
Google

#GeoForGood19

# Collections and Map()

## Harnessing the inner processing power of Google!!

Sai and Marisa | September 2019 | bit.ly/2lS31XQ

# Before We Start

- Learn what Collections are and how to Filter/Map over them
- The slides are at bit.ly/2lS31XQ
- Feel free to ask for any help from the TA's
- We are teaching this again tomorrow :)

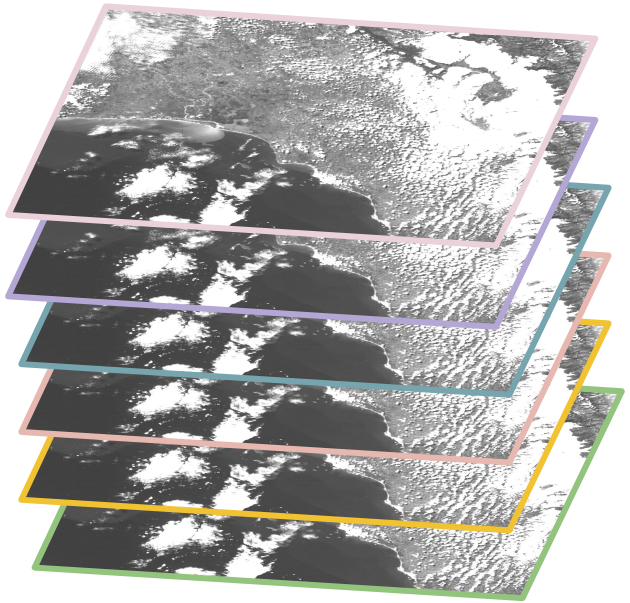
# Agenda

- Images and Features
- Collections
- Filtering
- Mapping
- Optional Hands-on Practice Examples

# Earth Engine Images

Each image has

- **Bands**
  - Band = 2D grid of pixels with a:
    - Name
    - CRS/Map Projection
    - Scale (crs_transform)
- **Properties**, including:
  - Date, Bounding-box, unique ID

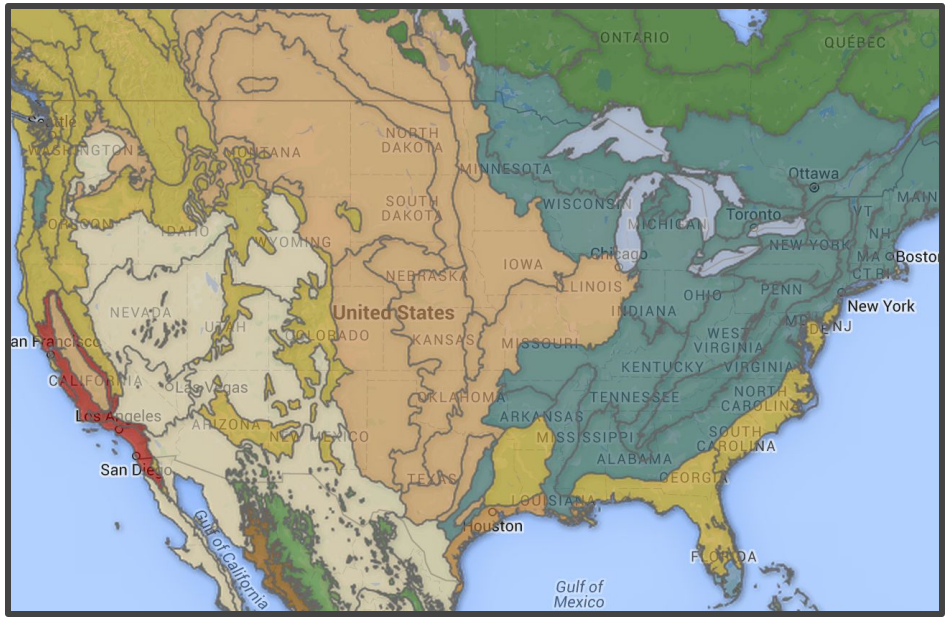


An Image with 6 bands

# Earth Engine Features

An element containing:

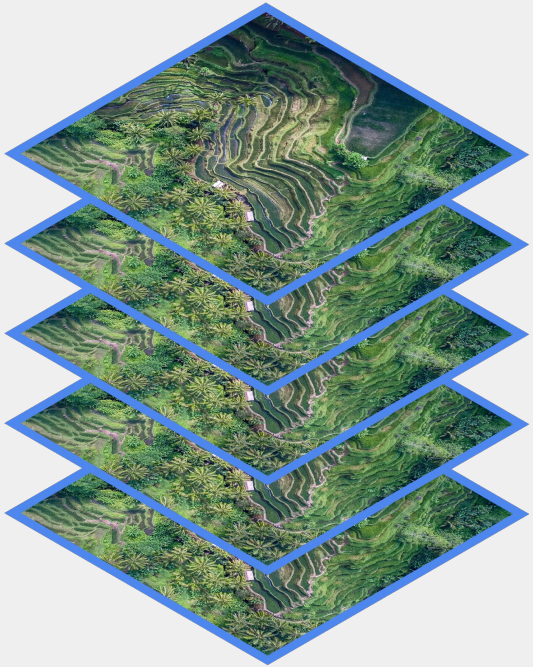
- A **Geometry**
  - Line / Point / Polygon
- List of **Properties**
  - Just like images
- Basically a **row** in a **table** with a geometry column




TNC Ecoregions

# Collection

- An **unmodifiable**, **homogenous** list of things!
- Can be a FeatureCollection or an ImageCollection



# Loading Earth Engine Images and Collection

All images and collections can be directly loaded by some unique ID.

```
var image = ee.Image('CGIAR/SRTM90_V4');
```

# Collection Examples

```
// A FeatureCollection (world boundaries).
var fc = ee.FeatureCollection(
  "USDOS/LSIB_SIMPLE/2017");

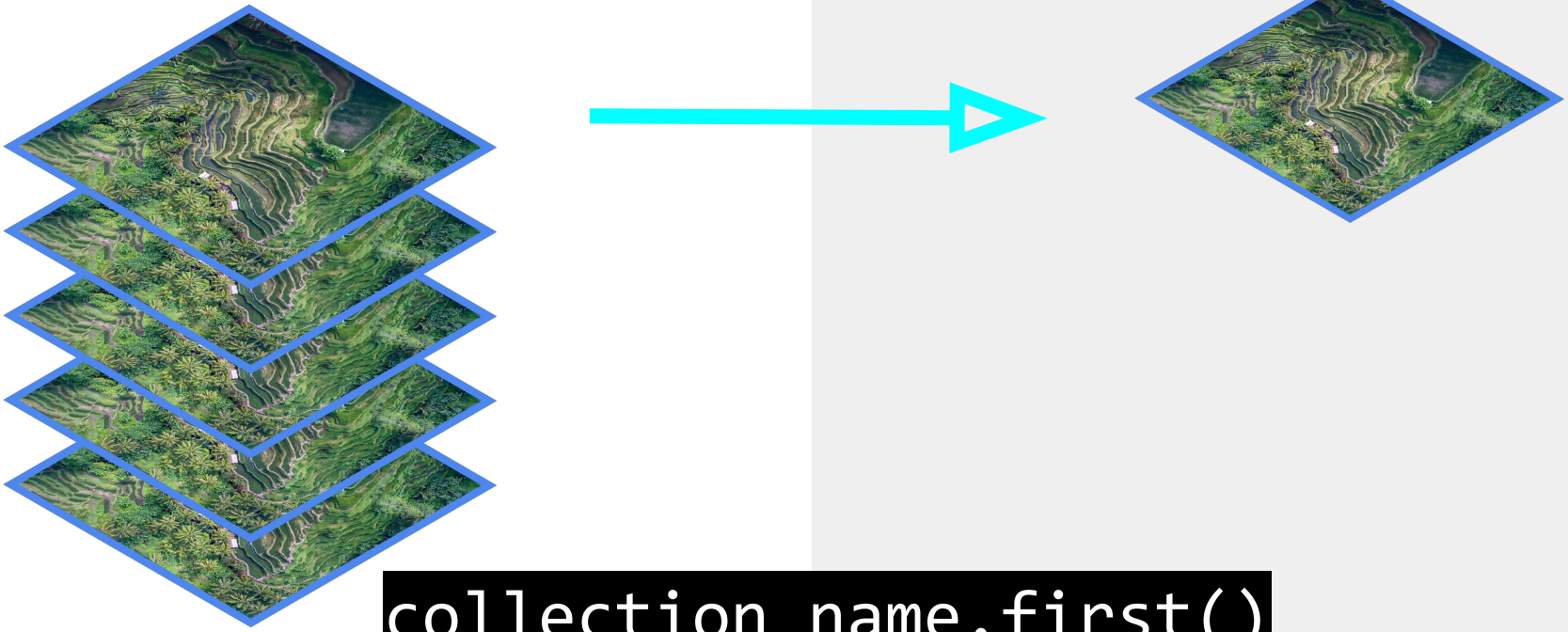
// An ImageCollection (Sentinel 2).
var ic = ee.ImageCollection(
  "COPERNICUS/S2");
```

# What can we do with Collections?

Some things we can do with Collections include but are not limited to:

- **Peek at the first value**
- Filter the Collection
- Map a function over the Collection

# Peek at the first element



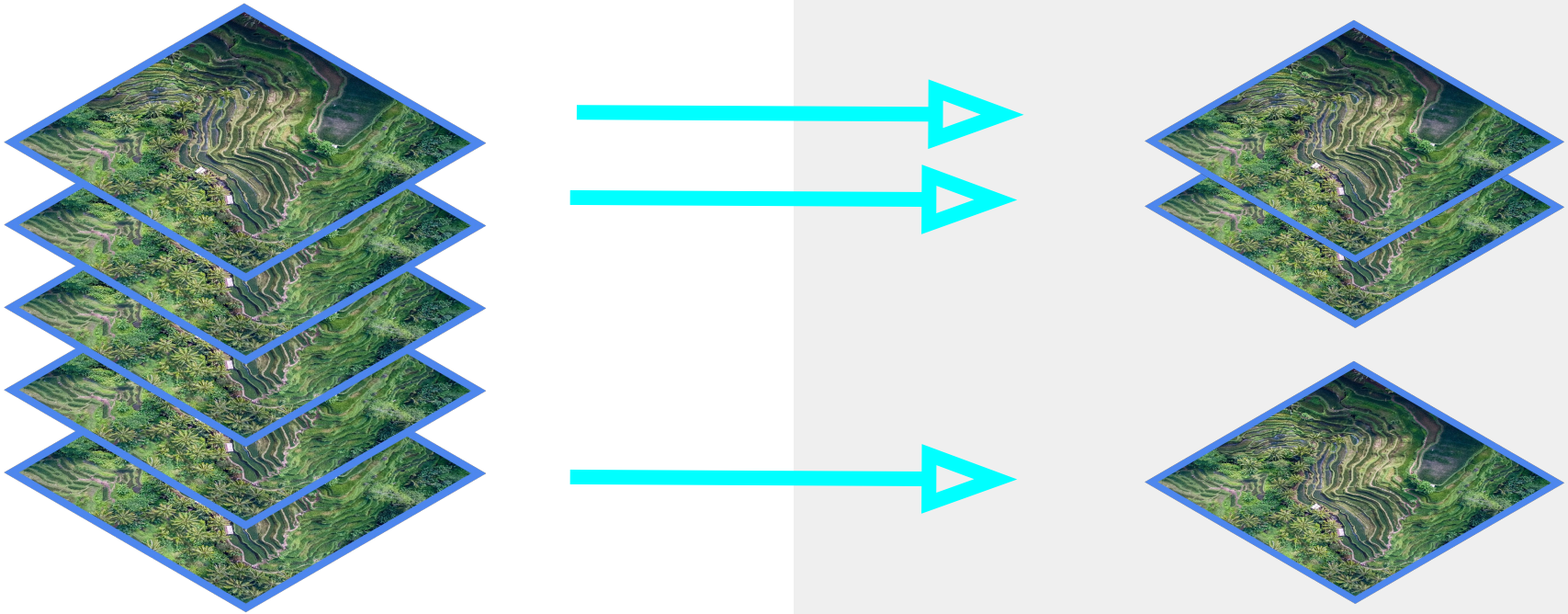
```
collection_name.first()
```

# What can we do with Collections?

Some things we can do with Collections include but are not limited to:

- Peek at the first value
- **Filter the Collection**
- Map a function over the Collection

# Filter for specific properties



```
collection_name.filter()
```

# Types of Filters

- Time (e.g., images from last month to today) - example script
- Space (e.g., images intersecting North Carolina) - example script
- Metadata (e.g., images with CLOUD_PERCENT < 10) - example script

# Filter Example - FeatureCollection

```
// A FeatureCollection.
var lsibSimple = ee.FeatureCollection(
  "USDOS/LSIB_SIMPLE/2017");

// Filter fc so we only have features
// with a specific property value.
var lsibSimpleUs = fc.filterMetadata(
  "country_co", "equals", "US");
```

# Filter Example - ImageCollection

```
// Filter a FeatureCollection
var lsibSimple = ee.FeatureCollection(
  'USDOS/LSIB_SIMPLE/2017');

var lsibSimpleUs = lsibSimple.filterMetadata(
  'country_co', equals', 'US');

// Filter an ImageCollection
var s2 = ee.ImageCollection('COPERNICUS/S2');

var s2Us2018 = s2.filterBounds(lsibSimpleUs).filterDate(
  '2018-01-01', '2019-01-01');
```

# Filter Example - ImageCollection

```
// Filter a FeatureCollection
var lsibSimple = ee.FeatureCollection(
  'USDOS/LSIB_SIMPLE/2017');

var lsibSimpleUs = lsibSimple.filterMetadata(
  'country_co', equals', 'US');

// Filter an ImageCollection
var s2 = ee.ImageCollection('COPERNICUS/S2');

var s2Us2018 = s2.filterBounds(us_fc).filterDate(
  '2018-01-01', '2019-01-01');
```

# Filter Example - ImageCollection

```
// Filter a FeatureCollection
var lsibSimple = ee.FeatureCollection(
  'USDOS/LSIB_SIMPLE/2017');

var lsibSimpleUs = lsibSimple.filterMetadata(
  'country_co', equals', 'US');

// Filter an ImageCollection
var s2 = ee.ImageCollection('COPERNICUS/S2');

var s2Us2018 = s2.filterBounds(us_fc).filterDate(
  '2018-01-01', '2019-01-01');
```

# Filter Example - ImageCollection

```
// Filter a FeatureCollection
var lsibSimple = ee.FeatureCollection(
  'USDOS/LSIB_SIMPLE/2017');

var lsibSimpleUs = lsibSimple.filterMetadata(
  'country_co', equals', 'US');

// Filter an ImageCollection
var s2 = ee.ImageCollection('COPERNICUS/S2');

var s2Us2018 = s2.filterBounds(us_fc).filterDate(
  '2018-01-01', '2019-01-01');
```

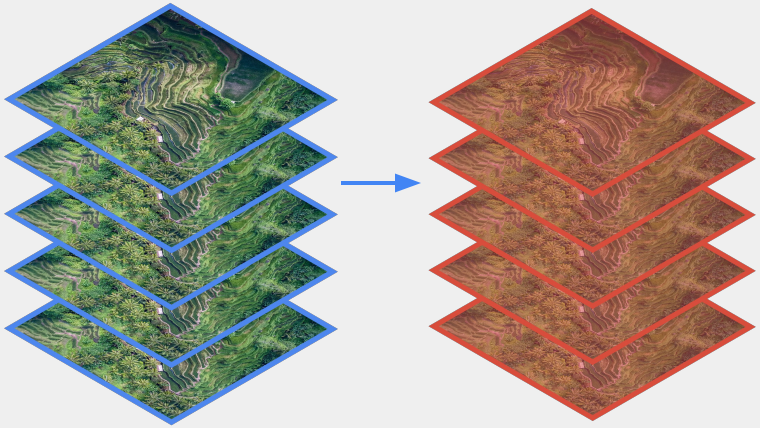
# What can we do with Collections?

Some things we can do with Collections include but are not limited to:

- Peek at the first value
- Filter the Collection
- **Map a function over the Collection**

# Map over a Collection

- Apply a function to all elements!



# What is a Function?

A function takes an input and produces an output.

- x -> [ return x + 1 ] -> y
- f(x) = x + 1;
- ```
  function(x) {
    x = ee.Number(x);
    return x.add(1);
  }
  ```
  ```

# Function Example - Function

```
// This function adds 10 to passed in number!
var addTen = function(number) {
  return number.add(10);
};

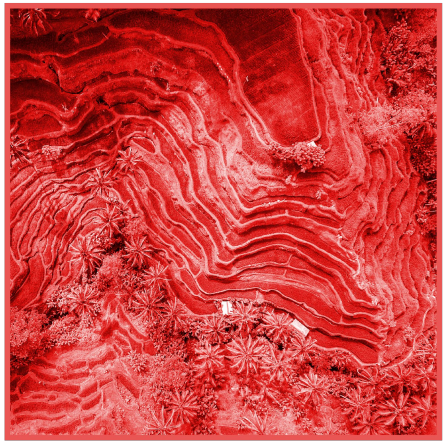
var myNumber = ee.Number(3);

// Prints 13.
print(addTen(myNumber));
```

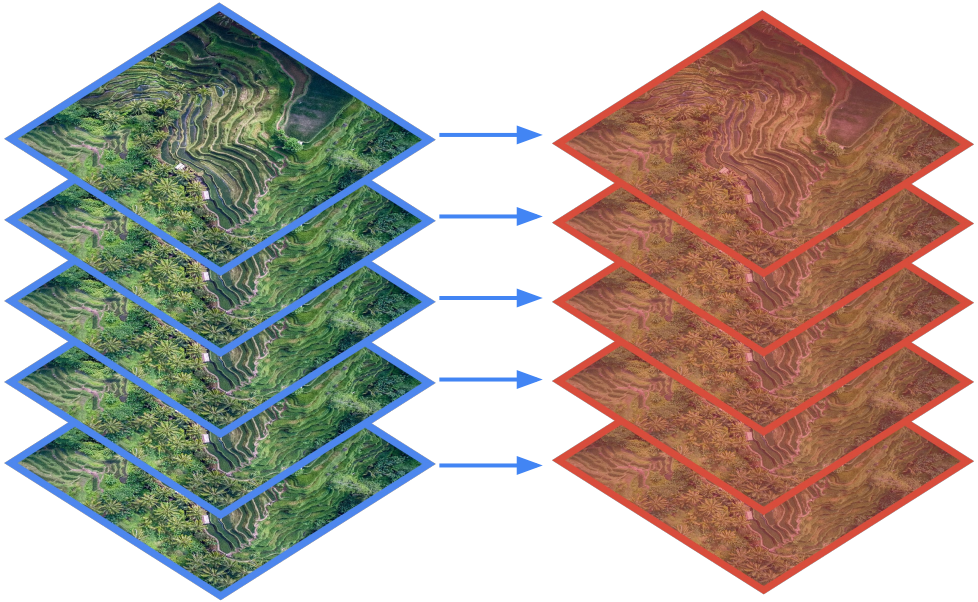
# *Get* an image

# *Apply* a function to an image

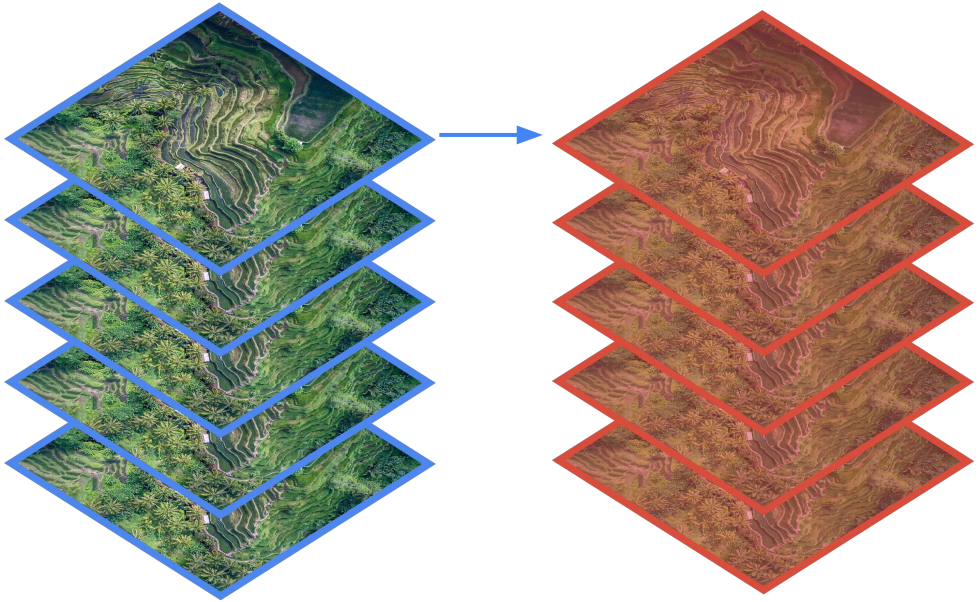
*This function alters the image*



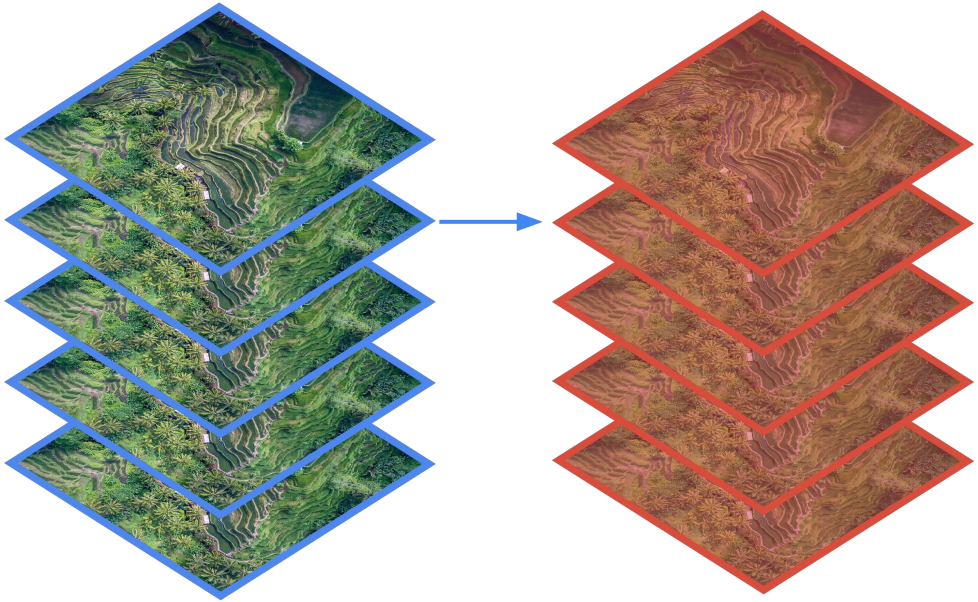
# When we have an Image Collection, we Map!



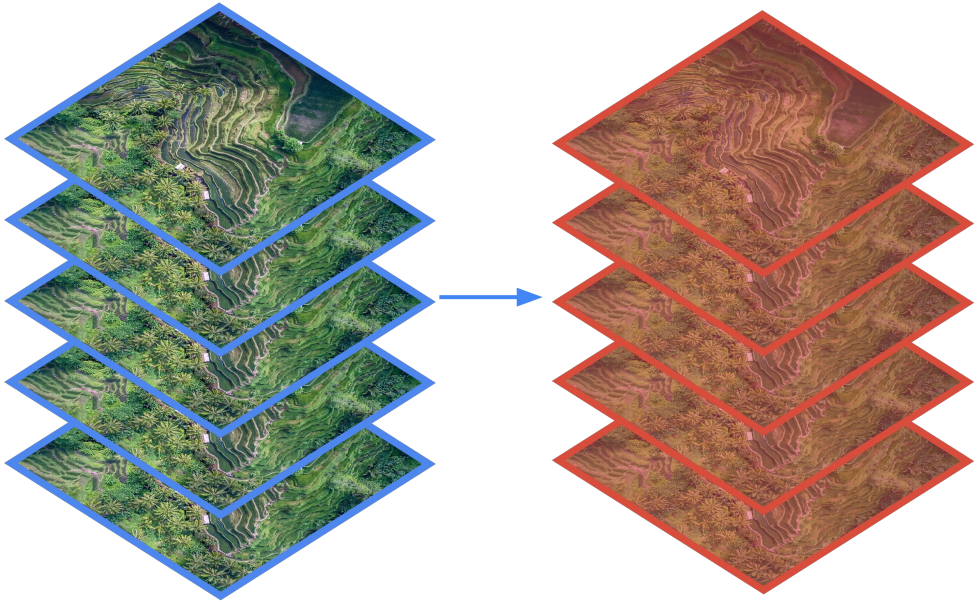
# No For-Loops!



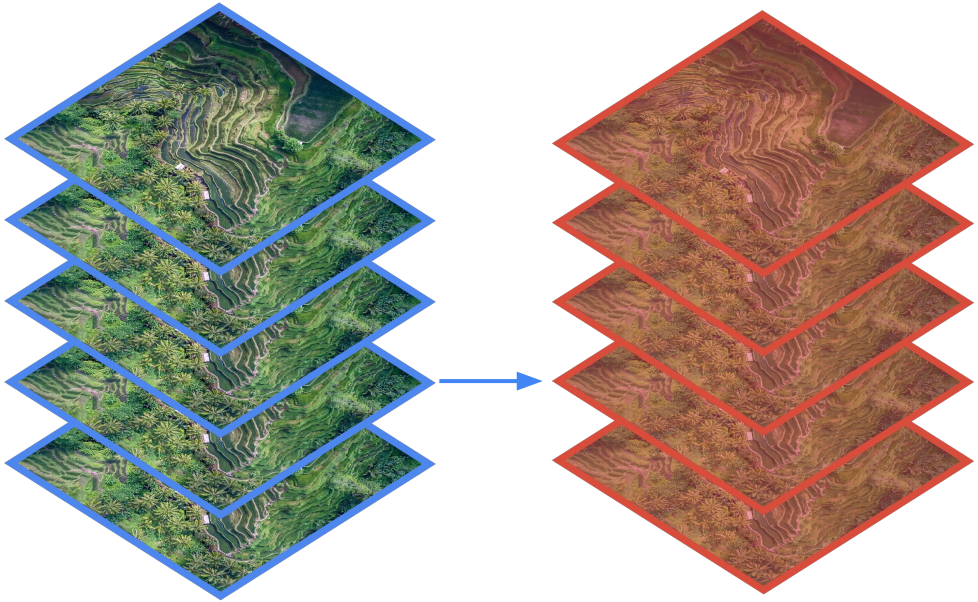
# No For-Loops!



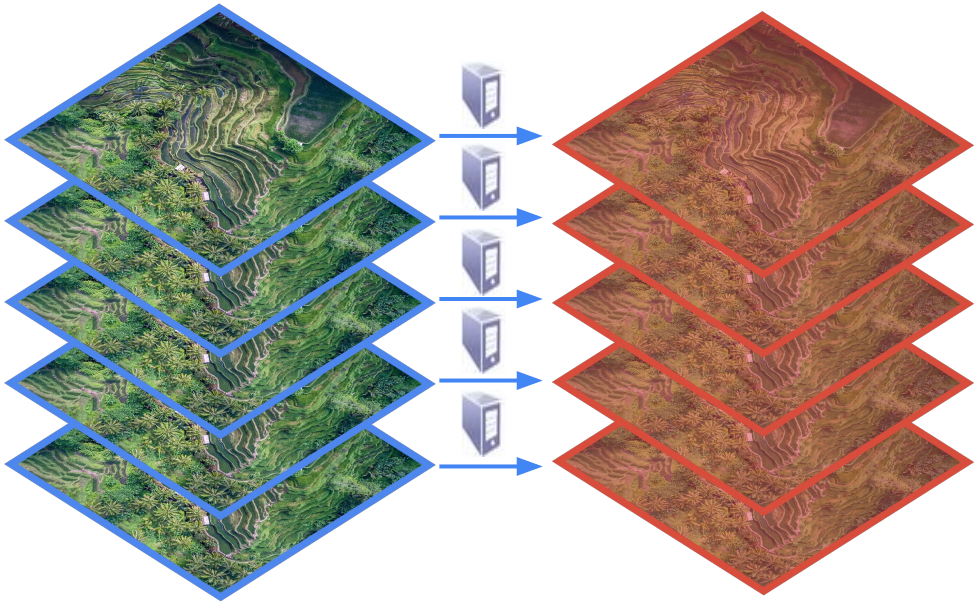
# No For-Loops!



# No For-Loops!



# We harness Google's processing power with Maps



Army of computers!!!!!!!

# What does Google see?


```
{
  "expression": {
    "result": "0",
    "values": {
      "0": {
        "functionInvocationValue": {
          "arguments": {
            "list": {
              "constantValue": [1, 2, 3]
            },
            "baseAlgorithm": {
              "functionDefinitionValue": {
                "argumentNames": ["_MAPPING_VAR_0_0"],
                "body": "1"
              }
            }
          },
          "functionName": "List.map"
        }
      },
      "1": {
        "functionInvocationValue": {
          "arguments": {
            "left": {
              "argumentReference": "_MAPPING_VAR_0_0"
            },
            "right": {
              "constantValue": 3
            }
          },
          "functionName": "Number.add"
        }
      }
```


```
ee.List([1, 2, 3]).map(function(x) {
  x = ee.Number(x);
  return x.add(3);
});
```

# Workflow tip

Filter your Collections before Mapping over them, it reduces the amount of work the server does per tile.

# But what really is a Map?

# Mapping Example: FeatureCollection

Codelab Walkthrough

# Mapping Example - FeatureCollection

```
// Load state collection.
var states = ee.FeatureCollection("TIGER/2018/States");
```

# Mapping Example - FeatureCollection

```
// Load state collection.
var states = ee.FeatureCollection("TIGER/2018/States");


// This function adds a band representing the image timestamp.
var statesWithArea = states.map(function(feature) {
  var area = feature.area();
  return feature.set({area: area});
});
```

# Mapping Example - FeatureCollection

```
// Load state collection.
var states = ee.FeatureCollection("TIGER/2018/States");


// This function adds a band representing the image timestamp.
var statesWithArea = states.map(function(feature) {
  var area = feature.area();
  return feature.set({area: area});
});

// Map the function over the collection and display result.
print(statesWithArea.sort('area'));
```

# You can Map over ImageCollections too!

Codelab Walkthrough

# Control Flow == No Go!

- If statements
- For loops may not work the way you expect.
- While the Code Editor lets you write Javascript, Earth Engine computations are **not** Javascript expressions.

You can dance if you want to!
You can leave your friends behind!

# We can also Map over an ImageCollection and produce a FeatureCollection

# And now you are ready

We have learned the following!

- Images and Features
- Collections
- Filtering
- Mapping

# Additional scripts

- Image Collections
- Mapping
- Functions (computing NDVI)