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

An array is an EE variable object that represents an ordered sequence of either numbers or other arrays. A two-dimensional array (i.e. an array of arrays of numbers)

is often called a “matrix.” Arrays 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** ARRAYS [ee.Array](#Array) [ee.Array.identity](#identity)

**EDITING** ARRAYS

BY **REPLICATING** PARTS [ee.Array.repeat](#repeat)

BY **COMBINING** THEM [ee.array.cat](#cat) [array.mask](#mask)

BY **REORGANIZING** THEM [array.sort](#sort) [array.transpose](#transpose) [array.project](#project)

BY **RECASTING** DATA TYPES [array.uint8](#uint8_) [array.Uint8](#uint8_)

[array.uint16](#uint8_) [array.Uint16](#uint8_)

[array.uint32](#uint8_) [array.Uint32](#uint8_)

[array.int8](#uint8_) [array.toInt8](#uint8_) [array.byte](#uint8_) [array.toByte](#uint8_)

[array.int16](#uint8_) [array.toInt16](#uint8_) [array.short](#uint8_) [array.toShort](#uint8_)

[array.int32](#uint8_) [array.toInt32](#uint8_) [array.int](#uint8_) [array.toInt](#uint8_)

[array.int64](#uint8_) [array.toInt64](#uint8_) [array.long](#uint8_) [array.toLong](#uint8_)

[array.float](#uint8_) [array.toFloat](#uint8_) [array.double](#uint8_) [array.toDouble](#uint8_)

**TRANSFORMING** ARRAYS

WITH **LOGICAL** OPERATIONS [array.eq](#eq_) [array.gt](#eq_) [array.lt](#eq_)  [array.and](#and_or)

[array.neq](#eq_) [array.gte](#eq_) [array.lte](#eq_) [array.or](#and_or)

[array.not](#not)

WITH **MATHEMATICAL** OPERATIONS [array.abs](#abs_) [array.ceil](#abs_) [array.floor](#abs_) [array.log](#abs_)

[array.round](#abs_) [array.sqrt](#abs_) [array.exp](#abs_) [array.log10](#abs_)

[array.add](#add_) [array.subtract](#add_) [array.multiply](#add_) [array.divide](#add_) [array.max](#add_) [array.min](#add_) [array.mod](#add_) [array.pow](#add_) [array.hypot](#add_) [array.first](#add_) [array.first\_nonzero](#add_)

WITH **TRIGONOMETRIC** OPERATIONS [array.sin](#sin_) [array.cos](#sin_) [array.tan](#sin_)

[array.sinh](#sin_) [array.cosh](#sin_) [array.tanh](#sin_)

[array.acos](#sin_) [array.asin](#sin_) [array.atan](#sin_) [array.atan2](#atan2)

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

( **TRANSFORMING** ARRAYS )

WITH **BITWISE** OPERATIONS [array.bitwiseAnd](#bitwise_) [array.bitwiseOr](#bitwise_) [array.bitwise\_xor](#bitwise_) [array.bitwiseNot](#bitwise_)

[array.bitwise\_and](#bitwise_) [array.bitwise\_or](#bitwise_) [array.bitwiseXor](#bitwise_) [array.bitwise\_not](#bitwise_)

[array.leftShift](#leftshift_) [array.left\_shift](#leftshift_) [array.rightShift](#leftshift_) [array.right\_shift](#leftshift_)

WITH **MATRIX** OPERATIONS [array.matrixDeterminant](#matrixD_T_F) [array.matrixTrace](#matrixD_T_F) [array.matrixFnorm](#matrixD_T_F)

[array.matrixDiagonal](#matrixDiag) [array.matrixToDiag](#matrixToDiag) [array.matrixTranspose](#matrixTranspose)

[array.matrixInverse](#matrixInverse) [array.matrixPseudoInverse](#matrixPseudo)

[array.matrixMultiply](#matrixMult) [array.matrixSolve](#matrixSolve) [array.eigen](#eigen)

WITH **REDUCERS** [array.reduce](#reduce) [array.accum](#accum)

**REPRODUCING** ARRAYS

AS **LISTS** [array.toList](#toList)

AS **CONFUSION MATRICES** [ee.ConfusionMatrix](#ConfusionMatrix)

**QUERYING** ARRAYS

FOR **VALUES**  [array.get](#get)

FOR **SUBARRAYS**  [array.slice](#slice) [array.cut](#cut)

**COMPARING** ARRAYS [ee.Algorithms.IsEqual(array)](#IsEqual)

**MEASURING** ARRAYS [array.length](#length)

**DOCUMENTING** ARRAYS [array.getInfo](#Describe_getInfo) [ee.Algorithms.Describe(array)](#Describe_getInfo)

[array.toString](#toString_serialize)  [array.serialize](#toString_serialize)

**PRESENTING** ARRAYS

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

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

IN **CHARTS** [Chart.array.values](#Chart)

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

ee.Array creates a new array from a specified set of numerical values of a specified numerical type.

newArray = ee.Array ( values, *pixelType* )

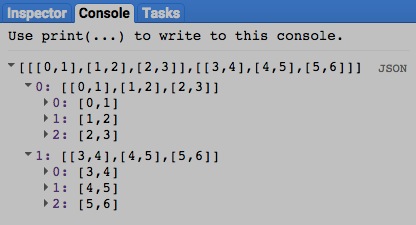
The specified value(s), given as a list of numbers, an array of numbers, or a single number

A specified numerical type, given as a pixelType object. Default: Inferred from the format of **values**

The new array

var TheARRAY = ee.Array( [ [ [0,1], [1,2], [2,3] ], [ [3,4], [4,5], [5,6] ] ] );

print( TheARRAY );



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

ee.Array.identity creates a new array representing a two-dimensional identity matrix of a specified size.

An identity matrix has a value of 1 at positions (1,1), (2,2), (3,3), etc. and 0 elsewhere.

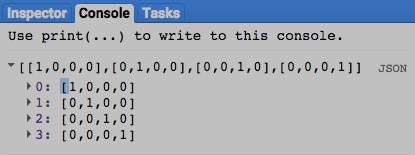
newArray = ee.Array.identity ( size )

The specified size of the new array, given as an integer indicating the number of rows and the number of columns to be generated

The new array

var TheARRAY = ee.Array.identity( 4 );

print( TheARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **REPLICATING** PARTS

ee.Array.repeat creates a new array by replicating the values of a specified array along a specified axis a specified number of times.

newArray = oldArray.repeat ( *axis*, repetitions )

The specified axis, given as its index number. Default: 0

The specified number of times, given as an integer. Default: 2

The specified array

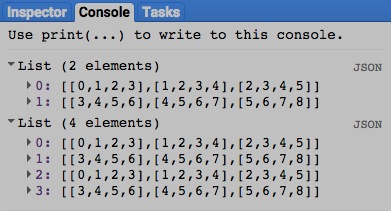
The new array

var OldARRAY = ee.Array( [ [ [0,1,2,3], [1,2,3,4], [2,3,4,5] ],

[ [3,4,5,6], [4,5,6,7], [5,6,7,8] ] ] );

var NewARRAY = OldARRAY.repeat( 0,2 );

print( OldARRAY, 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)

**EDITING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **COMBINING** THEM

ee.array.cat creates a new array by concatenating (joining) two or more specified arrays along a specified axis.

newArray = ee.Array.cat ( *arrayList, concatenationAxis* )

The specified axis, given as an integer referring to an index beginning with 0. Default: 0

The specified arrays, given as a list

The new array

var FirstARRAY = ee.Array( [ [ [111,112,113,114], [121,122,123,124], [131,132,133,134] ],

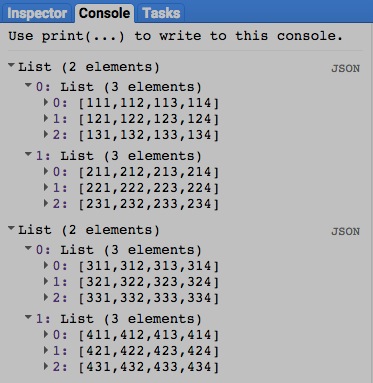
[ [211,212,213,214], [221,222,223,224], [231,232,233,234] ] ] );

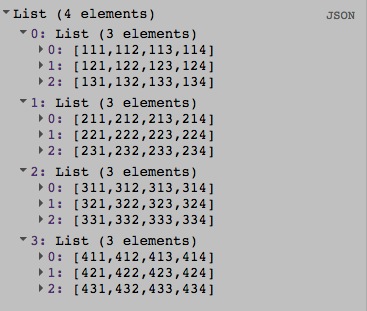
var SecondARRAY = ee.Array( [ [ [311,312,313,314], [321,322,323,324], [331,332,333,334] ],

[ [411,412,413,414], [421,422,423,424], [431,432,433,434] ] ] );

var NewARRAY = ee.Array.cat( [FirstARRAY, SecondARRAY],0 );

print( FirstARRAY, SecondARRAY, 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)

**EDITING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **COMBINING** THEM

array.mask creates a new array by removing, from the first of two specified arrays, all elements situated parallel to any non-zero element of the second of those arrays.

newArray = oldArray.mask ( *maskingArray* )

The second specified array

The first specified array

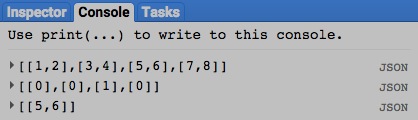
The new array

var FirstARRAY = ee.Array( [ [1,2],[3,4],[5,6],[7,8]] );

var SecondARRAY = ee.Array( [ [ 0],[ 0],[ 1],[ 0]] );

var NewARRAY = FirstARRAY.mask( SecondARRAY );

print( FirstARRAY, SecondARRAY, 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)

**EDITING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **REORGANIZING** THEM

array.sort creates a new array by sorting the elements of a specified array in ascending order along a specified axis.

newArray = oldArray.sort ( *arrayOfSortingKeys* )

The specified array

An array (of the same dimensions as **oldArray**) whose values are to be used to sort the elements of **oldArray** at similar positions

The new array

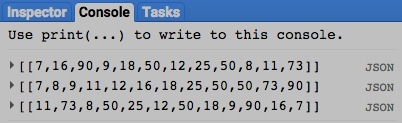
var OldARRAY = ee.Array( [ [ 7, 16, 90, 9, 18, 50, 12, 25, 50, 8, 11, 73 ] ] );

var KeyARRAY = ee.Array( [ [12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 1, 2 ] ] ); // Note those final two numbers

var SortedARRAY = OldARRAY.sort( );

var KeySortedARRAY = OldARRAY.sort( KeyARRAY );

print( OldARRAY, SortedARRAY, KeySortedARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **REORGANIZING** THEM

array.transpose creates a new array by transposing (swapping) two specified axes of a specified array.

newArray = oldArray.sort ( *firstAxis, secondAxis* )

The second specified axis, given as an integer referring to an index beginning with 0. Default: 1

The specified array

The new array

The first specified axis, given as an integer referring to an index beginning with 0. Default: 0

var TheARRAY = ee.Array( [ [111,112,113,114],

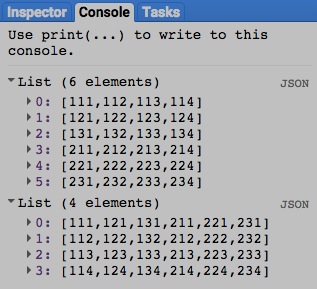
[121,122,123,124],

[131,132,133,134],

[211,212,213,214],

[221,222,223,224],

[231,232,233,234]

 ]

);

var TransposedARRAY = TheARRAY.transpose( 0,1 );

print( TheARRAY, TransposedARRAY );

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **REORGANIZING** THEM

array.project creates a new array by replicating the values of a specified array after dropping specified axes that have no more that a single index position.

newArray = oldArray.project ( axes )

The specified axes, given as a list of integers that refer to axis indices.

All axes containing more than one position must be specified.

The specified array

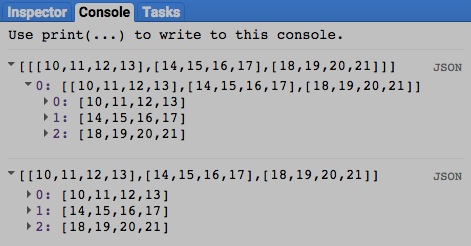
The new array

var TheARRAY = ee.Array( [ [ [10,11,12,13], [14,15,16,17], [18,19,20,21] ] ] );

var ProjectedARRAY = TheARRAY.project( [1,2] );

print( TheARRAY );

print( ProjectedARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) BY **RECASTING** DATA TYPES

array.uint8 , .toUint8 , .byte , and .toByte unsigned 8-bit integers

.uint16 and .toUint16 unsigned 16-bit integers

.uint32 and .toUint32 unsigned 32-bit integers

Each of these operations replicates a specified array (of any numerical type) to create a new one whose pixel type is as indicated in red.

.int8 and .toInt8 signed 8-bit integers

.int16 , .toInt16 , .short , and .toShort signed 16-bit integers

.int32 , .toInt32 , .int , and .toInt signed 32-bit integers

.int64 , .toInt64 , .long , and .toLong signed 64-bit integers

.float and .toFloat 32-bit floating-point numbers

.double and .toDouble 64-bit floating-point numbers

newArray = oldArray.uint8( ) or .toUint8( ) or .byte( ) or .toByte( )

or .uint16( ) or .toUint16( )

or .int16( ) or .toInt8( ) or .short( ) or .toShort( )

The new array

or .uint32( ) or .toUint32( )

or .int8( ) or .toInt8( )

The specified array

or .int32( ) or .toInt32( ) or .int( ) or .toInt( )

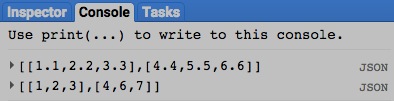
or .int64( ) or .toInt64( ) or .long( ) or .toLong( )

or .float( ) or .toFloat( )

or .double( ) or .toDouble( )

var OldARRAY = ee.Array( [ [1.1, 2.2, 3.3], [4.4, 5.5, 6.6] ] );

var NewARRAY = OldARRAY.byte();

print( OldARRAY, 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)

**TRANSFORMING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **LOGICAL** OPERATIONS

array.eq , .neq , .gt , .gte , .lt , and .lte

create a new array of 0s and 1s that characterize a specified relationship

between corresponding elements of two specified arrays.

newArray = 1stArray.eq( 2ndArray ) or .neq( ) or .gt( ) or .gte( ) or .lt( ) or .lte( )

The second specified array, which must have the same numbers of axes and elements as the first array.

The first specified array

The new array, in which each element is set to 1 if the specified relationship is true or to 0 if it is false

The specified relationship, questioning whether each element of the first specified array is equal to (**eg**), not equal to (**neq**),

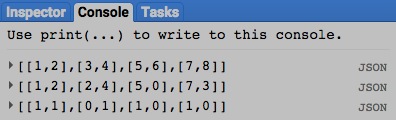
greater than (**gt**), greater than or equal to (**gte**), less than (**lt**), or less than or equal to (**lte**) the corresponding element of the second.

var FirstARRAY = ee.Array( [ [1,2],[3,4],[5,6],[7,8]] );

var SecondARRAY = ee.Array( [ [1,2],[2,4],[5,0],[7,3]] );

var NewARRAY = FirstARRAY.eq( SecondARRAY );

print( FirstARRAY, SecondARRAY, 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)

**TRANSFORMING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **LOGICAL** OPERATIONS

array.and and .or creates a new array of 0s or 1s that indicate whether either or both of each pair of corresponding elements in two specified arrays are non-zero.

newArray = 1stArray.and( 2ndArray ) or .or( )

The second specified array, which must have the same numbers of axes and elements as the first array.

The first specified array

The new array, in which each element is set to 1 if the specified relationship is true or to 0 if it is false

The specified relationship, questioning whether

*either* specified array is non-zero.

The specified relationship, questioning whether

*both* specified arrays are non-zero.

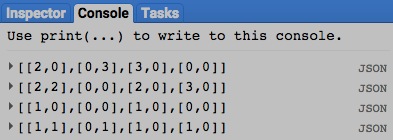
var FirstARRAY = ee.Array( [ [2,0],[0,3],[3,0],[0,0]] );

var SecondARRAY = ee.Array( [ [2,2],[0,0],[2,0],[3,0]] );

var AndARRAY = FirstARRAY.and( SecondARRAY );

var OrARRAY = FirstARRAY.or( SecondARRAY );

print( FirstARRAY, SecondARRAY, AndARRAY, OrARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **LOGICAL** OPERATIONS

array.not creates a new array in which each element is set to 1 if the corresponding element a specified array is equal to 0. Otherwise, the new element is set to 0.

newArray = oldArray.not( )

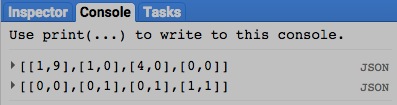
The specified array

The new array

var OldARRAY = ee.Array( [ [1,9],[1,0],[4,0],[0,0]] );

var NewARRAY = OldARRAY.not( );

print( OldARRAY, 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)

**TRANSFORMING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATHEMATICAL** OPERATIONS

array.abs , .round , .floor , .ceil , .sqrt , .exp , .log, and .log10 all create a new array by applying a specified math-

ematical function to each element in a specified array.

newArray = oldArray.abs( ) or .round( ) or .floor( ) or .ceil( ) or .sqrt( ) or .exp( ) or .log( ) or .log10( )

The specified

array

The new

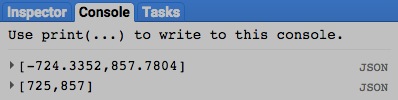
array

The specified mathematical function

var OldARRAY = ee.Array( [-724.3352, 857.7804] );

var NewARRAY = OldARRAY.floor().abs();

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)

**TRANSFORMING** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATHEMATICAL** OPERATIONS

array.add , .subtract , .multiply , .divide , .max , .min ,

all create a new number by applying a specified mathematical function to the corresponding elements of two specified arrays.

.mod , .pow , .hypot , .first , and .first\_nonzero

newArray = 1stArray.add( 2ndArray ) or .subtract( ) or .multiply( ) or .divide( ) or .max( ) or .min( )

The new array

or .mod( ) or .pow( ) or .hypot( ) or .first( ) or .first\_nonzero( )

The second specified array

The first specified array

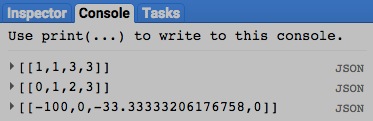
The specified mathematical function

var FirstARRAY = ee.Array( [ [1,1,3,3] ] );

var SecondARRAY = ee.Array( [ [0,1,2,3] ] );

var PctChangeARRAY = SecondARRAY.subtract(FirstARRAY).multiply(100.0).divide(FirstARRAY);

print ( FirstARRAY, SecondARRAY, PctChangeARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **TRIGONOMETRIC** OPERATIONS

array.sin , .cos , .tan , .sinh , .cosh , .tanh , .acos , .asin , and .atan all create a new array by applying a specified

trigonometric function to each element of a specified array.

newArray = oldArray.sin( ) or .cos( ) or .tan( )

or .sinh( ) or .cosh( ) or .tanh( ) or .acos( ) or .asin( ) or .atan( )

The new array

The specified array, whose values are assumed to be in radians

The specified trigonometric function

var DegreeARRAY = ee.Array( [30, 45, 60] );

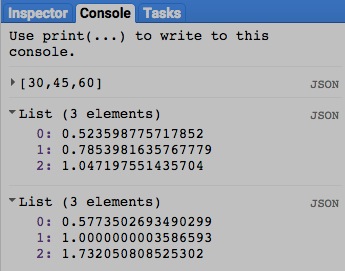
var RadianARRAY = ee.Array( DegreeARRAY.divide( ee.Array( [57.2957795, 57.2957795,57.2957795] ) ) );

var TangentARRAY = RadianARRAY.tan();

print( DegreeARRAY );

print( RadianARRAY );

print( TangentARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **TRIGONOMETRIC** OPERATIONS

array.atan2 creates a new array by calculating the arctangent of corresponding elements from two specified arrays that respectively hold numerators and denominators.

newArray = 1stArray.atan2( 2ndArray )

The specified array

of denominators

The specified array

of numerators

The new array

var FirstARRAY = ee.Array( [0, 1, 2, 3] );

var SecondARRAY = ee.Array( [2, 2, 2, 0] );

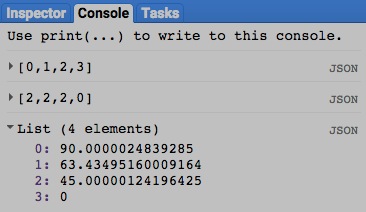
var RadianARRAY = FirstARRAY.atan2( SecondARRAY );

var DegreeARRAY = RadianARRAY.multiply( 57.2957795 );

print( FirstARRAY );

print( SecondARRAY );

print( DegreeARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **BITWISE** OPERATIONS

array.bitwiseAnd , .bitwiseOr , .bitwiseXor , .bitwiseNot

all create a new array by applying a specified bitwise function to the corresponding elements of two specified arrays.

.bitwise\_and , .bitwise\_or , .bitwise\_xor , and .bitwise\_not

newArray = 1stArray.bitwiseAnd( 2ndArray ) or .bitwiseOr( ) or .bitwiseXOr( ) or .bitwiseNot( )

or .bitwise\_and( ) or .bitwise\_or( ) or .bitwise\_xor( ) or .bitwise\_not( )

The new array

The second specified array

The first specified array

The specified bitwise function

var FirstARRAY = ee.Array( [1,1,1] );

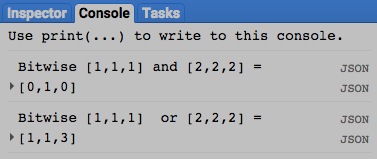
var SecondARRAY = ee.Array( [0,1,2] );

var BothBitARRAY = FirstARRAY.bitwiseAnd( SecondARRAY );

var EitherBitARRAY = FirstARRAY.bitwise\_or( SecondARRAY );

print ( 'Bitwise [1,1,1] and [2,2,2] =', BothBitARRAY );

print ( 'Bitwise [1,1,1] or [2,2,2] =', EitherBitARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **BITWISE** OPERATIONS

array.leftShift , .rightShift , left\_shift , and .right\_shift create a new array by shifting the bits of a specified array

left or right by a specified amount.

newArray = oldArray.leftShift( numberOfPositions ) or .rightShift( )

The new array

The specified array

The specified amount, given as an integer referring to bit positions

The specified function, indicating whether bits are to be shifted left or right

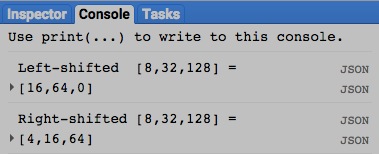
var OriginalARRAY = ee.Array( [8,32,128] );

var LeftwardARRAY = OriginalARRAY.leftShift( [1,1,1] );

var RightwardARRAY = OriginalARRAY.rightShift( [1,1,1] );

print ( 'Left-shifted [8,32,128] =', LeftwardARRAY );

print ( 'Right-shifted [8,32,128] =', RightwardARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixDeterminant , .matrixTrace , and .matrixFnorm create a new number equal to the determinant,

the trace, or the Frobenius norm of the matrix represented by a specified array.

newArray = oldArray.matrixDeterminant( ) or .matrixTrace( ) or .matrixFnorm( )

The new array

The specified array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixDiagonal creates a new single-column array of values drawn from the diagonal of the matrix represented by a specified array.

newArray = oldArray.matrixDiagonal( )

The new array

The specified array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixToDiag creates a new array representing a square diagonal matrix of values drawn from a specified array representing a single column matrix.

newArray = oldArray.matrixToDiag( )

The new array

The specified array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixTranspose creates a new array by transposing (swapping) two specified axes of a specified array.

newArray = oldArray.matrixTranspose( firstAxis, secondAxis )

The new array

The second specified axis, given as an integer referring to an index beginning with 0. Default: 0

The specified array

The first specified axis, given as an integer referring to an index beginning with 0. Default: 1

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixInverse creates a new array representing the inverse of the matrix represented by a specified array.

newArray = oldArray.matrixInverse( )

The new array

The specified array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixPseudoInverse creates a new array representing the Moore-Penrose pseudoinverse of the matrix represented by a specified array.

newArray = oldArray.matrixPseudoInverse( )

The new array

The specified array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixMultiply creates a new array representing the matrix multiplication of specified left- and right-hand arrays.

newArray = oldLeftArray.matrixTranspose( oldRightArray )

The specified right-hand array

The specified left-hand array

The new array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.matrixSolve creates a new array representing the solution for X in the matrix equation LEFT \* X = RIGHT where LEFT and RIGHT

are matrices represented by specified left- and right-hand arrays. If LEFT is over-determined. a least-squares solution is found.

newArray = oldLeftArray.matrixSolve( oldRightArray )

The specified right-hand array

The specified left-hand array

The new array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **MATRIX** OPERATIONS

array.eigen a new array containing the real eigenvectors and eigenvalues of a specified two-dimensional array of A rows and A columns.

The new array will have A rows and A+1 columns, with each row containing an eigenvalue in its first column, and the corresponding eigenvector in the remaining A columns. The rows are sorted by eigenvalue in descending order.

newArray = oldArray.eigen( )

The new array

The specified array

ADD EXAMPLE

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **REDUCERS**

array.reduce creates a new array by applying a specified reducer to specified axes of a specified array

such that the input values along each specified axis are reduced to a single output value.

newArray = oldArray.reduce( reducer, axesToBeReduced*, axisForFieldNames* )

An axis to be used for names of the reducer’s input

and output field names, given as an index number.

Default: null.

The specified reducer

The new array

The specified axes, given as a list of index numbers

The specified array

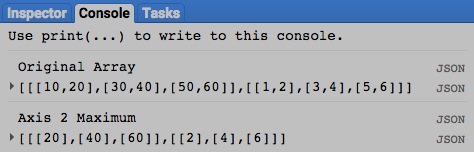
var TheREDUCER = ee.Reducer.max( );

var OriginalARRAY = ee.Array( [ [ [10,20], [30,40], [50,60] ], [ [1,2], [3,4], [5,6] ] ] );

var Axis2MaxARRAY = OriginalARRAY.reduce( TheREDUCER, [2] );

print( 'Original Array', OriginalARRAY );

print( 'Axis 2 Maximum', Axis2MaxARRAY );



[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) WITH **REDUCERS**

array.accum creates a new array by using a specified reducer to accumulate the values of elements along a specified axes of a specified array such that each new

value is computed by applying the specified reducer to the existing values of all elements at the same or lower positions along a specified axis.

newArray = oldArray.accum( *axis, reducer* )

The specified array

A specified reducer. Default: Sum.

The new array

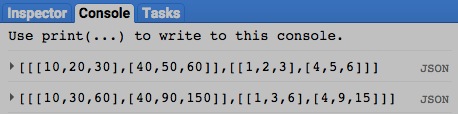
A specified axis, given as an index number. Default: 0

var OriginalARRAY = ee.Array( [ [ [10,20,30], [40,50,60] ], [ [1,2,3], [4,5,6] ] ] );

var Axis2SumARRAY = OriginalARRAY.accum( 2, ee.Reducer.sum( ) );

print( OriginalARRAY );

print( Axis2SumARRAY );



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

array.toList creates a new hierarchical set of lists replicating the structure and content of a specified array.

newList = oldArray.toList ( )

The new list(s)

The specified array

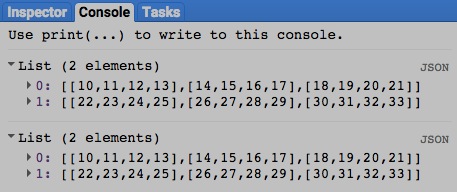
var TheARRAY = ee.Array( [ [ [10,11,12,13], [14,15,16,17], [18,19,20,21] ],

[ [22,23,24,25], [26,27,28,29], [30,31,32,33] ] ] );

var TheLIST = TheARRAY.toList( );

print( TheARRAY );

print( TheLIST );

****

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) AS **CONFUSION MATRICES**

ee.ConfusionMatrix creates a new confusion matrix from a specified two-dimensional array

- whose horizontal rows (axis 1) represent known classes,

- whose vertical columns (axis 0) represent predicted classes, and

- whose values indicate the number cases in which a given known

value was classified as a given predicted value.

newConfusionMatrix = ee.ConfusionMatrix( array, *order* )

The row and column size and order of a

non-contiguous or non-zero matrix, given as a list

The specified array

The new confusion matrix

var TheARRAY = ee.Array( [ [0,2,0,0],

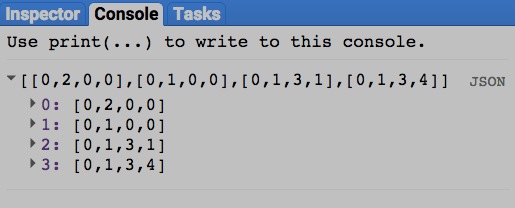
[0,1,0,0],

[0,1,3,1],

[0,1,3,4] ] );

var TheCONFUSIONMATRIX = ee.ConfusionMatrix( TheARRAY );

print( TheCONFUSIONMATRIX );



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

array.get creates a new number by copying the value at a specified position in a specified array.

newNumber = oldArray.get ( position )

The specified position, given as a list of integers representing array axis indices (each starting at 0)

The specified array

The new number

var OldARRAY = ee.Array( [ [ [10,11,12,13], [14,15,16,17], [18,19,20,21] ],

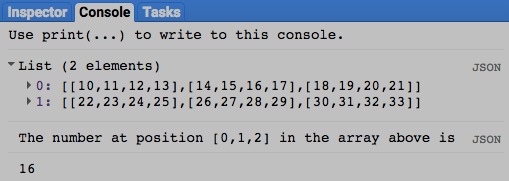
[ [22,23,24,25], [26,27,28,29], [30,31,32,33] ] ] );

var NewNUMBER = OldARRAY.get( [0,1,2] );

print( OldARRAY );

print( 'The number at position [0,1,2] in the array above is');

print( 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)

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

array.slice creates a new array by replicating a specified sequence of values along a specified axis of a specified array.

newArray = oldArray.slice ( axis, *startingPosition*, *stoppingPosition*, *everyNth* )

The specified axis, given as an integer indicating its index: Default: 0

The new array

The (positive) number of positions between each selected position and the one following it. Default: 1

The specified array

The position immediately following the last of those whose values are to be replicated, given as an integer starting with zero (or as a negative integer to be counted backward from the end of the axis. Default: Number of list items

The first of the positions whose values are to be replicated, given as an integer starting with zero (or as a negative integer to be counted backward from the end of the axis. Default: 0

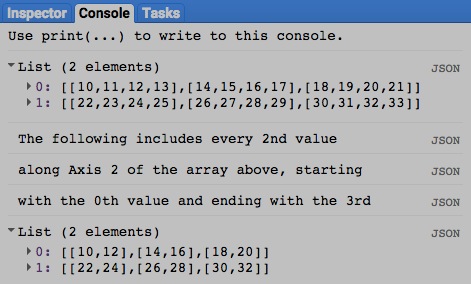
var OldARRAY = ee.Array( [ [ [10,11,12,13], [14,15,16,17], [18,19,20,21] ],

[ [22,23,24,25], [26,27,28,29], [30,31,32,33] ] ] );

var NewARRAY = OldARRAY.slice( 2, 0, 3, 2 );

print( OldARRAY );

print( 'The following includes every 2nd value' );

print( 'along Axis 2 of the array above, starting' );

print( 'with the 0th value and ending with the 3rd' );

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)

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

array.cut creates a new array by replicating the values at specified positions along the axes of a specified array.

newArray = oldArray.cut ( position )

The specified array

The specified positions, given as a list of integers such that the first identifies a position on axis 0, the next on axis 1, and so on until all axes are accounted for. A position given as -1 specifies all positions along that axis.

The new array

var OldARRAY = ee.Array( [ [ [10,11,12,13], [14,15,16,17], [18,19,20,21] ],

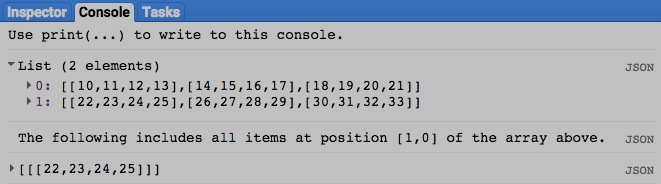
[ [22,23,24,25], [26,27,28,29], [30,31,32,33] ] ] );

var NewARRAY = OldARRAY.cut( [1,0,-1] );

print( OldARRAY );

print( 'The following includes all items at position [1,0] of the array above.' );

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

newBoolean = ee.Algorithms.IsEqual ( 1stArray, 2ndArray )

The first specified array

The second specified array

The new Boolean

var FirstARRAY = ee.Array( [ [1,9],[1,0],[4,0],[0,0]] );

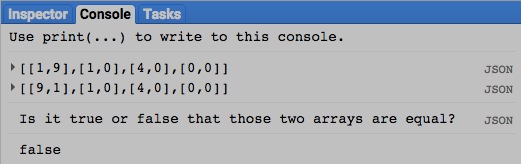
var SecondARRAY = ee.Array( [ [9,1],[1,0],[4,0],[0,0]] );

var TrueOrFalse = ee.Algorithms.IsEqual( FirstARRAY, SecondARRAY );

print( FirstARRAY, SecondARRAY );

print('Is it true or false that those two arrays 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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

array.length creates a new one-dimensional array of integers indicating the number of elements in each axis of a specified array, starting with axis 0, then 1, and so on.

newArray = oldArray.length ( )

The specified array

The new array

### var TheARRAY = ee.Array( [ [ [111,112,113,114], [121,122,123,124], [131,132,133,134] ],

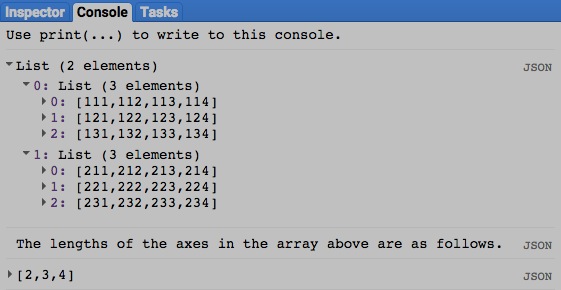
### [ [211,212,213,214], [221,222,223,224], [231,232,233,234] ] ] );

### var LengthARRAY = TheARRAY.length( );

### print( TheARRAY );

### print('The lengths of the axes in the array above are as follows.' );

### print( LengthARRAY );



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

ee.Algorithms.Describe and array.getInfo

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

newObject = ee.Algorithms.Describe( oldArray )

and oldArray.getInfo( )

The specified array

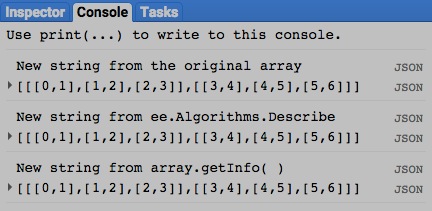
The new object

var OldARRAY = ee.Array( [ [ [0,1], [1,2], [2,3] ], [ [3,4], [4,5], [5,6] ] ] );

print( 'New string from the original array', OldARRAY );

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

print( 'New string from array.getInfo( )', OldARRAY.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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx)

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

newString = oldArray.toString ( )

and oldArray.serialize( )

The specified array

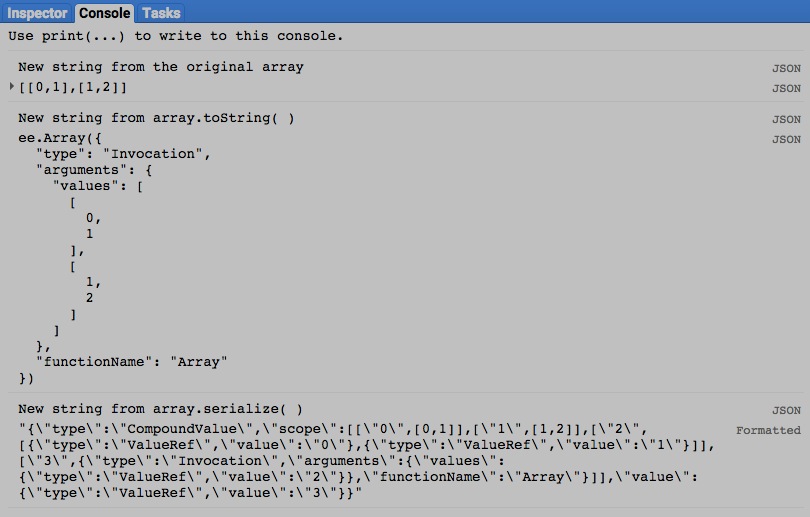
The new string

var OldARRAY = ee.Array( [ [ [0,1], [1,2], [2,3] ], [ [3,4], [4,5], [5,6] ] ] );

print( 'New string from the original array', OldARRAY );

print( 'New string from array.toString( )', OldARRAY.toString( ) );

print( 'New string from array.serialize( )', OldARRAY.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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) IN **PRINT**

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

print( oldArray ) or console.log( oldArray )

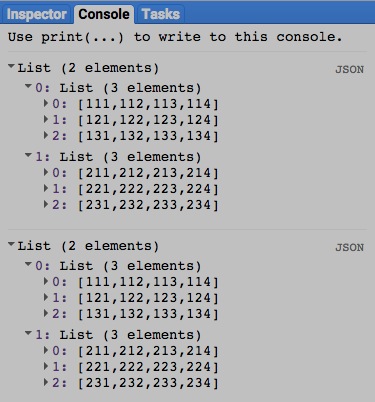
The specified array

var TheARRAY = ee.Array( [ [ [111,112,113,114], [121,122,123,124], [131,132,133,134] ],

[ [211,212,213,214], [221,222,223,224], [231,232,233,234] ] ] );

print( TheARRAY );

console.log( TheARRAY );



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

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

alert( oldArray ) or confirm( oldArray )

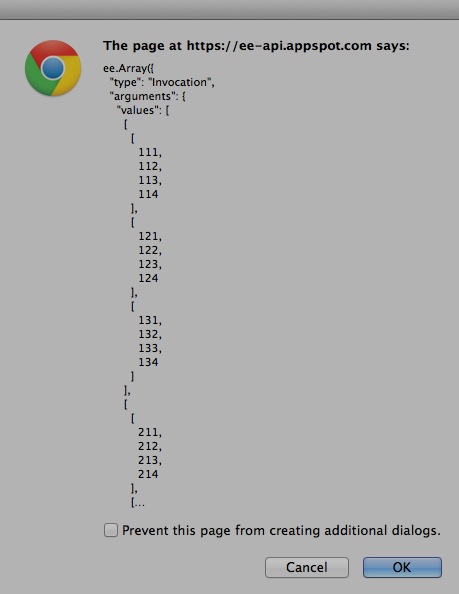
The specified array

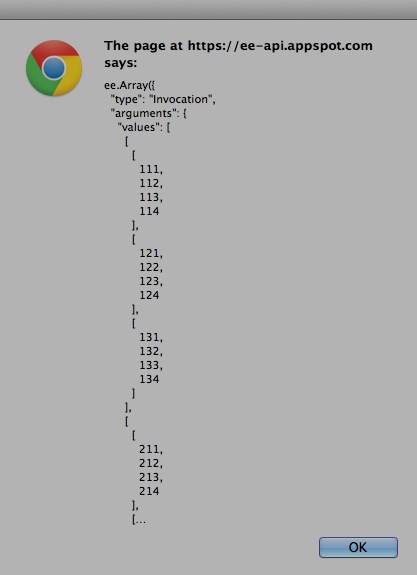
var TheARRAY = ee.Array( [ [ [111,112,113,114], [121,122,123,124], [131,132,133,134] ],

[ [211,212,213,214], [221,222,223,224], [231,232,233,234] ] ] );

alert( TheARRAY );

confirm( TheARRAY );



****

[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** [ARRAY](#_top) [VARIABLES](EE13%20%20%20%20%20%20%20%20%20Variables.docx) IN **CHARTS**

Chart.array.values presents a chart

- whose horizontal (X) axis identifies the indices along a specified dimension of a specified array;

- whose vertical (Y) axis identifies the range of values stored within that array; and

- whose charted data indicate the values associated with each of a series of array indices other than those on the X axis.

newChart = Chart.array.values( oldArray, xDimension, *xLabels* )

The specified array

The specified dimension,

given as a number

A list of numerical values to be used as labels

for the array indices presented on chart’s X-axis

The new chart

var TheARRAY = ee.Array( [ [ 0, 1, 2, 3],

[ 10, 11, 12, 13],

[ 20, 21, 22, 23] ] );

var TheCHART = Chart.array.values( TheARRAY, 1, [0,1,2,3]);

var TheCHART = TheCHART.setSeriesNames( ['Array Row 0','Array Row 1','Array Row 2'] );

var TheCHART = TheCHART.setOptions( { title: 'Arrayed Values',

hAxis: { title: 'Array Columns' },

vAxis: { title: 'Array Value' } } );

print( TheCHART );

