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

A feature is is an EE variable object that represents a geometry referenced to geographic space and associated with geographic attributes. Features 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** FEATURES [ee.Feature](#Feature) [ee.Algorithms.Feature](#Feature) [Onscreen Drawing](#Drawing)

**EDITING** FEATURES

BY **SELECTING** PROPERTIES [feature.select](#select)

BY **REPROJECTING** [feature.transform](#transform) [ee.Algorithms.ProjectionTransform](#ProjectionTrans)

BY **RESETTING** VALUES [feature.set](#set) [feature.setMulti](#setMulti)

**TRANSFORMING** FEATURES [feature.centroid](#centroid) [feature.simplify](#simplify)

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

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

[feature.symmetricDifference](#union_)

**REPRODUCING** FEATURES

AS **GEOMETRIES** [feature.geometry](#geometry)

AS **FEATURE COLLECTIONS** [ee.Algorithms.Collection](#Collection) [ee.FeatureCollection(featureList)](#FeatureCollection)

AS **GOOGLEMAP OVERLAYS** [feature.getMap](#getMap)

**QUERYING** FEATURES

FOR **VALUES** [feature.get](#get) [feature.toArray](#toArray)

**COMPARING** FEATURES [ee.Algorithms.IsEqual(feature)](#IsEqual)

**MEASURING** FEATURES [feature.length](#length) [feature.perimeter](#perimeter) [feature.area](#area)

**DOCUMENTING** FEATURES [feature.getInfo](#Describe_) [ee.Algorithms.Describe(feature)](#Describe_)

[feature.toString](#toString_) [feature.serialize](#toString_)

**PRESENTING** FEATURES

IN **PRINT** [print(feature)](#print_console) [console.log(feature)](#print_console)

[alert(feature)](#alert_confirm) [confirm(feature)](#alert_confirm)

IN **MAPS** [Map.addLayer(feature)](#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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **ONSCREEN DRAWING**

DESCRIBE THE ONSCREEN DRAWING TOOLS, FOCUSING ONLY ON FEATURES

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

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

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

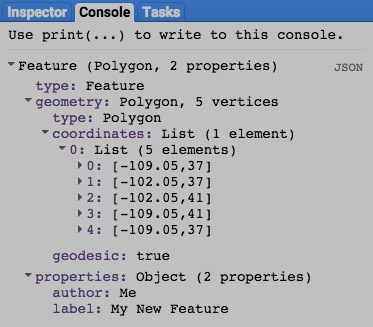
Map.centerObject( TheGEOMETRY );

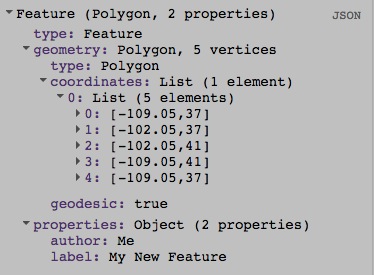
Map.addLayer( ThisFEATURE );

Map.addLayer( ThatFEATURE );

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)

**EDITING** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **SELECTING** PROPERTIES

feature.select creates a new feature by replicating a specified feature but only with a specified set of properties.

newFeature = oldFeature.select ( propertyList*, propertyNameList, retainGeometry?*  )

A Boolean set to rue (only) if the primary

geometry key is be regarded as selected

even when unselected. Default: true

New names for the selected

properties, given as a list

corresponding to **propertyList**

The specified properties,

given as a list of property

names

The specified feature

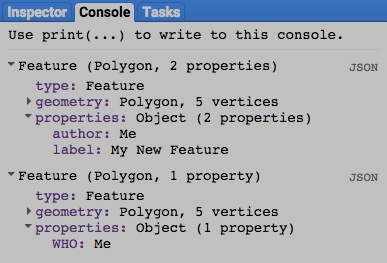
The new feature

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

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

var NewFEATURE = OldFEATURE.select( ['author'], ['WHO'] );

print( OldFEATURE, NewFEATURE );



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

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

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

newFeature = oldFeature.transform ( *coordinateSystem*, *errorMargin* )

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

feature

The new

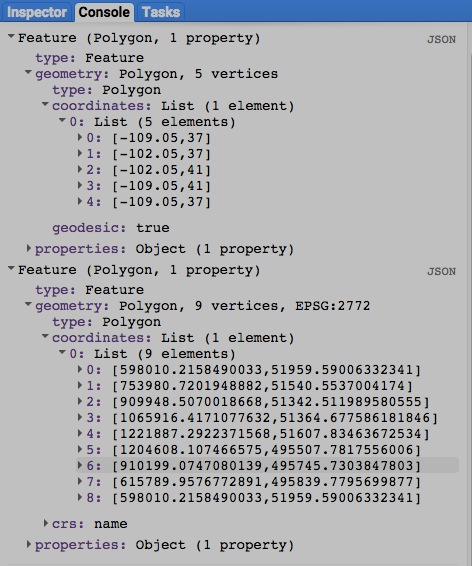
feature

A ErrorMargin object indicating the maximum

allowable reprojection error in meters

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

var OldFEATURE = ee.Feature( OldGEOMETRY, {name:'Colorado'} );

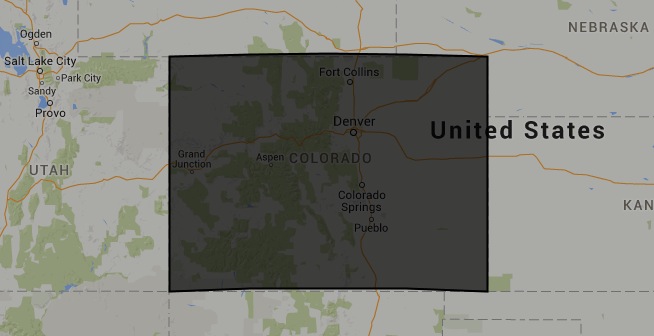
var NewFEATURE = OldFEATURE.transform( 'EPSG:2772', 100 );

print( OldFEATURE, NewFEATURE );

Map.centerObject( OldFEATURE );

Map.addLayer( OldFEATURE );

Map.addLayer( NewFEATURE );



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

ee.Algorithms.ProjectionTransform creates a new feature by reprojecting a specified feature to a specified coordinate system, with lines

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

newFeature = ee.Algorithms.ProjectionTransform ( oldFeature, *coordinateSystem*, *errorMargin* )

The new feature

The specified feature

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

A ErrorMargin object indicating the maximum

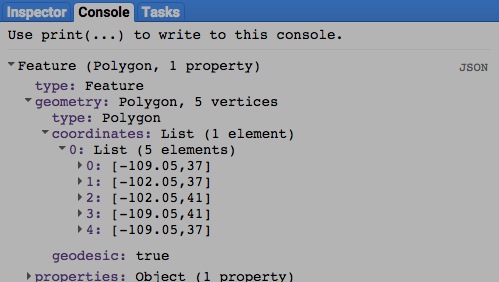
allowable reprojection error in meters

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

var OldFEATURE = ee.Feature( OldGEOMETRY, {name:'Colorado'} );

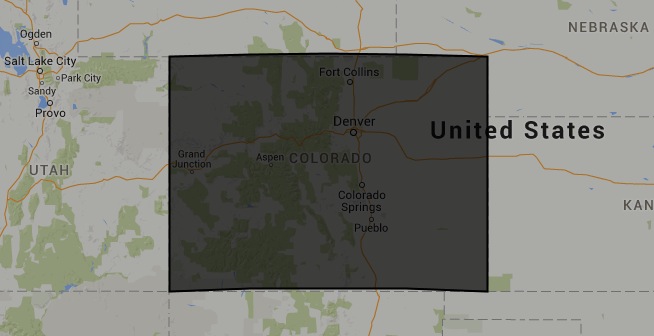
var NewFEATURE = ee.Algorithms.ProjectionTransform( OldFEATURE, 'EPSG:2772', 100 );

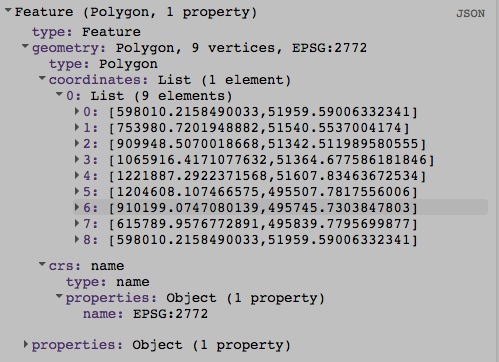
print( OldFEATURE, NewFEATURE );

Map.centerObject ( OldFEATURE );

Map.addLayer( OldFEATURE );

Map.addLayer( NewFEATURE );





[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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **RESETTING** VALUES

feature.set creates new feature by replicating a specified feature after setting or resetting one or more specified properties to specified values.

newFeature = oldFeature.set ( pairsOfPropertiesAndValues )

The specified feature

The new feature

The specified properties and new values, given as a comma-separated sequence

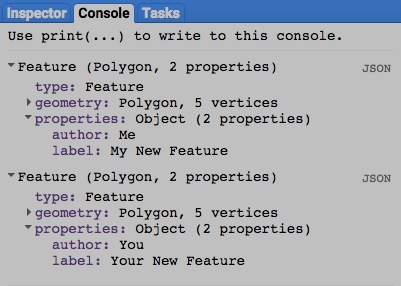
(or a dictionary) of property name strings, each immediately followed by its new value.

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

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

var NewFEATURE = OldFEATURE.set( 'author','You', 'label','Your New Feature' );

print( OldFEATURE.getInfo( ), NewFEATURE.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)

**EDITING** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **RESETTING** VALUES

feature.setMulti creates new feature by replicating a specified feature after setting or resetting one or more specified properties to specified values.

newFeature = oldFeature.set ( dictionaryOfPropertiesAndValues )

The new feature

The specified properties and new values, given as a

dictionary of property name strings and new values.

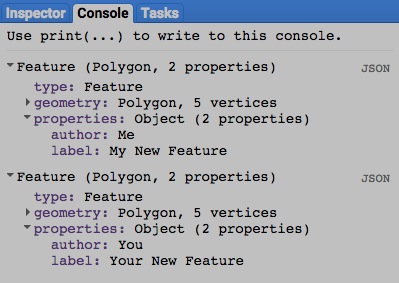
The specified feature

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

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

var NewFEATURE = OldFEATURE.setMulti( {'author':'You','label':'Your New Feature'} );

print( OldFEATURE.getInfo( ), NewFEATURE.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)

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

feature.centroid creates a new point feature at the centroid of a specified feature.

newFeature = oldFeature.centroid ( *errorMargin,* *coordinateSystem* )

The new

feature

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

feature

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 NewFEATURE = OneStateFEATURE.centroid( );

Map.centerObject( OneStateFEATURE, 6 );

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

Map.addLayer( NewFEATURE, {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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

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

newFeature = oldFeature.simplify ( errorMargin*,* *coordinateSystem* )

An ErrorMargin object indicating the maximum

allowable placement error in meters. This will

maintained regardless of zoom level.

The new

feature

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 **OldFeature**

The specified

feature

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

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

var OneStateELEMENT = OneStateCOLLECTION.first( );

var OneStateFEATURE = ee.Feature( OneStateELEMENT );

Map.centerObject( OneStateFEATURE, 5 ); Map.addLayer( OneStateFEATURE, {color:'331188'} );

var NewFEATURE = OneStateFEATURE.simplify( 50000 ); Map.addLayer( NewFEATURE, {color:'ffffff', opacity:0.1} );



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

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

newFeature = oldFeature.bounds ( *errorMargin,* *coordinateSystem* )

The new

feature

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

feature

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 NewFEATURE = OneStateFEATURE.bounds( );

Map.centerObject( OneStateFEATURE, 5 );

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

Map.addLayer( NewFEATURE, {color:'ffffff', opacity:0.1} );



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

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

newFeature = oldFeature.convexHull ( *errorMargin,* *coordinateSystem* )

The new

feature

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

feature

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 NewFEATURE = OneStateFEATURE.convexHull( );

Map.centerObject( OneStateFEATURE, 5 );

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

Map.addLayer( NewFEATURE, {color:'ffffff', opacity:0.1} );



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

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

newFeature = oldFeature.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 feature 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 feature

The specified feature

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 NewFEATURE = OneStateFEATURE.centroid( ).buffer( 100000 );

Map.centerObject( OneStateFEATURE, 5 );

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

Map.addLayer( NewFEATURE, {color:'ffffff', opacity:0.1} );

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

feature.union , .intersection , .symmetricDifference , and Difference

create a new polygonal feature encompassing locations that are determined by two specified polygonal features such that

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

newFeature =1stOldFeature.union ( 2ndOldFeature, *errorMargin,* *coordinateSystem* )

or .intersection

or .symmetricDifference

or .Difference

The new feature

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

indicating the

maximum allowable placement error

in meters.

The second

specified specified

The first specified feature

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 UnionGEOMETRY = OneStateFEATURE.union( OneRegionFEATURE );

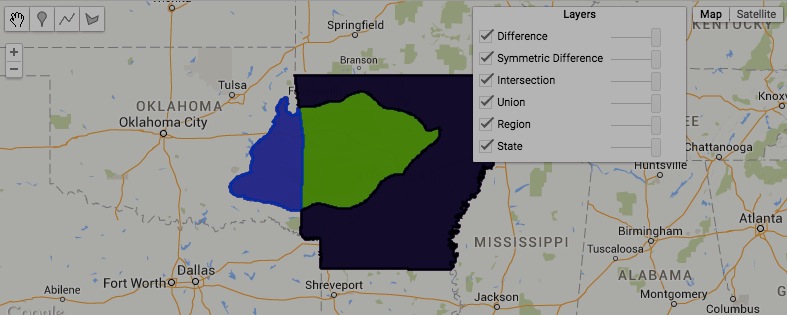
var IntersectionGEOMETRY = OneStateFEATURE.intersection( OneRegionFEATURE );

var SymmetricDifferenceGEOMETRY = OneStateFEATURE.symmetricDifference( OneRegionFEATURE );

var DifferenceGEOMETRY = OneStateFEATURE.difference( OneRegionFEATURE );

Map.centerObject( OneStateGEOMETRY, 6 );

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

Map.addLayer( OneRegionFEATURE, {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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **GEOMETRIES**

feature.geometry creates a new geometry by reproducing that of a specified feature.

*newGeometry = oldFeature.geometry ( errorMargin, coordinateSystem, geodesic )*

An error margin indicating the spatial precision to be maintained

A Boolean set to True (only) if the new geometry is

to have geodesic (rather than straight-line) edges

in its projected coordinate system. If a **coordSystem**

is specified, this option is ignored. Default: edges

will be handled as they are in the specified feature.

The new geometry

The specified feature

The coordinate system to be employed. Default: lon/lat

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

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

var OneStateELEMENT = OneStateCOLLECTION.first( );

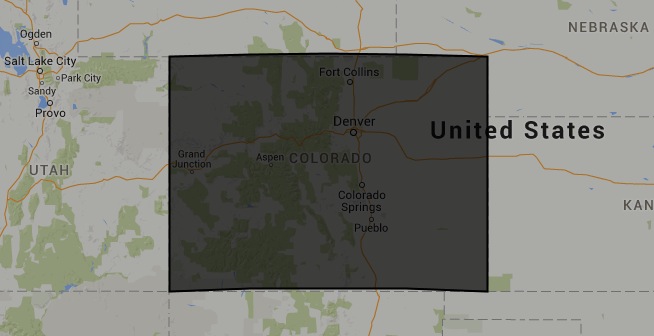
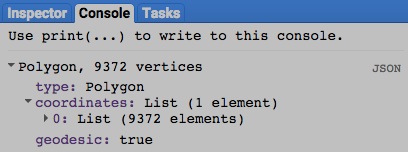
var OneStateFEATURE = ee.Feature( OneStateELEMENT );

var OneStateGEOMETRY = OneStateFEATURE.geometry();

Map.centerObject( OneStateGEOMETRY );

Map.addLayer( OneStateGEOMETRY );

print( OneStateGEOMETRY );



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

ee.Algorithms.Collection creates a new feature collection containing features drawn from a specified list.

newFeatureCollection = ee.Algorithms.Collection ( oldFeatures )

The new feature collection

The specified list of features

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

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

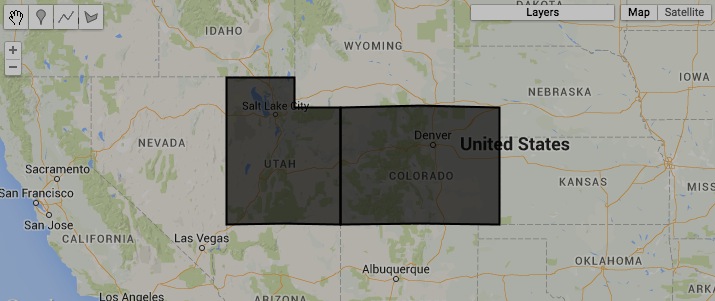
var UtahFEATURE = ee.Feature( UtahGEOMETRY, {name:'Utah'} );

var ColoFEATURE = ee.Feature( ColoGEOMETRY, {name:'Colorado'} );

var NewFEATURES = ee.Algorithms.Collection( [UtahFEATURE,ColoFEATURE] );

Map.centerObject( NewFEATURES, 5 );

Map.addLayer( NewFEATURES );

****

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

ee.FeatureCollection creates a new feature collection containing a specified set of one or more features.

newFeatureCollection = ee.FeatureCollection ( oldFeatureList, *geometryColumn* )

The name of the feature geometry column, given as a string. Default “geometry.”

The specified feature or list of features

The new feature collection

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

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

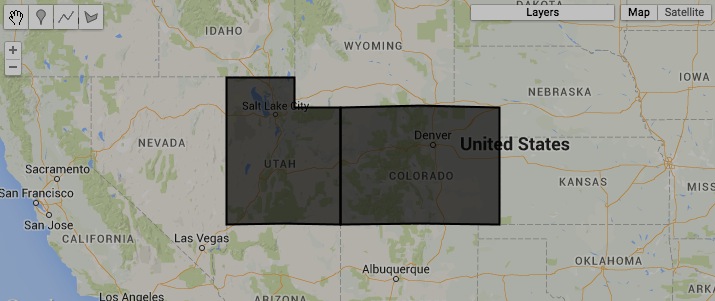
var UtahFEATURE = ee.Feature( UtahGEOMETRY, {name:'Utah'} );

var ColoFEATURE = ee.Feature( ColoGEOMETRY, {name:'Colorado'} );

var NewFEATURES = ee.FeatureCollection( [UtahFEATURE,ColoFEATURE], 'geometry' );

Map.centerObject( NewFEATURES, 5 );

Map.addLayer( NewFEATURES );

****

[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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **GOOGLE MAP OVERLAYS**

feature.getMap creates a new object containing a specified feature, Map ID, and token from which to create a Google Maps overlay.

newObject = oldFeature.getMap ( color)

**000000 505050**

**FF0000 800000**

**FFAA00 DDCCAA**

**FFFF00 808000**

**00FF00 008000**

**00FFFF 008080**

**0000FF 000080**

**FF00FF 800080**

**FFFFFF 808080**

The specified feature

The specified color, given as the name of a color property or 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 ten, B (or b) represents eleven, and so on while A0 represents 160, while A 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’.

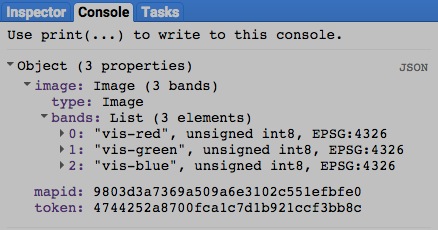
The new object

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

var TheFEATURE = ee.Feature( TheGEOMETRY, {name:'Colorado', fill:1} );

var ThePACKAGE = TheFEATURE.getMap( {color:'000099'} );

print( ThePACKAGE );



[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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) FOR **VALUES**

feature.get creates a new collection element replicating the value associated with a specified property of a specified feature.

newElement = oldFeature.get ( property )

The new element

The specified feature

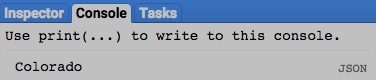
The specified property, given as a string indicating its name

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

var OldFEATURE = ee.Feature( OldGEOMETRY, {name: 'Colorado', fill: 1} );

var NewELEMENT = OldFEATURE.get( 'name' );

print(NewELEMENT );



[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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) FOR **VALUES**

feature.toArray creates a new one-dimensional array containing the values of specified numerical properties associated with a specified feature.

newArray = oldFeature.toArray ( properties )

The specified properties, given as a list of strings indicating property names

The new array

The specified feature

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

var OldFEATURE = ee.Feature( OldGEOMETRY, {name:'Colorado', area:104094, pop:5360000, income:57685} );

var NewARRAY = OldFEATURE.toArray( ['area','pop','income'] );

print( NewARRAY );

[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** [FEATURE](#_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 features is identical to the second in both structure and content.

newBoolean = ee.Algorithms.IsEqual ( 1stFeature, 2ndFeature )

The first specified feature

The second specified feature

The new Boolean

var The1stFEATURE = ee.Feature( ee.Geometry.Point( 31.134204, 29.979241 ) );

var The2ndFEATURE = ee.Feature( ee.Geometry.Point( 31.000000, 29.979241 ) );

var AreTheyIDENTICAL = ee.Algorithms.IsEqual( The1stFEATURE, The2ndFEATURE );

print( AreTheyIDENTICAL );

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

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

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

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

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

var UpperFEATURE = ee.Feature( UpperLINESTRING );

var LowerFEATURE = ee.Feature( LowerLINEARRING );

Map.setCenter( 31.1342, 29.9792, 15 ); Map.addLayer( UpperFEATURE ); Map.addLayer( LowerFEATURE );

var UpperNUMBER = UpperFEATURE.length( ); var LowerNUMBER = LowerFEATURE.length( );

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

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

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

newNumber = oldFeature.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

feature

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 UpperFEATURE = ee.Feature( UpperPOLYGON );

var LowerFEATURE = ee.Feature( LowerPOLYGON );

var ClockwiseFEATURE = ee.Feature( LowerPOLYGON );

Map.setCenter( 31.1342, 29.9792, 15 ); Map.addLayer( UpperFEATURE ); Map.addLayer( LowerFEATURE );

var UpperNUMBER = UpperFEATURE.perimeter( ); var LowerNUMBER = LowerFEATURE.perimeter( );

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

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

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

newNumber = oldFeature.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

feature

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 UpperFEATURE = ee.Feature( UpperPOLYGON );

var LowerFEATURE = ee.Feature( LowerPOLYGON );

var ClockwiseFEATURE = ee.Feature( LowerPOLYGON );

Map.setCenter( 31.1342, 29.9792, 15 ); Map.addLayer( UpperFEATURE ); Map.addLayer( LowerFEATURE );

var UpperNUMBER = UpperFEATURE.area( ); var LowerFEATURE = LowerPOLYGON.area( ); var ClockwiseNUMBER = ClockwiseFEATURE.area( );

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)

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

ee.Algorithms.Describe and feature.getInfo

each creates a JSON-compatible text object representing a specified feature.

newObject = ee.Algorithms.Describe( oldFeature )

and oldFeature.getInfo( )

The specified feature

The new object

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

var OldFEATURE = ee.Feature( OldGEOMETRY, {name:'Colorado', area:104094, pop:5360000, income:57685} );

print( 'New string from the original feature', OldFEATURE );

print( 'New string from ee.Algorithms.Describe', ee.Algorithms.Describe( OldFEATURE ) );

print( 'New string from feature.getInfo( )', OldFEATURE.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)

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

feature.toString and .serialize each creates a new string presenting information on a specified feature.

newString = oldFeature.toString ( )

and oldFeature.serialize( )

The specified feature

The new string

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

var OldFEATURE = ee.Feature( OldGEOMETRY, {name:'Colorado', area:104094, pop:5360000, income:57685} );

print( 'New string from the original feature', OldFEATURE );

print( 'New string from feature.toString( )', OldFEATURE.toString( ) );

print( 'New string from feature.serialize( )', OldFEATURE.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** [FEATURE](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) IN **PRINT**

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

print( oldFeature ) or console.log( oldFeature )

The specified feature

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

var TheFEATURE = ee.Feature( TheGEOMETRY, {name:'Colorado', area:104094, pop:5360000, income:57685} );

print( TheFEATURE );

console.log( TheFEATURE );

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

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

alert( oldFeature ) or confirm( oldFeature )

The specified feature

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

var TheFEATURE = ee.Feature( TheGEOMETRY, {name:'Colorado', area:104094, pop:5360000, income:57685} );

alert( TheFEATURE );

confirm( TheFEATURE );

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

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

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

The specified feature

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

var TheFEATURE = ee.Feature( TheGEOMETRY, {name:'Colorado', fill:1} );

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

Map.addLayer( TheFEATURE );

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

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

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