ADVANCED GIS AND REMOTE SENSING
(IRGN 490)

Ran Goldblatt

Lecture 9:
Google Earth Engine API-1
Images and Image Collections

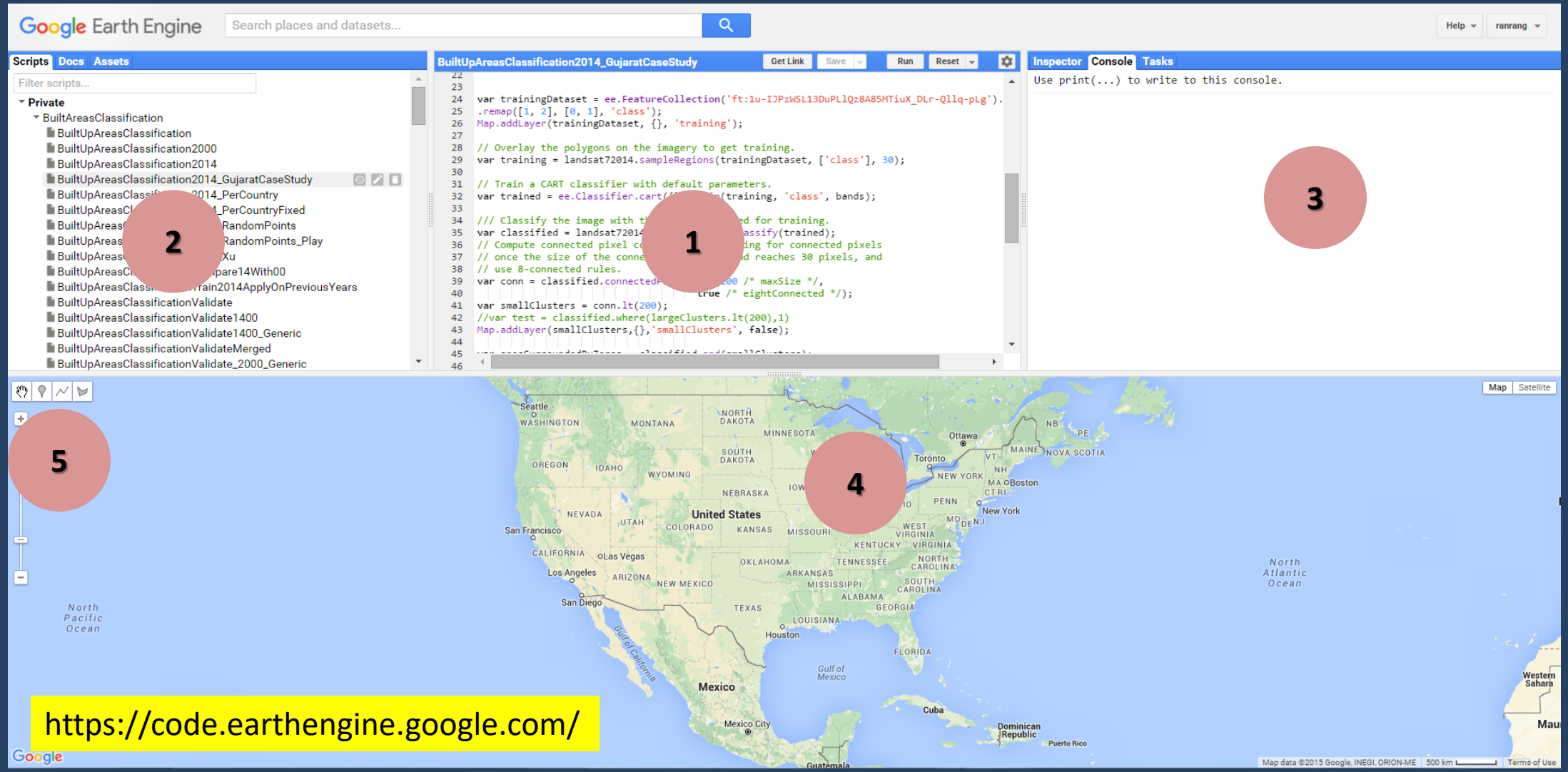
https://code.earthengine.google.com/

Google Earth Engine Coder (API)

- Used to run algorithms on georeferenced imagery and vectors.
- Includes a library of functions that can be applied to imagery for display and analysis.
- Vector datasets (feature collections) are used via Google Fusion Tables.
- MUCH more control on the analysis process

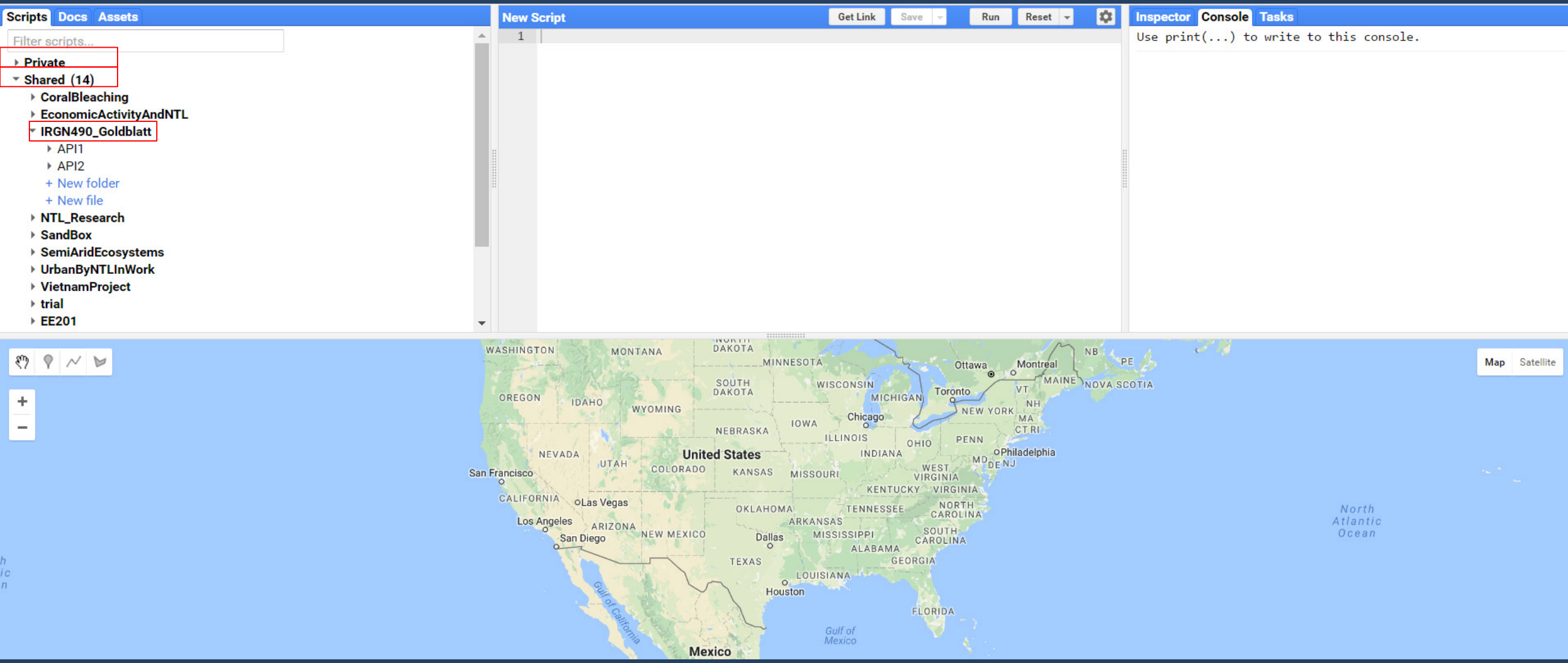
The Application Programming Interface (API) of Google Earth Engine (EE)

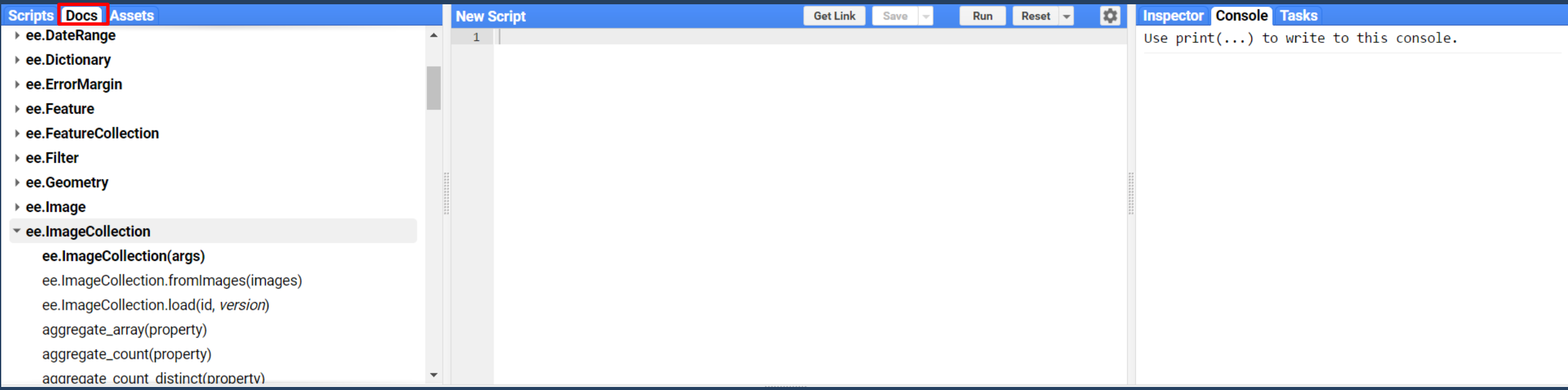
https://code.earthengine.google.com/

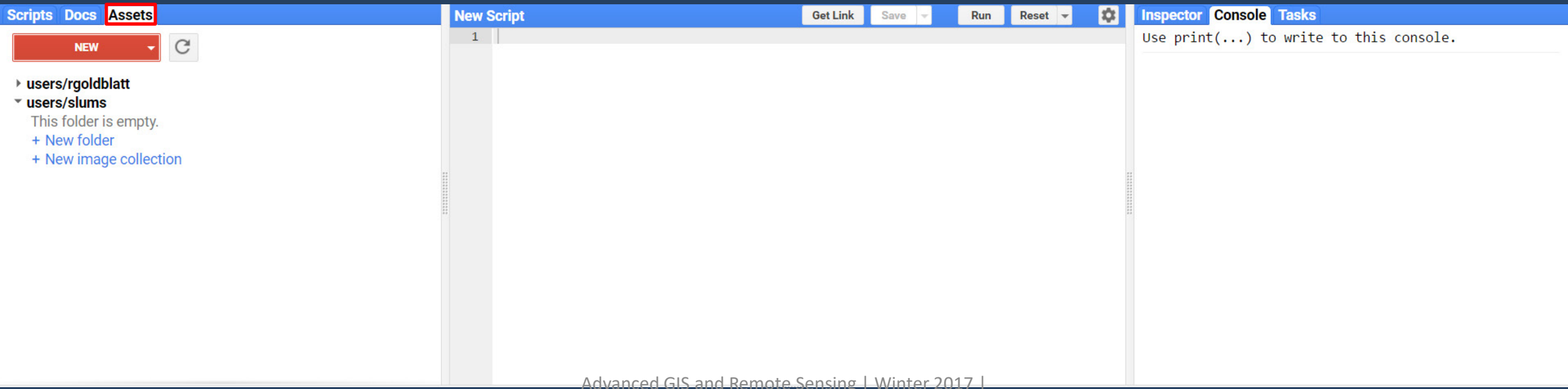


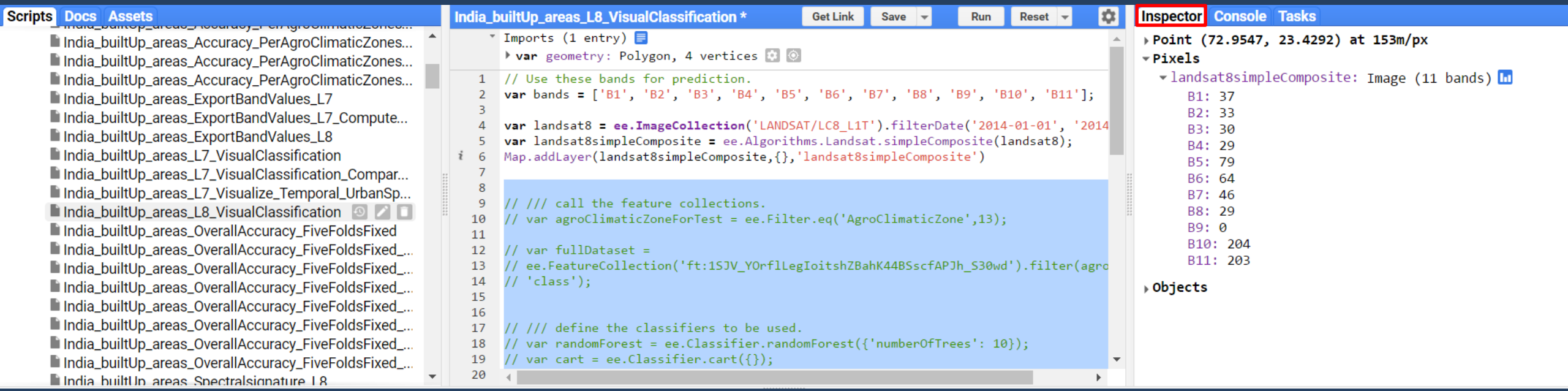
1- Code editor | **2**- Archive panel, documentation, examples | **3**- Report panel, debugging and output

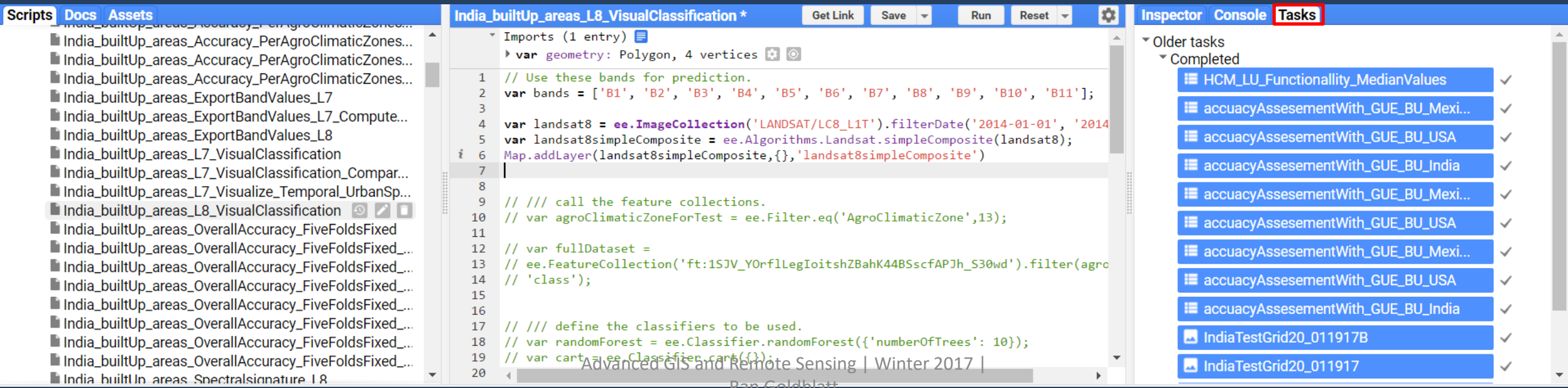
4- Display screen | **5**- Draw feature geometries


Scripts
Docs
Assets
Filter scripts...
Private
Shared (14)
CoralBleaching
EconomicActivityAndNTL
IRGN490_Goldblatt
API1
API2
+ New folder
+ New file
NTL_Research
SandBox
SemiAridEcosystems
UrbanByNTLInWork
VietnamProject
trial
EE201
New Script
Get Link
Save
Run
Reset
1
Inspector
Console
Tasks
Use print(...) to write to this console.
Map
Satellite
United States
Mexico
Canada
San Francisco
Los Angeles
San Diego
Las Vegas
Chicago
Toronto
Ottawa
Montreal
Philadelphia
Houston
Dallas
WASHINGTON
OREGON
CALIFORNIA
NEVADA
IDAHO
MONTANA
WYOMING
UTAH
ARIZONA
COLORADO
NEW MEXICO
NORTH DAKOTA
SOUTH DAKOTA
NEBRASKA
KANSAS
OKLAHOMA
TEXAS
MINNESOTA
IOWA
MISSOURI
ARKANSAS
LOUISIANA
WISCONSIN
ILLINOIS
MICHIGAN
INDIANA
OHIO
KENTUCKY
TENNESSEE
MISSISSIPPI
ALABAMA
GEORGIA
FLORIDA
SOUTH CAROLINA
NORTH CAROLINA
VIRGINIA
WEST VIRGINIA
PENN
NEW YORK
MAINE
VT
NH
MA
CT RI
NJ
DE
MD
NB
PE
NOVA SCOTIA
Gulf of California
Gulf of Mexico
North Atlantic Ocean

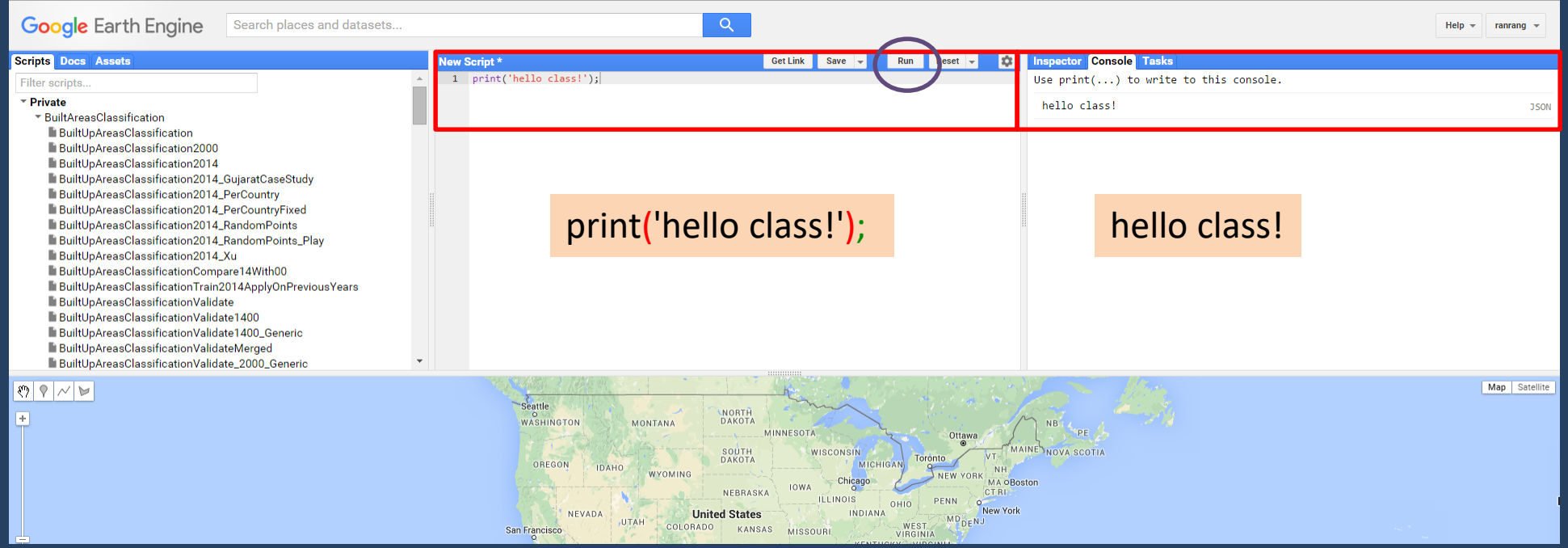

Scripts
Docs
Assets
ee.DateRange
ee.Dictionary
ee.ErrorMargin
ee.Feature
ee.FeatureCollection
ee.Filter
ee.Geometry
ee.Image
ee.ImageCollection
ee.ImageCollection(args)
ee.ImageCollection.fromImages(images)
ee.ImageCollection.load(id, version)
aggregate_array(property)
aggregate_count(property)
aggregate_count_distinct(property)
New Script
Get Link
Save
Run
Reset
1
Inspector
Console
Tasks
Use print(...) to write to this console.


Scripts
Docs
Assets
NEW
users/rgoldblatt
users/slums
This folder is empty.
+ New folder
+ New image collection
New Script
Get Link
Save
Run
Reset
1
Inspector
Console
Tasks
Use print(...) to write to this console.


Scripts
Docs
Assets
India_builtUp_areas_Accuracy_PerAgroClimaticZones...
India_builtUp_areas_Accuracy_PerAgroClimaticZones...
India_builtUp_areas_Accuracy_PerAgroClimaticZones...
India_builtUp_areas_ExportBandValues_L7
India_builtUp_areas_ExportBandValues_L7_Compute...
India_builtUp_areas_ExportBandValues_L8
India_builtUp_areas_L7_VisualClassification
India_builtUp_areas_L7_VisualClassification_Compar...
India_builtUp_areas_L7_Visualize_Temporal_UrbanSp...
India_builtUp_areas_L8_VisualClassification
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_Spectralsignature_L8
India_builtUp_areas_L8_VisualClassification *
Get Link
Save
Run
Reset
Imports (1 entry)
var geometry: Polygon, 4 vertices
1 // Use these bands for prediction.
2 var bands = ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'B11'];
3
4 var landsat8 = ee.ImageCollection('LANDSAT/LC8_L1T').filterDate('2014-01-01', '2014
5 var landsat8simpleComposite = ee.Algorithms.Landsat.simpleComposite(landsat8);
6 Map.addLayer(landsat8simpleComposite,{},'landsat8simpleComposite')
7
8
9 // /// call the feature collections.
10 // var agroClimaticZoneForTest = ee.Filter.eq('AgroClimaticZone',13);
11
12 // var fullDataset =
13 // ee.FeatureCollection('ft:1SJV_YOrflLegIoitshZBahK44BSscfAPJh_S30wd').filter(agro
14 // 'class');
15
16
17 // /// define the classifiers to be used.
18 // var randomForest = ee.Classifier.randomForest({'numberOfTrees': 10});
19 // var cart = ee.Classifier.cart({});
20
Inspector
Console
Tasks
Point (72.9547, 23.4292) at 153m/px
Pixels
landsat8simpleComposite: Image (11 bands)
B1: 37
B2: 33
B3: 30
B4: 29
B5: 79
B6: 64
B7: 46
B8: 29
B9: 0
B10: 204
B11: 203
Objects


Scripts
Docs
Assets
India_builtUp_areas_Accuracy_PerAgroClimaticZones...
India_builtUp_areas_Accuracy_PerAgroClimaticZones...
India_builtUp_areas_Accuracy_PerAgroClimaticZones...
India_builtUp_areas_ExportBandValues_L7
India_builtUp_areas_ExportBandValues_L7_Compute...
India_builtUp_areas_ExportBandValues_L8
India_builtUp_areas_L7_VisualClassification
India_builtUp_areas_L7_VisualClassification_Compar...
India_builtUp_areas_L7_Visualize_Temporal_UrbanSp...
India_builtUp_areas_L8_VisualClassification
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_OverallAccuracy_FiveFoldsFixed_...
India_builtUp_areas_Spectralsignature_L8
India_builtUp_areas_L8_VisualClassification *
Get Link
Save
Run
Reset
Imports (1 entry)
var geometry: Polygon, 4 vertices
1 // Use these bands for prediction.
2 var bands = ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10', 'B11'];
3
4 var landsat8 = ee.ImageCollection('LANDSAT/LC8_L1T').filterDate('2014-01-01', '2014
5 var landsat8simpleComposite = ee.Algorithms.Landsat.simpleComposite(landsat8);
6 Map.addLayer(landsat8simpleComposite,{},'landsat8simpleComposite')
7
8
9 // /// call the feature collections.
10 // var agroClimaticZoneForTest = ee.Filter.eq('AgroClimaticZone',13);
11
12 // var fullDataset =
13 // ee.FeatureCollection('ft:1SJV_YOrflLegIoitshZBahK44BSscfAPJh_S30wd').filter(agro
14 // 'class');
15
16
17 // /// define the classifiers to be used.
18 // var randomForest = ee.Classifier.randomForest({'numberOfTrees': 10});
19 // var cart = ee.Classifier.cart({});
20
Inspector
Console
Tasks
Older tasks
Completed
HCM_LU_Functionallity_MedianValues
accuacyAssesementWith_GUE_BU_Mexi...
accuacyAssesementWith_GUE_BU_USA
accuacyAssesementWith_GUE_BU_India
accuacyAssesementWith_GUE_BU_Mexi...
accuacyAssesementWith_GUE_BU_USA
accuacyAssesementWith_GUE_BU_Mexi...
accuacyAssesementWith_GUE_BU_USA
accuacyAssesementWith_GUE_BU_India
IndiaTestGrid20_011917B
IndiaTestGrid20_011917

Let`s begin…



JavaScript ***statements***

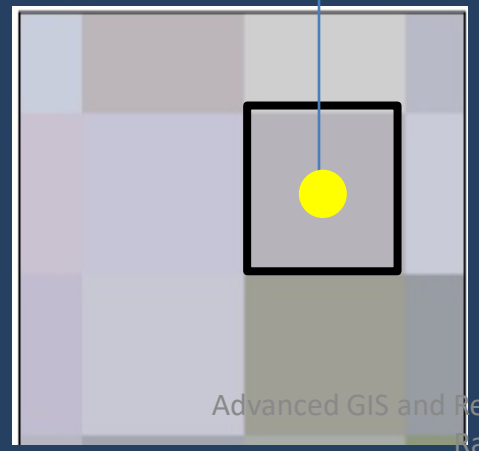
- JavaScript statements have a pair of parentheses ()
- They end with a semicolon ;
- Anything between single quotes is called a 'string'.
- **Strings:**
  - Used to print out readable information.
  - Define sets of characters that name datasets or parts of datasets

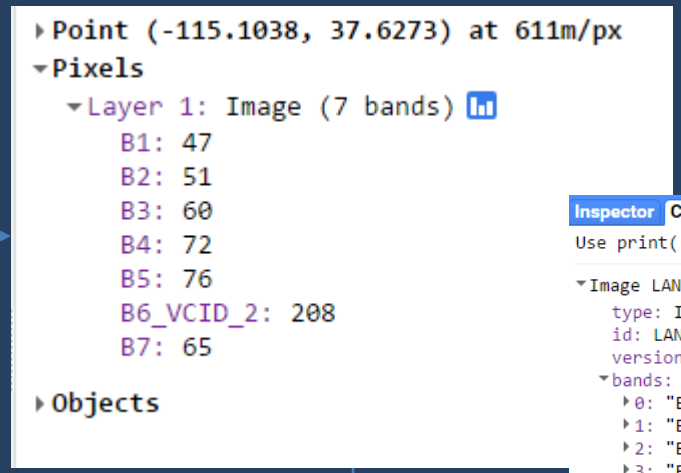
```
print('hello class');
```

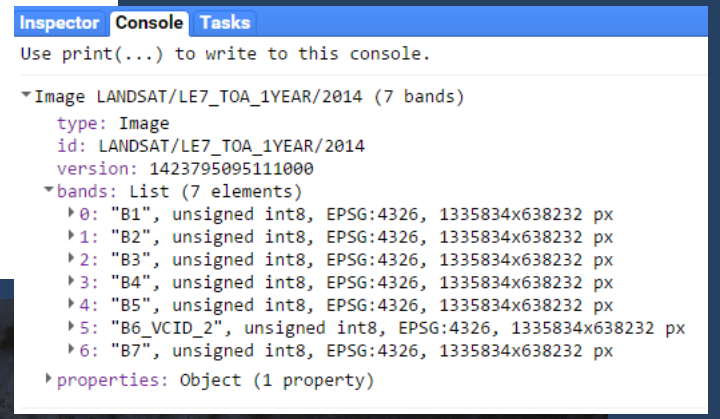
Image and Image collection = Rasters

Images

- Pixels with associated values.
- An image can have multiple bands
- Each band in an image contains:
  - A name
  - Pixel value
  - Resolution
  - Projection







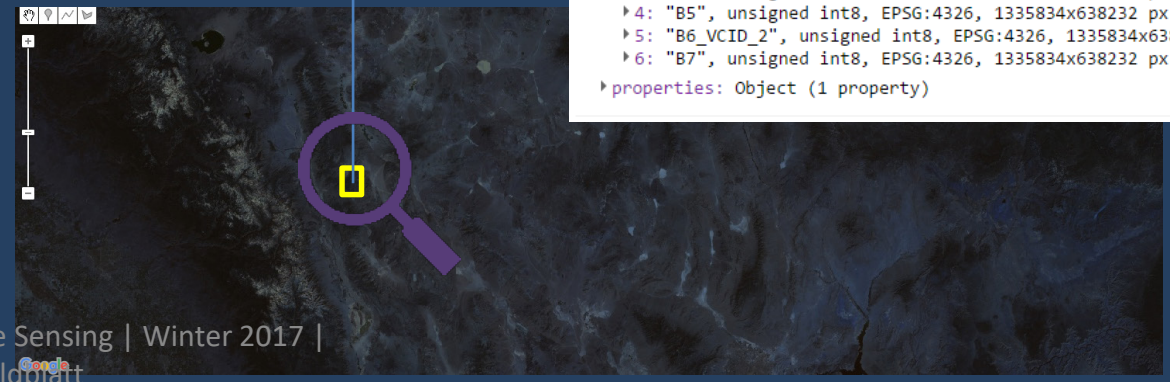
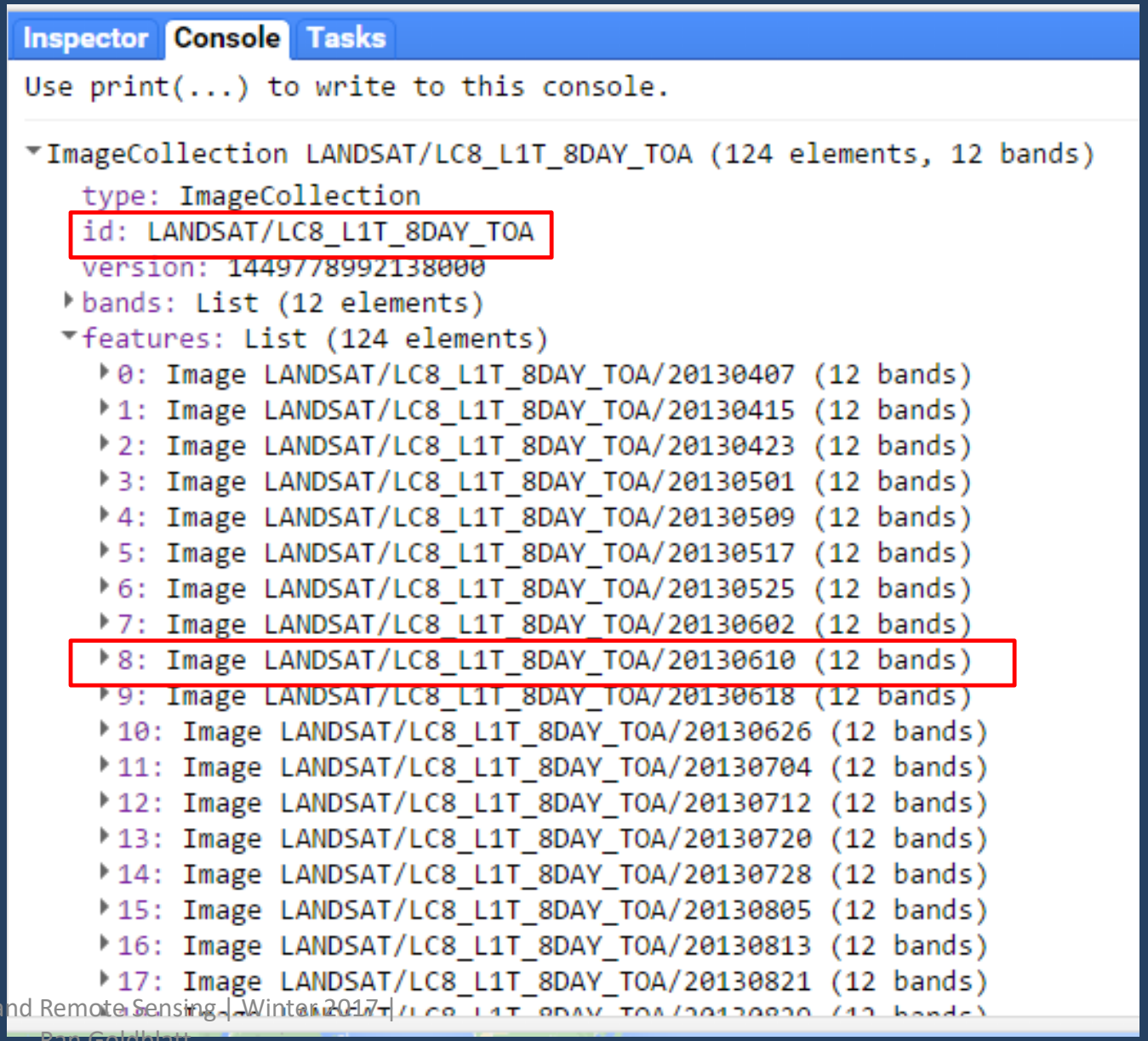


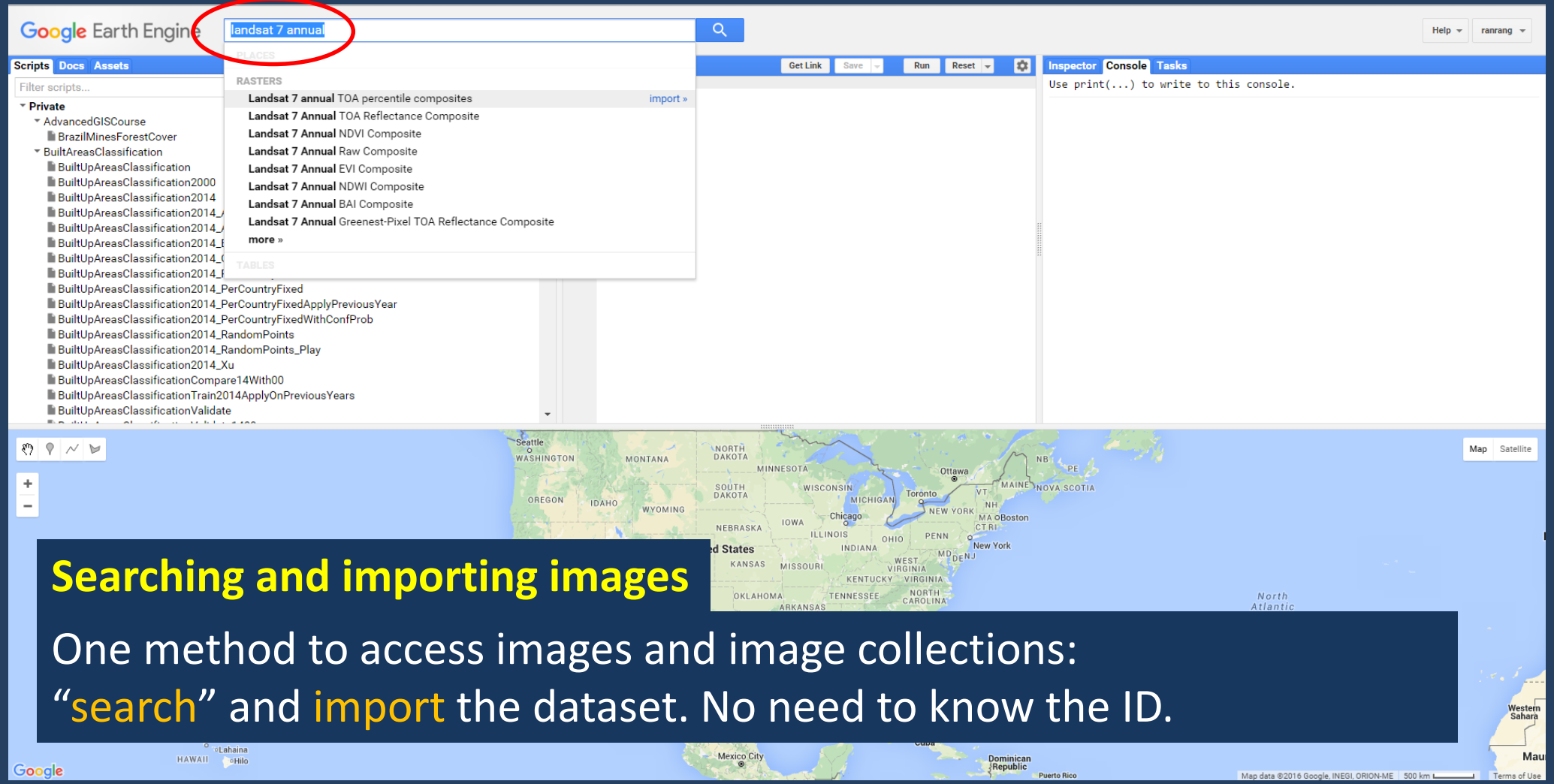
Image collection

- Collections/
  Mosaics of EE images

- *Example:*
  - A collection of all L8 images in a given time period
  - Each image collection has an ID.

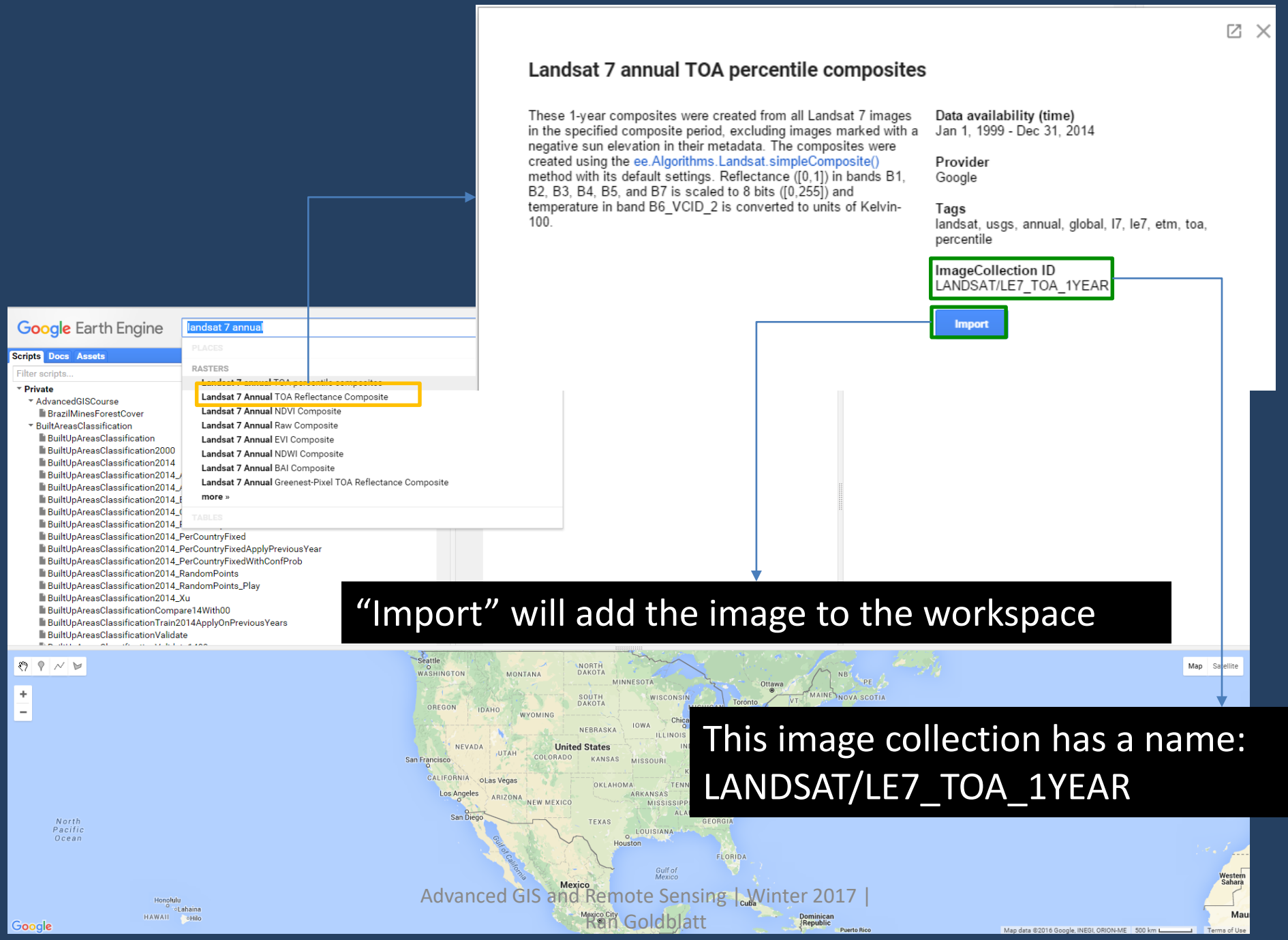


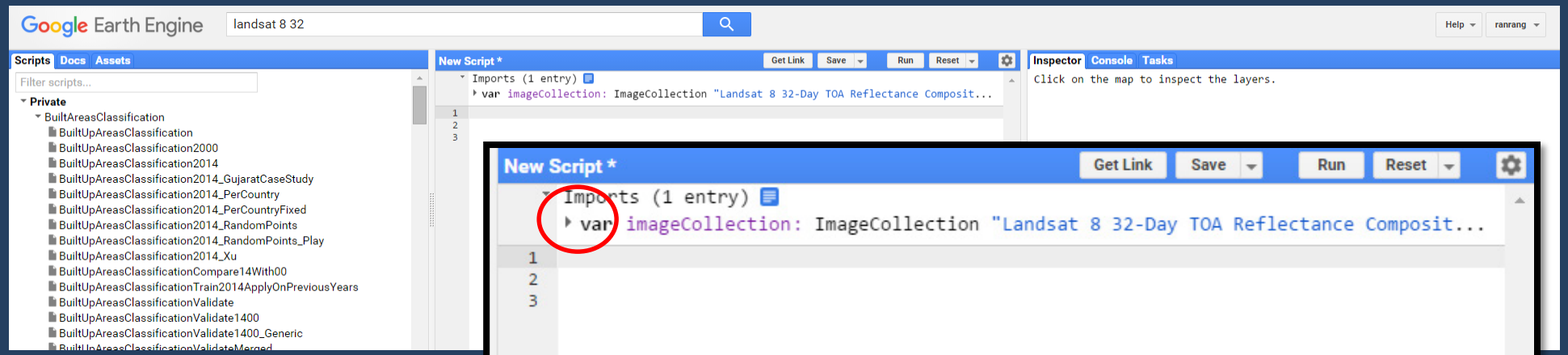
“Calling” Image Collections
Option 1



Searching and importing images

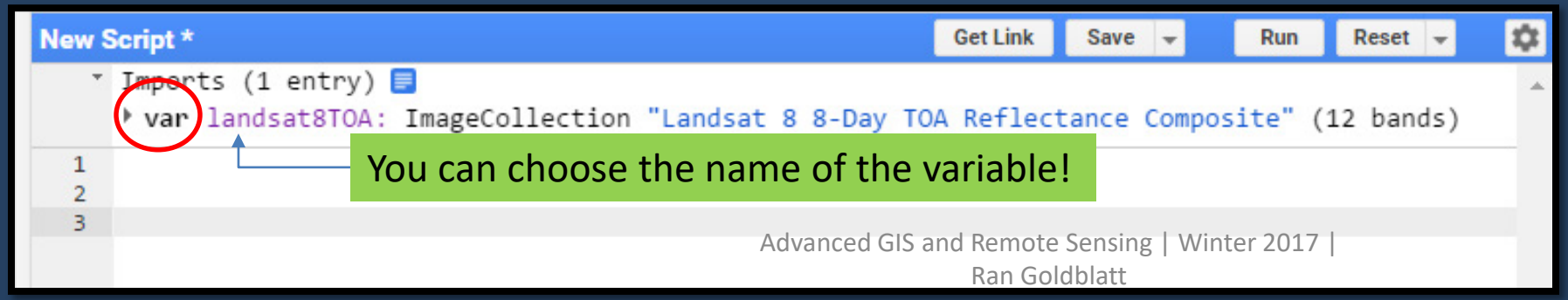
One method to access images and image collections:
“search” and import the dataset. No need to know the ID.


Landsat 7 annual TOA percentile composites
These 1-year composites were created from all Landsat 7 images in the specified composite period, excluding images marked with a negative sun elevation in their metadata. The composites were created using the ee.Algorithms.Landsat.simpleComposite() method with its default settings. Reflectance ([0,1]) in bands B1, B2, B3, B4, B5, and B7 is scaled to 8 bits ([0,255]) and temperature in band B6_VCID_2 is converted to units of Kelvin-100.
Data availability (time)
Jan 1, 1999 - Dec 31, 2014
Provider
Google
Tags
landsat, usgs, annual, global, l7, le7, etm, toa, percentile
ImageCollection ID
LANDSAT/LE7_TOA_1YEAR
Import
Google Earth Engine
landsat 7 annual
Scripts
Docs
Assets
Filter scripts...
Private
AdvancedGISCourse
BrazilMinesForestCover
BuiltAreasClassification
BuiltUpAreasClassification
BuiltUpAreasClassification2000
BuiltUpAreasClassification2014
BuiltUpAreasClassification2014_PerCountryFixed
BuiltUpAreasClassification2014_PerCountryFixedApplyPreviousYear
BuiltUpAreasClassification2014_PerCountryFixedWithConfProb
BuiltUpAreasClassification2014_RandomPoints
BuiltUpAreasClassification2014_RandomPoints_Play
BuiltUpAreasClassification2014_Xu
BuiltUpAreasClassificationCompare14With00
BuiltUpAreasClassificationTrain2014ApplyOnPreviousYears
BuiltUpAreasClassificationValidate
PLACES
RASTERS
Landsat 7 Annual TOA Reflectance Composite
Landsat 7 Annual NDVI Composite
Landsat 7 Annual Raw Composite
Landsat 7 Annual EVI Composite
Landsat 7 Annual NDWI Composite
Landsat 7 Annual BAI Composite
Landsat 7 Annual Greenest-Pixel TOA Reflectance Composite
more »
TABLES
“Import” will add the image to the workspace
This image collection has a name:
LANDSAT/LE7_TOA_1YEAR
Map
Satellite
United States
Mexico
Google



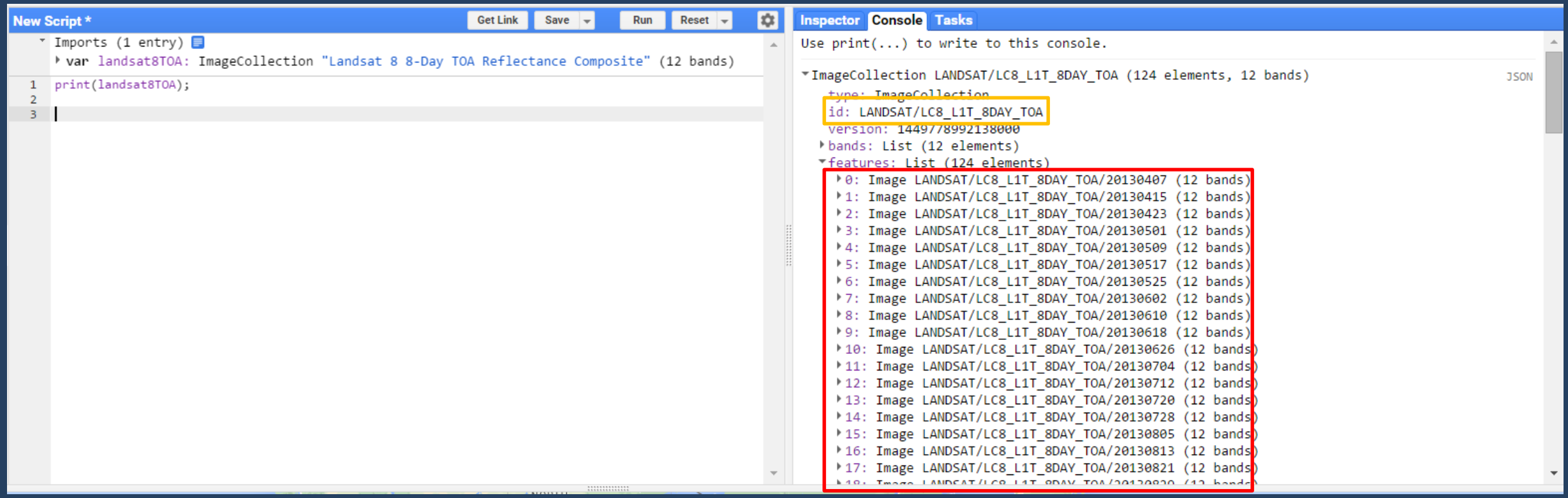
Variables

- Variables store values that make the code more readable.
- Must begin with a letter (or a $ or _)
- They are CASE-sensitive
- The first time you use a given variable, start with the word **var**.



The print() statement can be used to get information about the added image collection:

```
print(landsat8TOA);
```



This image collection has a name: LANDSAT/LC8_L1T_8DAY_TOA

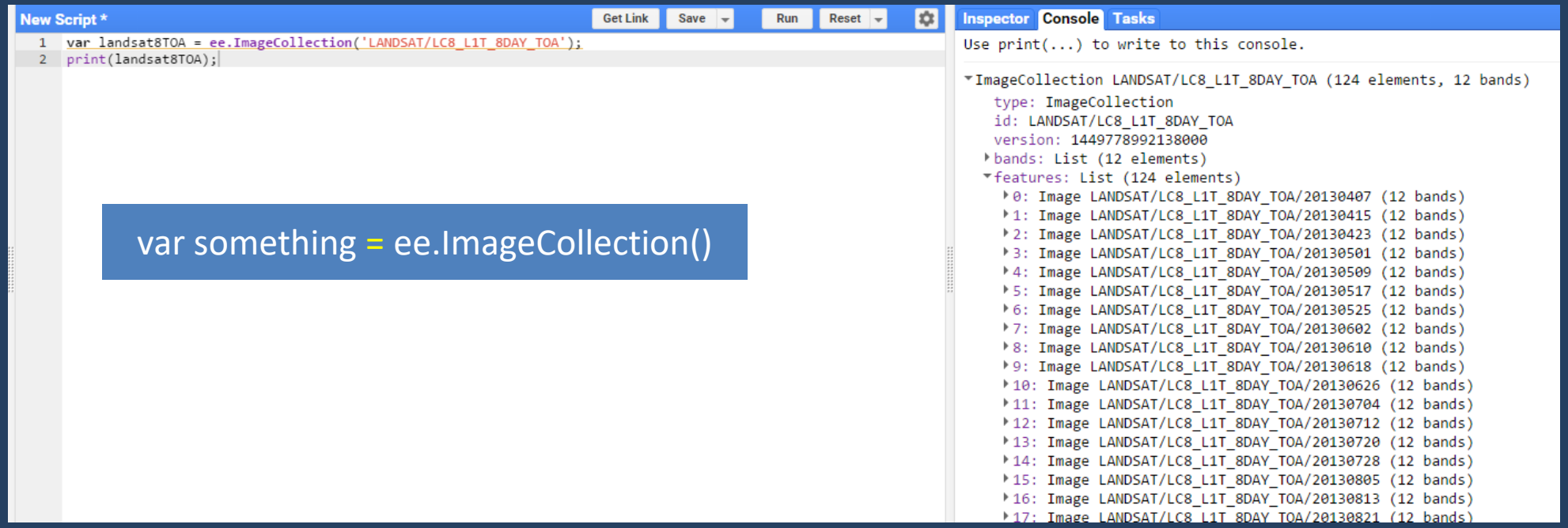
Another method to add images and image collections:

ee.ImageCollection()

This function “tells” EE that a given ID is a name of an image collection:

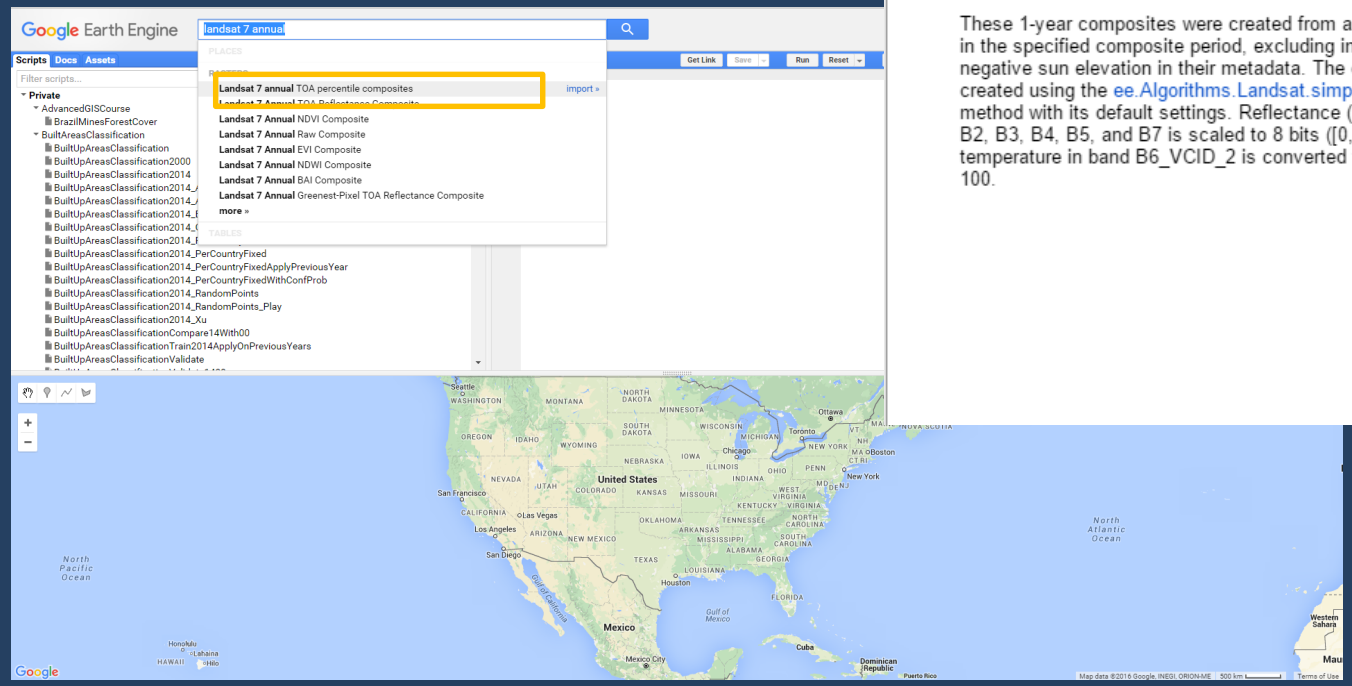
LANDSAT/LC8_L1T_8DAY_TOA is an Image collection- Access it as a parameter.

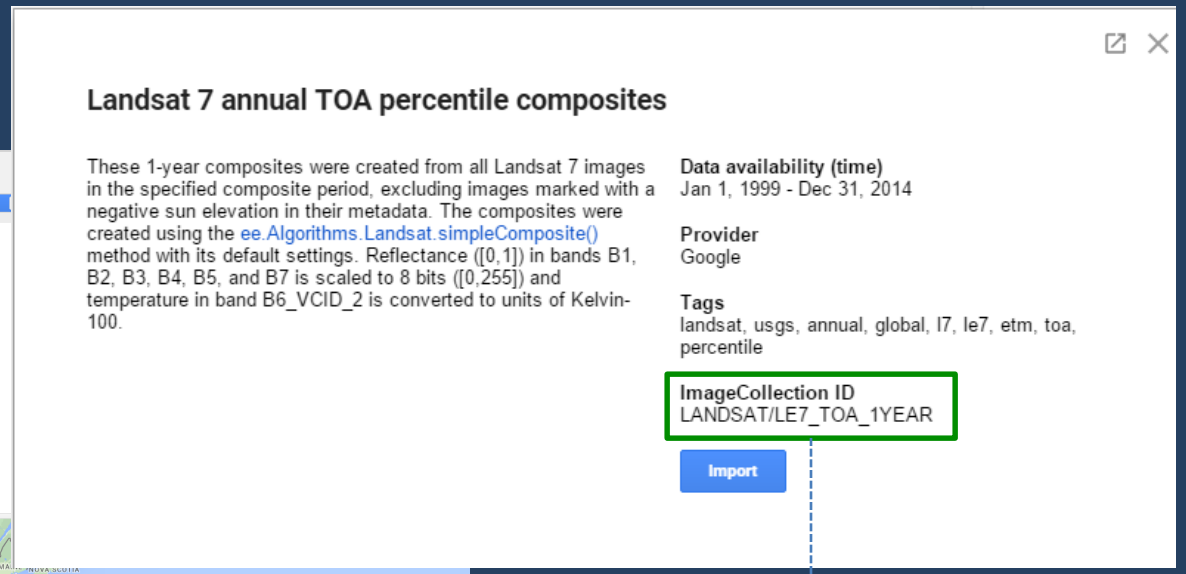
ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA'**)**




```
var landsat8TOA = ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA');
print(landsat8TOA);
```

Let`s search for another image collection:






```
var landsat7TOA = ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
print(landsat7TOA);
```

New Script *

Get Link | Save | Run | Reset

```
1  var landsat7TOA = ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
2  print(landsat7TOA);
```

var landsat7TOA =
ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
print(landsat7TOA);

Inspector | Console | Tasks

```
Use print(...) to write to this console.

▼ ImageCollection LANDSAT/LE7_TOA_1YEAR (16 elements, 7 bands)
    type: ImageCollection
    id: LANDSAT/LE7_TOA_1YEAR
    version: 1449779677092000
  ▸ bands: List (7 elements)
  ▼ features: List (16 elements)
    ▸ 0: Image LANDSAT/LE7_TOA_1YEAR/1999 (7 bands)
    ▸ 1: Image LANDSAT/LE7_TOA_1YEAR/2000 (7 bands)
    ▸ 2: Image LANDSAT/LE7_TOA_1YEAR/2001 (7 bands)
    ▸ 3: Image LANDSAT/LE7_TOA_1YEAR/2002 (7 bands)
    ▸ 4: Image LANDSAT/LE7_TOA_1YEAR/2003 (7 bands)
    ▸ 5: Image LANDSAT/LE7_TOA_1YEAR/2004 (7 bands)
    ▸ 6: Image LANDSAT/LE7_TOA_1YEAR/2005 (7 bands)
    ▸ 7: Image LANDSAT/LE7_TOA_1YEAR/2006 (7 bands)
    ▸ 8: Image LANDSAT/LE7_TOA_1YEAR/2007 (7 bands)
    ▸ 9: Image LANDSAT/LE7_TOA_1YEAR/2008 (7 bands)
    ▸ 10: Image LANDSAT/LE7_TOA_1YEAR/2009 (7 bands)
    ▸ 11: Image LANDSAT/LE7_TOA_1YEAR/2010 (7 bands)
    ▸ 12: Image LANDSAT/LE7_TOA_1YEAR/2011 (7 bands)
    ▸ 13: Image LANDSAT/LE7_TOA_1YEAR/2012 (7 bands)
    ▸ 14: Image LANDSAT/LE7_TOA_1YEAR/2013 (7 bands)
    ▼ 15: Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)
        type: Image
        id: LANDSAT/LE7_TOA_1YEAR/2014
        version: 1423795095111000
      ▸ bands: List (7 elements)
      ▸ properties: Object (1 property)
  ▸ properties: Object (21 properties)
```

The Image collection includes features → Images

Every image in Earth Engine has a name. Example:

LANDSAT/LE7_TOA_1YEAR/2014

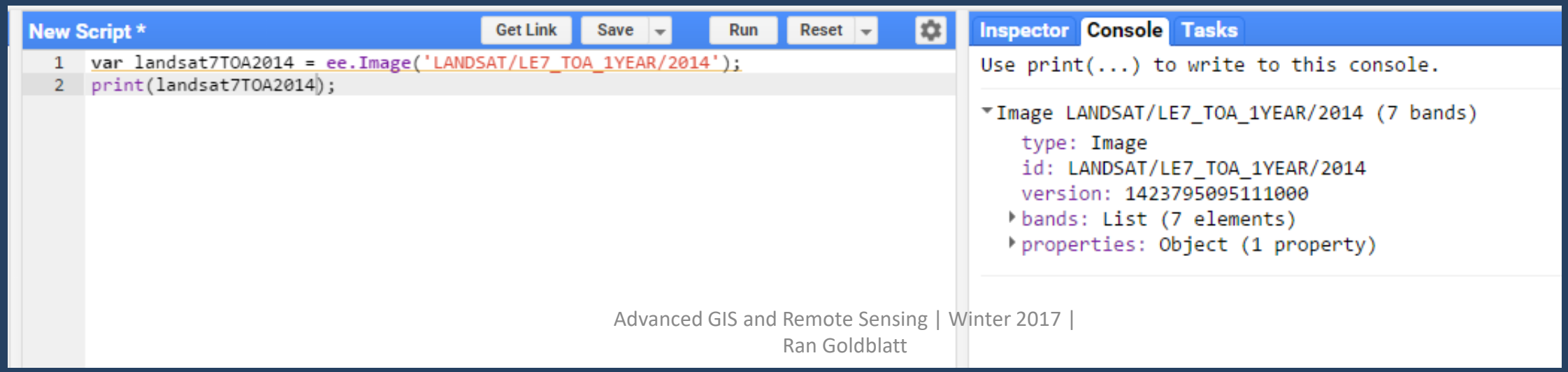
This is an image

The image ID that we want to call: LANDSAT/LE7_TOA_1YEAR/2014

We need to explicitly tell Earth Engine that LANDSAT/LE7_TOA_1YEAR/2014 is an image.

We will use the function: **ee.Image()**
"Take the name of the image and access it as a parameter".

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
print(landsat7TOA2014);
```

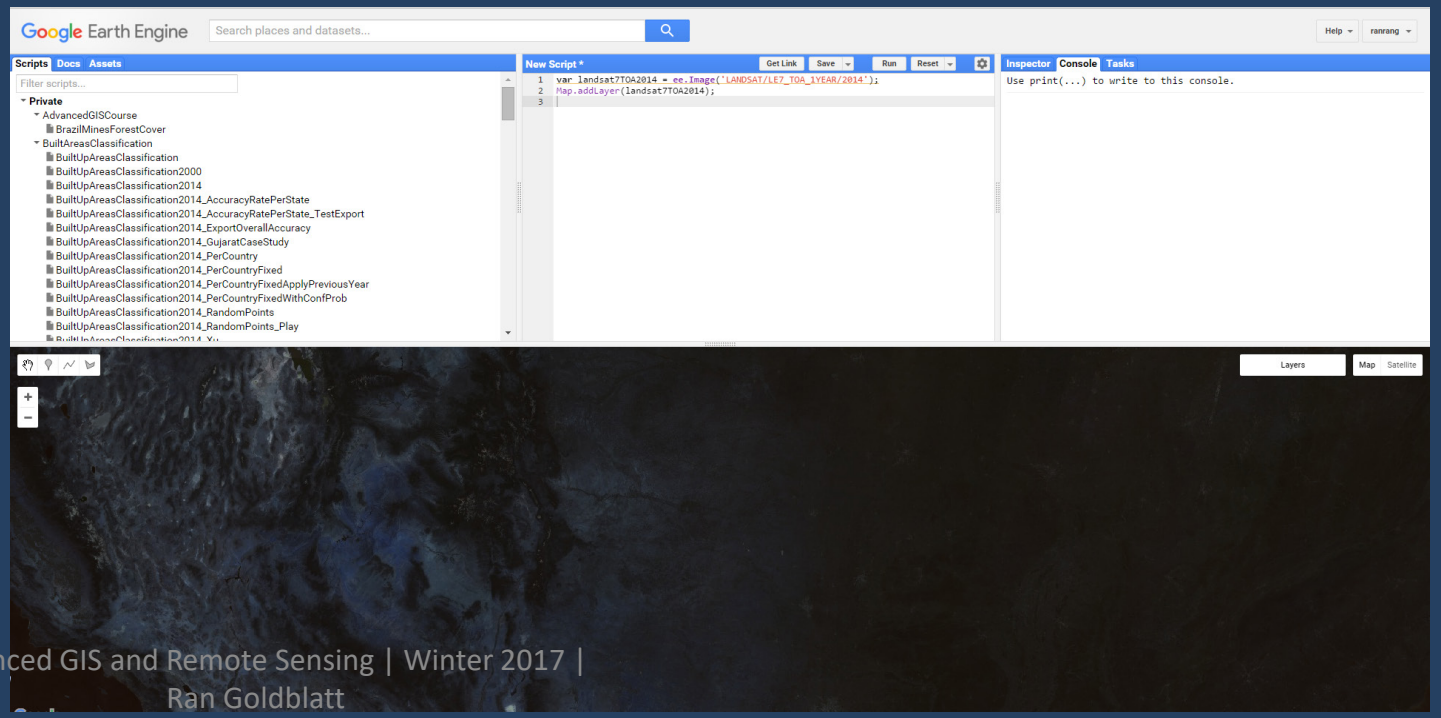


Adding an image to the map:

Map.addLayer(something);

This statement says that we want the **Map.addLayer()** function to add something to the map.

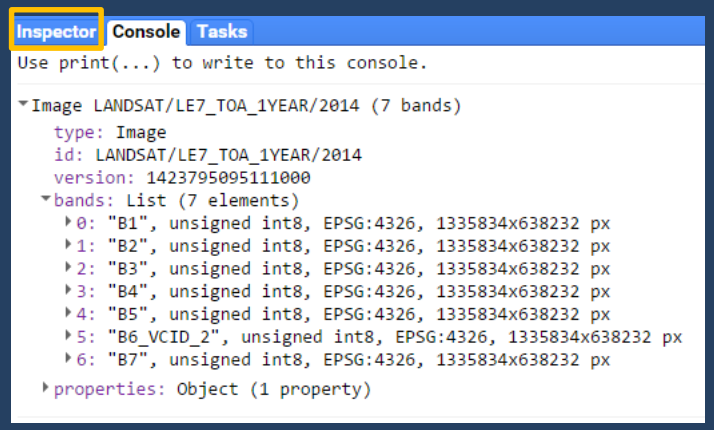
```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
Map.addLayer(landsat7TOA2014);
```

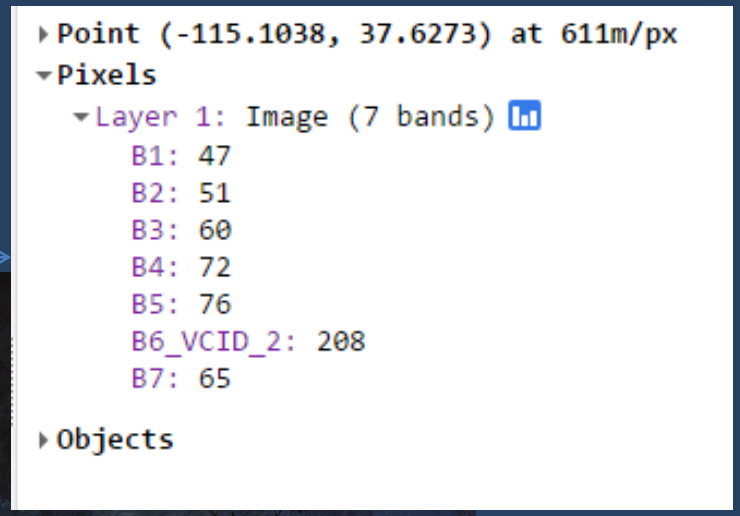


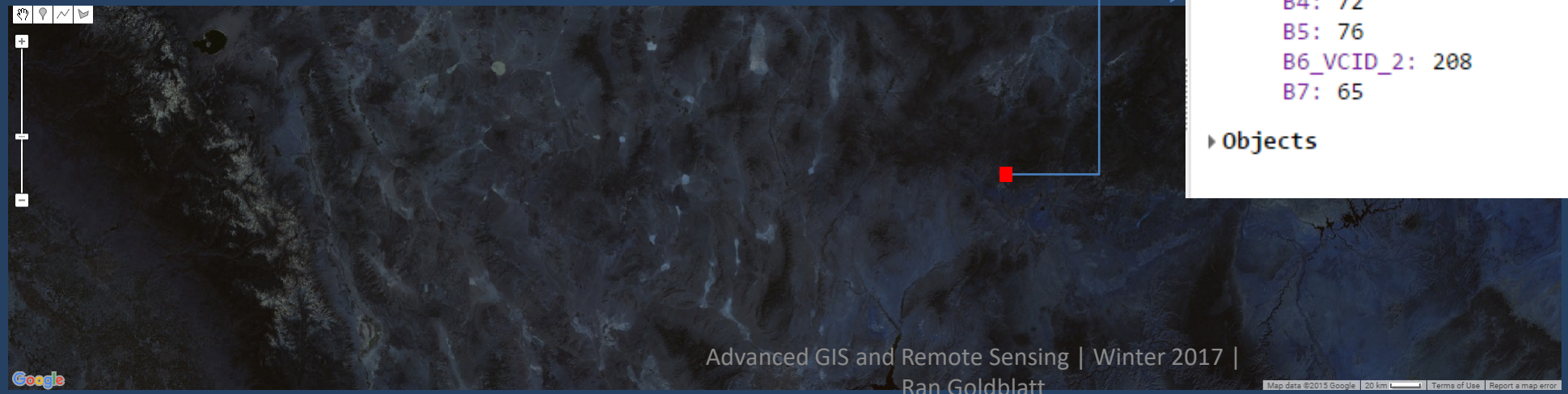
What is wrong with the visualization?

Many images in Earth Engine have one or more bands.
For example, images from the Landsat 7 satellite include:

Band 1 - blue
Band 2 - green
Band 3 - red
Band 4 - Near Infrared
Band 5 - Short-wave Infrared
Band 6 - Thermal Infrared
Band 7 - Short-wave Infrared







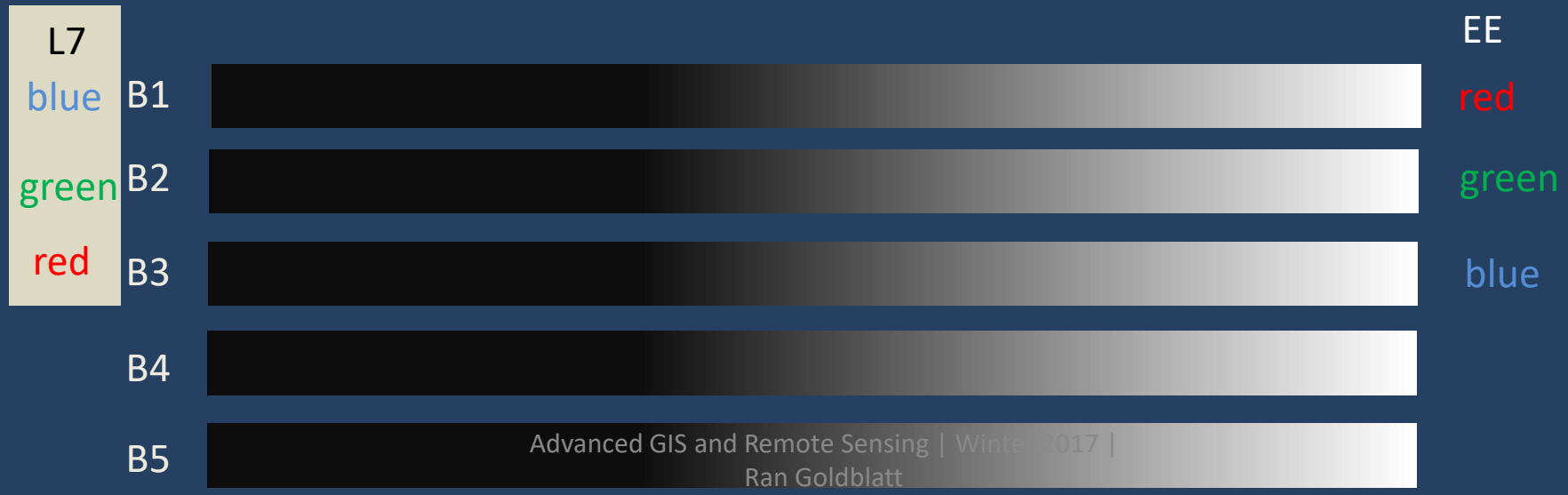
How does EE map the values in the image bands to colors on the map?

One band: value 0 is rendered as black and value 255 as white, with a linear gray gradient in between.

0 255

More than one band:

EE takes the first three bands and maps the first to red, the second to green, and the third to blue. The value of the pixel determines the intensity of the color.



The Map.addLayer() function can take a second parameter {} which describes how the image should be visualized.

```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014'),
{}
);
```

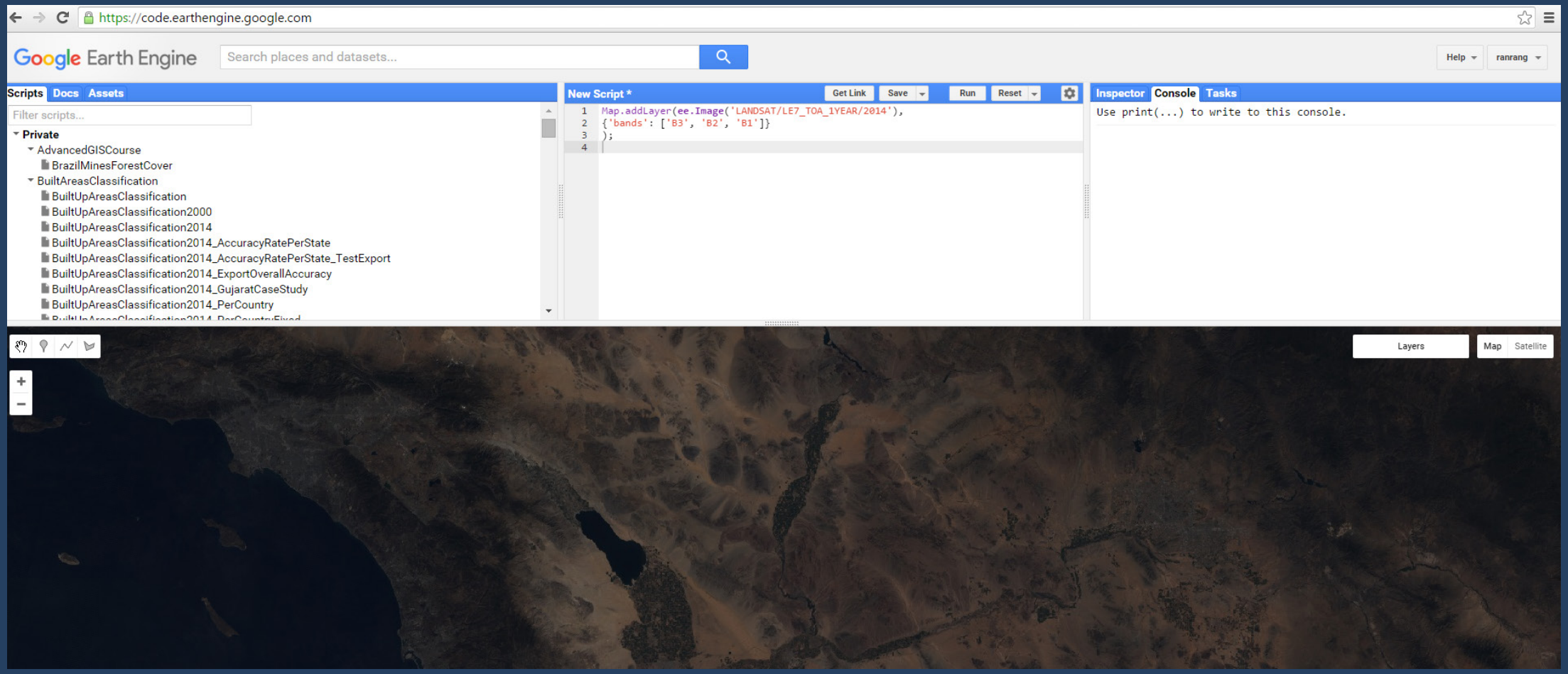
The format of this second parameter (visualization factor) is:
{'string': value, 'string': value, ... , 'string': value}

each string describes one visualization factor:
For example: 'min', 'max', 'bands'.

We want to add a *list* of bands to visualize:

```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014'),
{'bands': ['B3', 'B2', 'B1']}
);
```

A visualization of a single Landsat image in Earth Engine.

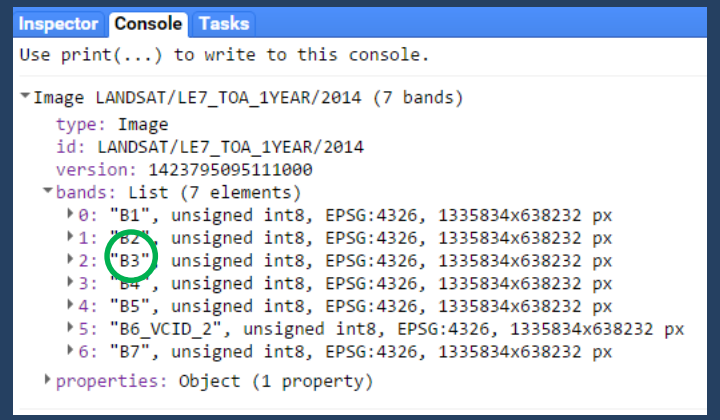



```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014'),
{'bands': ['B3', 'B2', 'B1']}
);
```

	R	G	B
L7	B3	B2	B1
L8	B4	B3	B2

Selecting specific band(s) in an Image

.select('band name')

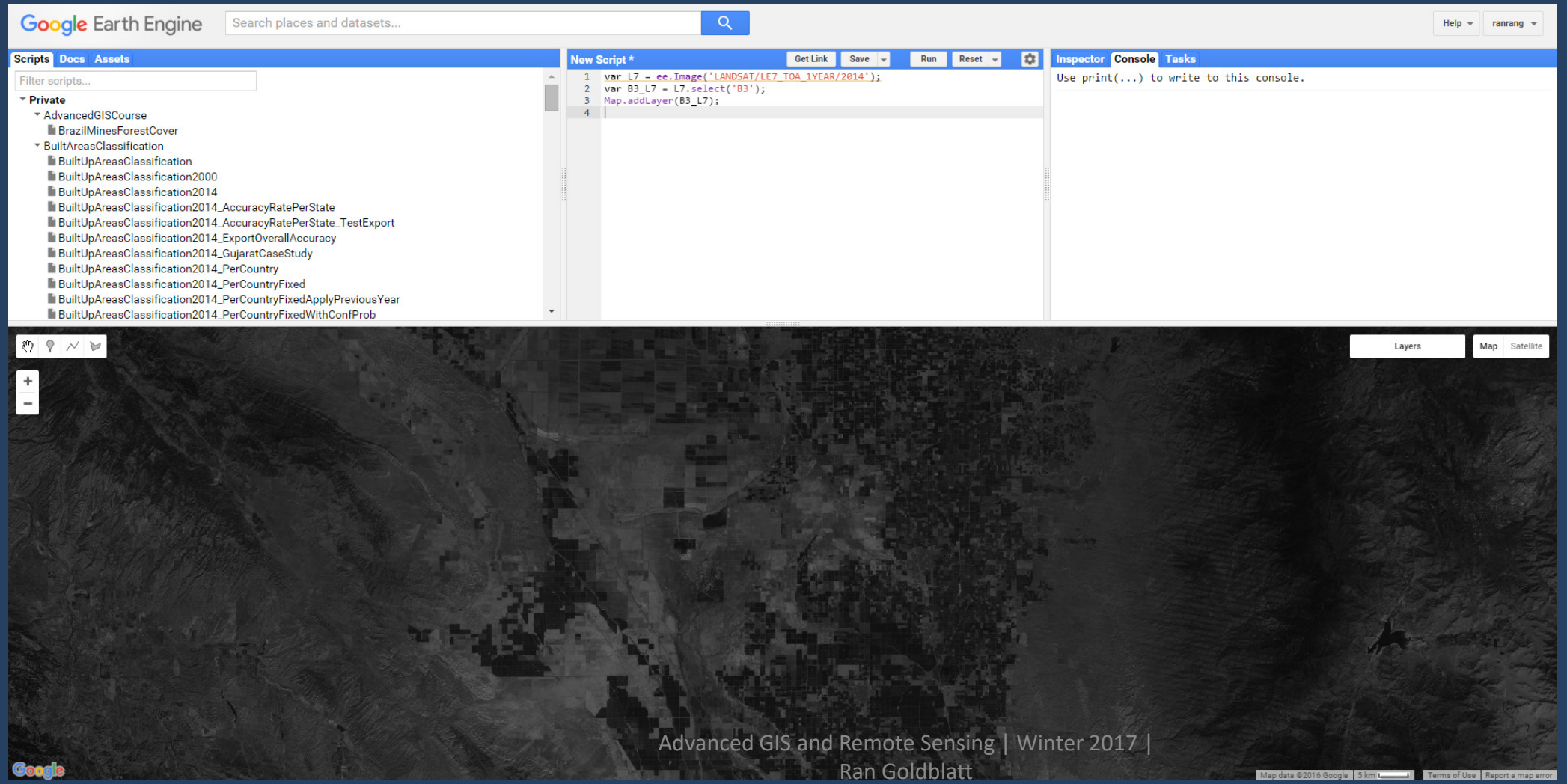



```
Map.addLayer(ee.Image('LANDSAT/LE7_TOA_1YEAR/2014')
.select('B3'));
```

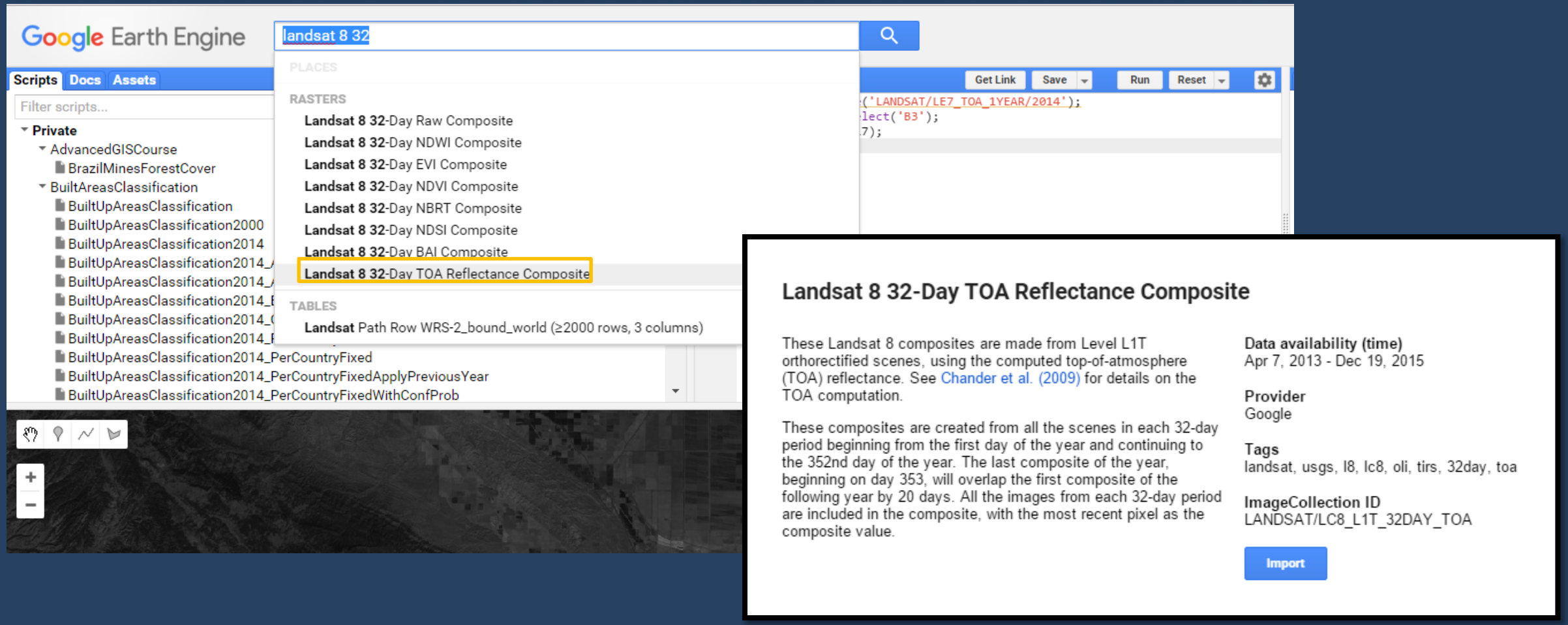
Which is the same as:

```
var L7 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
var B3_L7 = L7.select('B3');
Map.addLayer(B3_L7);
```

```
var L7 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
var B3_L7 = L7.select('B3');
Map.addLayer(B3_L7);
```

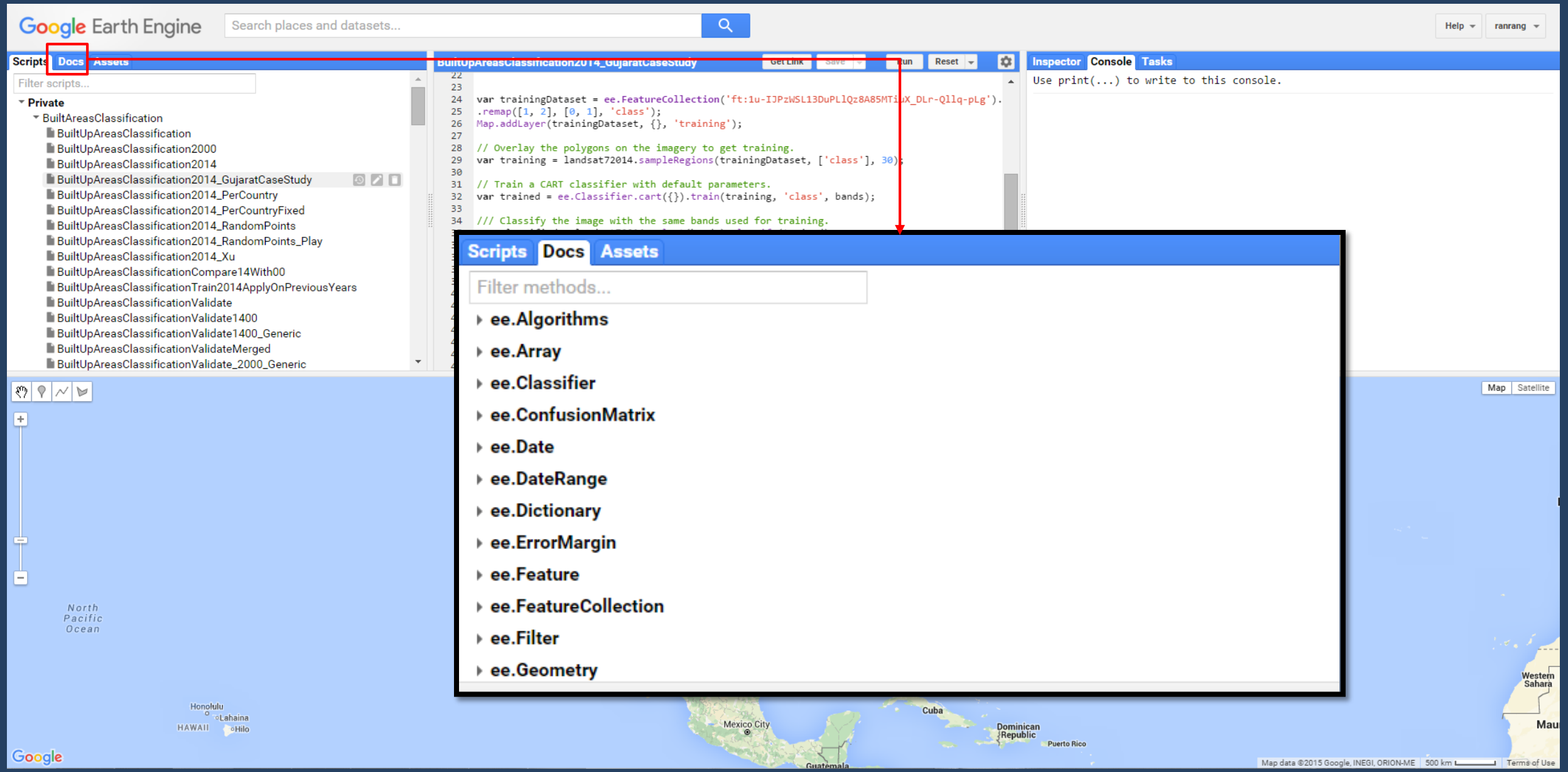


Let`s call another image collection:

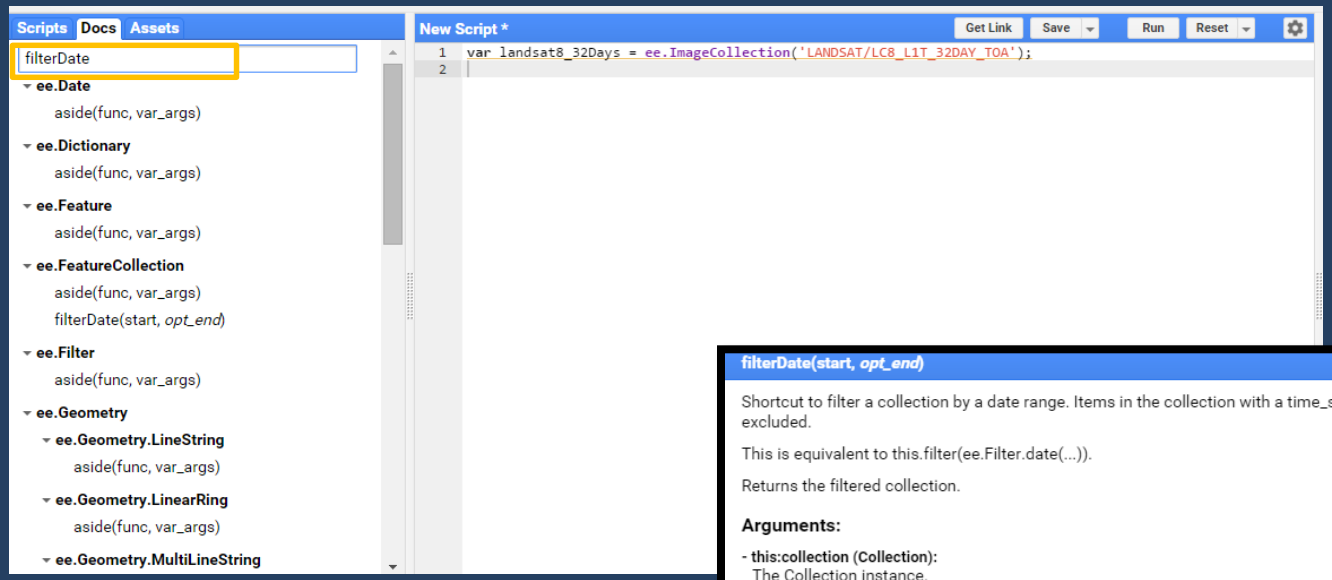


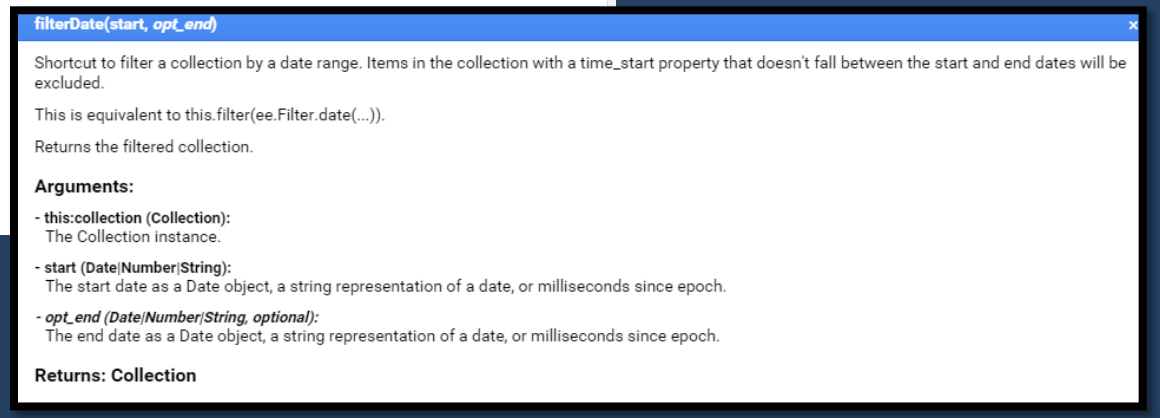
Task:

Filter an image collection only for scenes in a specific date (second half of 2014)


Google Earth Engine
Search places and datasets...
Help
ranrang
Scripts
Docs
Assets
Filter scripts...
Private
BuiltAreasClassification
BuiltUpAreasClassification
BuiltUpAreasClassification2000
BuiltUpAreasClassification2014
BuiltUpAreasClassification2014_GujaratCaseStudy
BuiltUpAreasClassification2014_PerCountry
BuiltUpAreasClassification2014_PerCountryFixed
BuiltUpAreasClassification2014_RandomPoints
BuiltUpAreasClassification2014_RandomPoints_Play
BuiltUpAreasClassification2014_Xu
BuiltUpAreasClassificationCompare14With00
BuiltUpAreasClassificationTrain2014ApplyOnPreviousYears
BuiltUpAreasClassificationValidate
BuiltUpAreasClassificationValidate1400
BuiltUpAreasClassificationValidate1400_Generic
BuiltUpAreasClassificationValidateMerged
BuiltUpAreasClassificationValidate_2000_Generic
BuiltUpAreasClassification2014_GujaratCaseStudy
Get Link
Save
Run
Reset
var trainingDataset = ee.FeatureCollection('ft:1u-IJPzWSL13DuPLlQz8A85MTiuX_DLr-Qllq-pLg').
.remap([1, 2], [0, 1], 'class');
Map.addLayer(trainingDataset, {}, 'training');
// Overlay the polygons on the imagery to get training.
var training = landsat72014.sampleRegions(trainingDataset, ['class'], 30);
// Train a CART classifier with default parameters.
var trained = ee.Classifier.cart({}).train(training, 'class', bands);
/// Classify the image with the same bands used for training.
Inspector
Console
Tasks
Use print(...) to write to this console.
Scripts
Docs
Assets
Filter methods...
ee.Algorithms
ee.Array
ee.Classifier
ee.ConfusionMatrix
ee.Date
ee.DateRange
ee.Dictionary
ee.ErrorMargin
ee.Feature
ee.FeatureCollection
ee.Filter
ee.Geometry
Map
Satellite
North Pacific Ocean
Honolulu
Lahaina
Hilo
HAWAII
Mexico City
Guatemala
Cuba
Dominican Republic
Puerto Rico
Western Sahara
Google
Map data ©2015 Google, INEGI, ORION-ME
500 km
Terms of Use

.filterDate()





filterDate() *method* on an image collection.

Each action on an image collection is a *method*.

A method is applied on an image collection by .methodName(arguments) after the image collection

```
ee.imageCollection.filterDate(date1, date2)
```

'2014-06-01', '2014-12-31'

'YYYY-MM-DD', 'YYYY-MM-DD'

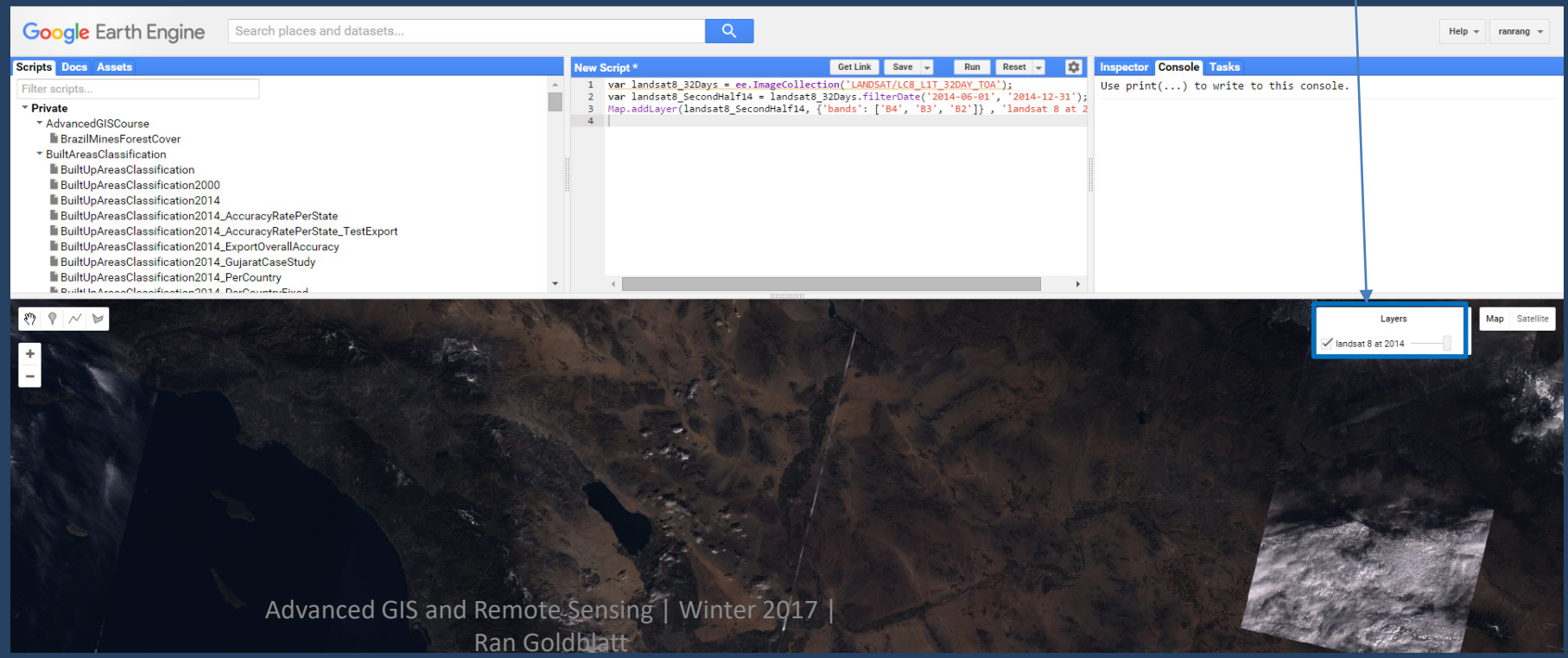
```
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA')
.filterDate('2014-06-01', '2014-12-31');
```

Which is the same as:

```
var L8 = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');

var L8_SecondHalf = L8.filterDate('2014-06-01', '2014-12-31');
```

All together:

```
var L8 = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');

var L8_SecondHalf14 = L8.filterDate('2014-06-01', '2014-12-31');
Map.addLayer(L8_SecondHalf14, {'bands': ['B4', 'B3', 'B2']} , 'landsat 8 at 2014' );
```



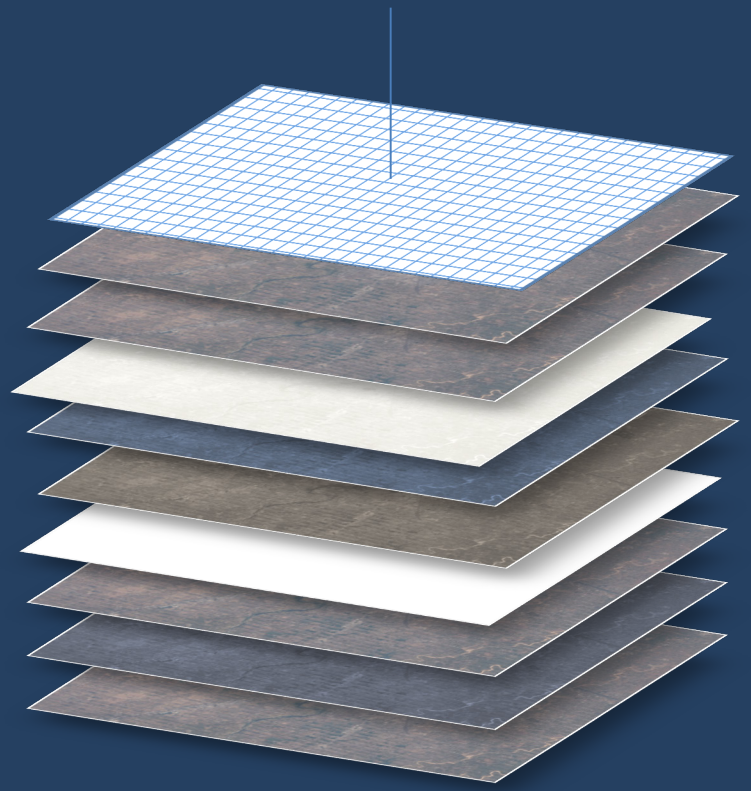
Reducers

- Landsat 8 visits the same spot on the Earth every 16 days.
- Over a 6 month period, there will be approximately 12 images.
- Each pixel on the map is derived from a stack of pixels
  (a pixel from each image in the collection).
- The default behavior is to select the most recent pixel- the one from the recent scene in the stack.
- We can alter this behavior by using a **reducer**

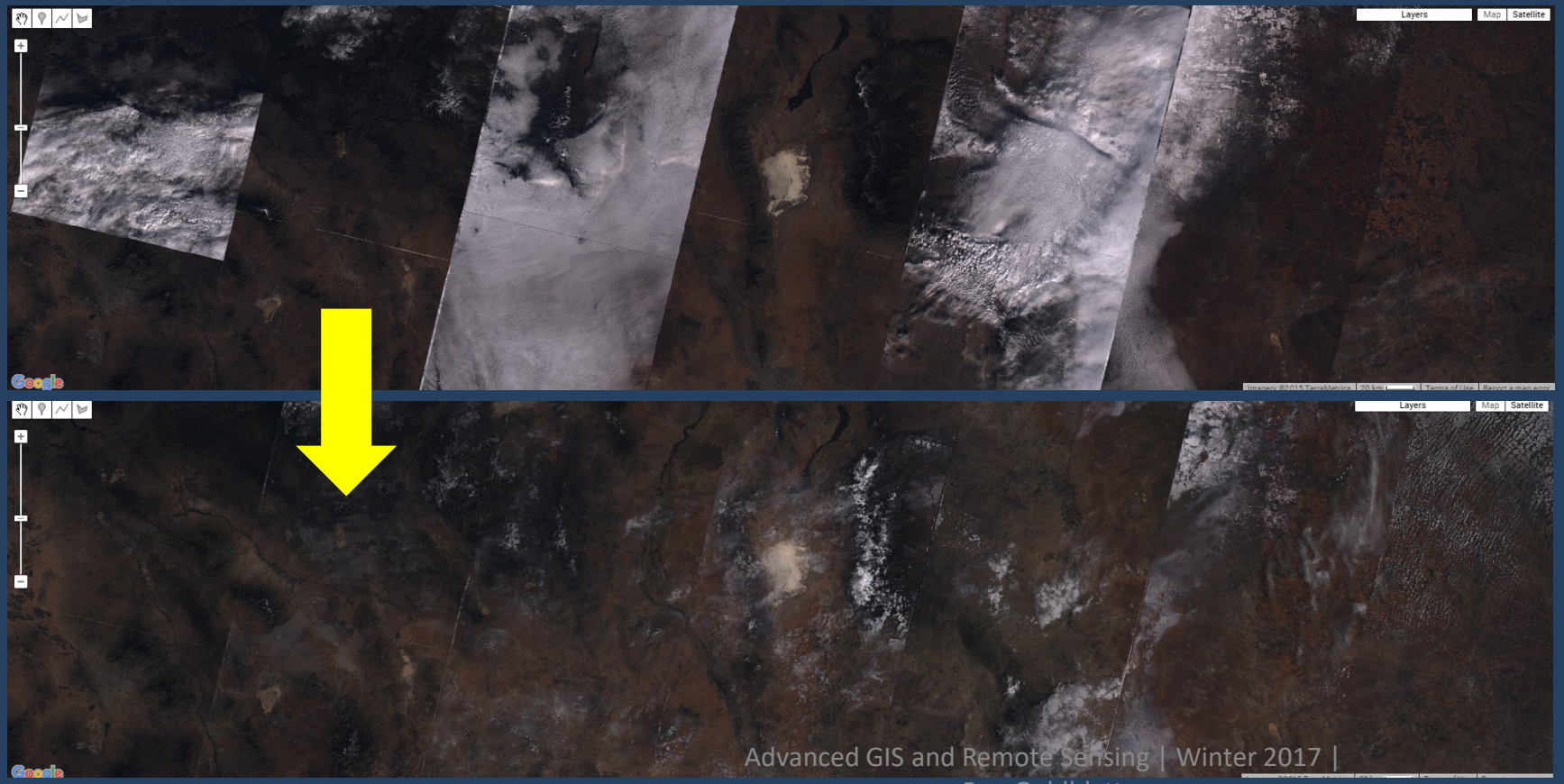
For example: pick the median value in the stack.

Reducers

```
var imageCollection = ee.ImageCollection()

var minVal = imageCollection.min()
var maxVal = imageCollection.max()
var meanVal = imageCollection.mean ()
var medianVal = imageCollection.median ()
var firstVal = imageCollection.first ()
var sumVal = imageCollection.sum ()
```



```
var L8_32Days = ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');
var L8_SecondHalf14 = L8_32Days
.filterDate('2014-06-01', '2014-12-31').median();
Map.addLayer(L8_SecondHalf14, {'bands': ['B4', 'B3', 'B2']} , 'landsat 8 at 2014' );
```



Creating a TOA calibrated composite of “raw” Landsat scenes

USGS Landsat 8 Raw Scenes (Orthorectified)

Landsat 8 DN values, representing scaled, calibrated at-sensor radiance, orthorectified scenes only.

This dataset contains the following bands:

- B1: Coastal aerosol (0.43 - 0.45 µm)
- B2: Blue (0.45 - 0.51 µm)
- B3: Green (0.53 - 0.59 µm)
- B4: Red (0.64 - 0.67 µm)
- B5: Near Infrared (0.85 - 0.88 µm)
- B6: Short-wave Infrared 1 (1.57 - 1.65 µm)
- B7: Short-wave infrared 2 (2.11 - 2.29 µm)
- B8: Panchromatic (0.50 - 0.68 µm)
- B9: Cirrus (1.36 - 1.38 µm)
- B10: Thermal Infrared 1 (10.60 - 11.19 µm)
- B11: Thermal Infrared 2 (11.50 - 12.51 µm)
- BQA: Data quality assessment band

The panchromatic band B8 has a spatial resolution of 15 meters per pixel, and all other bands have a spatial resolution of 30 meters per pixel, although the thermal bands B10 and B11 are generated from sensor data whose native resolution is 100 meters per pixel.

For more information, see Landsat 8 Handbook

Data availability (time)
Apr 11, 2013 - Nov 21, 2016

Provider
USGS

Tags
landsat, usgs, l8, lc8, oli, tirs, radiance

ImageCollection ID
LANDSAT/LC8_L1T

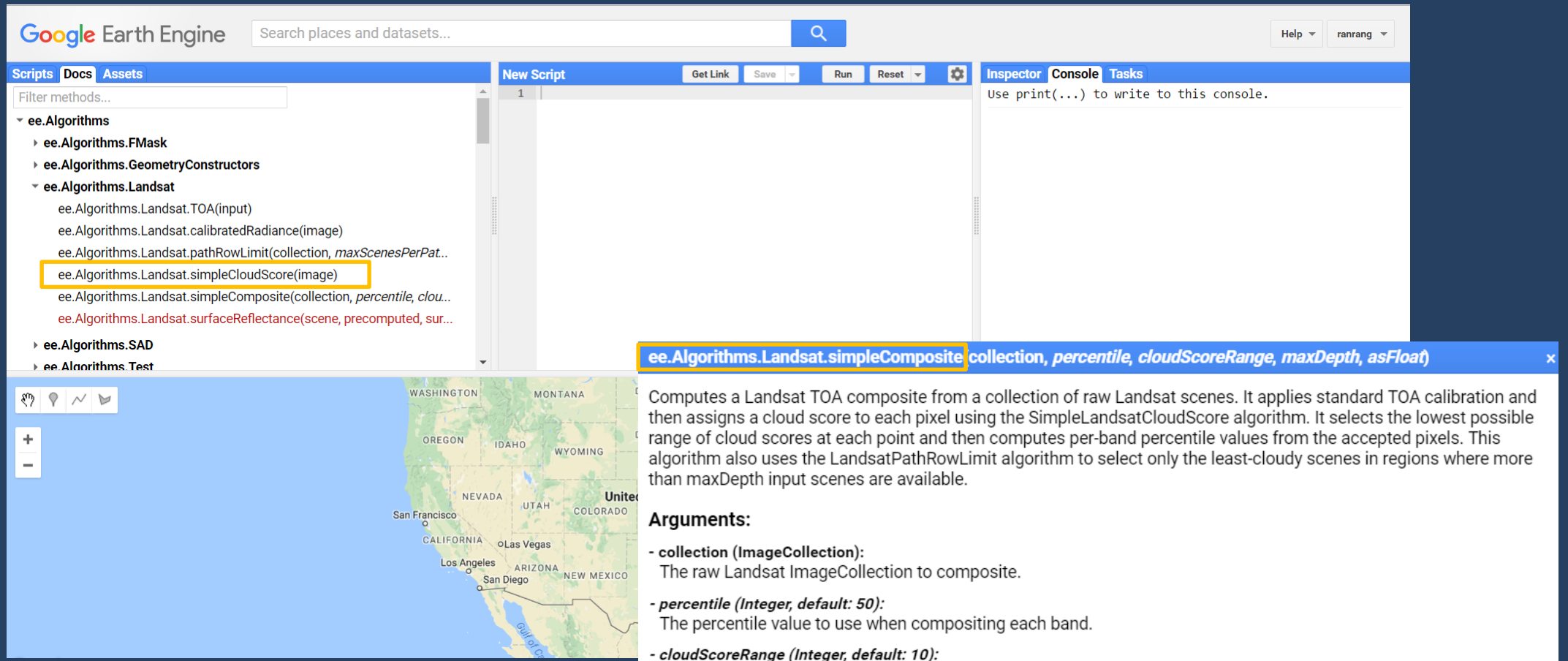
Import

Creating a composite of “raw” Landsat scenes:

1. Take the collection of the raw scenes
2. Apply TOA calibration
3. Assign a cloud score to each pixel
4. Select the lowest possible range of cloud scores at each point
5. Compute per-band percentile values from the accepted pixels.

Reminder:
Top of Atmosphere Reflectance (%)
Consider the angle and distance between the sun, surface and satellite.

ee.Algorithms.Landsat.simpleComposite(raw image collection)



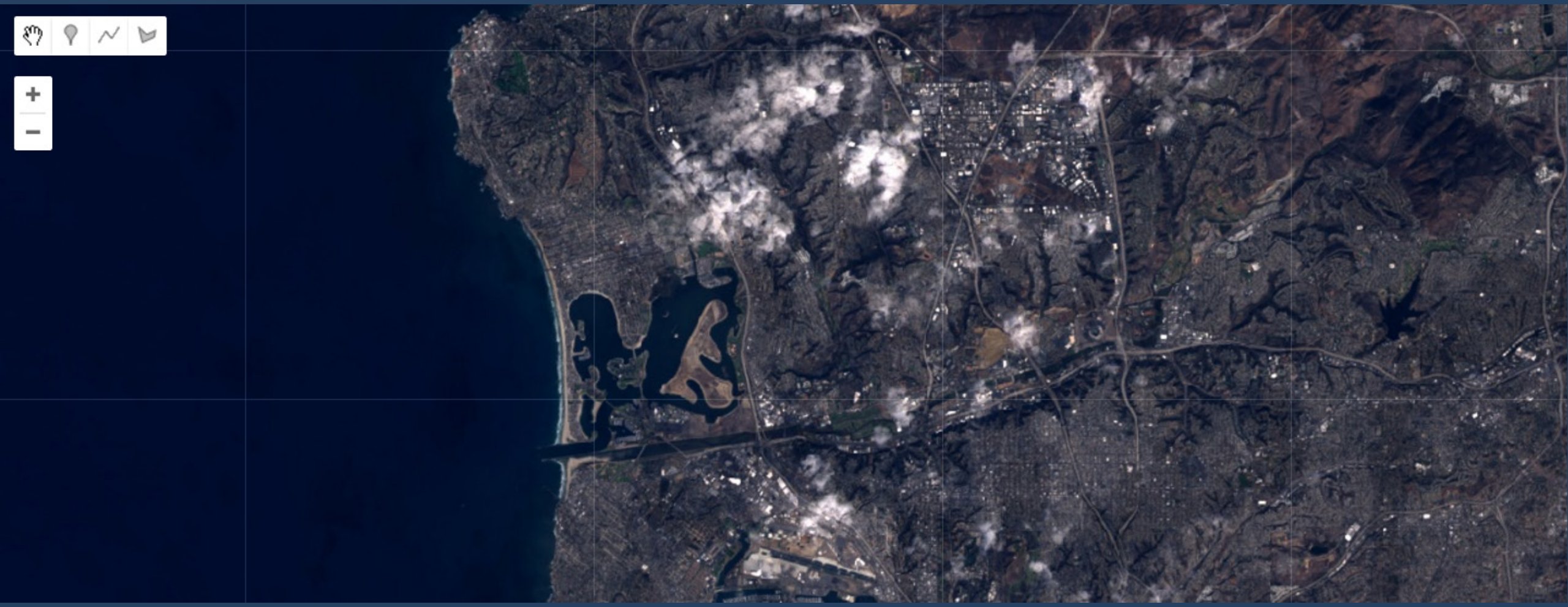
Simple composite arguments:

```
var image = ee.Algorithms.Landsat.simpleComposite(
{collection:   //The raw Landsat ImageCollection to composite.
percentile:   //The percentile value to use when compositing each band. default: 50
cloudScoreRange:   // Range of cloud scores to accept per pixel. default: 10
maxDepth:  //Maximum number of scenes used to compute each pixel. default: 40
asFloat: // Units of output bands (true: same as the Landsat.TOA algorithm; false: convert to uint8: multiply the reflective bands by 255, subtract 100 from the thermal bands, round to the nearest integer. default: false
}
)
```

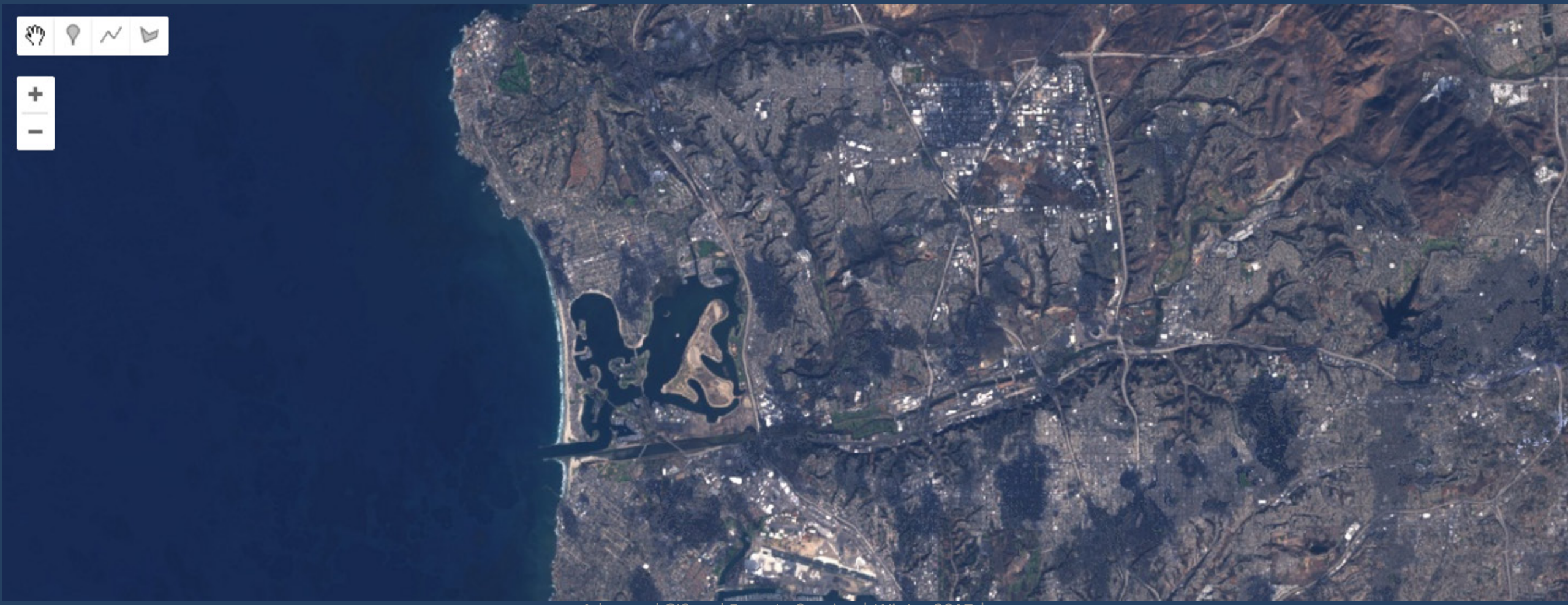
Example:

```
var L8_Collection = ee.ImageCollection('LANDSAT/LC8_L1T');

var L8Filtered = L8_Collection.filterDate('2014-10-01', '2014-12-31');

var L8Median = L8Filtered.median();

Map.addLayer(L8Median,  {'bands': ['B4', 'B3', 'B2'], min: 6000, max:17000}, 'landsat 8
median val' );

var l8_Composite= ee.Algorithms.Landsat.simpleComposite({
  collection: L8Filtered,
  asFloat: true
});

Map.addLayer(l8_Composite , {'bands': ['B4', 'B3', 'B2'], min: 0, max:0.3}, 'landsat 8
simple composite' );
```

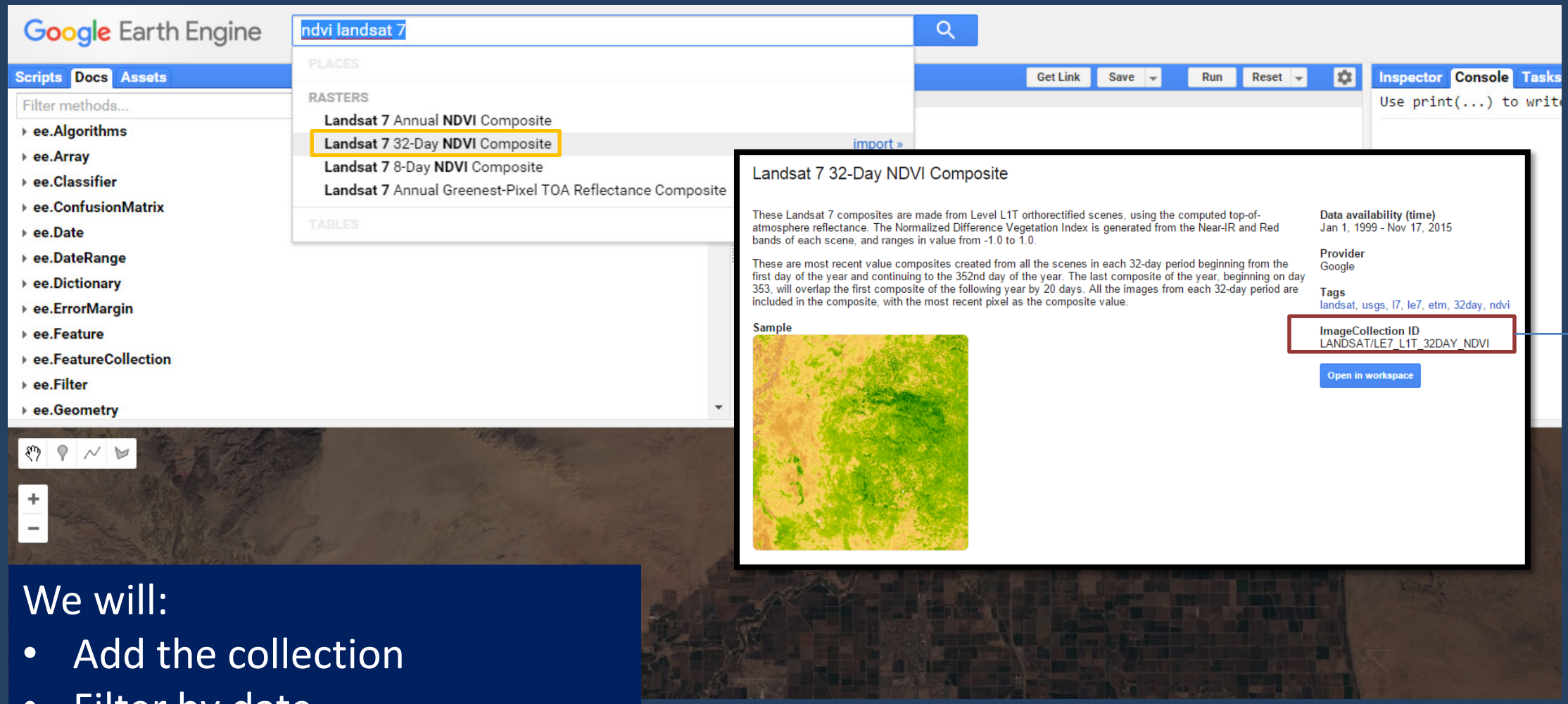
Before….



And after (simple composite)….



NDVI

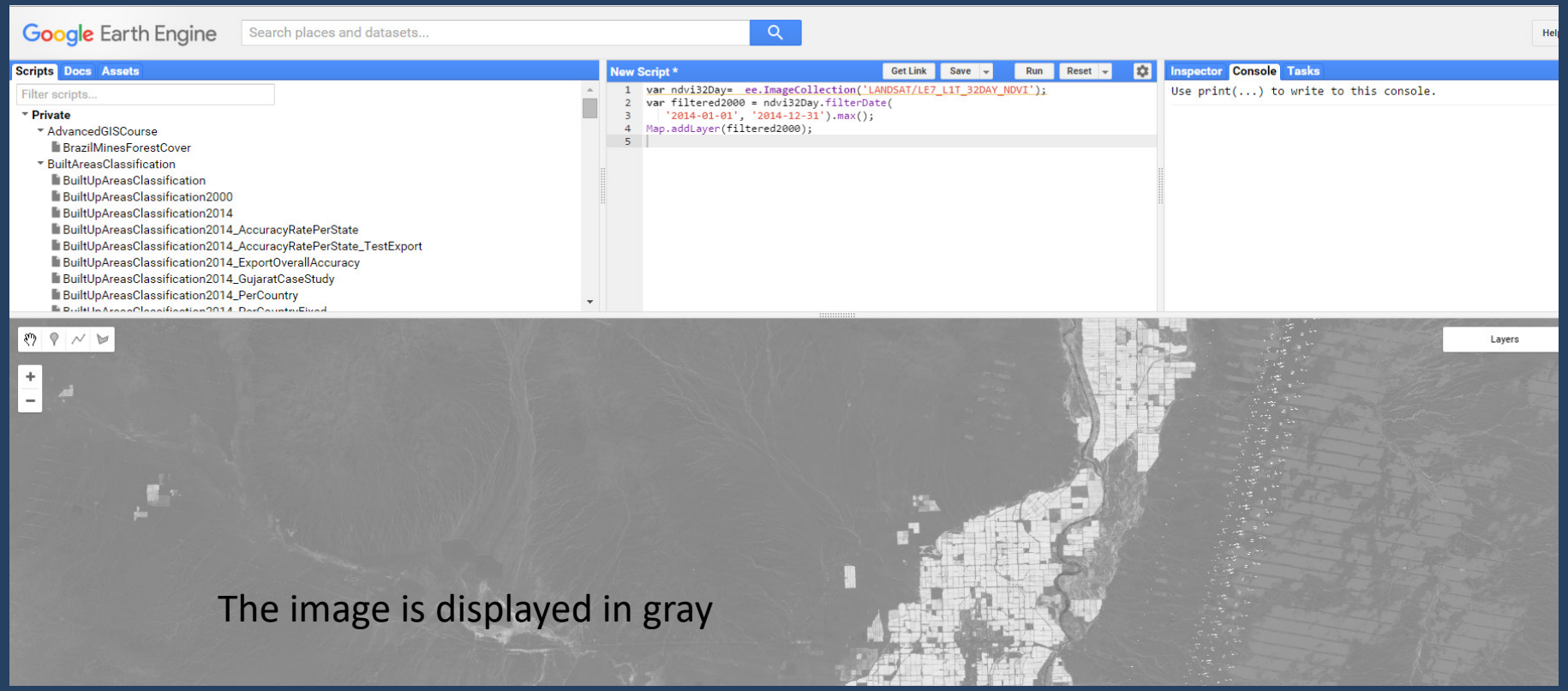


We will:

- Add the collection
- Filter by date
- Find the maximum ndvi value
- Visualize it.

LANDSAT/LE7_L1T_32DAY_NDVI

```
var ndvi32Day=  ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
var filtered2000 = ndvi32Day
.filterDate('2014-01-01', '2014-12-31').max();
Map.addLayer(filtered2000);
```



Palettes

- Palettes define the color scheme an image.

- A palette is a comma delimited string of colors, stretched over a band's maximum and minimum pixel values.

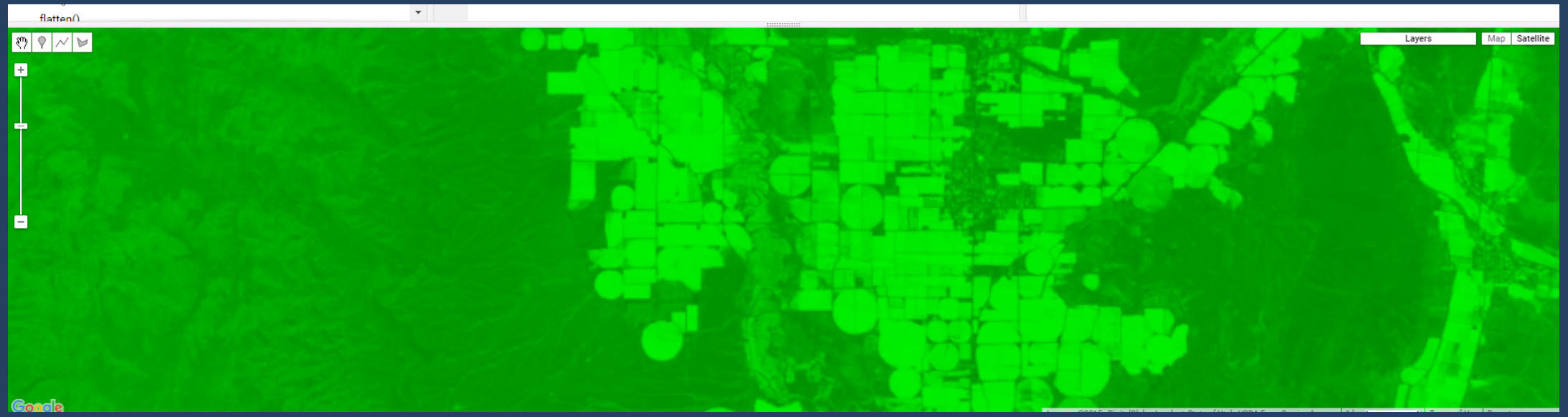
The string '000000' is black
'FFFFFF' is white
'FF0000' is red
'00FF00' is green
'0000FF' is blue.

http://www.w3schools.com/colors/colors_names.asp

We need to add a *visualization parameter* (just as we did with 'bands') that will specify a palette

```
var collection =  ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
var filtered2000 =
collection.filterDate('2014-01-01', '2014-12-31').max();
Map.addLayer(filtered2000, {'palette':'000000, 00FF00'});
```

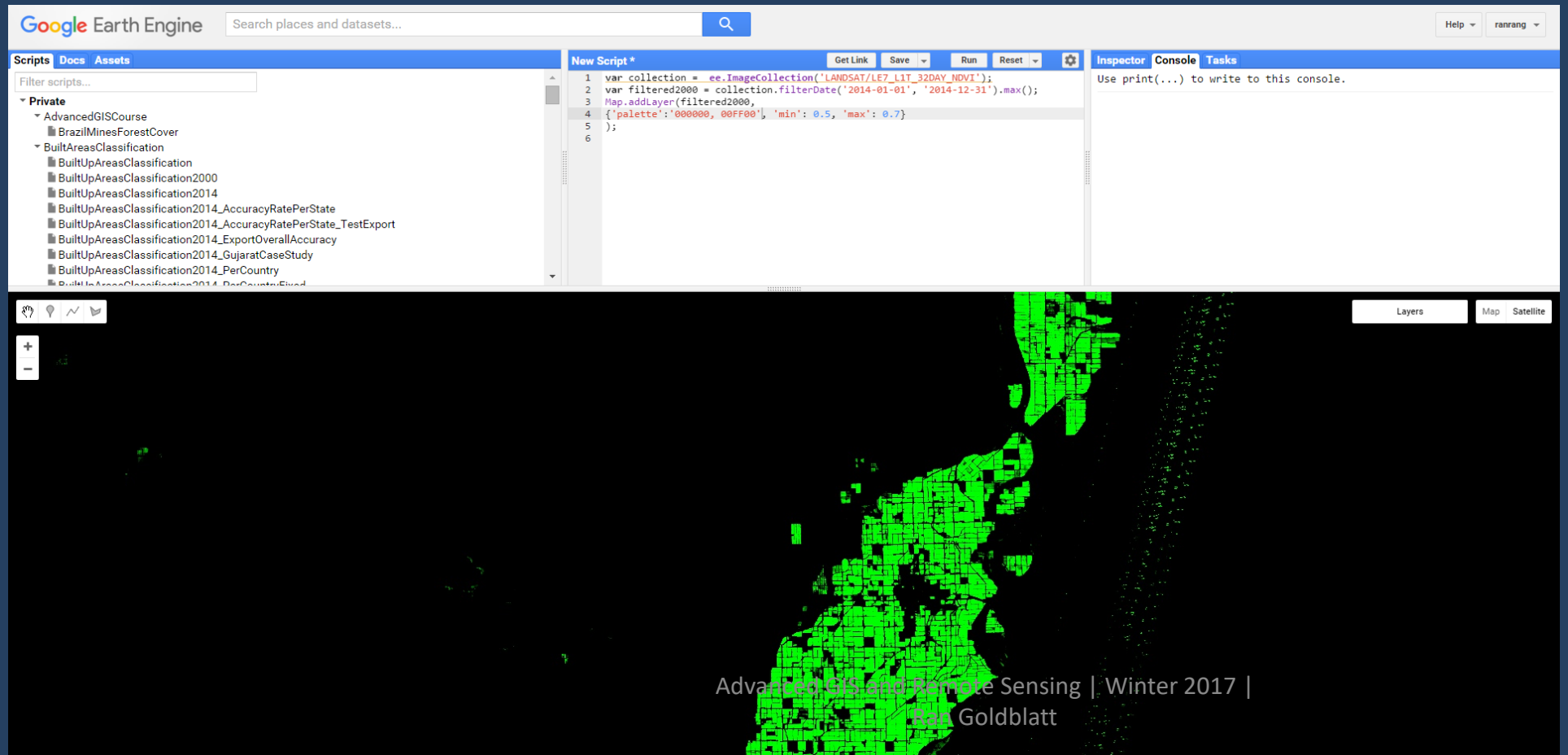
000000 00FF00

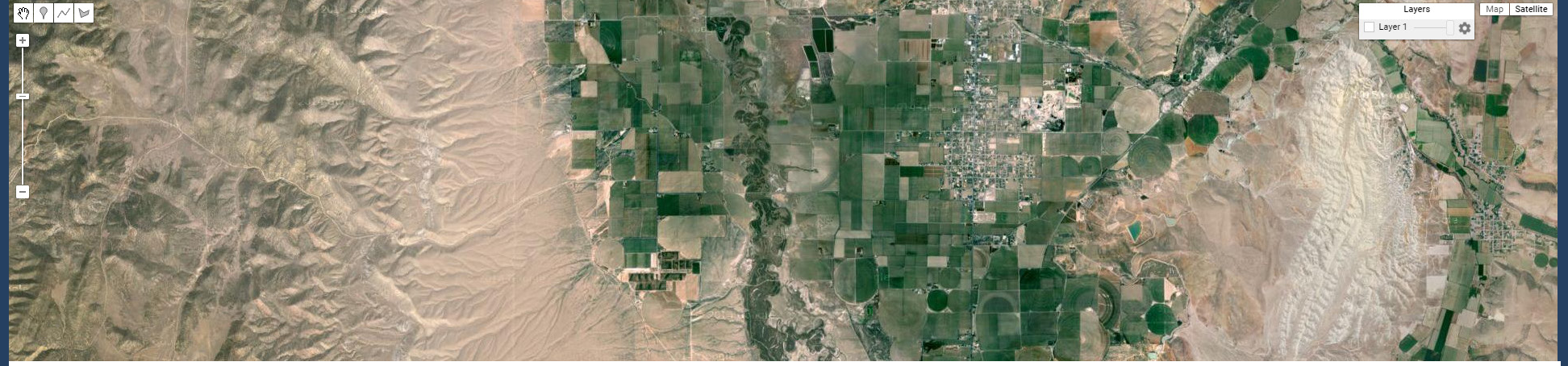


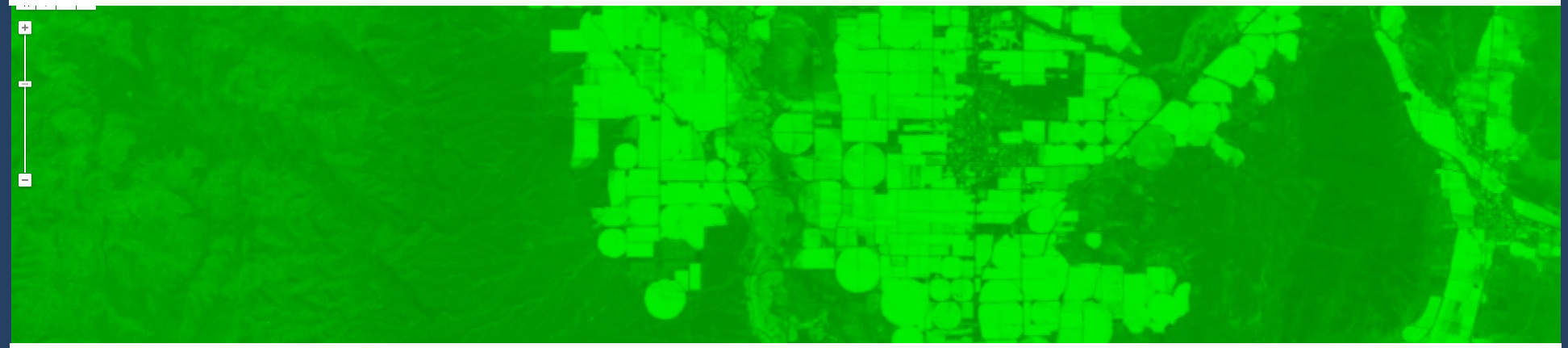
Why is the entire image green?

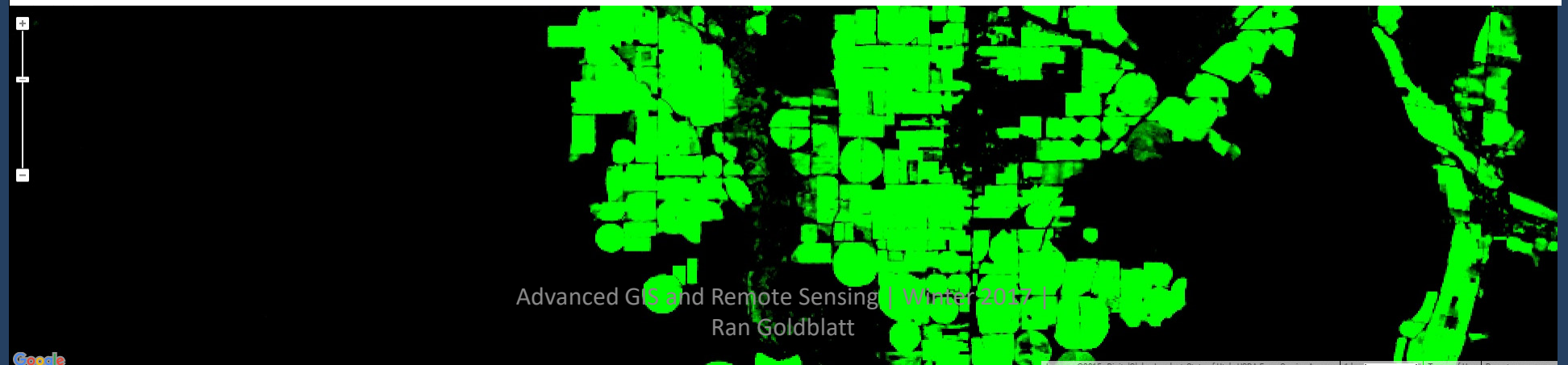
-1 +1

```
var collection =  ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');
var filtered2000 = collection.filterDate('2014-01-01', '2014-12-31').max();
Map.addLayer(filtered2000,
{'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}
);
```




Layers
Layer 1
Map
Satellite





DEMO

https://code.earthengine.google.com/

```
print('hello class!');
```

1. Navigate to: El centro, California
2. Search for: landsat 7 annual TOA Percentile composites
3. Copy the ID of the image collection: LANDSAT/LE7_TOA_1YEAR
4. Call this image collection and print it to the console:

```
var landsat7TOA = ee.ImageCollection('LANDSAT/LE7_TOA_1YEAR');
print(landsat7TOA);
```

5. Copy the image ID of the 2014 image.
Call it, print it, and add it to the map:

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
print(landsat7TOA2014);
Map.addLayer(landsat7TOA2014);
```

Define the bands for the visualization:

```
var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');
Map.addLayer(landsat7TOA2014,
{'bands': ['B3', 'B2', 'B1']}
);
```

1. Search for: Landsat 8 32-Day TOA Reflectance Composite.
2. Copy the ID of the collection (LANDSAT/LC8_L1T_32DAY_TOA)
3. Filter by date to the second half of 2014, and add this image to the map

```
var landsat8_32Days =
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');

var landsat8_SecondHalf14 = landsat8_32Days
.filterDate('2014-06-01', '2014-12-31');

Map.addLayer(landsat8_SecondHalf14,
{'bands': ['B4', 'B3', 'B2']}, 'landsat 8 at 2014');
```

1. Define a variable for Band 2 (Blue)
2. Add Band 2 to the map

```
var landsat8_32Days =
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');

var landsat8_SecondHalf14 = landsat8_32Days
.filterDate('2014-06-01', '2014-12-31');

Map.addLayer(landsat8_SecondHalf14,
{'bands': ['B4', 'B3', 'B2']}, 'landsat 8 at 2014');

var L8_Band2 = landsat8_SecondHalf14.select('B2');
Map.addLayer(L8_Band2,{}, 'band 2');
```

Calculate the median value of this collection and add it to the map

```
var L8_32Days =
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA');

var L8_SecondHalf14 = L8_32Days
.filterDate('2014-06-01', '2014-12-31')

var L8_SecondHalf14Median = L8_SecondHalf14.median();

Map.addLayer(L8_SecondHalf14, {'bands': ['B4', 'B3', 'B2']} ,
'landsat 8 ' );

Map.addLayer(L8_SecondHalf14Median, {'bands': ['B4', 'B3', 'B2']} ,
'landsat 8 median' );
```

Simple composite of Landsat scenes

```
var L8_Collection = ee.ImageCollection('LANDSAT/LC8_L1T');

var L8Filtered = L8_Collection.filterDate('2014-10-01', '2014-12-31');

var L8Median = L8Filtered.median();

Map.addLayer(L8Median,  {'bands': ['B4', 'B3', 'B2'], min: 6000, max:17000});

var image = ee.Algorithms.Landsat.simpleComposite({
  collection: L8Filtered,
  asFloat: true
});

Map.addLayer(image , {'bands': ['B4', 'B3', 'B2'], min: 0, max:0.3}, 'landsat 8
Median Witner 2014' );
```

1. Search for the ID of Landsat 7 32-Day NDVI Composite
LANDSAT/LE7_L1T_32DAY_NDVI
2. Filter by date (2014 and 2000) and extract the max NDVI value

```
var ndvi32Day=  ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');

var filtered2014 = ndvi32Day.filterDate('2014-01-01', '2014-12-31').max();
var filtered2000 = ndvi32Day.filterDate('2000-01-01', '2000-12-31').max();


Map.addLayer(filtered2014, {}, 'max ndvi 2014');
Map.addLayer(filtered2000, {}, 'max ndvi 2000');
```

Add a visualization parameter:

```
var ndvi32Day=  ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');

var filtered2014 = ndvi32Day.filterDate('2014-01-01', '2014-12-31').max();
var filtered2000 = ndvi32Day.filterDate('2000-01-01', '2000-12-31').max();


Map.addLayer(filtered2014, {'palette':'000000, 00FF00'}, 'max ndvi 2014');
Map.addLayer(filtered2000, {'palette':'000000, 00FF00'}, 'max ndvi 2000');
```

Define the minimum and maximum value for visualization

```
var ndvi32Day=  ee.ImageCollection('LANDSAT/LE7_L1T_32DAY_NDVI');

var filtered2014 = ndvi32Day.filterDate('2014-01-01', '2014-12-31').max();
var filtered2000 = ndvi32Day.filterDate('2000-01-01', '2000-12-31').max();


Map.addLayer(filtered2014,
{'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}, 'max ndvi 2014');

Map.addLayer(filtered2000,
{'palette':'000000, 00FF00', 'min': 0.5, 'max': 0.7}, 'max ndvi 2000');
```

Exercise:

Navigate to any region and present 2 Landsat images for the rainy season of 2013 and one for 2016, using the median value of the composite