[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 **ALGORITHM** [[[[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

An algorithm is an Earth Engine parameter object that represents a user-authored procedure. Algorithms 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** ALGORITHMS [function](#function)

**INVOKING** ALGORITHMS [<algorithmName>](#algorithmName) [algorithm.call](#call) [algorithm.apply](#apply)

[featureCollection.map](EE21%20%20%20%20%20%20%20%20%20%20%20%20Feature%20Collections.docx#map) [imageCollection.map](EE23%20%20%20%20%20%20%20%20%20%20%20%20Image%20Collections.docx#map)

[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 **ALGORITHM** [PARAMETERS](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

function creates a new algorithm of a specified name that will apply a specified sequence of statements to a specified set of input arguments

when the algorithm is later called by invoking **ee.call**, **ee.apply**, **map** or whatever arbitrary **algorithmName** has been specified.

.

function algorithmName( inputArguments ) { statements }

The specified

algorithm name

The specified sequence of JavaScript statements, ending with a **return** statement

that indicates the value to be returned when this algorithm is later called.

The specified input arguments,

given as a comma-separated list

function TheALGORITHM( InitialNUMBER, FinalNUMBER ) // Calculate % Increase in Value

{ var DifferenceNUMBER = FinalNUMBER.subtract( InitialNUMBER );

var PctIncreaseNUMBER = DifferenceNUMBER.divide( InitialNUMBER ).multiply(100);

return PctIncreaseNUMBER;

}

var FirstNUMBER = ee.Number( 12 );

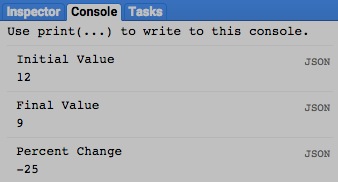
var SecondNUMBER = ee.Number( 9 );

var NewNUMBER = TheALGORITHM ( FirstNUMBER, SecondNUMBER );

print( 'Initial Value', FirstNUMBER );

print( 'Final Value', SecondNUMBER );

print( 'Percent Change', NewNUMBER );



[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 **ALGORITHM** [[[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

<algorithmName> creates a new object by subjecting a specified set of input arguments to a specified algorithm.

This can be a standard Earth Engine algorithm or a user-defined algorithm created with a prior **function** statement.

newObject = algorithmName( inputArguments )

The specified algorithm,

given by its name.

The specified input arguments, given as a comma-separated sequence of input values

The new object

function goGet( RequestedIMAGE )

{ var ReturnedIMAGE = ee.Image( RequestedIMAGE );

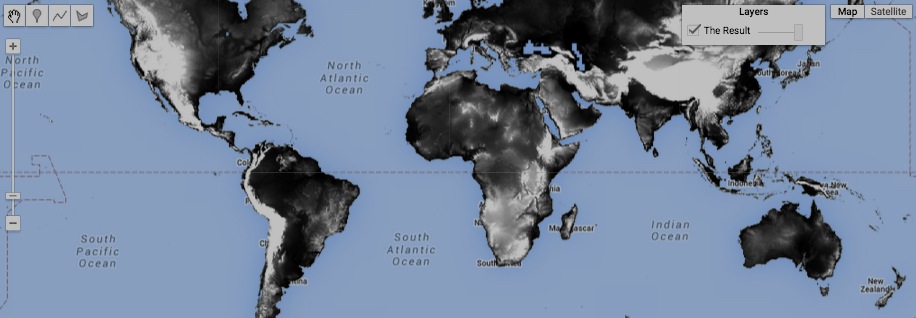
return ReturnedIMAGE;

}

var TheRESULT = goGet("CGIAR/SRTM90\_V4");

Map.centerObject( TheRESULT, 2 );

Map.addLayer( TheRESULT, {min:0, max:1500},'The Result' );



[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 **ALGORITHM** [[[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

algorithm.call creates a new object by subjecting a specified set of input arguments to a specified algorithm.

This can be a standard Earth Engine algorithm or a user-defined algorithm created with a prior **function** statement.

newObject = ee.call( algorithmName, inputArguments )

The new object

The specified input arguments, given as a dictionary of input values

The specified algorithm,

given by its name.

function goGet( RequestedIMAGE )

{ var ReturnedIMAGE = ee.Image( RequestedIMAGE );

return ReturnedIMAGE;

}

var TheRESULT = ee.call(goGet, {RequestedIMAGE:'CGIAR/SRTM90\_V4'} );

Map.centerObject( TheRESULT, 2 );

Map.addLayer( TheRESULT, {min:0, max:1500},'The Result' );

Unable to make this work as advertised

[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 **ALGORITHM** [[[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

algorithm.apply creates a new object by subjecting a specified set of input arguments to a specified algorithm.

This can be a standard Earth Engine algorithm or a user-defined algorithm created with a prior **function** statement.

newObject = ee.apply( algorithmName, inputArguments )

The specified input arguments, given as a comma-separated sequence of input values

The new object

The specified algorithm,

given by its name.

function goGet( RequestedIMAGE )

{ var ReturnedIMAGE = ee.Image( RequestedIMAGE );

return ReturnedIMAGE;

}

//var TheRESULT = goGet("CGIAR/SRTM90\_V4");

var TheRESULT = goGet.apply( "CGIAR/SRTM90\_V4" );

Map.centerObject( TheRESULT, 3 );

Map.addLayer( TheRESULT, {min:0, max:1500},'The Result' );

Unable to make this work as advertised