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

A pixelType is an Earth Engine parameter object that represents one of the digital formats available for the storage of numerical values associated with pixels.. PixelTypes can be processed by using the operations listed below, which vary according to the nature of that processing. Each operation name is linked to a separate page describing that operation.

**CREATING** PIXELTYPES [ee.PixelType](#PixelType)

[ee.PixelType.uint8](#uint8_) [ee.PixelType.uint16](#uint8_)  [ee.PixelType.uint32](#uint8_)

[ee.PixelType.int8](#uint8_) [ee.PixelType.int16](#uint8_) [ee.PixelType.int32](#uint8_) [ee.PixelType.int64](#uint8_)

[ee.PixelType.double](#uint8_) [ee.PixelType.float](#uint8_)

**DOCUMENTING** PIXELTYPES pixelType.getInfo ee.Algorithms.Describe(pixelType)

pixelType.toString pixelType.serialize

**PRESENTING** PIXELTYPES

IN **PRINT** print(pixelType) console.log(pixelType)

alert(pixelType) confirm(pixelType)

[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** [PIXELTYPE](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

ee.PixelType creates a new pixel type of a specified precision with specified minimum and maximum values and dimensionality..

newPixelType = ee.PixelType( precision*, minimumValue, maximumValue, numberOfDimensions* )

The new pixelType

A specified number of dimensions in which pixels values can vary,

given as 0 for scalar, 1 for vector, 2 for matrix, etc. Default: null.

The specified precision, given as one

of the strings **'int'**, **'float'**, or **'double'**.

A specified minimum value for pixels of this type. If precision is **'float'** or

**'double'**, this can be null, signifying negative infinity. Default: null.

A specified minimum value for pixels of this type. If precision is **'float'** or

**'double'**, this can be null, signifying negative infinity. Default: null.

var ThePIXELTYPE = ee.PixelType( 'double' );

print( ThePIXELTYPE );

[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** [PIXELTYPE](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

ee.PixelType.uint8 , .uint16 , and .uint32 unsigned 8, 16, and 32-bit integers

.int8 , .int16 , .int32, and .int64 signed 8, 16, 32, and 64-bit integers

Each of these operations creates a pixelType of a particular numerical type as indicated in red.

.float and .double 32 and 64-bit floating-point numbers

newNumber = oldNumber.uint8( ) or .touint8( ) or .uint16( ) or .uint32( )

or .int8( ) or .int16( ) or .int32( ) or .int64( )

or .float( ) or .double( )

The new pixelType

var

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)

**DOCUMENTING** [PIXELTYPE](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

ee.Algorithms.Describe and pixelType.getInfo

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

newObject = ee.Algorithms.Describe( oldPixelType )

and oldPixelType.getInfo( )

The specified pixelType

The new object

var ThePIXELTYPE = ee.PixelType( 'double' );

print( 'From print:', ThePIXELTYPE );

print( 'From ee.Algorithms.Describe( ):', ee.Algorithms.Describe(ThePIXELTYPE ) );

print( 'From getInfo( ):', ThePIXELTYPE.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** [PIXELTYPE](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx)

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

newString = oldPixelType.toString ( )

and oldPixelType.serialize( )

The specified pixelType

The new string

var ThePIXELTYPE = ee.PixelType( 'double' );

print( 'From print:', ThePIXELTYPE );

print( 'From toString( ):', ThePIXELTYPE.toString( ) );

print( 'From serialize( ):', ThePIXELTYPE.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** [PIXELTYPE](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) IN **PRINT**

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

print( oldPixelType ) or console.log( oldPixelType )

The specified pixelType

var ThePIXELTYPE = ee.PixelType( 'double' );

print( 'From print:', ThePIXELTYPE );

console.log( 'From console.log:', ThePIXELTYPE );

[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** [PIXELTYPE](#_top) [[PARAMETERS](EE26%20%20%20%20%20%20%20%20%20Parameters.docx)](EE25%20%20%20%20%20%20%20%20%20Parameters.docx) IN **PRINT**

alert ( pixelType ) and confirm ( pixelType ) present JSON-formatted text renditions of a specified

pixelType in a pop-up message box.

alert( oldPixelType ) or confirm( oldPixelType )

The specified pixelType

var ThePIXELTYPE = ee.PixelType( 'double' );

alert( ThePIXELTYPE );

confirm( ThePIXELTYPE );