

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/>

The screenshot displays the Google Earth Engine web interface. At the top, the Google Earth Engine logo and a search bar are visible. Below the search bar, the interface is divided into several panels:

- Scripts Panel (Left):** Contains a list of scripts under the 'Private' tab. A red circle with the number '2' highlights this panel.
- Code Editor (Center):** Displays a script titled 'BuiltUpAreasClassification2014_GujaratCaseStudy'. A red circle with the number '1' highlights the code editor.
- Inspector/Console/Logs Panel (Right):** Shows the 'Inspector' tab with a message: 'Use print(...) to write to this console.' A red circle with the number '3' highlights this panel.
- Map Panel (Bottom):** Displays a map of the United States and Mexico. A red circle with the number '4' highlights the map display area.
- Map Tools Panel (Bottom Left):** Contains various map interaction tools like pan, zoom, and draw. A red circle with the number '5' highlights this panel.

At the bottom left, a yellow box contains the URL: <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

ScriptsDocsAssets

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 LinkSaveRunReset

1

InspectorConsoleTasks

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



ScriptsDocsAssets

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 LinkSaveRunReset

1

InspectorConsoleTasks

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

ScriptsDocsAssets

NEW

users/rgoldblatt

users/slums

This folder is empty.

+ New folder

+ New image collection

New Script

Get LinkSaveRunReset

1

InspectorConsoleTasks

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

ScriptsDocsAssets

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 LinkSaveRunReset

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_Y0rflLegIoitshZBahK44BSscfAPJh_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

InspectorConsoleTasks

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

ScriptsDocsAssets

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 *

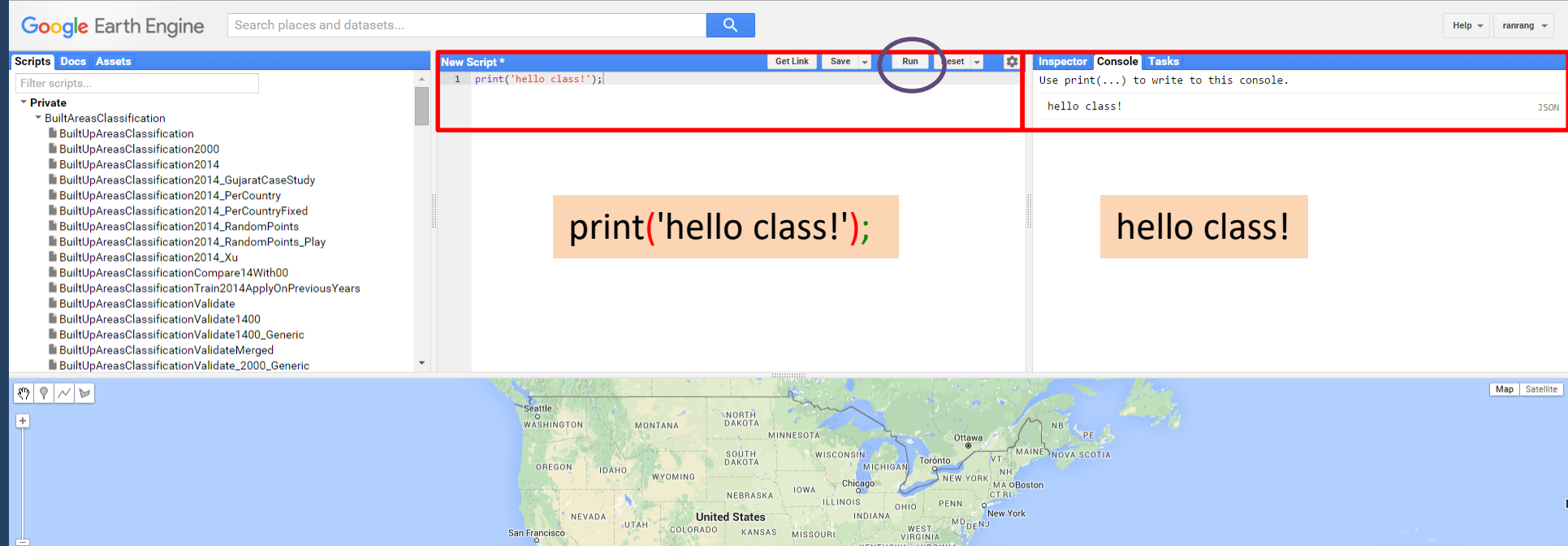
Get LinkSaveRunReset

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_Y0rflLegIoitshZBahK44BSscfAPJh_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

InspectorConsoleTasks

Older tasks
Completed
HCM_LU_Functionallity_MedianValues ✓
accuracyAssesementWith_GUE_BU_Mexi... ✓
accuracyAssesementWith_GUE_BU_USA ✓
accuracyAssesementWith_GUE_BU_India ✓
accuracyAssesementWith_GUE_BU_Mexi... ✓
accuracyAssesementWith_GUE_BU_USA ✓
accuracyAssesementWith_GUE_BU_Mexi... ✓
accuracyAssesementWith_GUE_BU_USA ✓
accuracyAssesementWith_GUE_BU_India ✓
IndiaTestGrid20_011917B ✓
IndiaTestGrid20_011917 ✓

Let`s begin...



JavaScript *statements*

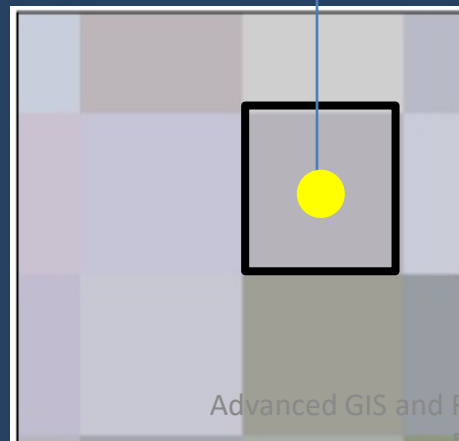
- JavaScript statements have a pair of **parentheses ()**
- They end with a **semicolon ;**
- Anything between **single quotes** is called a **'string'**.
- **Strings:**
 - Used to print out readable information.
 - Define sets of characters that name datasets or parts of datasets

```
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



```
► Point (-115.1038, 37.6273) at 611m/px
▼ Pixels
  ▼ Layer 1: Image (7 bands) 
    B1: 47
    B2: 51
    B3: 60
    B4: 72
    B5: 76
    B6_VCID_2: 208
    B7: 65
  ► Objects
```

```
Inspector Console Tasks
Use print(...) to write to this console.

▼ Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)
  type: Image
  id: LANDSAT/LE7_TOA_1YEAR/2014
  version: 1423795095111000
  ▼ bands: List (7 elements)
    ►0: "B1", unsigned int8, EPSG:4326, 1335834x638232 px
    ►1: "B2", unsigned int8, EPSG:4326, 1335834x638232 px
    ►2: "B3", unsigned int8, EPSG:4326, 1335834x638232 px
    ►3: "B4", unsigned int8, EPSG:4326, 1335834x638232 px
    ►4: "B5", unsigned int8, EPSG:4326, 1335834x638232 px
    ►5: "B6_VCID_2", unsigned int8, EPSG:4326, 1335834x638232 px
    ►6: "B7", unsigned int8, EPSG:4326, 1335834x638232 px
  ► properties: Object (1 property)
```

Image collection

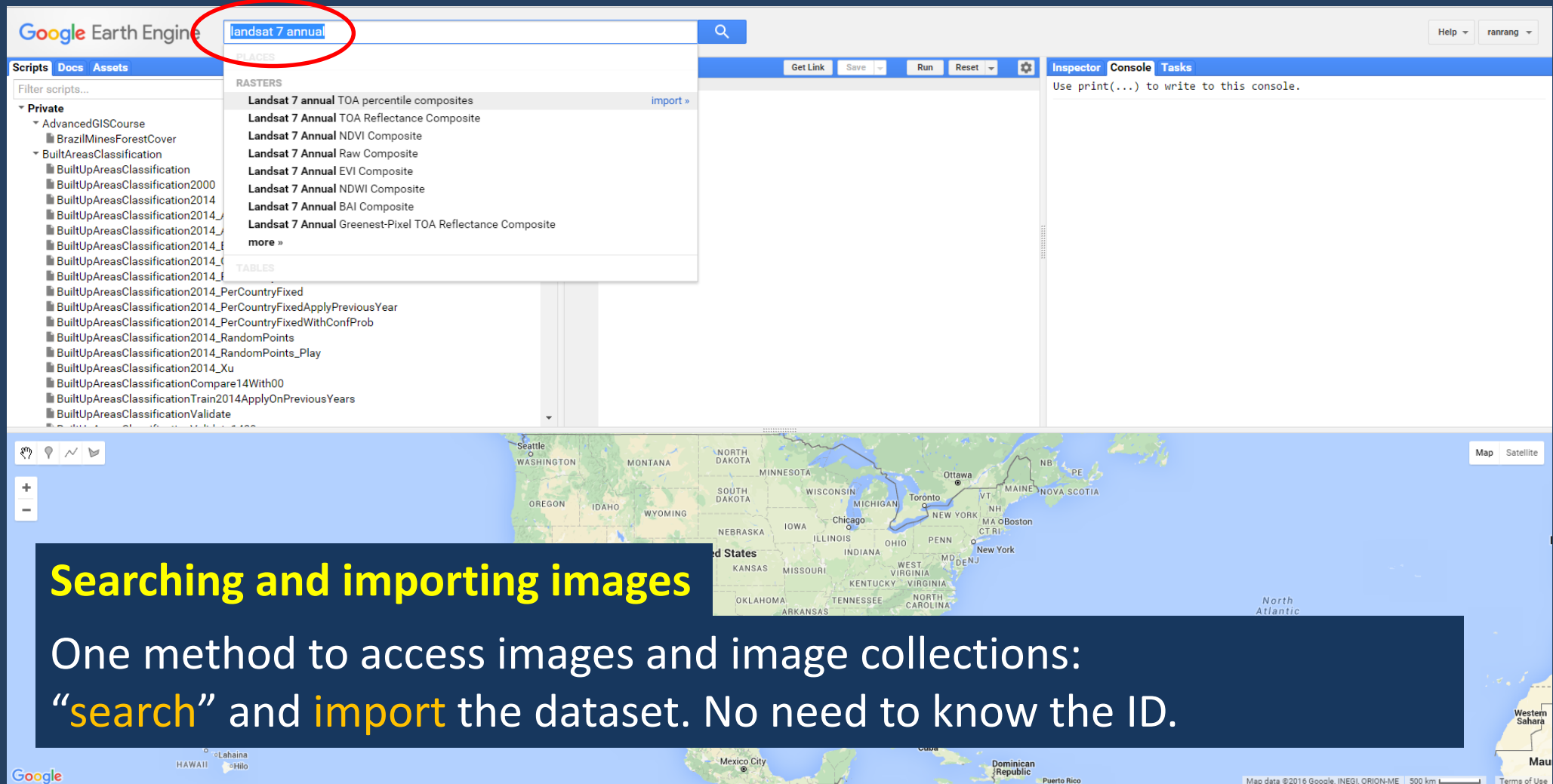
- Collections/
Mosaics of EE images
- *Example:*
 - A collection of all L8 images in a given time period
 - Each image collection has an ID.

```
Inspector Console Tasks
Use print(...) to write to this console.

▼ ImageCollection LANDSAT/LC8_L1T_8DAY_TOA (124 elements, 12 bands)
  type: ImageCollection
  id: LANDSAT/LC8_L1T_8DAY_TOA
  version: 1449778992138000
  ▶ bands: List (12 elements)
  ▼ features: List (124 elements)
    ▶ 0: Image LANDSAT/LC8_L1T_8DAY_TOA/20130407 (12 bands)
    ▶ 1: Image LANDSAT/LC8_L1T_8DAY_TOA/20130415 (12 bands)
    ▶ 2: Image LANDSAT/LC8_L1T_8DAY_TOA/20130423 (12 bands)
    ▶ 3: Image LANDSAT/LC8_L1T_8DAY_TOA/20130501 (12 bands)
    ▶ 4: Image LANDSAT/LC8_L1T_8DAY_TOA/20130509 (12 bands)
    ▶ 5: Image LANDSAT/LC8_L1T_8DAY_TOA/20130517 (12 bands)
    ▶ 6: Image LANDSAT/LC8_L1T_8DAY_TOA/20130525 (12 bands)
    ▶ 7: Image LANDSAT/LC8_L1T_8DAY_TOA/20130602 (12 bands)
    ▶ 8: Image LANDSAT/LC8_L1T_8DAY_TOA/20130610 (12 bands)
    ▶ 9: Image LANDSAT/LC8_L1T_8DAY_TOA/20130618 (12 bands)
    ▶ 10: Image LANDSAT/LC8_L1T_8DAY_TOA/20130626 (12 bands)
    ▶ 11: Image LANDSAT/LC8_L1T_8DAY_TOA/20130704 (12 bands)
    ▶ 12: Image LANDSAT/LC8_L1T_8DAY_TOA/20130712 (12 bands)
    ▶ 13: Image LANDSAT/LC8_L1T_8DAY_TOA/20130720 (12 bands)
    ▶ 14: Image LANDSAT/LC8_L1T_8DAY_TOA/20130728 (12 bands)
    ▶ 15: Image LANDSAT/LC8_L1T_8DAY_TOA/20130805 (12 bands)
    ▶ 16: Image LANDSAT/LC8_L1T_8DAY_TOA/20130813 (12 bands)
    ▶ 17: Image LANDSAT/LC8_L1T_8DAY_TOA/20130821 (12 bands)
```

“Calling” Image Collections

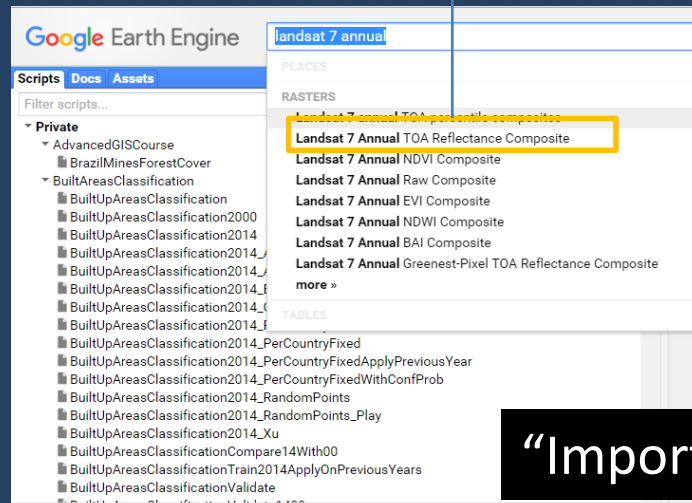
Option 1



The screenshot shows the Google Earth Engine web interface. In the top-left search bar, the text "landsat 7 annual" is entered and circled in red. Below the search bar, a dropdown menu displays a list of image collections under the "RASTERS" tab. The first item, "Landsat 7 annual TOA percentile composites", is highlighted, and an "import »" link is visible to its right. Other collections listed include "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", and "Landsat 7 Annual Greenest-Pixel TOA Reflectance Composite". The left sidebar shows a tree view of the user's workspace, including a "Private" folder and various "BuiltUpAreasClassification" datasets. The main map area displays a satellite view of the United States, with state boundaries and major cities labeled. The bottom of the interface features a Google logo, a scale bar, and a "Terms of Use" link.

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

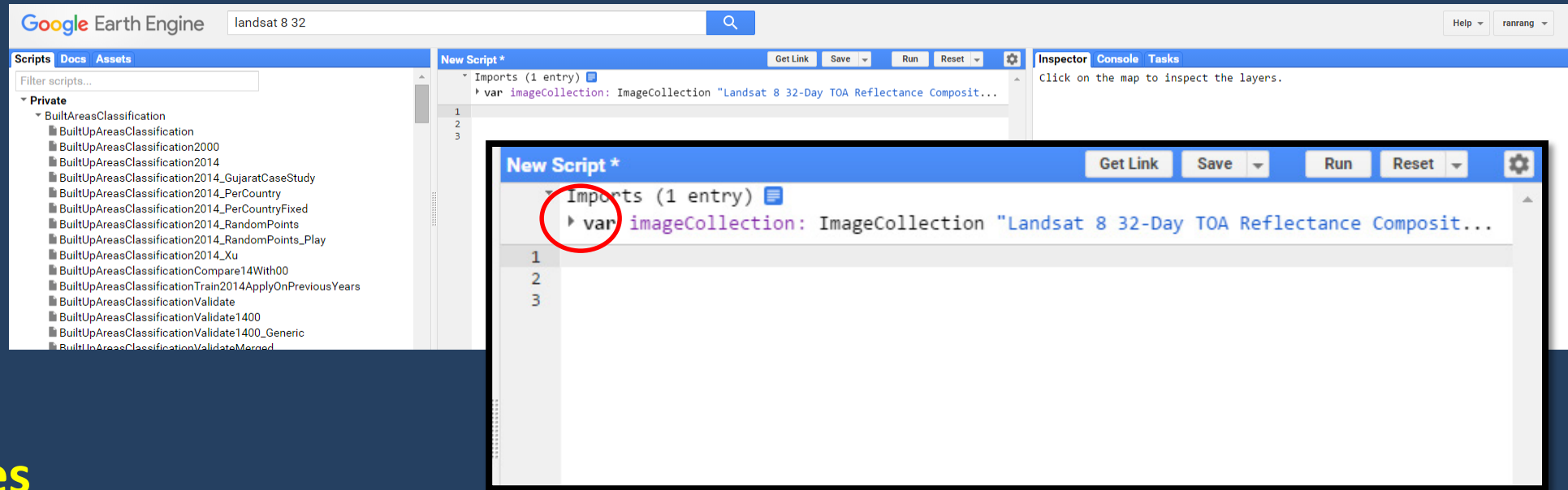
ImageCollection ID
LANDSAT/LE7_TOA_1YEAR

Import

“Import” will add the image to the workspace

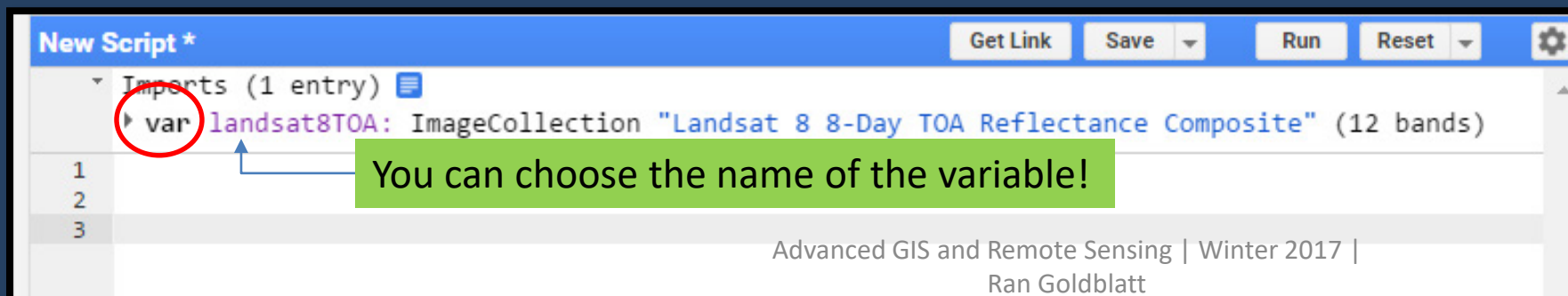
This image collection has a name:
LANDSAT/LE7_TOA_1YEAR





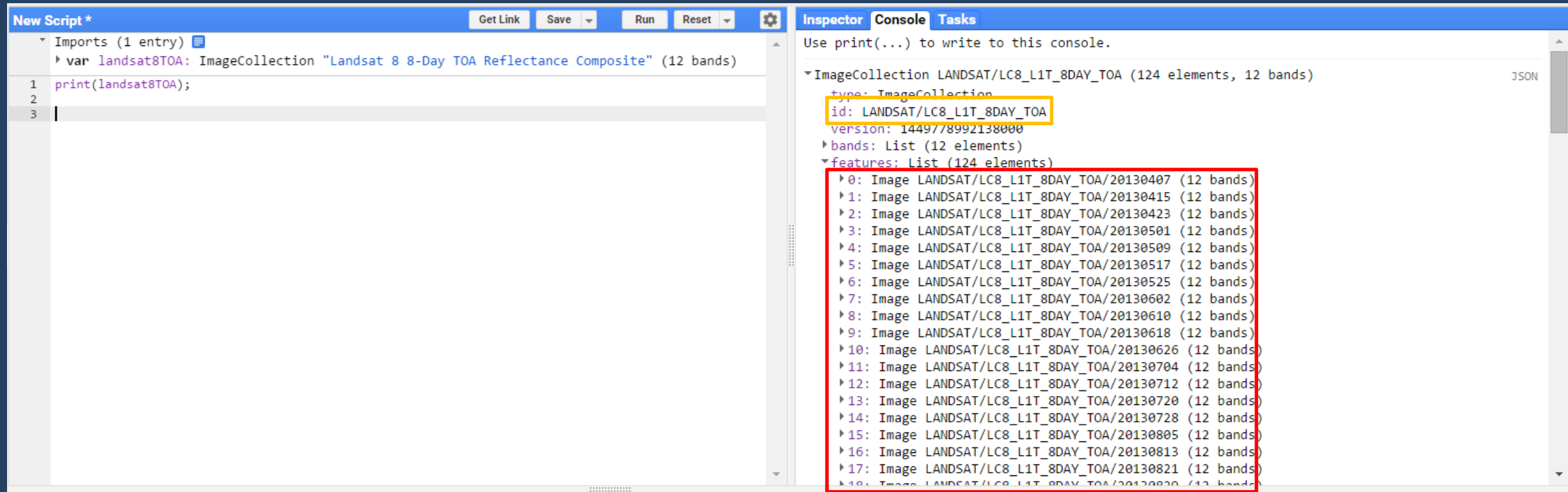
Variables

- 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);
```



The screenshot shows a QGIS console window with a script that prints the details of an image collection. The script in the left pane is:

```
1 print(landsat8TOA);
```

The console output on the right shows the following structure:

- ImageCollection LANDSAT/LC8_L1T_8DAY_TOA (124 elements, 12 bands)
 - type: ImageCollection
 - id: LANDSAT/LC8_L1T_8DAY_TOA
 - version: 1449778992138000
 - bands: List (12 elements)
 - features: List (124 elements)
 - 0: Image LANDSAT/LC8_L1T_8DAY_TOA/20130407 (12 bands)
 - 1: Image LANDSAT/LC8_L1T_8DAY_TOA/20130415 (12 bands)
 - 2: Image LANDSAT/LC8_L1T_8DAY_TOA/20130423 (12 bands)
 - 3: Image LANDSAT/LC8_L1T_8DAY_TOA/20130501 (12 bands)
 - 4: Image LANDSAT/LC8_L1T_8DAY_TOA/20130509 (12 bands)
 - 5: Image LANDSAT/LC8_L1T_8DAY_TOA/20130517 (12 bands)
 - 6: Image LANDSAT/LC8_L1T_8DAY_TOA/20130525 (12 bands)
 - 7: Image LANDSAT/LC8_L1T_8DAY_TOA/20130602 (12 bands)
 - 8: Image LANDSAT/LC8_L1T_8DAY_TOA/20130610 (12 bands)
 - 9: Image LANDSAT/LC8_L1T_8DAY_TOA/20130618 (12 bands)
 - 10: Image LANDSAT/LC8_L1T_8DAY_TOA/20130626 (12 bands)
 - 11: Image LANDSAT/LC8_L1T_8DAY_TOA/20130704 (12 bands)
 - 12: Image LANDSAT/LC8_L1T_8DAY_TOA/20130712 (12 bands)
 - 13: Image LANDSAT/LC8_L1T_8DAY_TOA/20130720 (12 bands)
 - 14: Image LANDSAT/LC8_L1T_8DAY_TOA/20130728 (12 bands)
 - 15: Image LANDSAT/LC8_L1T_8DAY_TOA/20130805 (12 bands)
 - 16: Image LANDSAT/LC8_L1T_8DAY_TOA/20130813 (12 bands)
 - 17: Image LANDSAT/LC8_L1T_8DAY_TOA/20130821 (12 bands)
 - 18: Image LANDSAT/LC8_L1T_8DAY_TOA/20130829 (12 bands)

This image collection has a name: **LANDSAT/LC8_L1T_8DAY_TOA**

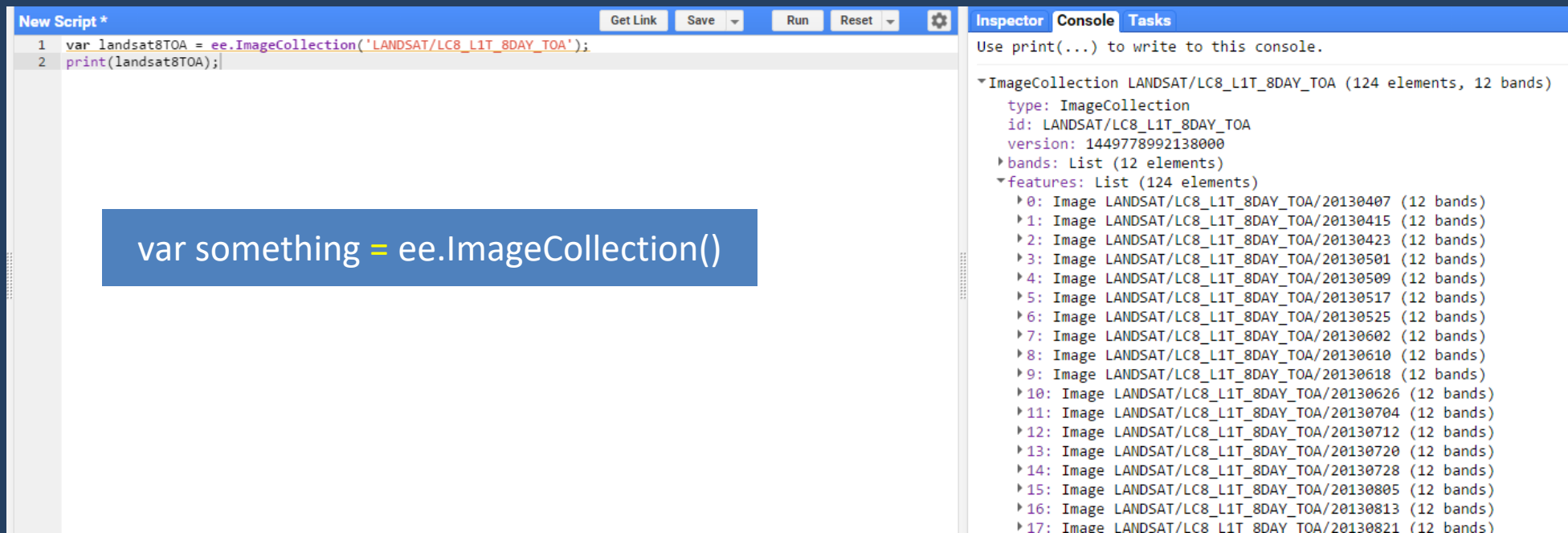
Another **method** to add images and image collections:

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.

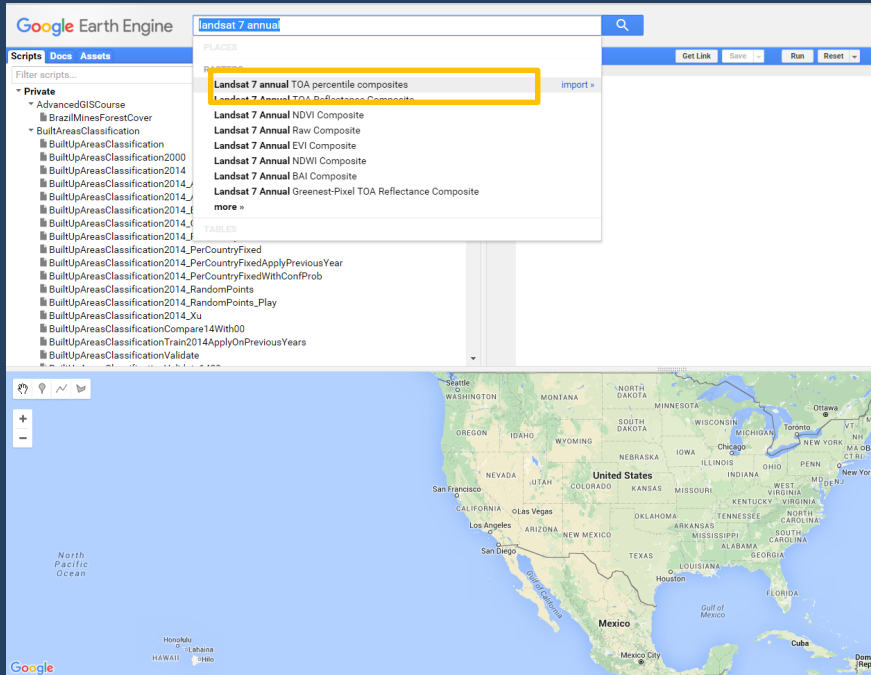
ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA')



The screenshot displays the Google Earth Engine interface. On the left, the 'New Script' editor contains two lines of code: `var landsat8TOA = ee.ImageCollection('LANDSAT/LC8_L1T_8DAY_TOA');` and `print(landsat8TOA);`. A blue callout box highlights the function signature `var something = ee.ImageCollection()`. On the right, the 'Inspector' panel shows the output of the print statement. The output is an `ImageCollection` object for `LANDSAT/LC8_L1T_8DAY_TOA`, containing 124 elements and 12 bands. The `features` list is expanded, showing a series of 17 image objects, each representing a specific date and time (e.g., `Image LANDSAT/LC8_L1T_8DAY_TOA/20130407 (12 bands)`).

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

Let's search for another image collection:



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

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

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

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

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

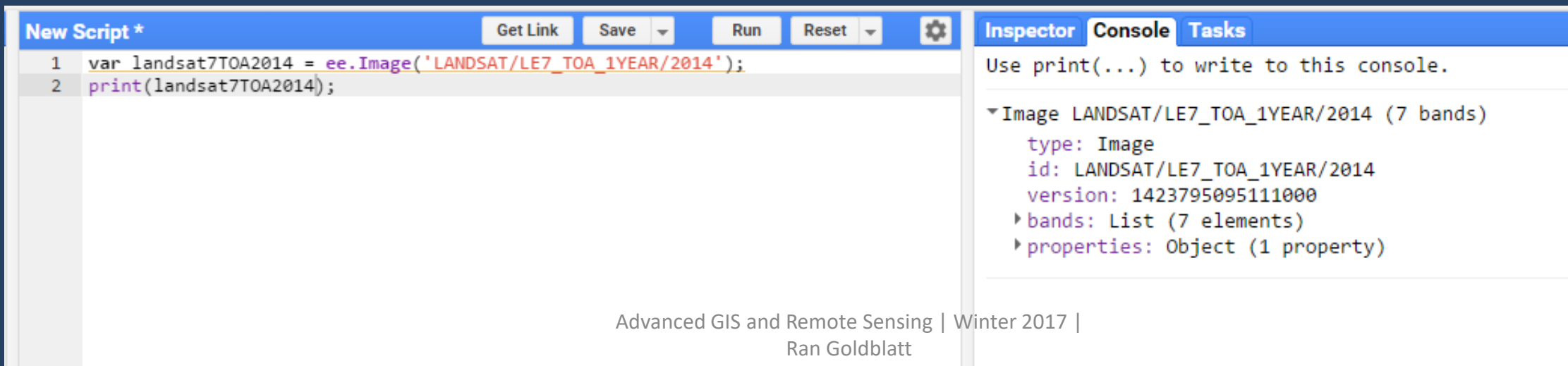
The image ID that we want to call: **LANDSAT/LE7_TOA_1YEAR/2014**

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);
```



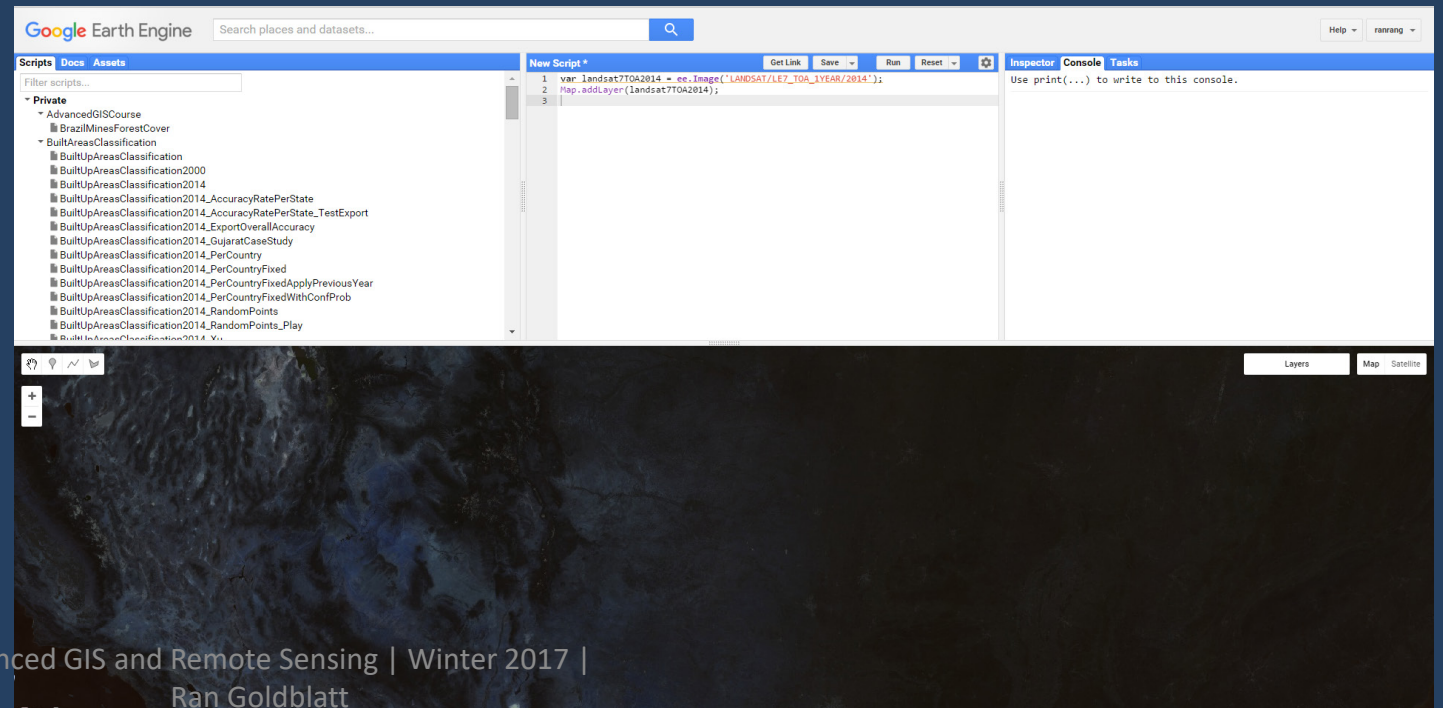
The screenshot shows the Google Earth Engine interface. On the left, a 'New Script' window contains two lines of code: `1 var landsat7TOA2014 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');` and `2 print(landsat7TOA2014);`. The 'Run' button is visible. On the right, the 'Console' tab is active, displaying the output: `Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)`. Below this, the properties are listed: `type: Image`, `id: LANDSAT/LE7_TOA_1YEAR/2014`, `version: 1423795095111000`, `bands: List (7 elements)`, and `properties: Object (1 property)`. At the bottom of the slide, the text 'Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt' is displayed.

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

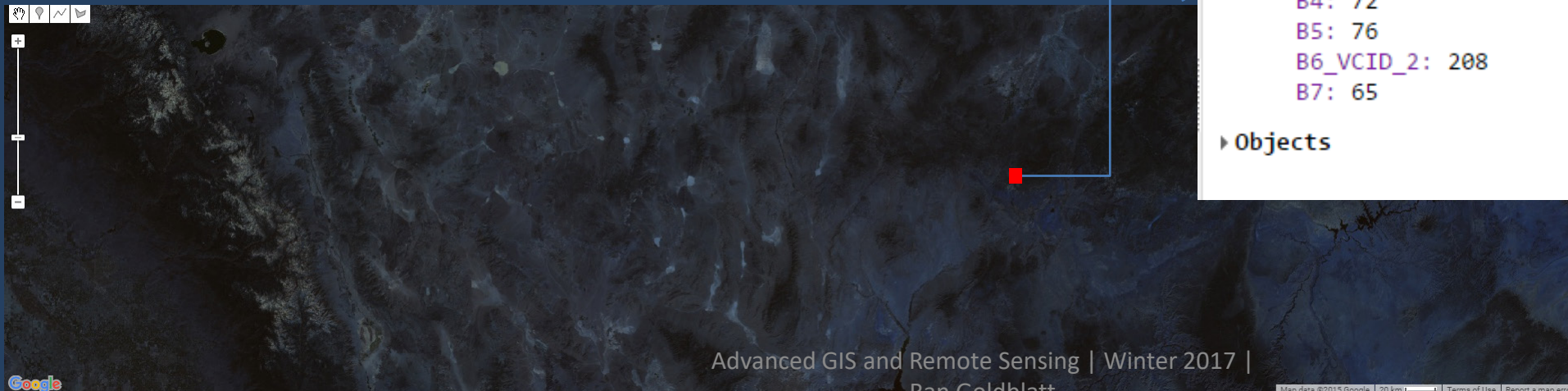
Band 3 - red

Band 4 - Near Infrared

Band 5 - Short-wave Infrared

Band 6 - Thermal Infrared

Band 7 - Short-wave Infrared



```
Inspector Console Tasks
Use print(...) to write to this console.

Image LANDSAT/LE7_TOA_1YEAR/2014 (7 bands)
  type: Image
  id: LANDSAT/LE7_TOA_1YEAR/2014
  version: 1423795095111000
  bands: List (7 elements)
    0: "B1", unsigned int8, EPSG:4326, 1335834x638232 px
    1: "B2", unsigned int8, EPSG:4326, 1335834x638232 px
    2: "B3", unsigned int8, EPSG:4326, 1335834x638232 px
    3: "B4", unsigned int8, EPSG:4326, 1335834x638232 px
    4: "B5", unsigned int8, EPSG:4326, 1335834x638232 px
    5: "B6_VCID_2", unsigned int8, EPSG:4326, 1335834x638232 px
    6: "B7", unsigned int8, EPSG:4326, 1335834x638232 px
  properties: Object (1 property)
```

```
Point (-115.1038, 37.6273) at 611m/px
  Pixels
    Layer 1: Image (7 bands)
      B1: 47
      B2: 51
      B3: 60
      B4: 72
      B5: 76
      B6_VCID_2: 208
      B7: 65
    Objects
```

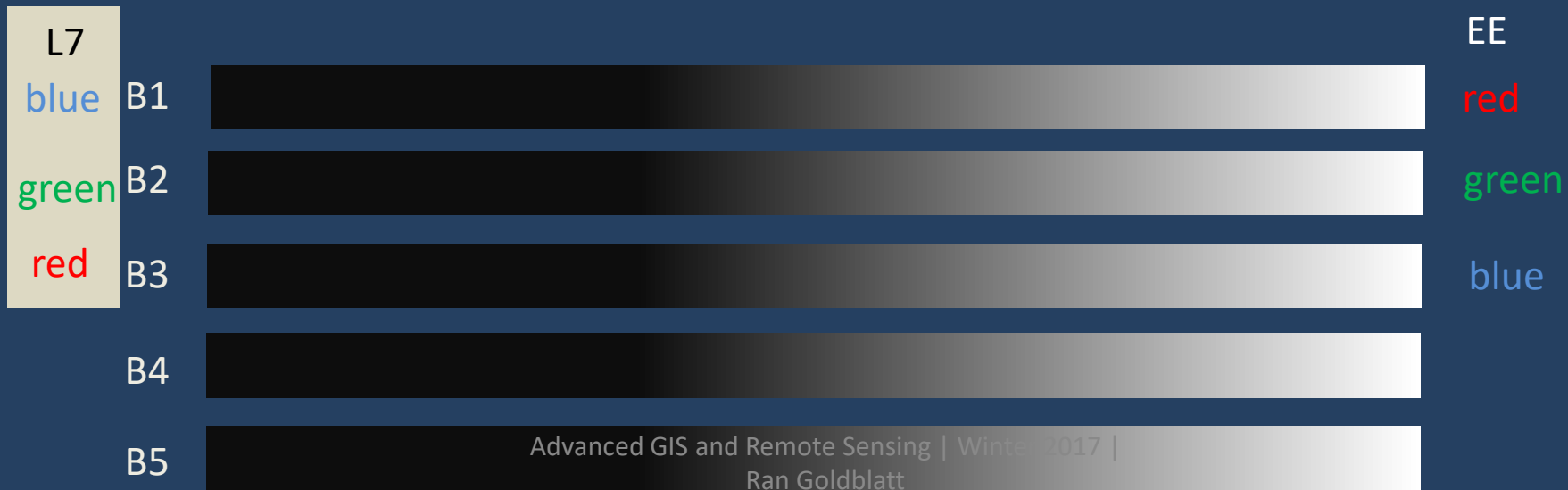
How does EE map the values in the image bands to colors on the map?

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



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}
```

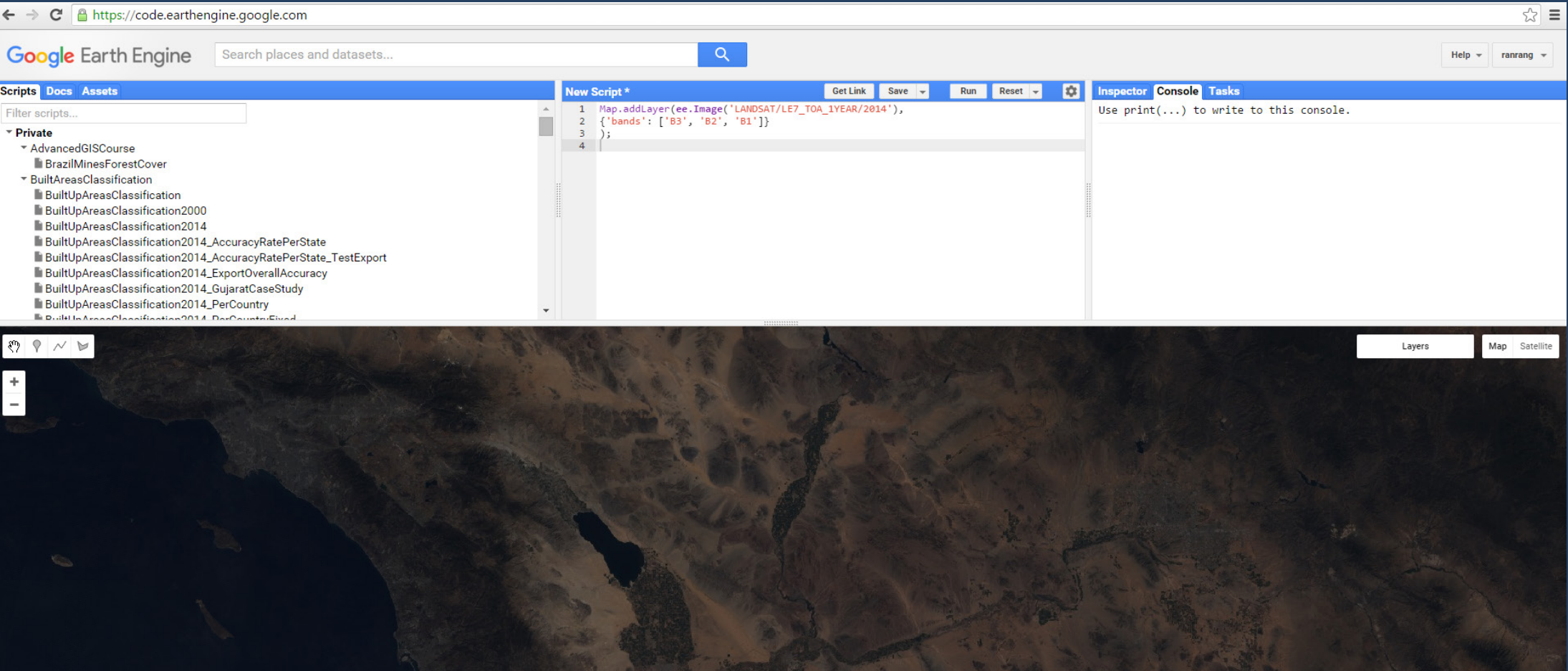
each string describes one **visualization factor**:

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

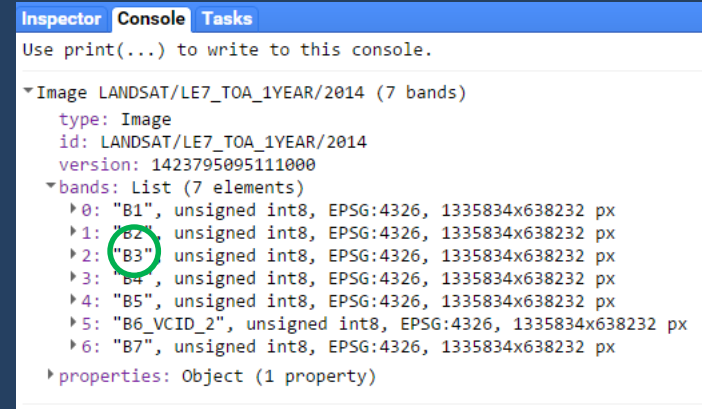
Selecting specific band(s) in an Image

`.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);
```

The screenshot displays the Google Earth Engine web interface. At the top, the Google Earth Engine logo is on the left, and a search bar with the text "Search places and datasets..." is in the center. On the right, there are "Help" and "ranrang" buttons. Below the search bar, the interface is divided into three main sections. The left section, titled "Scripts", contains a "Filter scripts..." input field and a list of scripts under a "Private" folder, including "AdvancedGISCourse" and "BuiltUpAreasClassification". The middle section, titled "New Script *", shows a code editor with the following JavaScript code:

```
1 var L7 = ee.Image('LANDSAT/LE7_TOA_1YEAR/2014');  
2 var B3_L7 = L7.select('B3');  
3 Map.addLayer(B3_L7);  
4
```

 Above the code editor are buttons for "Get Link", "Save", "Run", and "Reset". The right section, titled "Inspector Console Tasks", shows the "Console" tab with the text "Use print(...) to write to this console." Below these sections is a map view showing a satellite image of a mountainous region. In the bottom right corner of the map, there are buttons for "Layers", "Map", and "Satellite". At the very bottom of the interface, there is a footer with the text "Advanced GIS and Remote Sensing | Winter 2017 | Ran Goldblatt" and a small Google logo on the left. On the right side of the footer, there is a scale bar indicating "5 km" and links for "Map data ©2016 Google", "Terms of Use", and "Report a map error".

Let's call another image collection:

The screenshot shows the Google Earth Engine web interface. The search bar at the top contains the text 'landsat 8 32'. Below the search bar, the 'PLACES' and 'RASTERS' sections are visible. In the 'RASTERS' section, the item 'Landsat 8 32-Day TOA Reflectance Composite' is highlighted with a yellow box. The 'TABLES' section below it shows 'Landsat Path Row WRS-2_bound_world (≥2000 rows, 3 columns)'. On the left side, the 'Scripts' tab is active, showing a list of scripts under the 'Private' section. The 'Assets' tab is also visible. The bottom of the interface shows a map view with a dark, textured surface.

Landsat 8 32-Day TOA Reflectance Composite

These Landsat 8 composites are made from Level L1T orthorectified scenes, using the computed top-of-atmosphere (TOA) reflectance. See [Chander et al. \(2009\)](#) for details on the TOA computation.

These composites are created from all the scenes in each 32-day period beginning from the first day of the year and continuing to the 352nd day of the year. The last composite of the year, beginning on day 353, will overlap the first composite of the following year by 20 days. All the images from each 32-day period are included in the composite, with the most recent pixel as the composite value.

Data availability (time)
Apr 7, 2013 - Dec 19, 2015

Provider
Google

Tags
landsat, usgs, l8, lc8, oli, tirs, 32day, toa

ImageCollection ID
LANDSAT/LC8_L1T_32DAY_TOA

Import

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

- BuiltUpAreasClassification
 - 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

22
23
24
25
26
27
28
29
30
31
32
33
34

```
var trainingDataset = ee.FeatureCollection('ft:1u-IJPzWSL13DuPL1Qz8A85MTiux_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

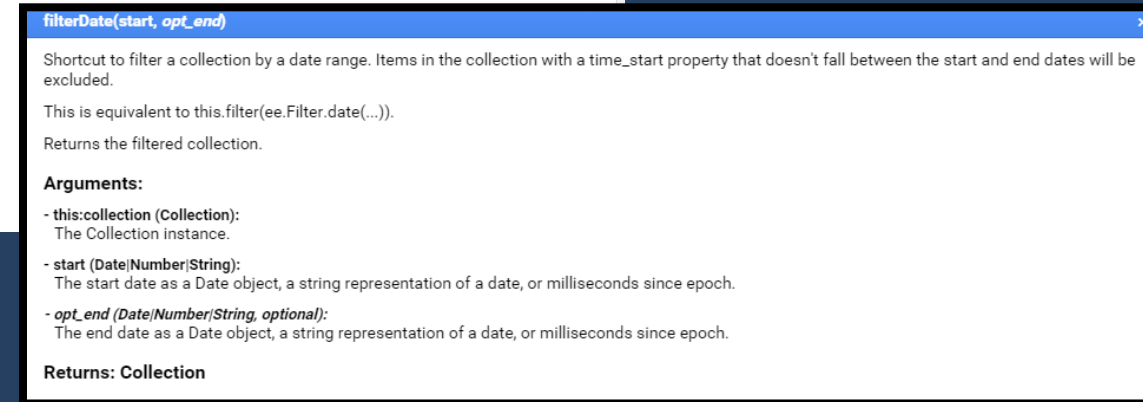
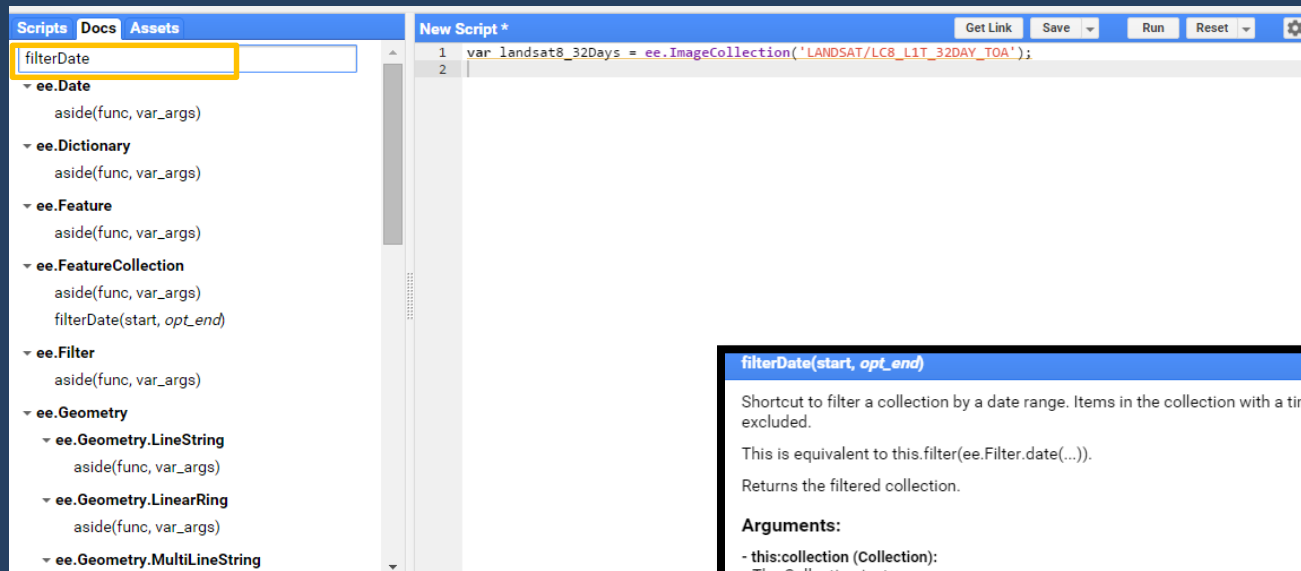
Honolulu
Lahaina
HAWAII
Hilo

Mexico City
Guatemala
Cuba
Dominican Republic
Puerto Rico

Western Sahara
Mau

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'

Advanced GIS and Remote Sensing (Winter 2014)
Ran Goldblatt

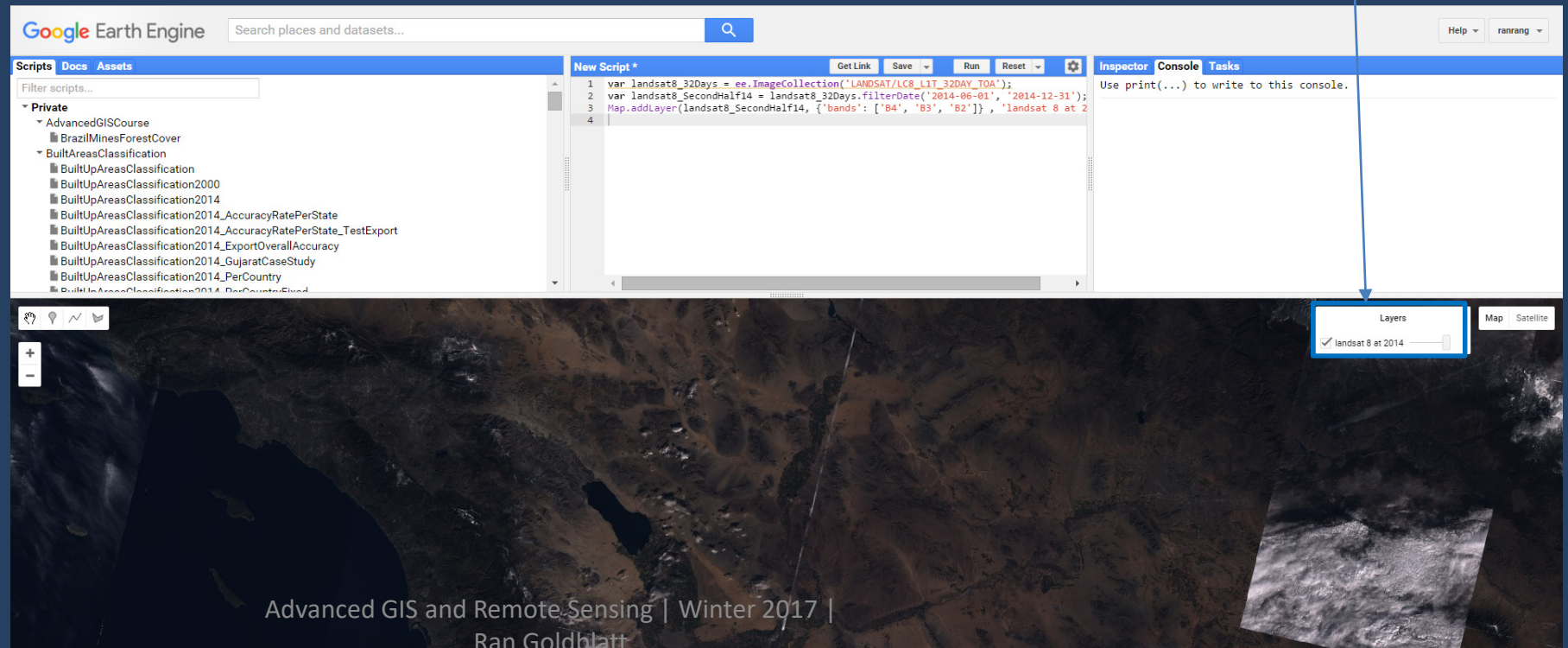
```
ee.ImageCollection('LANDSAT/LC8_L1T_32DAY_TOA')  
.filterDate('2014-06-01', '2014-12-31');
```

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()
```

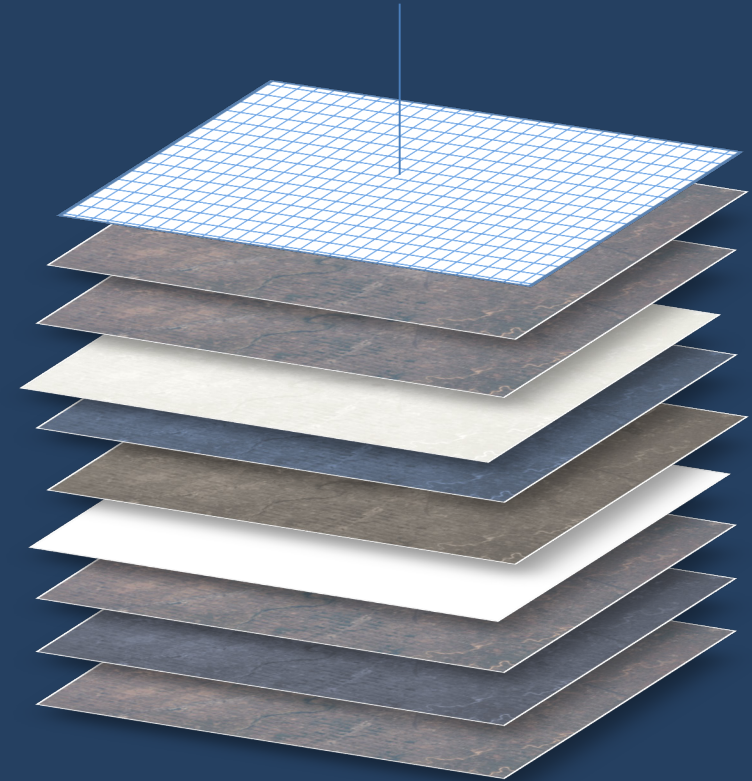
```
var maxVal = imageCollection.max()
```

```
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:

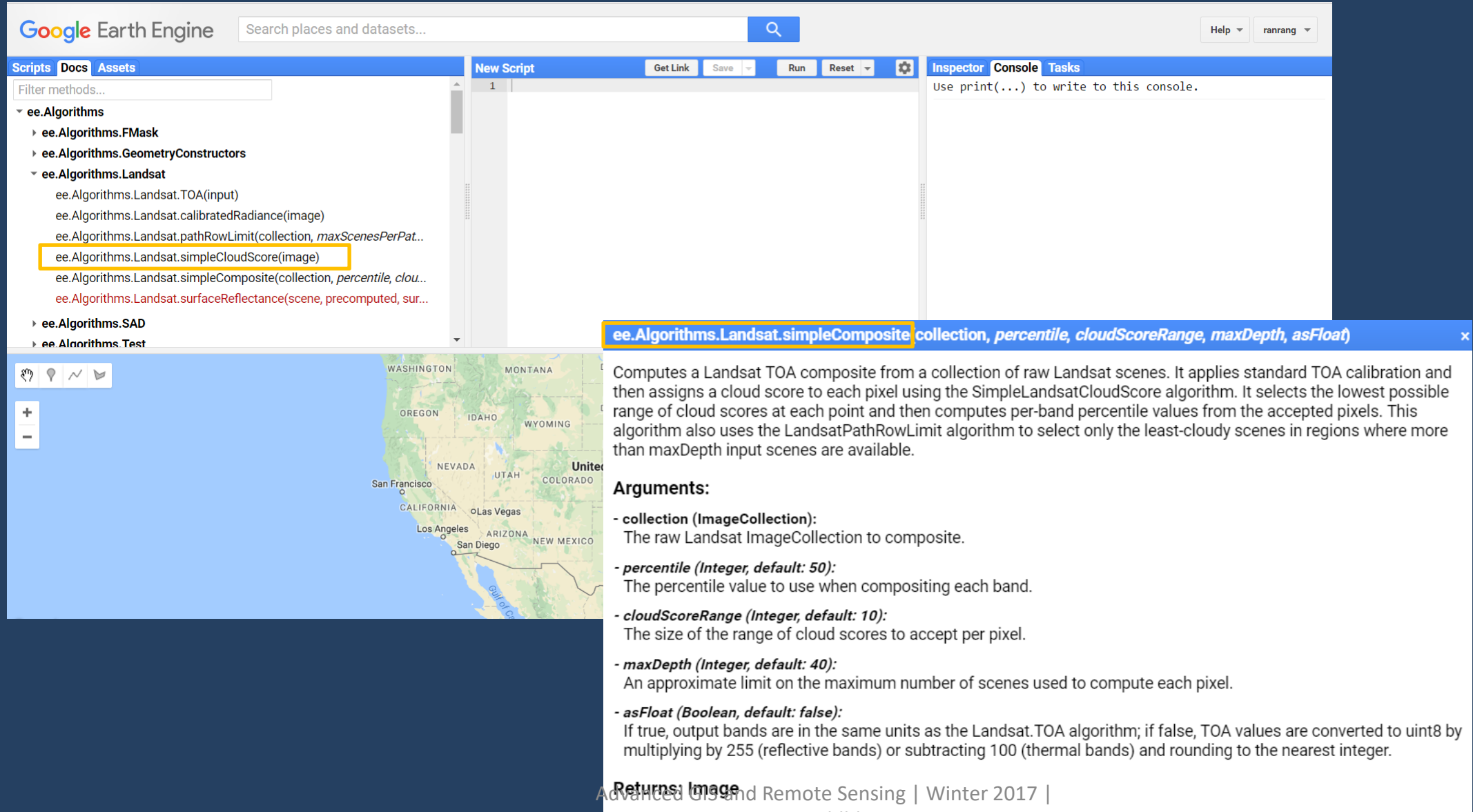
1. Take the collection of the **raw scenes**
2. Apply TOA **calibration**
3. Assign a **cloud score** to each pixel
4. Select the **lowest possible range of cloud** scores at each point
5. Compute per-band **percentile values** from the accepted pixels.

Reminder:

Top of Atmosphere Reflectance (%)

Consider the angle and distance between the sun, surface and satellite.

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



The screenshot shows the Google Earth Engine web interface. In the left sidebar, under 'Scripts', the 'ee.Algorithms.Landsat' folder is expanded, and the 'ee.Algorithms.Landsat.simpleComposite' function is highlighted with a yellow box. A tooltip is visible over this function, displaying its signature and description. The main map area shows a view of the western United States, including California, Nevada, and Arizona. The right sidebar contains the 'Inspector', 'Console', and 'Tasks' tabs.

ee.Algorithms.Landsat.simpleComposite *collection, percentile, cloudScoreRange, maxDepth, asFloat*

Computes a Landsat TOA composite from a collection of raw Landsat scenes. It applies standard TOA calibration and then assigns a cloud score to each pixel using the SimpleLandsatCloudScore algorithm. It selects the lowest possible range of cloud scores at each point and then computes per-band percentile values from the accepted pixels. This algorithm also uses the LandsatPathRowLimit algorithm to select only the least-cloudy scenes in regions where more than maxDepth input scenes are available.

Arguments:

- **collection** (*ImageCollection*):
The raw Landsat ImageCollection to composite.
- **percentile** (*Integer, default: 50*):
The percentile value to use when compositing each band.
- **cloudScoreRange** (*Integer, default: 10*):
The size of the range of cloud scores to accept per pixel.
- **maxDepth** (*Integer, default: 40*):
An approximate limit on the maximum number of scenes used to compute each pixel.
- **asFloat** (*Boolean, default: false*):
If true, output bands are in the same units as the Landsat TOA algorithm; if false, TOA values are converted to uint8 by multiplying by 255 (reflective bands) or subtracting 100 (thermal bands) and rounding to the nearest integer.

Returns: Image

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');
```

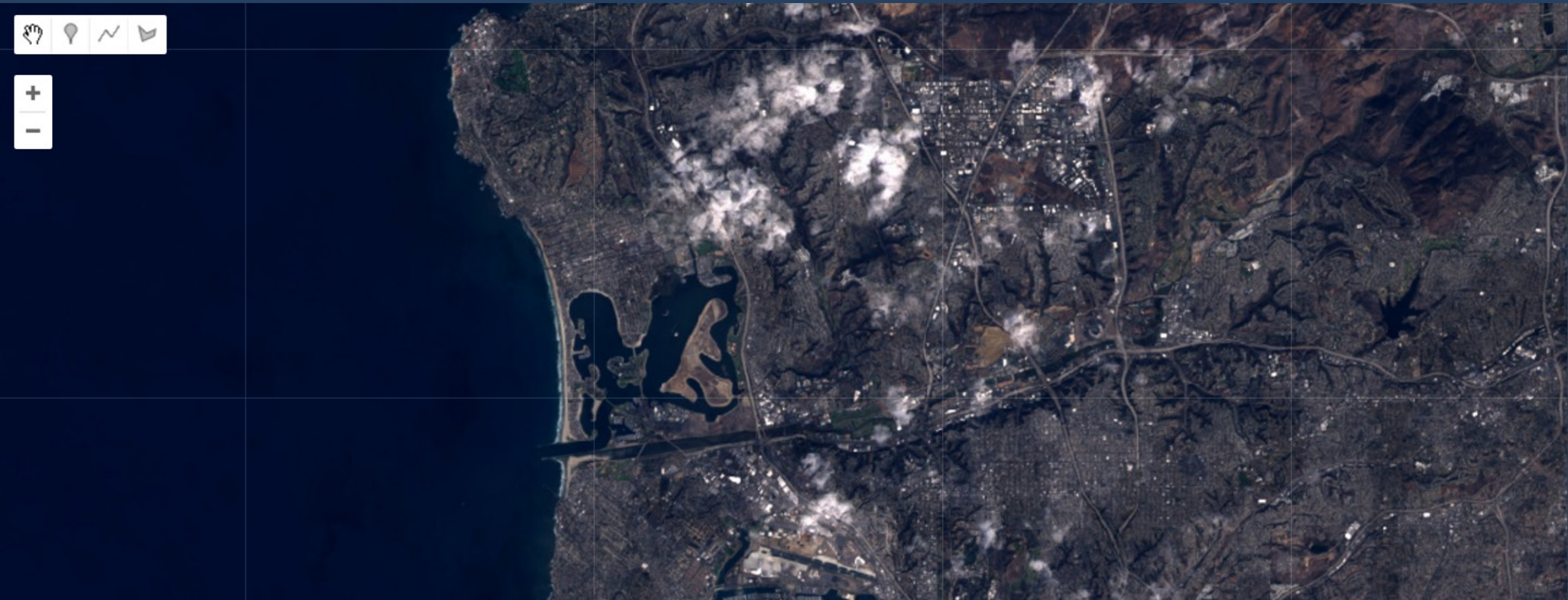
```
var L8Median = L8Filtered.median();
```

```
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

The screenshot shows the Google Earth Engine web interface. In the top search bar, 'ndvi landsat 7' is entered. The left sidebar shows a list of methods under 'ee.Algorithms'. The right sidebar shows the 'Inspector' tab with details for the selected 'Landsat 7 32-Day NDVI Composite'.

Google Earth Engine

Search:

PLACES

RASTERS

- Landsat 7 Annual NDVI Composite
- Landsat 7 32-Day NDVI Composite**
- Landsat 7 8-Day NDVI Composite
- Landsat 7 Annual Greenest-Pixel TOA Reflectance Composite

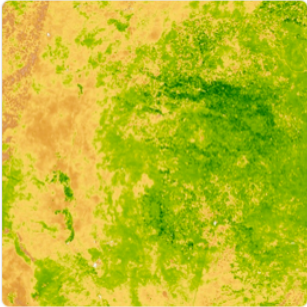
TABLES

Landsat 7 32-Day NDVI Composite

These Landsat 7 composites are made from Level L1T orthorectified scenes, using the computed top-of-atmosphere reflectance. The Normalized Difference Vegetation Index is generated from the Near-IR and Red bands of each scene, and ranges in value from -1.0 to 1.0.

These are most recent value composites created from all the scenes in each 32-day period beginning from the first day of the year and continuing to the 352nd day of the year. The last composite of the year, beginning on day 353, will overlap the first composite of the following year by 20 days. All the images from each 32-day period are included in the composite, with the most recent pixel as the composite value.

Sample



Data availability (time)
Jan 1, 1999 - Nov 17, 2015

Provider
Google

Tags
landsat, usgs, l7, le7, etm, 32day, ndvi

ImageCollection ID
LANDSAT/LE7_L1T_32DAY_NDVI

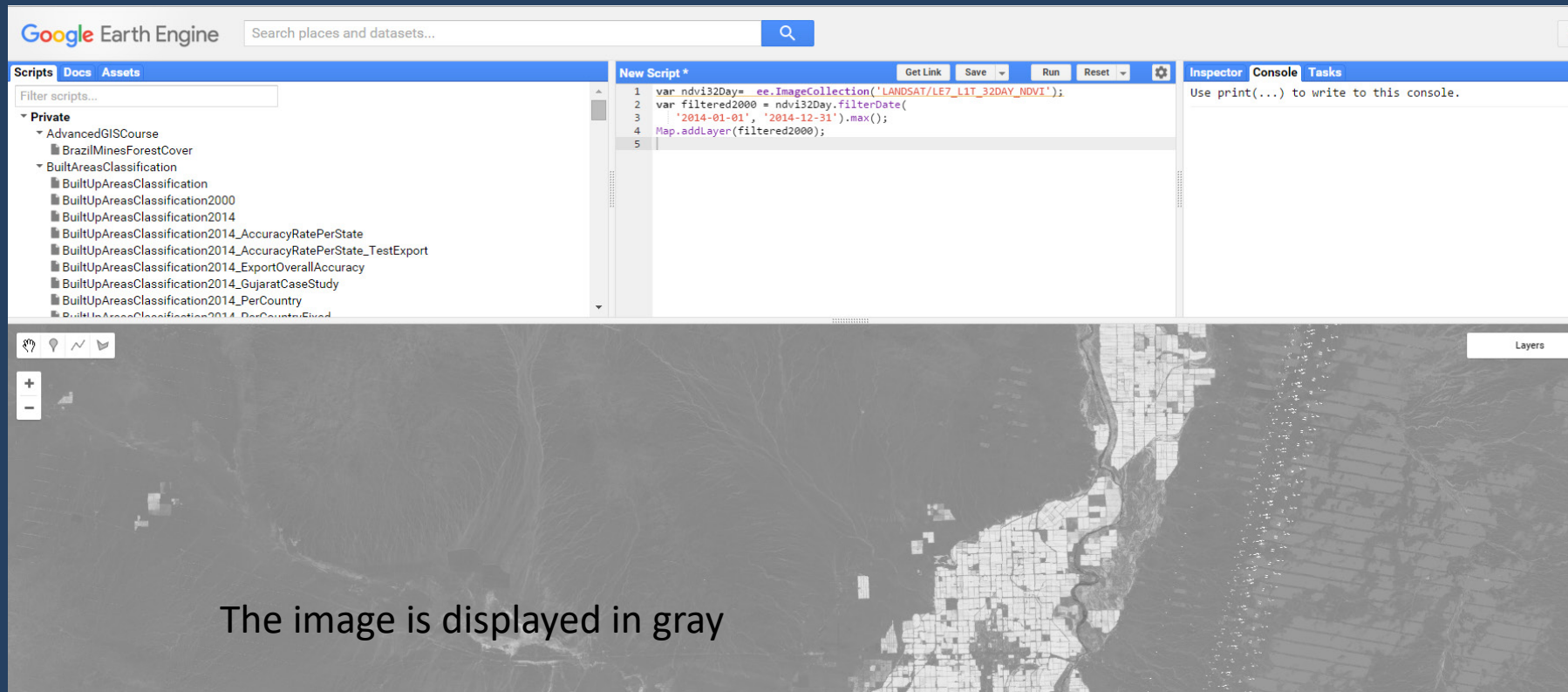
[Open in workspace](#)

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);
```



The image is displayed in gray

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

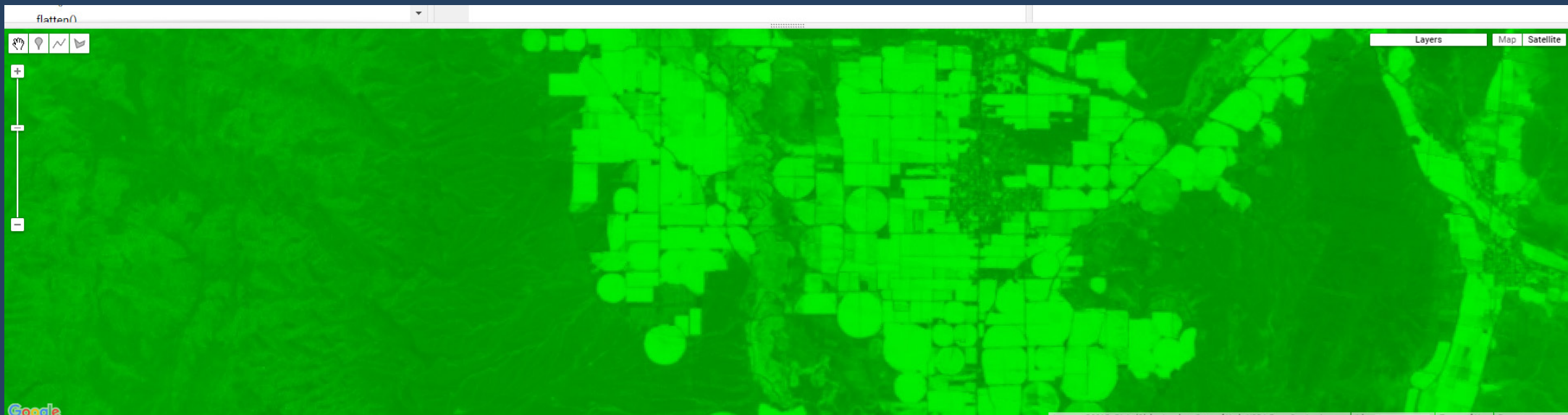
'0000FF' is blue.

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?



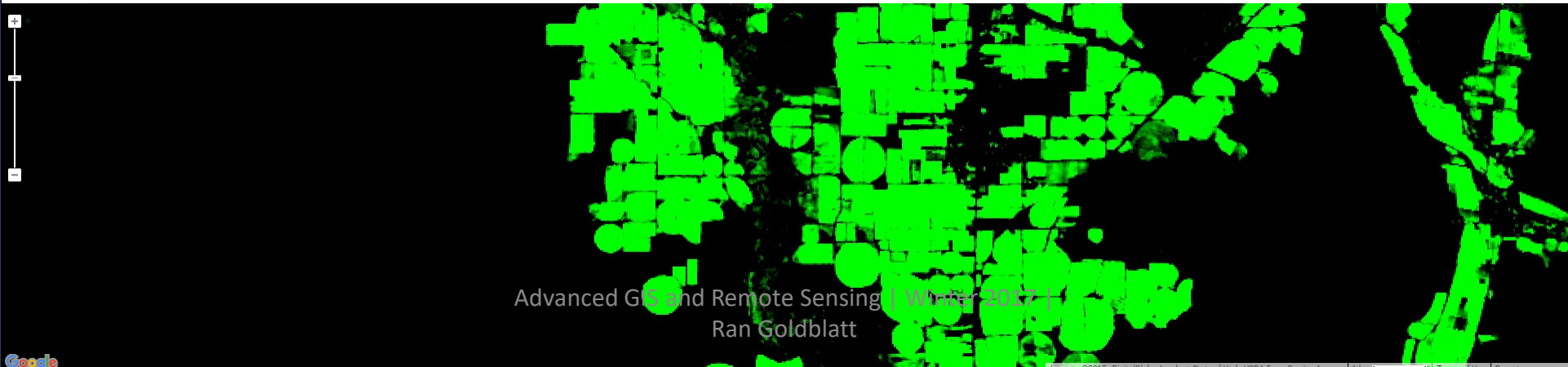
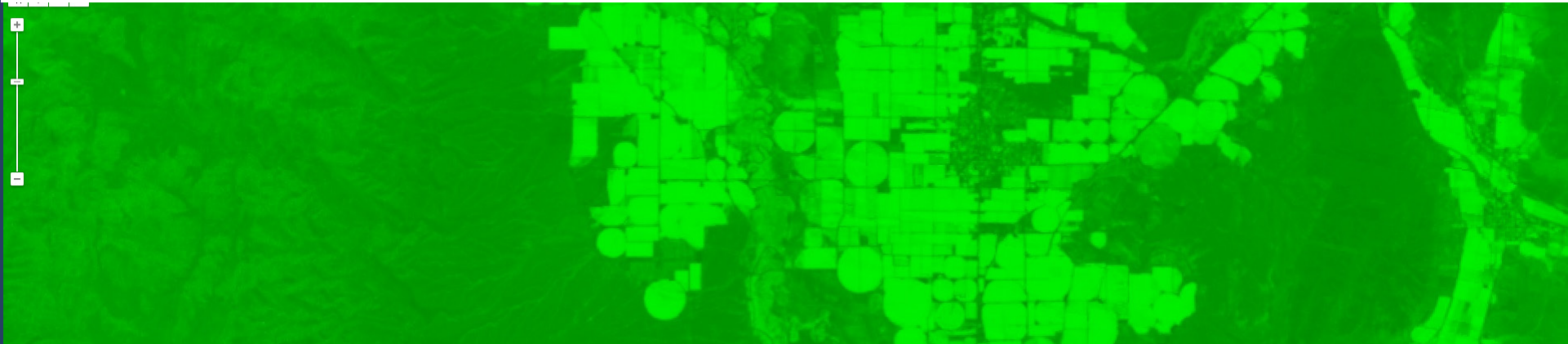
-1

+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}
);
```

The screenshot displays the Google Earth Engine web interface. The top navigation bar includes the Google Earth Engine logo, a search bar, and user controls. The left sidebar shows a list of scripts under the 'Private' section. The main area is divided into three panels: 'New Script' (containing the code from the previous block), 'Inspector' (showing the current layer's properties), and 'Console' (showing the output of the script). The map at the bottom shows a satellite image of a coastal area with a green overlay representing the filtered data.

Advanced GIS and Remote Sensing | Winter 2017 |
Ran Goldblatt



DEMO

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

1

```
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