOMETRY = OldGEOMETRY.transform( 'EPSG:2772', ee.ErrorMargin(100) );

print( OldGEOMETRY, NewGEOMETRY );

Map.centerObject ( NewGEOMETRY, 5 );

Map.addLayer( OldGEOMETRY );

Map.addLayer( NewGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.centroid creates a new point geometry at the centroid of (the highest-dimensional components of) a specified geometry.

newGeometry = oldGeometry.centroid ( *errorMargin,* *coordinateSystem* )

The new

geometry

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum

allowable placement error in meters

The specified

geometry

var AllStateCOLLECTION = ee.FeatureCollection('ft:1fRY18cjsHzDgGiJiS2nnpUU3v9JPDc2HNaR7Xk8');

var OneStateCOLLECTION = AllStateCOLLECTION.filter( ee.Filter.eq('Name', 'Iowa') );

var OneStateELEMENT = OneStateCOLLECTION.first( );

var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

Map.centerObject( OneStateGEOMETRY, 6 ); Map.addLayer( OneStateGEOMETRY, {color:'331188'} );

var NewGEOMETRY = OneStateGEOMETRY.centroid( ); Map.addLayer( NewGEOMETRY, {color:'ff0000'} );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.simplify creates a new geometry by removing points from a specified geometry without exceeding a specified error margin.

newGeometry = oldGeometry.centroid ( errorMargin*,* *coordinateSystem* )

An ErrorMargin object indicating the maximum

allowable placement error in meters. This will

maintained regardless of zoom level.

The new

geometry

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: Same as **OldGeometry**

The specified

geometry

var AllStateCOLLECTION = ee.FeatureCollection('ft:1fRY18cjsHzDgGiJiS2nnpUU3v9JPDc2HNaR7Xk8');

var OneStateCOLLECTION = AllStateCOLLECTION.filter( ee.Filter.eq('Name', 'Idaho') );

var OneStateELEMENT = OneStateCOLLECTION.first( );

var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

Map.centerObject( OneStateGEOMETRY, 6 ); Map.addLayer( OneStateGEOMETRY, {color:'331188'} );

var NewGEOMETRY = OneStateGEOMETRY.simplify( 50000 ); Map.addLayer( NewGEOMETRY, {color:'ffffff'} );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.bounds creates a new rectangular geometry extending from the northerly to the southerly and the westerly to the easterly extent of a specified geometry.

newGeometry = oldGeometry.bounds ( *errorMargin,* *coordinateSystem* )

The new

geometry

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum

allowable placement error in meters

The specified

geometry

var AllStateCOLLECTION = ee.FeatureCollection('ft:1fRY18cjsHzDgGiJiS2nnpUU3v9JPDc2HNaR7Xk8');

var OneStateCOLLECTION = AllStateCOLLECTION.filter( ee.Filter.eq('Name', 'Florida') );

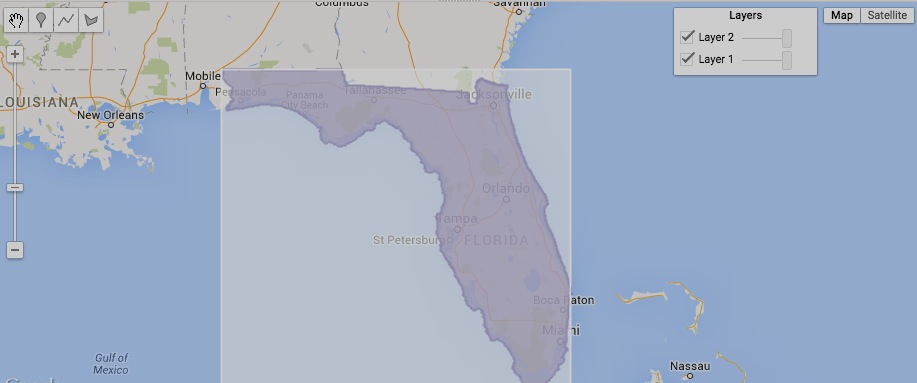
var OneStateELEMENT = OneStateCOLLECTION.first( );

var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

Map.centerObject( OneStateGEOMETRY, 6 ); Map.addLayer( OneStateGEOMETRY, {color:'331188'} );

var NewGEOMETRY = OneStateGEOMETRY.bounds( ); Map.addLayer( NewGEOMETRY, {color:'ffffff'} );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.convexHull creates a new geometry encompassing a specified geometry by connecting its outermost vertices.

newGeometry = oldGeometry.convexHull ( *errorMargin,* *coordinateSystem* )

The new

geometry

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum

allowable placement error in meters

The specified

geometry

var AllStateCOLLECTION = ee.FeatureCollection('ft:1fRY18cjsHzDgGiJiS2nnpUU3v9JPDc2HNaR7Xk8');

var OneStateCOLLECTION = AllStateCOLLECTION.filter( ee.Filter.eq('Name', 'New York') );

var OneStateELEMENT = OneStateCOLLECTION.first( );

var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

Map.centerObject( OneStateGEOMETRY, 6 ); Map.addLayer( OneStateGEOMETRY, {color:'331188'} );

var NewGEOMETRY = OneStateGEOMETRY.convexHull( ); Map.addLayer( NewGEOMETRY, {color:'ffffff'} );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.buffer creates a new polygonal geometry whose boundaries are all at a specified distance from those of a specified geometry.

newGeometry = oldGeometry.buffer ( distanceOutward, *errorMargin,* *coordinateSystem* )

An ErrorMargin indicating the maximum allowable placement error in meters. Default: **distance** \* 0.1

The specified distance, given as a number in meters

or, if specified, the units of **coordinateSystem**. Positive distances are measured outward (and negative distances inward) from specified geometry boundaries.

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

The new geometry

The specified geometry

var AllStateCOLLECTION = ee.FeatureCollection('ft:1fRY18cjsHzDgGiJiS2nnpUU3v9JPDc2HNaR7Xk8');

var OneStateCOLLECTION = AllStateCOLLECTION.filter( ee.Filter.eq('Name', 'Colorado') );

var OneStateELEMENT = OneStateCOLLECTION.first( );

var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

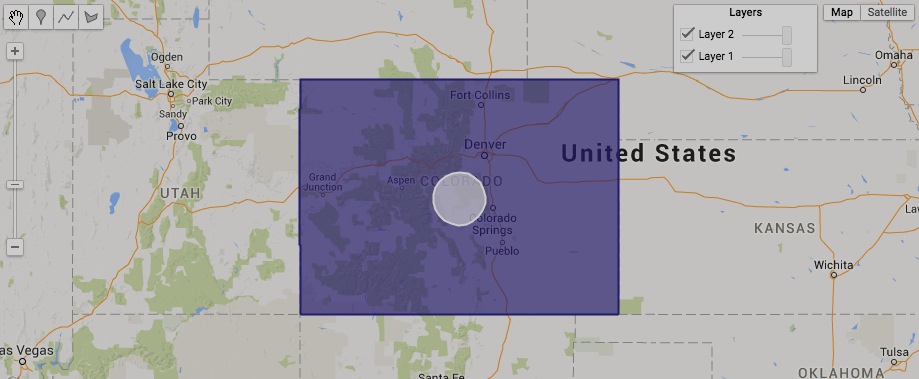
var NewGEOMETRY = OneStateGEOMETRY.centroid().buffer( 50000 );

Map.centerObject( OneStateGEOMETRY, 6 );

Map.addLayer( OneStateGEOMETRY, {color:'331188'} );

Map.addLayer( NewGEOMETRY, {color:'ffffff'} );

can buffer point but not polygon



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**TRANSFORMING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.union , .intersection , .symmetricDifference , and Difference

create a new polygonal geomety encompassing locations that are determined by two specified polygonal geometries such that

* **union** calls for locations included in either or both of the specified geometries;
* **intersection** calls for locations included both of the specified geometries;
* **symmetricDifference** calls for locations included in either but not both of the specified geometries; and
* **Difference** calls for locations included in the 1st but not the 2nd of the specified geometries;

newGeometry =1stOldGeometry.union ( 2ndOldGeometry, *errorMargin,* *coordinateSystem* )

or .intersection

or .Difference

or .symmetricDifference

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

The new geometry

An ErrorMargin

indicating the

maximum allowable placement error

in meters.

The second

specified geometry

The first specified geometry

var AllStateCOLLECTION = ee.FeatureCollection('ft:1fRY18cjsHzDgGiJiS2nnpUU3v9JPDc2HNaR7Xk8');

var OneStateCOLLECTION = AllStateCOLLECTION.filter( ee.Filter.eq('Name', 'Arkansas') );

var OneStateELEMENT = OneStateCOLLECTION.first( ); var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

var AllRegionCOLLECTION = ee.FeatureCollection('ft:1Ec8IWsP8asxN-ywSqgXWMuBaxI6pPaeh6hC64lA');

var OneRegionCOLLECTION = AllRegionCOLLECTION.filter(ee.Filter().eq('ECO\_NAME', 'Ozark Mountain forests'));

var OneRegionELEMENT = OneRegionCOLLECTION.first( ); var OneRegionFEATURE = ee.Feature( OneRegionELEMENT );

var OneRegionGEOMETRY = OneRegionFEATURE.geometry( );

var UnionGEOMETRY = OneStateGEOMETRY.union( OneRegionGEOMETRY );

var IntersectionGEOMETRY = OneStateGEOMETRY.intersection( OneRegionGEOMETRY );

var SymmetricDifferenceGEOMETRY = OneStateGEOMETRY.symmetricDifference( OneRegionGEOMETRY );

var DifferenceGEOMETRY = OneStateGEOMETRY.difference( OneRegionGEOMETRY );

Map.centerObject( OneStateGEOMETRY, 6 );

Map.addLayer( OneStateGEOMETRY, {color:'ffff00'}, 'State' );

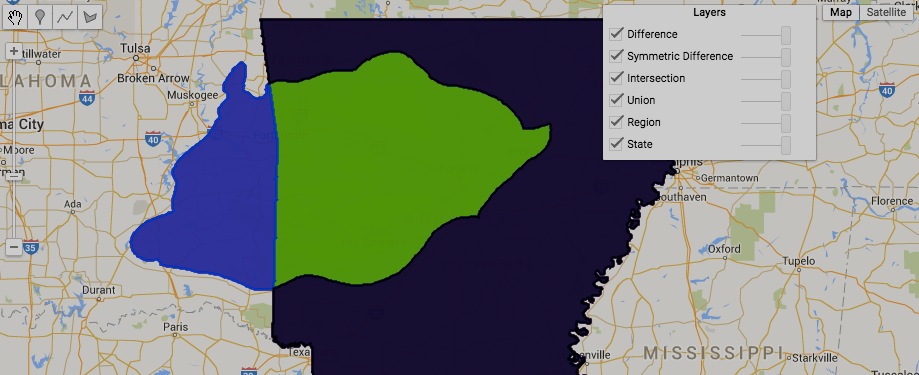
Map.addLayer( OneRegionGEOMETRY, {color:'00ffff'}, 'Region' );

Map.addLayer( UnionGEOMETRY, {color:'ff0000'}, 'Union' );

Map.addLayer( IntersectionGEOMETRY, {color:'00ff00'}, 'Intersection' );

Map.addLayer( SymmetricDifferenceGEOMETRY, {color:'0000ff'}, 'Symmetric Difference' );

Map.addLayer( DifferenceGEOMETRY, {color:'000000'}, 'Difference' );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**REPRODUCING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **LISTS**

geometry.geometries creates a list of the component geometries within a specified geometry.

newList = oldGeometry.geometries ( )

The specified geometry

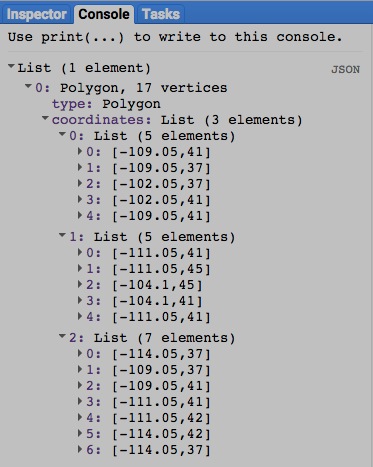
The new list

var TheGEOMETRY = ee.Geometry.Polygon([ [ [-109.05,41],[-109.05,37],[-102.05,37],[-102.05,41] ],

[ [-111.05,41],[-111.05,45],[-104.10,45],[-104.10,41] ],

[ [-114.05,37],[-109.05,37],[-109.05,41],[-111.05,41],[-111.05,42],[-114.05,42.0] ]

] );

var GeometryLIST = TheGEOMETRY.geometries( );

print( GeometryLIST );

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**REPRODUCING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **LISTS**

geometry. coordinates creates a new list containing the vertex coordinates for a specified geometry in a GeoJSON format.

newList = oldGeometry.coordinates ( )

The new list

The specified geometry

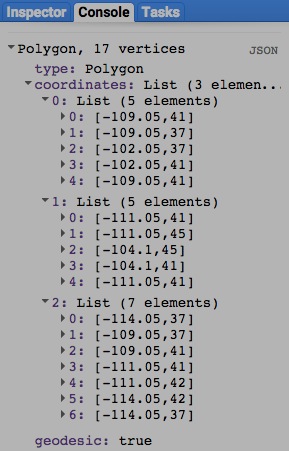
var OldGEOMETRY = ee.Geometry.Polygon(

[ [ [-109.05,41],[-109.05,37],[-102.05,37],[-102.05,41] ],

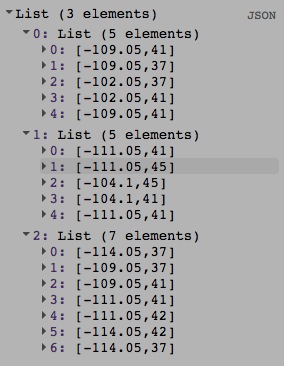
[ [-111.05,41],[-111.05,45],[-104.10,45],[-104.10,41] ],

[ [-114.05,37],[-109.05,37],[-109.05,41],[-111.05,41],[-111.05,42],[-114.05,42.0] ]

] );

var NewLIST = OldGEOMETRY.coordinates( );

print( OldGEOMETRY, NewLIST );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**REPRODUCING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **FEATURES**

ee. Algorithms.Feature and ee.Feature both create a new feature from a specified geometry and (optional) set of metadata properties.

newFeature = ee.AlgorithmsFeature ( oldGeometry, *properties* )

or .ee.Feature(oldGeometry, *properties* )

The new feature

The specified geometry, given as an EE geometry object or as a GeoJSON object representing either a geometry or a feature.

A dictionary of properties to be ascribed to the new feature. Default: { }

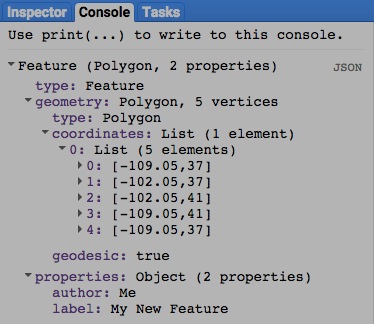
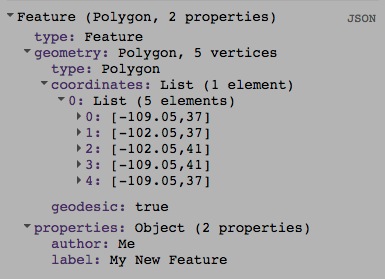
var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05,41],[-109.05,37],[-102.05,37],[-102.05,41] ] ); // Colorado

var ThatFEATURE = ee.Algorithms.Feature( TheGEOMETRY, {label:'My New Feature', author:'Me'} );

var ThisFEATURE = ee.Feature( TheGEOMETRY, {label:'My New Feature', author:'Me'} );

print( ThisFEATURE.getInfo( ) );

print( ThatFEATURE.getInfo( ) );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**REPRODUCING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **FEATURE COLLECTIONS**

ee.FeatureCollection creates a new feature collection from a specified geometry or list of geometries.

newFeatureCollection = ee.FeatureCollection ( oldGeometry )

The new feature

The specified geometry or list of geometries

var ColoGEOMETRY = ee.Geometry.Polygon([[-109.05,41],[-109.05,37],[-102.05,37],[-102.05,41] ] );

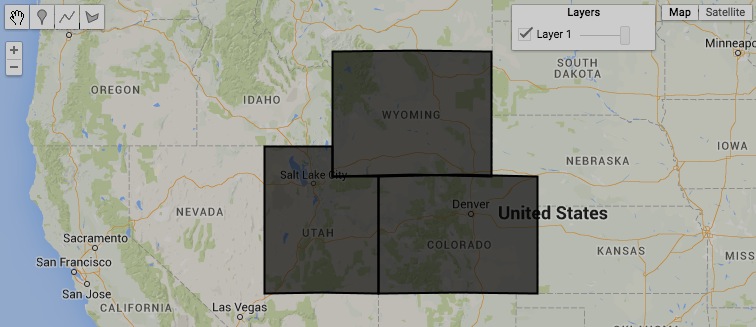
var WyomGEOMETRY = ee.Geometry.Polygon([[-111.05,41],[-111.05,45],[-104.10,45],[-104.10,41] ] );

var UtahGEOMETRY = ee.Geometry.Polygon([[-114.05,37],[-109.05,37],[-109.05,41],[-111.05,41],[-111.05,42],[-114.05,42] ] );

var TheFEATURES = ee.FeatureCollection( [ColoGEOMETRY, WyomGEOMETRY, UtahGEOMETRY] );

Map.setCenter( -109.05, 41, 5 );

Map.addLayer( TheFEATURES );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**QUERYING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) FOR **PROJECTIONS**

geometry.projection creates a new projection matching that of a specified geometry.

newProjection = oldGeometry.projection ( )

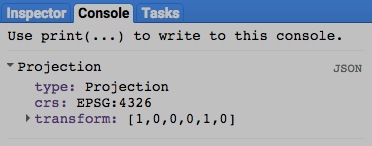
The new projection

The specified geometry

var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

var ThePROJECTION = TheGEOMETRY.projection ( );

print( ThePROJECTION );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**COMPARING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Algorithms.IsEqual creates a new Boolean set to True (only) if the first of two specified geometries is identical to the second in both structure and content.

newBoolean = ee.Algorithms.IsEqual ( 1stGeometry, 2ndGeometry )

The first specified geometry

The second specified geometry

The new Boolean

var FirstGEOMETRY = ee.Geometry.Point( 31.134204, 29.979241 );

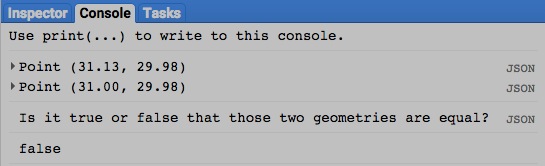
var SecondGEOMETRY = ee.Geometry.Point( 31.000000, 29.979241 );

var TrueOrFalse = ee.Algorithms.IsEqual( FirstGEOMETRY, SecondGEOMETRY );

print( FirstGEOMETRY, SecondGEOMETRY );

print( 'Is it true or false that those two geometries are equal?' );

print( TrueOrFalse );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**MEASURING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.length creates a new floating-point number indicating the length of (all parts of) a specified linestring or linearring feature, measured in the units of

a specified coordinate system (or meters if no coordinate system is specified).

newNumber = oldFeature.length ( *errorMargin,* *coordinateSystem* )

The new

number

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum

allowable reprojection error in meters

The specified

feature

var UpperGEOMETRY = ee.Geometry.LineString( 31.1330,29.9802, 31.1353,29.9802, 31.1353,29.9782, 31.1330,29.9782 );

var LowerGEOMETRY = ee.Geometry.LinearRing( 31.1319,29.9770, 31.1296,29.9770, 31.1296,29.9750, 31.1319,29.9750 );

var UpperNUMBER = UpperGEOMETRY.length( );

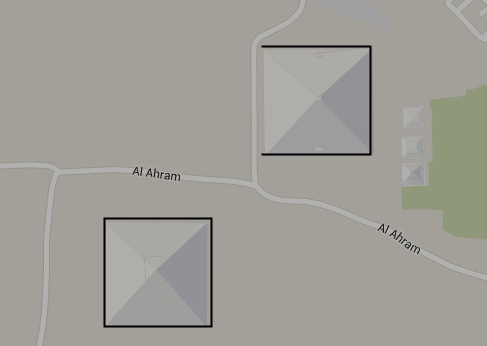
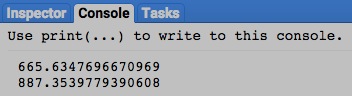
var LowerNUMBER = LowerGEOMETRY.length( );

Map.setCenter( 31.1342, 29.9792, 15 );

Map.addLayer( UpperGEOMETRY );

Map.addLayer( LowerGEOMETRY );

print( UpperNUMBER, LowerNUMBER );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**MEASURING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.perimeter creates a new floating-point number indicating the perimeter of (all parts of) a specified polygonal geometry, measured in the units of

a specified coordinate system (or meters if no coordinate system is specified). Point and line geometries result in a perimeter of 0.

newNumber = oldGeometry.area ( *errorMargin,* *coordinateSystem* )

The new

number

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum

allowable reprojection error in meters

The specified

geometry

var UpperPOLYGON = ee.Geometry( { "type": "Polygon",

"coordinates": [ [ [31.1330,29.9782],[31.1353,29.9782],[31.1353,29.9802],[31.1330,29.9802] ],

[ [31.1342,29.9792],[31.1352,29.9801],[31.1331,29.9801] ] ] } );

var LowerPOLYGON = ee.Geometry( { "type": "Polygon",

"coordinates": [ [ [31.1319,29.9770],[31.1296,29.9770],[31.1296,29.9750],[31.1319,29.9750] ] ] } );

var UpperNUMBER = UpperPOLYGON.perimeter( );

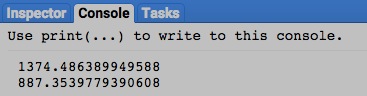
var LowerNUMBER = LowerPOLYGON.perimeter( );

Map.setCenter( 31.1342, 29.9792, 15 );

Map.addLayer( UpperPOLYGON );

Map.addLayer( LowerPOLYGON );

print( UpperNUMBER, LowerNUMBER );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**MEASURING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.area creates a new floating-point number indicating the total area of (all parts of) a specified polygonal geometry, measured in the squared units of

Polygons with clockwise vertices are regarded as holes in a global polygon for purposes of area measurement

a specified coordinate system (or square meters if no coordinate system is specified). Point and line geometries result in an area of 0.

newNumber = oldGeometry.area ( *errorMargin,* *coordinateSystem* )

The new

number

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum

allowable reprojection error in meters

The specified

geometry

var UpperPOLYGON = ee.Geometry( { "type": "Polygon",

"coordinates": [ [ [31.1330,29.9782],[31.1353,29.9782],[31.1353,29.9802],[31.1330,29.9802] ],

[ [31.1342,29.9792],[31.1352,29.9801],[31.1331,29.9801] ] ] } );

var LowerPOLYGON = ee.Geometry( { "type": "Polygon",

"coordinates": [ [ [31.1319,29.9770],[31.1296,29.9770],[31.1296,29.9750],[31.1319,29.9750] ] ] } );

var ClockwisePOLYGON = ee.Geometry(

{ "type": "Polygon", "coordinates": [ [ [31.1330,29.9802],[31.1353,29.9802],[31.1353,29.9782],[31.1330,29.9782] ] ] } );

var UpperNUMBER = UpperPOLYGON.area( );

var LowerNUMBER = LowerPOLYGON.area( );

var ClockwiseNUMBER = ClockwisePOLYGON.area( );

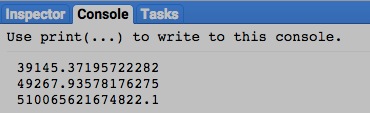
Map.setCenter( 31.1342, 29.9792, 15 );

Map.addLayer( UpperPOLYGON, null, 'Upper Polygon' );

Map.addLayer( LowerPOLYGON, null, 'Lower Polygon' );

Map.addLayer( ClockwisePOLYGON, null, 'Clockwise Polygon' );

print( UpperNUMBER, LowerNUMBER, ClockwiseNUMBER );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**MEASURING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.distance creates a new floating-point number indicating the distance in meters between the closest parts of two specified geometries.

newNumber = 1stGeometry.distance ( 2ndGeometry, *errorMargin,* *coordinateSystem* )

The first specified geometry

The second specified geometry

The new number

A specified coordinate system, given as an EPSG code ( as described [here](http://spatialreference.org) ) or as a WKT string ( as described [here](http://en.wikibooks.org/wiki/Geospatial_Data_in_SQL_Server/WKT) ). Default: 'EPSG4326' (WGS84)

An ErrorMargin object indicating the maximum allowable reprojection error in meters

var UpperPOLYGON = ee.Geometry( { "type": "Polygon",

"coordinates": [ [ [31.1330,29.9782],[31.1353,29.9782],[31.1353,29.9802],[31.1330,29.9802] ],

[ [31.1342,29.9792],[31.1352,29.9801],[31.1331,29.9801] ] ] } );

var LowerPOLYGON = ee.Geometry( { "type": "Polygon",

"coordinates": [ [ [31.1319,29.9770],[31.1296,29.9770],[31.1296,29.9750],[31.1319,29.9750] ] ] } );

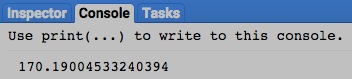
var TheNUMBER = UpperPOLYGON.distance( LowerPOLYGON );

Map.setCenter( 31.1342, 29.9792, 15 );

Map.addLayer( UpperPOLYGON );

Map.addLayer( LowerPOLYGON );

print( TheNUMBER



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DESCRIBING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.type creates a new string indicating the GeoJSON type of a specified geometry.

newString = oldGeometry.type ( )

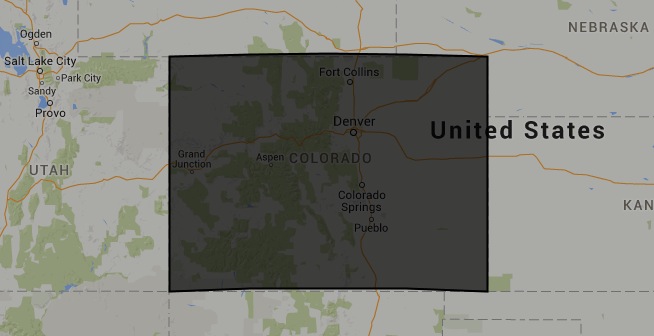
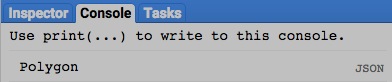
The specified geometry

The new string

var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

var TheSTRING = TheGEOMETRY.type( );

print( TheSTRING );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DESCRIBING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.edgesAreGeodesics creates a new Boolean set to True (only) if the lines in a specified geometry are spheroidal rather than planar.

newBoolean = oldGeometry.edgesAreGeodesics ( )

The specified geometry

The new Boolean

var OldGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

var OldBOOLEAN = OldGEOMETRY.edgesAreGeodesics ( );

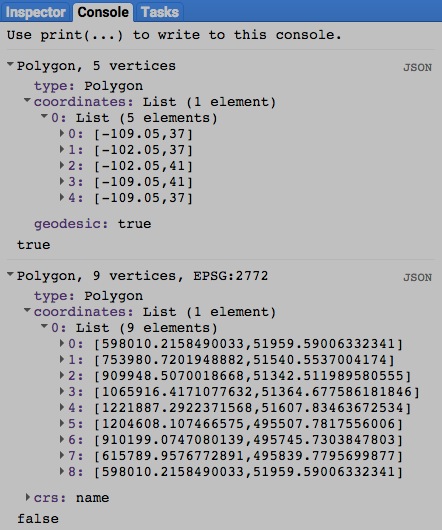
var NewGEOMETRY = OldGEOMETRY.transform( 'EPSG:2772', 100 );

var NewBOOLEAN = NewGEOMETRY.edgesAreGeodesics ( );

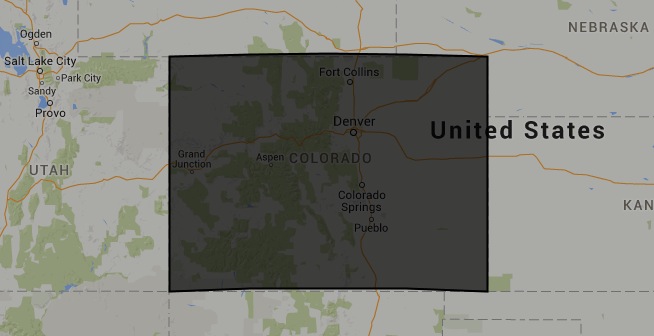
Map.centerObject( OldGEOMETRY );

Map.addLayer( OldGEOMETRY );

Map.addLayer( NewGEOMETRY );

print( OldGEOMETRY, OldBOOLEAN );

print( NewGEOMETRY, NewBOOLEAN );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DESCRIBING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.isUnbounded creates a new Boolean set to True (only) a specified geometry is unbounded.

newBoolean = oldGeometry.isUnbounded ( )

The specified geometry

The new Boolean

var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

var TheBOOLEAN = TheGEOMETRY.isUnbounded ( );

print( TheBOOLEAN );

ADD EXAMPLE AFTER DETERMINING HOW TO GENERATE AN UNBOUNDED GEOMETRY?

[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DOCUMENTING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

ee.Algorithms.Describe , geometry.getInfo , and geometry.toGeoJSON

each creates a JSON-compatible text object

representing a specified geometry.

newObject = ee.Algorithms.Describe ( oldGeometry )

oldGeometry.getInfo( )

and oldGeometry.toGeoJSON( )

The new object

The specified geometry

var OldGEOMETRY = ee.Geometry.Polygon(

[ [ [-109.05,41],[-109.05,37],[-102.05,37],[-102.05,41] ],

[ [-111.05,41],[-111.05,45],[-104.10,45],[-104.10,41] ],

[ [-114.05,37],[-109.05,37],[-109.05,41],[-111.05,41],[-111.05,42],[-114.05,42.0] ]

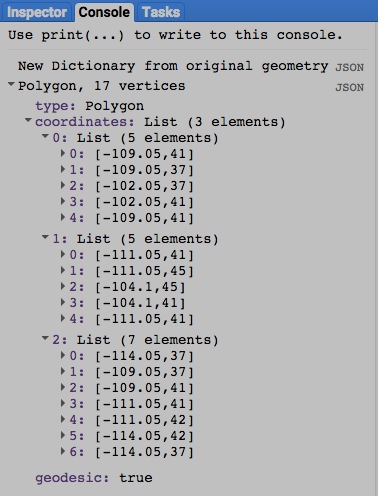
] );

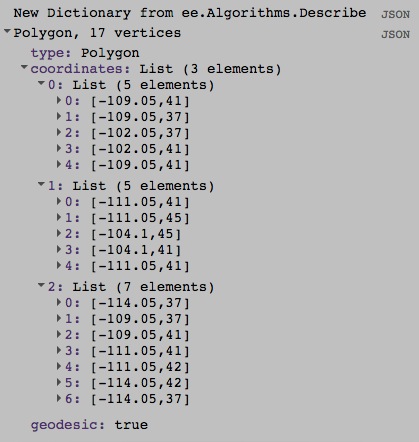
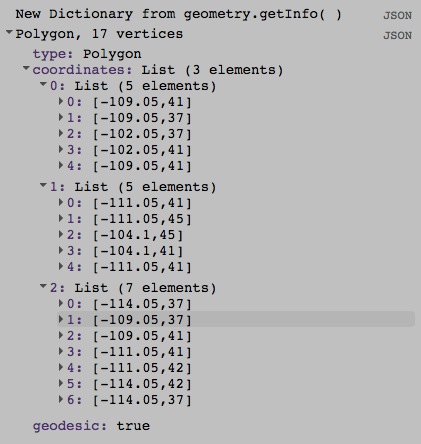
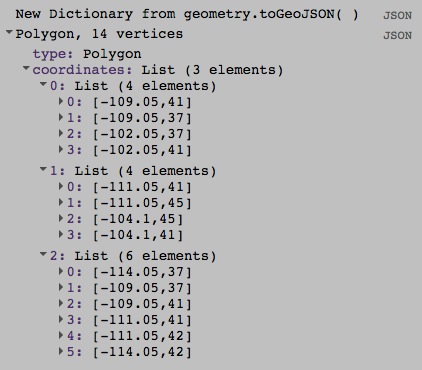
print( 'New Dictionary from original geometry', OldGEOMETRY );

print( 'New Dictionary from ee.Algorithms.Describe', ee.Algorithms.Describe( OldGEOMETRY ) );

print( 'New Dictionary from geometry.getInfo( )', OldGEOMETRY.getInfo( ) );

print( 'New Dictionary from geometry.toGeoJSON( )', OldGEOMETRY. toGeoJSON( ) );





[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**DOCUMENTING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

geometry.toString , . toGeoJSONString , and .serialize all create a new strings presenting

information on a specified geometry

newString = oldGeometry.toString ( )

oldGeometry.toGeoJSONString( )

and oldGeometry.serialize( )

The specified geometry

The new string

var OldGEOMETRY = ee.Geometry.Polygon(

[ [ [-109.05,41],[-109.05,37],[-102.05,37],[-102.05,41] ],

[ [-111.05,41],[-111.05,45],[-104.10,45],[-104.10,41] ],

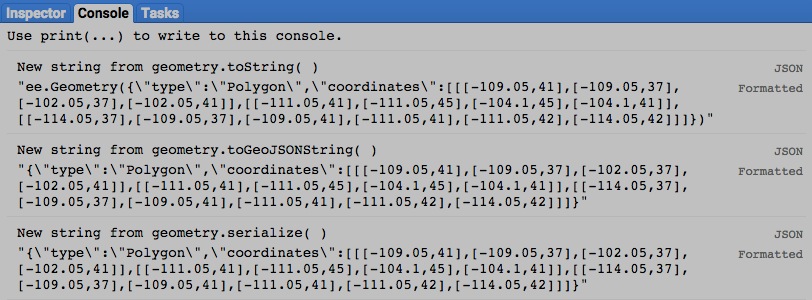
[ [-114.05,37],[-109.05,37],[-109.05,41],[-111.05,41],[-111.05,42],[-114.05,42.0] ]

] );

print( 'New string from geometry.toString( )', OldGEOMETRY.toString( ) );

print( 'New string from geometry.toGeoJSONString( )', OldGEOMETRY.toGeoJSONString( ) );

print( 'New string from geometry.serialize( )', OldGEOMETRY.serialize( ) );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**PRESENTING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) IN **PRINT**

print ( geometry ) and console.log ( geometry ) present JSON-formatted text renditions of a specified geometry in the console.

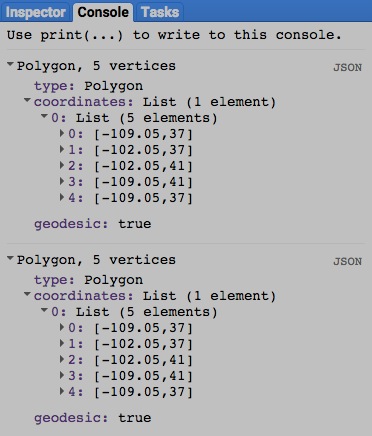
print( oldGeometry ) or console.log( oldGeometry )

The specified geometry

var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

print( TheGEOMETRY );

console.log( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**PRESENTING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) IN **PRINT**

alert ( geometry ) and confirm( geometry ) present JSON-formatted text renditions a specified geometry in a pop-up message box.

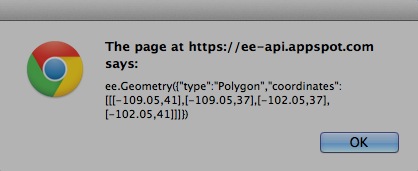
alert( oldGeometry ) or confirm( oldGeometry )

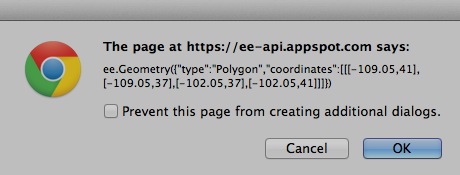
The specified geometry

var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

alert( TheGEOMETRY );

confirm( TheGEOMETRY );



[GOOGLE EARTH ENGINE](EE01%20Earth%20Engine%20(EE).docx) [APPLICATION PROGRAMMING INTERFACE](EE05%20%20%20The%20EE%20API.docx) [CAPABILITIES](EE07%20%20%20%20%20%20API%20Capabilities.docx)

**PRESENTING** [GEOMETRY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) IN **MAPS**

Map.addLayer ( geometry ) adds a specified geometry to the map as a layer.

Map.addLayer ( geometry*,  color, name, visibility* )

The specified geometry

A Boolean set to TRUE (only) if the layer being displayed should initially be toggled on (visible) rather than off (invisible).Visibility can also be adjusted interactively by using the map’s layer listing.

The name for the layer to be displayed, given as a string. This name will appear on the map’s layer list listing.Default: "Layer N," where N is 1, 2, 3, and so on according to the order in which layers are added to the display.

**000000 505050**

**FF0000 800000**

**FFAA00 DDCCAA**

**FFFF00 808000**

**00FF00 008000**

**00FFFF 008080**

**0000FF 000080**

**FF00FF 800080**

**FFFFFF 808080**

The specified color, given as **{ color: X }** where X is a string of three two-character codes

that indicate the amount of redness, greenness, and blueness (in that order) comprising

the color. The characters in each two-character code are digits that represent quantities

ranging from 0 to 15 as follows: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Thus, A (or a) represents 10, B (or b) represents 11, and so on while A0 represents 160,

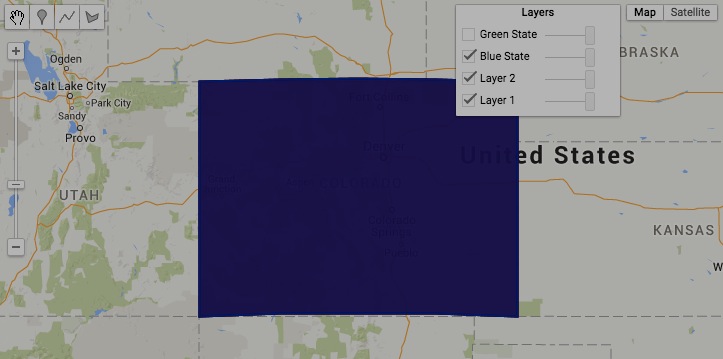
while A1 represents 161, and FF represents 255. To the right are some examples;

each six-character code appears in the color it represents as described [here](http://www.w3schools.com/tags/ref_colormixer.asp).

Default: ‘000000’.

var TheGEOMETRY = ee.Geometry.Polygon( [ [-109.05, 41], [-109.05, 37], [-102.05, 37], [-102.05, 41] ] ); // Colorado

Map.setCenter( -105.4907, 39.1301, 6 );

Map.addLayer( TheGEOMETRY );

Map.addLayer( TheGEOMETRY, {color:'990000'} );

Map.addLayer( TheGEOMETRY, {color:'000099'}, 'Blue State' );

Map.addLayer( TheGEOMETRY, {color:'009900'}, 'Green State', false );