



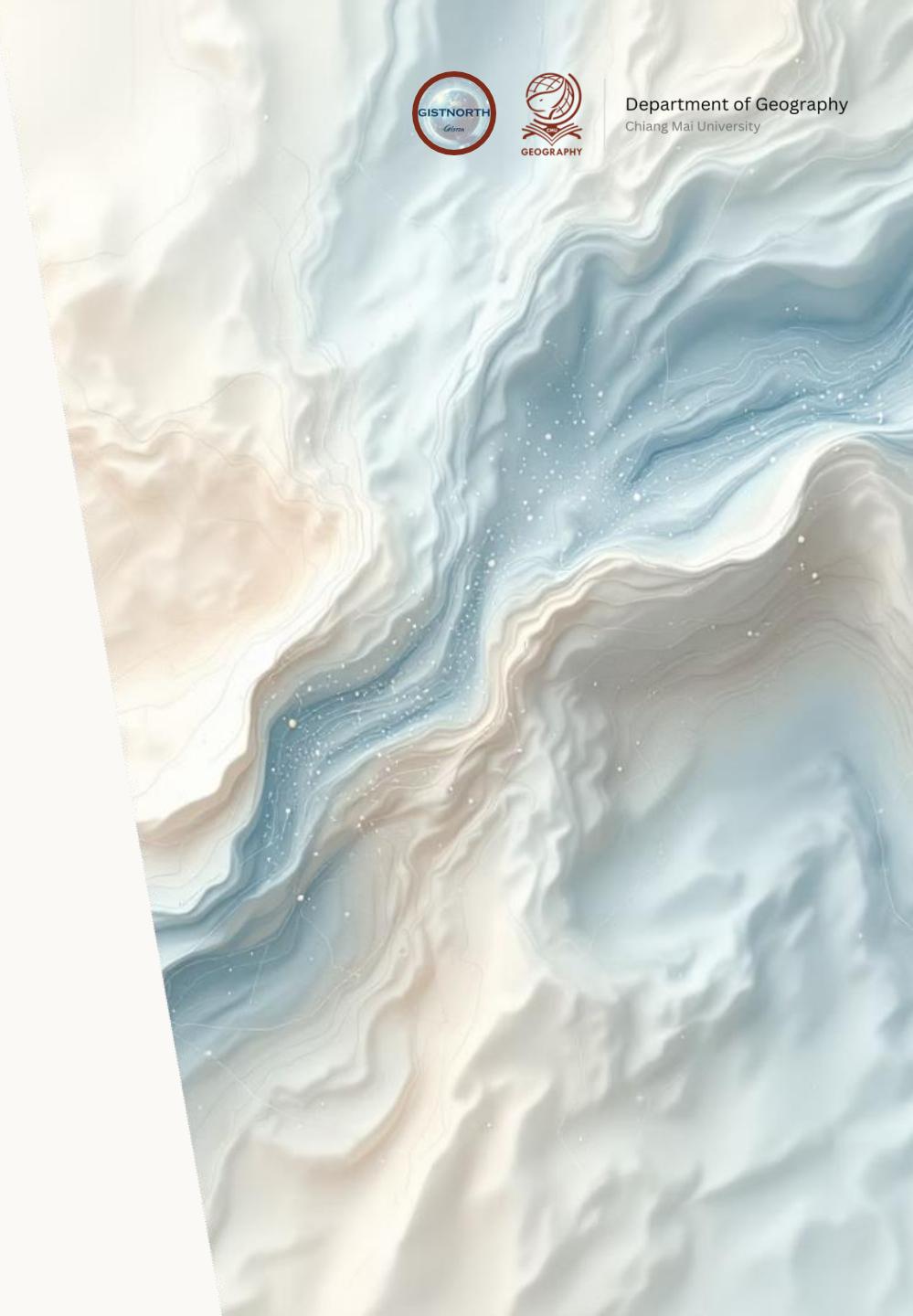
⚙️ Google Earth Engine

เพื่อสนับสนุนการบริหารงานท้องถิ่น

ศูนย์ภูมิภาคเทคโนโลยีอวกาศและภูมิสารสนเทศ

(ภาคเหนือ) (GISTNORTH)

ภาควิชาภูมิศาสตร์ คณะสังคมศาสตร์ มหาวิทยาลัยเชียงใหม่





What is Google Earth Engine?

Cloud-based Platform

Google Earth Engine (GEE) เป็นแพลตฟอร์มบนคลาวด์ที่ทรงประสิทธิภาพ ที่พัฒนาโดย Google และได้รับการออกแบบมาโดยเฉพาะสำหรับการวิเคราะห์ภูมิสารสนเทศขนาดใหญ่

Geospatial Big Data Analysis

เหมาะสมอย่างยิ่งในการประมวลผลและวิเคราะห์ชุดข้อมูลภูมิสารสนเทศขนาดใหญ่ โดยมุ่งเน้นที่ภาพถ่ายดาวเทียมและข้อมูลเชิงพื้นที่อื่น ๆ เป็นหลัก

Streamlined Operations

GEE ดำเนินการในระบบคลาวด์ทั้งหมด จึงไม่จำเป็นต้องให้ผู้ใช้ดาวน์โหลดหรือจัดเก็บชุดข้อมูลจำนวนมากไว้ในเครื่อง ทำให้กระบวนการทำงานมีประสิทธิภาพขึ้น และลดความต้องการโครงสร้างพื้นฐาน

สามารถเข้าถึงแพลตฟอร์มอันทรงพลังนี้ได้อย่างง่ายดายผ่าน earthengine.google.com ซึ่งมอบความสามารถทันทีเพื่อตอบสนองความต้องการในการวิเคราะห์ที่หลากหลาย

The Genesis of GEE



2010 Launch

เปิดตัวครั้งแรก 2 ธันวาคม พ.ศ. 2553 โดย Rebecca Moore และทีม

Google Earth Outreach

1

2

3

4

Research Facilitation

มีจุดประสงค์เพื่อช่วยนักวิจัยวิเคราะห์ข้อมูลภาพถ่ายดาวเทียมขนาดใหญ่
ได้ง่ายและรวดเร็ว

Global Impact
แพลตฟอร์มนี้ได้รับการพัฒนาเพื่อสนับสนุนการประยุกต์ใช้ด้านสิ่งแวดล้อม
และสังคมในระดับโลก โดยให้ข้อมูลเชิงลึกที่สำคัญต่อความยั่งยืนและการ
ดำเนินงานด้านมนุษยธรรม สามารถถูกรายละเอียดเพิ่มเติมได้จากบทความ
ที่เกี่ยวข้อง:

<https://www.sciencedirect.com/science/article/pii/S0034429317307224> Made with GEE

ข้อดีและจุดเด่นของ GEE



Cloud Computing Leveraging

GEE ใช้เทคโนโลยีคลาวด์ คอมพิวติ้งที่ล้ำสมัยของ Google ทำให้มีจำเป็นต้องดาวน์โหลดหรือจัดเก็บข้อมูลไว้ในเครื่อง ซึ่งช่วยลดต้นทุนด้านโครงสร้างพื้นฐานได้อย่างมาก



Vast Data Repository

คลังข้อมูลขนาดมหาศาล GEE มีคลังข้อมูลขนาดใหญ่กว่า 80 เพ特ะไบต์ ซึ่งรวมถึงข้อมูลดาวเทียมที่สำคัญจาก Landsat (ย้อนหลังถึงปี 1972), Sentinel และ MODIS ทำให้เป็นแหล่งข้อมูลที่มีคุณค่าสำหรับการวิเคราะห์ในหลากหลายด้าน



Parallel Processing Power

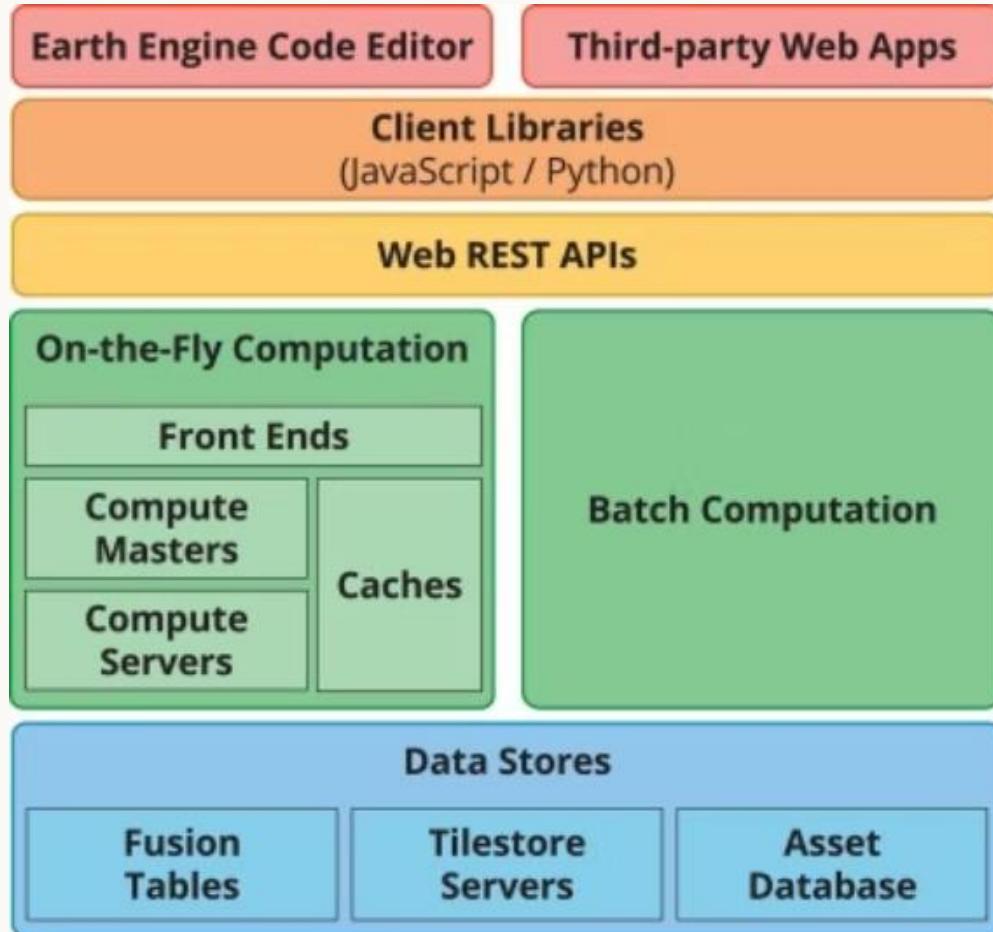
พลังการประมวลผลแบบขนานแพลตฟอร์มนี้มีความสามารถในการประมวลผลแบบขนานซึ่งช่วยลดเวลาในการวิเคราะห์ชุดข้อมูลขนาดใหญ่ได้อย่างมาก ทำให้สามารถได้ผลลัพธ์อย่างรวดเร็ว ซึ่งมีความสำคัญอย่างยิ่งสำหรับโครงการที่ต้องการข้อมูลทันเวลา



Flexible API Access

การเข้าถึง API ที่ยืดหยุ่น GEE รองรับการใช้งาน API ได้ทั้งผ่าน JavaScript (ผ่าน Code Editor) และ Python ซึ่งมอบความยืดหยุ่นให้กับนักพัฒนาแพลตฟอร์มนี้เปิดให้ใช้งานพร้อมทันทีไม่ต้องติดตั้งซอฟต์แวร์ใดๆ ก็ได้ ทำให้สามารถนำข้อมูลและวิเคราะห์จาก GEE ไปใช้ในงานวิจัย โครงการเพื่อประโยชน์สาธารณะ

System Architecture Overview



องค์ประกอบหลัก

โครงสร้างของ Google Earth Engine ถูกออกแบบระบบโครงสร้างพื้นฐานที่แข็งแกร่ง รองรับการประมวลผลเชิงพื้นที่ในระดับที่สามารถขยายได้ โดยมีองค์ประกอบหลักคือ คลังข้อมูลขนาดใหญ่ ทรัพยากรการประมวลผลแบบกระจาย และชั้น API ที่ซับซ้อน

ข้อมูลทั้งหมดถูกจัดเก็บในระบบคลาวด์ของ Google ซึ่งช่วยให้สามารถเข้าถึงได้อย่างรวดเร็วและรองรับการประมวลผลแบบขนาดผ่านเซิร์ฟเวอร์หลายตัว วิธีการแบบกระจายนี้ช่วยให้ระบบมีประสิทธิภาพสูงและเชื่อถือได้

ส่วนติดต่อผู้ใช้

ผู้ใช้สามารถโต้ตอบกับ GEE ได้หลัก ๆ ผ่าน JavaScript Code Editor หรือ Python API ซึ่งเปิดให้เข้าถึงข้อมูลและอัลกอริทึมโดยตรง โดยแพลตฟอร์มจะจัดการความซับซ้อนของการประมวลผลเบื้องหลังไว้ทั้งหมด การออกแบบเช่นนี้ช่วยให้ผู้ใช้และนักพัฒนาสามารถมุ่งเน้นไปที่การวิเคราะห์ข้อมูล โดยไม่ต้องจัดการกับระบบประมวลผลหรือการจัดเก็บข้อมูลขนาดใหญ่ เอง ถือเป็นโซลูชันแบบครบวงจรสำหรับการสังเกตการณ์โลกในระดับขนาดใหญ่

ความสำคัญของ GEE ในการบริหารงานท้องถิ่น

Budgetary Constraints Mitigation

การเข้าถึง GEE ได้ฟรีสำหรับวัตถุประสงค์ที่ไม่ใช่เชิงพาณิชย์ ซึ่งถือเป็นทางออกที่สำคัญต่อข้อจำกัดด้านงบประมาณของหน่วยงานท้องถิ่น โดยเปิดโอกาสให้สามารถวิเคราะห์ข้อมูลภูมิสารสนเทศขั้นสูงได้โดยไม่ต้องลงทุนจำนวนมากกับซอฟต์แวร์หรือฮาร์ดแวร์ที่มีลิขสิทธิ์

Reduced Data Management Burden

แนวทางแบบคลาวด์เน็ตฟ์ของแพลตฟอร์มนี้ช่วยขจัดความจำเป็นในการจัดเก็บและจัดการชุดข้อมูลขนาดใหญ่ในเครื่อง ทำให้หน่วยงานท้องถิ่นสามารถลดภาระด้านทรัพยากริโอที และลดต้นทุนในการดำเนินงานได้อย่างมีประสิทธิภาพ

Addressing Skill and Infrastructure Gaps

ช่วยเหลือเลี้ยงข้อจำกัดในด้านบุคลากรและโครงสร้างพื้นฐานการประมวลผลประสิทธิภาพสูง และทำให้การเข้าถึงความสามารถในการวิเคราะห์ที่มีประสิทธิภาพสำหรับส่วนราชการต่าง ๆ ขนาดเล็กมีความเท่าเทียมกันมากขึ้น

Timely and Continuous Data

ให้การเข้าถึงข้อมูลสตรีมที่ทันสมัยและต่อเนื่อง ซึ่งมีความสำคัญต่อการตรวจสอบแบบเรียลไทม์และการตอบสนองอย่างทันท่วงที ต่อสถานการณ์ที่เปลี่ยนแปลงไป ซึ่งช่วยเพิ่มความคล่องตัวและประสิทธิภาพของการบริหารจัดการในท้องถิ่น

ตัวอย่างการประยุกต์ใช้งาน GEE



ด้านการวางแผนเมืองและการติดตามการใช้ประโยชน์ที่ดิน

- วิเคราะห์การเปลี่ยนแปลงพื้นที่สีเขียว
 - การขยายตัวของเมือง
 - การตรวจสอบพื้นที่สาธารณะ
- ด้านการจัดการภัยพิบัติ
- การประเมินพื้นที่น้ำท่วม
 - การติดตามพื้นที่เสี่ยงดินถล่ม
 - ประเมินความเสียหายจากการถล่มหรือไฟป่า



การประยุกต์ใช้ GEE: ทรัพยากรธรรมชาติและโครงสร้างพื้นฐาน

Natural Resource & Environmental Management



ด้านการจัดการทรัพยากรธรรมชาติและสิ่งแวดล้อม

- ติดตามการเปลี่ยนแปลงพื้นที่ป่าไม้
- การประเมินคุณภาพน้ำเบื้องต้น (เช่น ความชุ่ม)
- ติดตามผลกระทบจากมลพิษทางสิ่งแวดล้อม



Infrastructure Development & Monitoring

ด้านโครงสร้างพื้นฐาน

- ติดตามการเปลี่ยนแปลงเส้นทางคมนาคม
- การประเมินผลกระทบจากการก่อสร้างใหม่

GEE provides the granular detail and historical context necessary for comprehensive environmental stewardship and strategic infrastructure development, underpinning robust local governance.

Real-World Projects Utilising GEE

JRC Global Surface Water Explorer (GSWE) (GSWE)

- วิเคราะห์แหล่งน้ำผิวดินทั่วโลก ด้วยข้อมูลภาพถ่าย Landsat ตั้งแต่ปี 1984
- ประมวลผลมากกว่า 4 ล้านภาพที่ความละเอียดเชิงพื้นที่ 30 เมตร
- แสดงการเปลี่ยนแปลงของแหล่งน้ำทั่วโลกในแต่ละเดือน
- แหล่งที่มา: <https://global-surface-water.appspot.com/map>
- อ้างอิง: European Commission (2020)



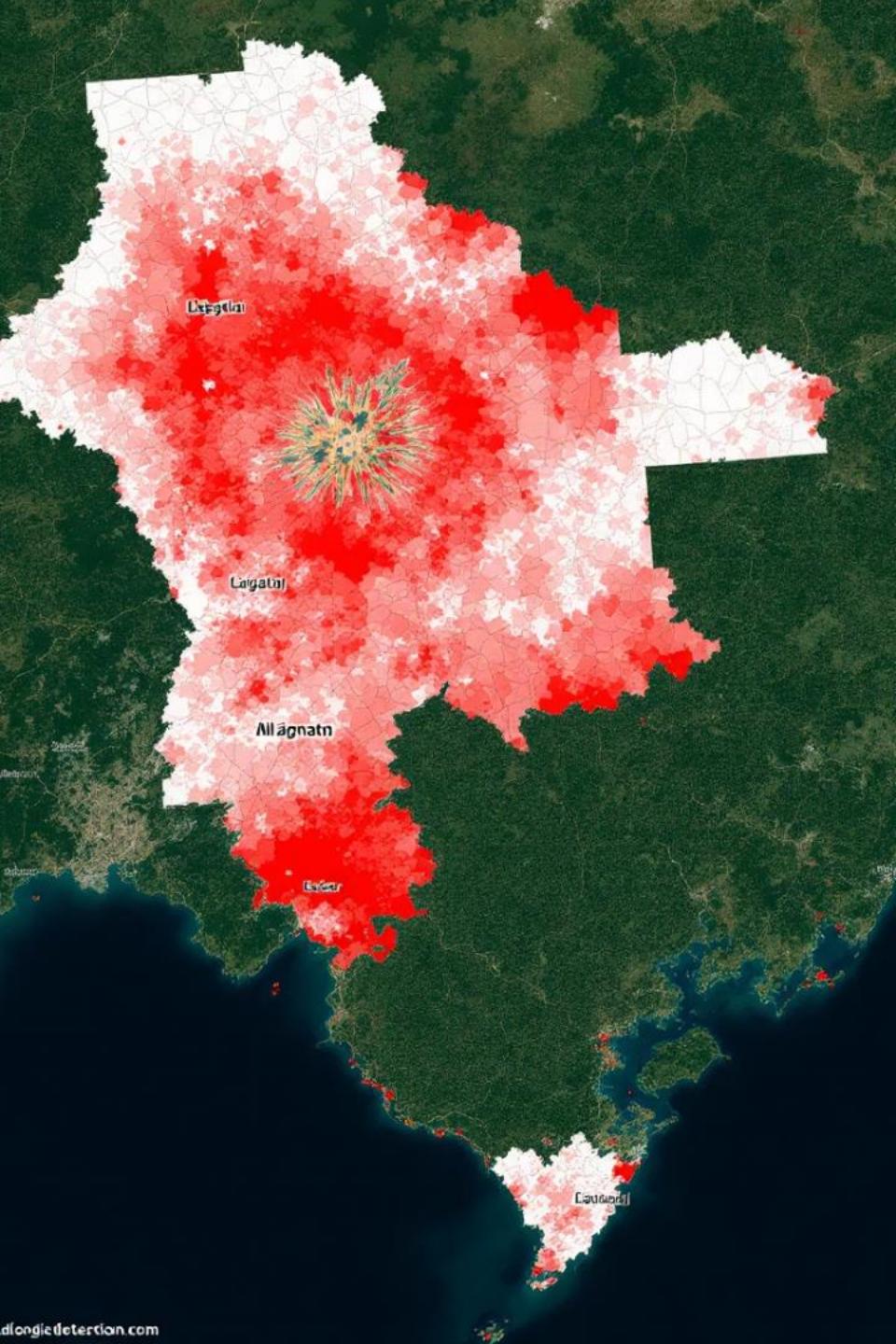
โครงการ GSWE เป็นตัวอย่างศักยภาพเชิงเปลี่ยนแปลงของ Google Earth Engine ในการดำเนินการตรวจสอบสิ่งแวดล้อมในระดับโลกและแจ้งการตัดสินใจด้านนโยบายโดยอิงจากข้อมูลจำนวนมากที่อัปเดตอย่างต่อเนื่อง

Monthly Water Extent Mapping

GSWE แสดงภาพการเปลี่ยนแปลงรายเดือนของระดับน้ำผิวดินทั่วโลก ซึ่งให้ข้อมูลเชิงลึกที่สำคัญเกี่ยวกับพลาตฟอร์มอุทกวิทยาและการจัดการทรัพยากร่น้ำ การติดตามอย่างต่อเนื่องนี้ มีค่าอย่างยิ่งสำหรับการศึกษาด้านสภาพภูมิอากาศ

Access & Reference

Interactive map ของโครงการเข้าชมได้ที่ <https://global-surface-water.appspot.com/map>,



Global Forest Watch และ GLAD Alerts



ติดตามการตัดไม้ทำลายป่า



แจ้งเตือนทันที

โครงการ Global Forest Watch และ GLAD Alerts แจ้งเตือนการเปลี่ยนแปลงป่ารายสัปดาห์

มีการแจ้งเตือนการบุกรุกพื้นที่ป่าด้วยความละเอียด 30 เมตร



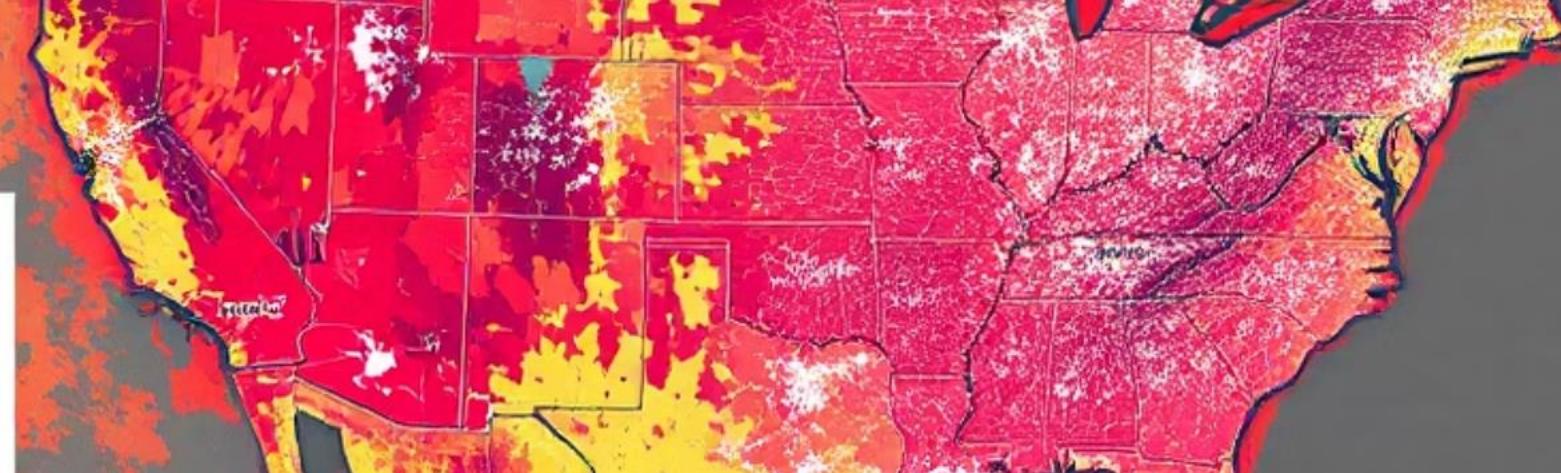
ความร่วมมือระดับโลก

พัฒนาโดย University of Maryland, World Resources Institute และ FAO

แหล่งข้อมูล: [Global Forest Watch](#) และ GLAD Alerts

Land Cover Map

Color coding, indicated the canotec categories.



โครงการ SERVIR (NASA & USAID Initiative)

สนับสนุนการเฝ้าระวัง

โครงการ SERVIR ช่วยประเทศต่างๆ เฝ้าระวังทรัพยากรและสิ่งแวดล้อม

การทำแผนที่การใช้ประโยชน์ที่ดิน

ใช้ข้อมูลดาวเทียมเพื่อจัดทำแผนที่การใช้ประโยชน์ที่ดิน (Land Cover Mapping)

การสนับสนุนทางเทคนิค

ให้การสนับสนุนทางเทคนิคและข้อมูลเชิงพื้นที่ผ่าน Google Earth Engine และหน่วยงานท้องถิ่น

แหล่งข้อมูล: SERVIR-SEA

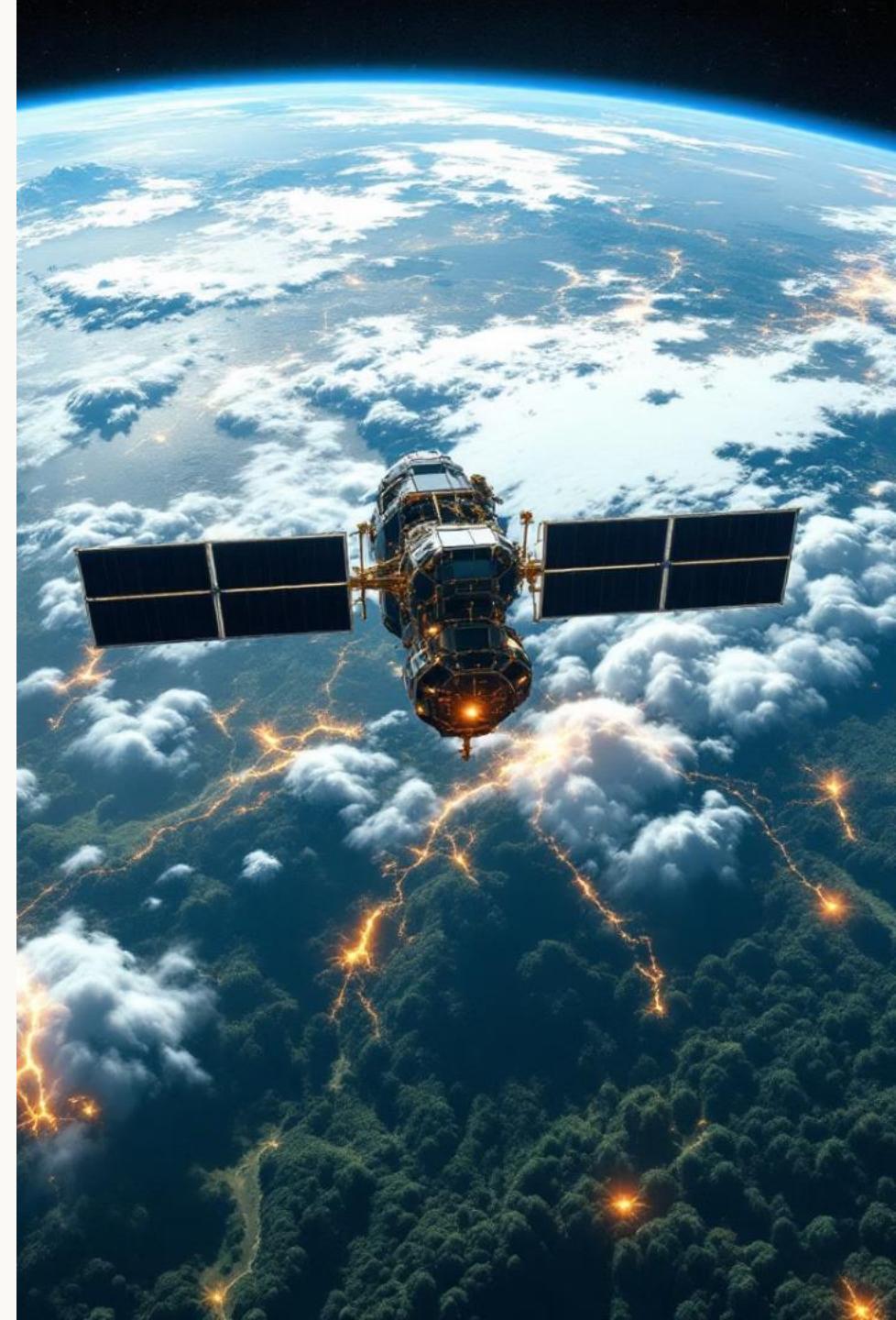
การสำรวจระยะไกล (Remote Sensing)

การนำเสนอฉบับนี้จะสำรวจบทบาทของการสำรวจระยะไกลและเทคโนโลยีอวกาศ

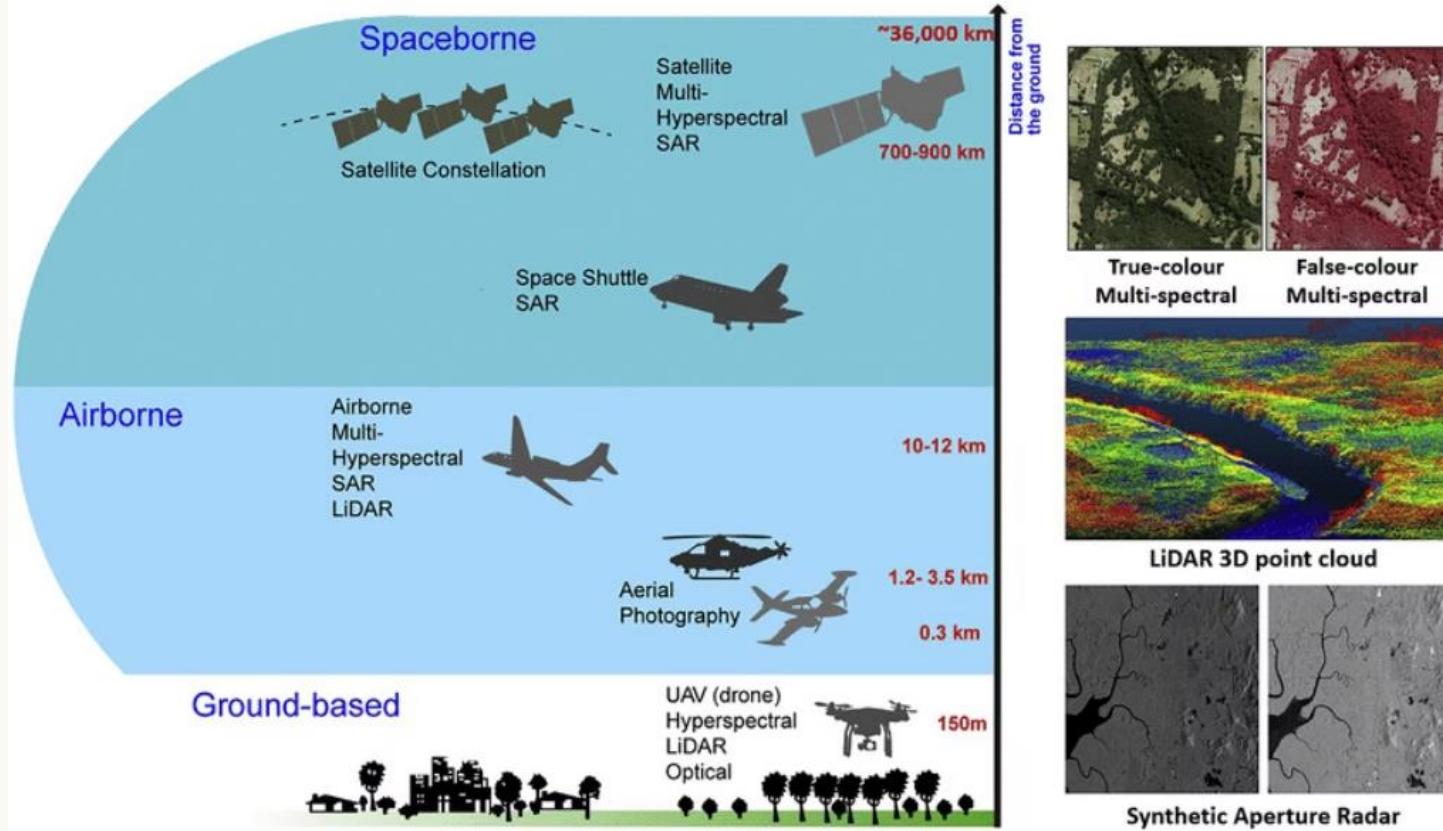
ในการติดตามและจัดการทรัพยากรธรรมชาติของประเทศไทยอย่างยั่งยืน



by ชวิต ศรีมณี



นิยามของการสำรวจระยะไกล



การเก็บข้อมูลโดยไม่สัมผัส

การสำรวจระยะไกล คือ การเก็บรวบรวมข้อมูล เกี่ยวกับวัตถุหรือปรากฏการณ์โดยไม่มีการสัมผัส โดยตรง

แพลตฟอร์มการเก็บข้อมูล

มักใช้ดาวเทียม เครื่องบิน หรือโดรนในการเก็บข้อมูลจากระยะห้ามเข้าใกล้ที่ได้เรียกว่าข้อมูลภาพจากดาวเทียม หรือข้อมูลรีโมตเซนซิ่ง

ข้อมูลภาพถ่ายดาวเทียม

การสำรวจระยะไกลแบบดิจิทัล



ภาพถ่ายดาวเทียมดิจิทัล

ใช้เซ็นเซอร์บนดาวเทียมเพื่อบันทึกข้อมูลแบบดิจิทัล



ภาพถ่ายทางอากาศดิจิทัล

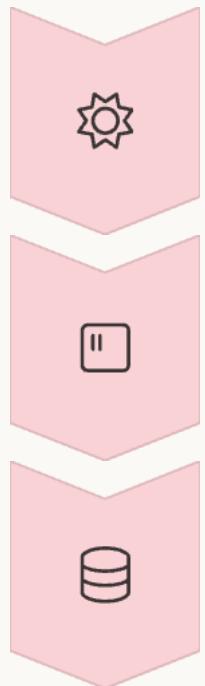
ใช้กล้องดิจิทัลบนเครื่องบินเพื่อเก็บภาพความละเอียดสูง



ภาพถ่ายจากโดรน

โดรนให้ภาพระยะใกล้และมีรายละเอียดสูง
เหมาะกับพื้นที่ขนาดเล็ก

หลักการพื้นฐานของการสำรวจระยะไกล



การสะท้อนพลังงาน

วัตถุบนพื้นโลกสะท้อนหรือปล่อยพลังงานแม่เหล็กไฟฟ้า

การเก็บข้อมูลโดยเซ็นเซอร์

เซ็นเซอร์บนดาวเทียมหรือโดรนจะเก็บพลังงานในช่วงความยาวคลื่นต่างๆ

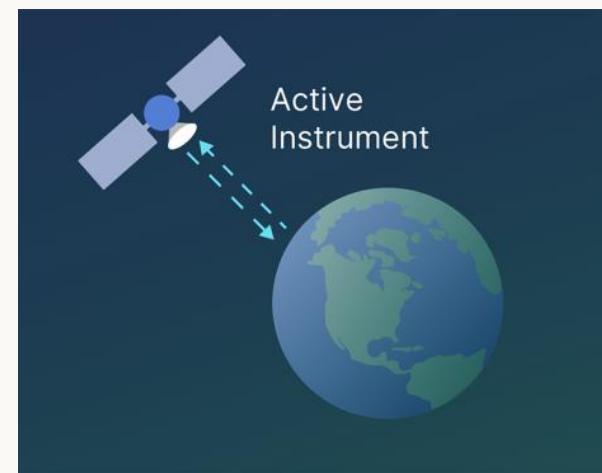
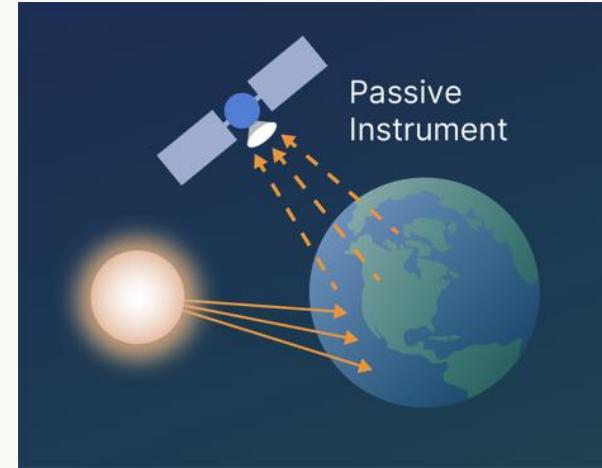
การวิเคราะห์ข้อมูล

ข้อมูลจะถูกส่งกลับสถานีภาคพื้นดิน เพื่อวิเคราะห์และตีความ



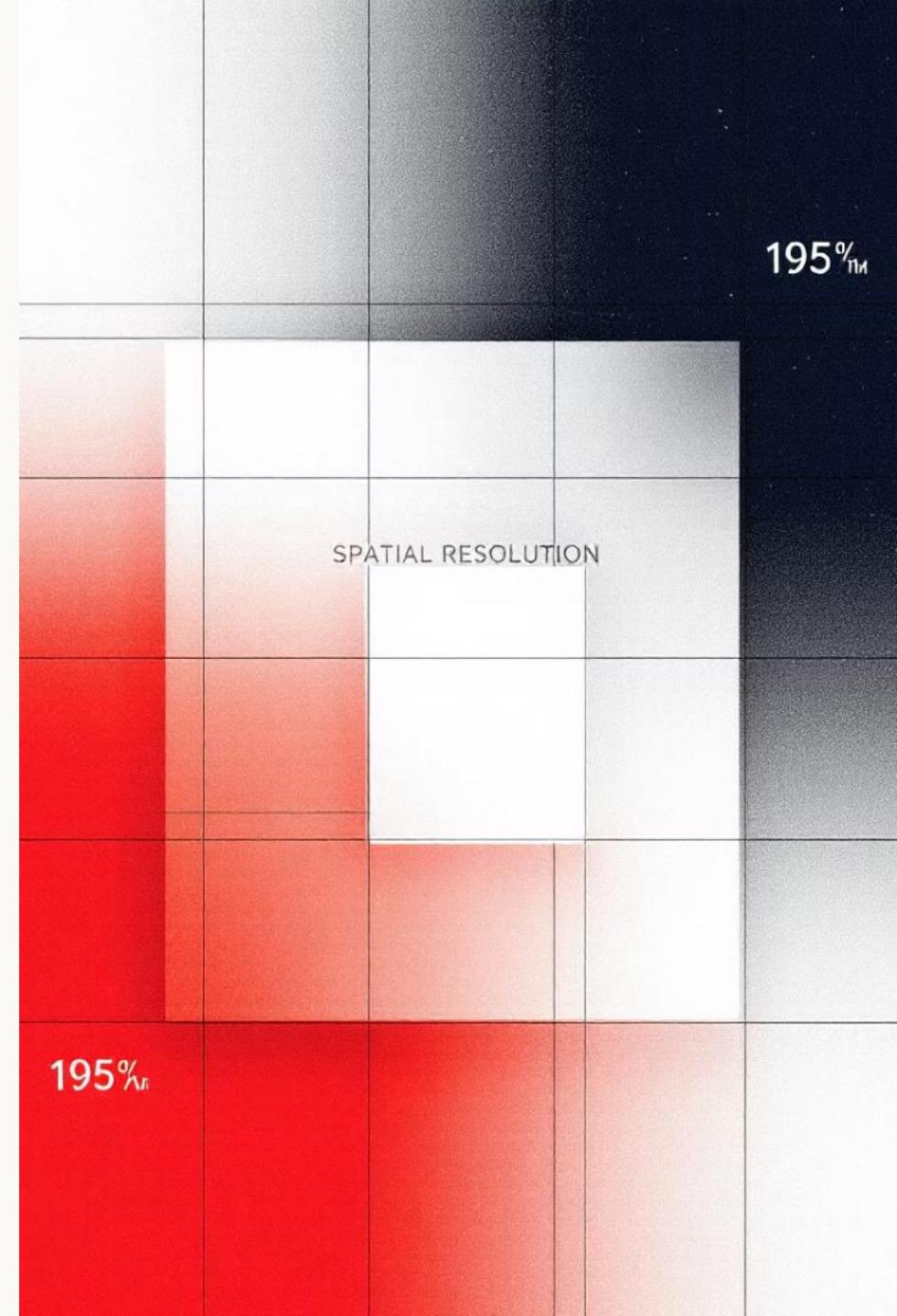
ประเภทของการสำรวจระยะไกล

- **Passive Remote Sensing** (การสำรวจแบบรับพลังงาน)
 - ดาวเทียมบันทึกพลังงานที่สะท้อนหรือแผ่ออกจากวัตถุ เช่น แสงจากดวงอาทิตย์
 - ตัวอย่างเช่น ดาวเทียม Landsat, Sentinel-2, MODIS
- **Active Remote Sensing** (การสำรวจแบบส่งพลังงาน)
 - ดาวเทียมส่งพลังงาน (เช่น ไมโครเวฟ) ลงสู่พื้นโลกแล้วบันทึกสัญญาณที่สะท้อนกลับ
 - ตัวอย่างเช่น ดาวเทียม Sentinel-1 (SAR)

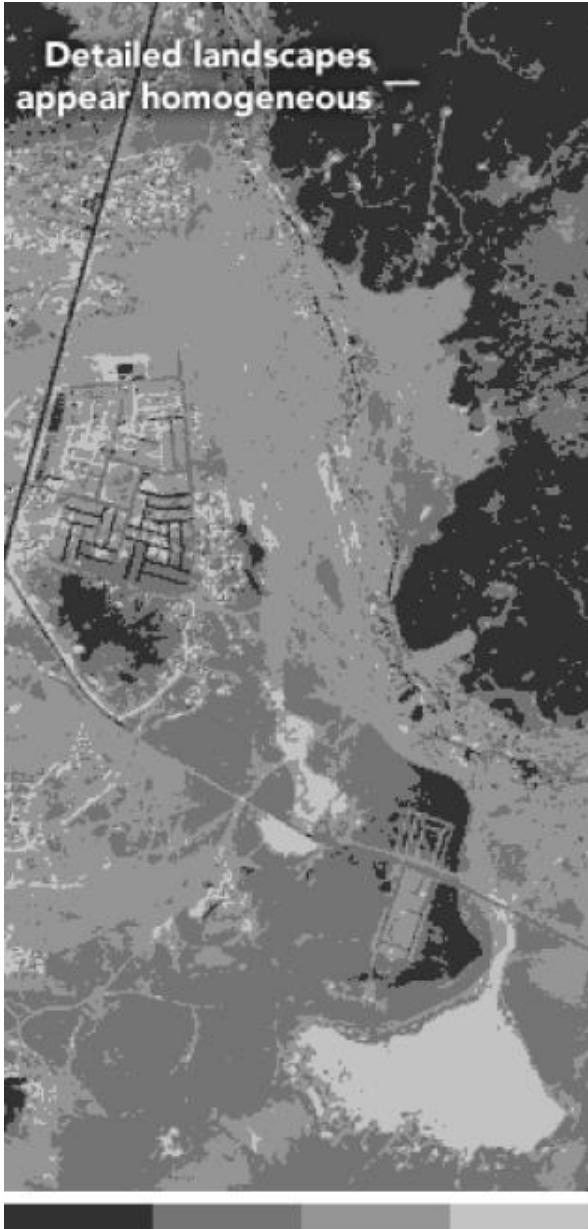


องค์ประกอบสำคัญของข้อมูลภาพถ่ายดาวเทียม

- 1 Spatial Resolution (ความละเอียดเชิงพื้นที่)
ขนาดเล็กที่สุดของพื้นที่ที่เซ็นเซอร์สามารถแยกแยะได้ เช่น 10 เมตร หรือ 30 เมตร
- 2 Spectral Resolution (ความละเอียดเชิงสเปกตรัม)
จำนวนและช่วงความยาวคลื่นที่เซ็นเซอร์บันทึกได้ เช่น RGB หรือ NIR
- 3 Temporal Resolution (ความละเอียดเชิงเวลา)
ความถี่ของการเก็บข้อมูลในตำแหน่งเดิมซ้ำๆ เช่น รายวัน หรือ รายสัปดาห์
- 4 Radiometric Resolution (ความละเอียดเชิงรังสี)
ความสามารถในการแยกแยะระดับความสว่าง เช่น 8-bit หรือ 16-bit







ตัวอย่างดาวเทียมที่นิยมใช้ใน GEE

ดาวเทียม	ความละเอียด	ความถี่ในการเก็บข้อมูล	ตัวอย่างการใช้งาน
Landsat 8	30 เมตร	16 วัน	วิเคราะห์การเปลี่ยนแปลงการใช้ที่ดิน
Sentinel-2	10-20 เมตร	5 วัน	การประเมินสุขภาพพืชพรรณ
MODIS	250-1000 เมตร	รายวัน	วิเคราะห์ไฟป่า นำท่วม
Sentinel-1 (SAR)	10 เมตร	6-12 วัน	วิเคราะห์พื้นที่นำท่วม พื้นที่การเกษตร

ดัชนีที่สำคัญจากภาพถ่ายดาวเทียม

- **NDVI (Normalized Difference Vegetation Index)**

- วิเคราะห์ความหนาแน่นและสุขภาพของพืช
- คำนวณจากแบนด์ Near-Infrared (NIR) และ Red

- **NDWI (Normalized Difference Water Index)**

- วิเคราะห์ความชื้นในพืชและแหล่งน้ำ
- คำนวณจากแบนด์ NIR และ Shortwave-Infrared (SWIR)

- **NDBI (Normalized Difference Built-up Index)**

- วิเคราะห์พื้นที่สิ่งปลูกสร้างและเขตเมือง
- คำนวณจากแบนด์ SWIR และ NIR

ความสำคัญของ Remote Sensing กับ GEE

- GEE ช่วยให้การวิเคราะห์ข้อมูล Remote Sensing สะดวกและรวดเร็ว
- มีข้อมูลย้อนหลังหลายสิบปีให้ใช้งานได้ทันที
- รองรับการคำนวณดัชนีต่างๆ ได้ง่าย เช่น NDVI, NDWI, NDBI
- ประยุกต์ใช้ในงานด้านการจัดการทรัพยากรธรรมชาติ, สิ่งแวดล้อม, ภัยพิบัติ และการเกษตร

ขั้นตอนทั่วไปในการวิเคราะห์ข้อมูล Remote Sensing ด้วย GEE

1. กำหนดพื้นที่ศึกษา (Region of Interest: ROI)
2. เลือกข้อมูลดาวเทียม (Image Collection)
3. กรองข้อมูลตามช่วงเวลาและพื้นที่
4. ประมวลผลและวิเคราะห์ (เช่น คำนวณ NDVI, NDWI)
5. แสดงผลบนแผนที่ (Visualization)
6. ส่งออกข้อมูลหรือผลลัพธ์ (Export)

วิธีในการใช้งาน Google Earth Engine

Google Earth Engine (GEE) ให้ผู้ใช้งานเข้าถึงและวิเคราะห์ข้อมูลภูมิสารสนเทศได้ผ่าน 3 ช่องทางหลัก ได้แก่

1. ใช้งานผ่าน JavaScript API (GEE Code Editor)

- **รูปแบบ:** Web-based IDE (Integrated Development Environment)
- **วิธีการใช้งาน:** เข้าผ่านเว็บไซต์ code.earthengine.google.com
- **จุดเด่น:**
 - ใช้งานง่าย เหมาะสำหรับผู้เริ่มต้น
 - ไม่ต้องติดตั้งโปรแกรมเพิ่มเติม
 - รองรับการเขียนโค้ด JavaScript โดยตรงในเบราว์เซอร์

ตัวอย่างการใช้งานผ่าน GEE Code Editor

Scripts Docs Assets New Script * Get Link Save Run Reset Apps Inspector Console Tasks

ee.Array
ee.Blob
ee.Classifier
ee.Clusterer
ee.ConfusionMatrix
ee.Date
ee.DateRange
ee.Dictionary
ee.ErrorMargin

```
75 // 9. Display the results
76 Map.centerObject(roi);
77 Map.addLayer(comp1, {bands: ['B4','B3','B2'], min:0, max:3000}, 'True Color 1', false)
78 Map.addLayer(classified1, {min:0, max:3, palette: palette}, 'Classified Jan-Mar', true)
79
80 Map.addLayer(comp2, {bands: ['B4','B3','B2'], min:0, max:3000}, 'True Color 2', false)
81 Map.addLayer(classified2, {min:0, max:3, palette: palette}, 'Classified Jul-Sep', true)
82
83
84 // 10. Optional: accuracy assessment for period 1
85 var trainTest1 = samples1.randomColumn('rnd', 42);
86 var split = 0.7;
```

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

FeatureCollection pr... JSON

Confusion matrix (1st... JSON

[[21,0,0,0],[0,1,1,0,... JSON

Overall accuracy: JSON

0.9989462592202318

Geometry Imports Layers Map Satellite

Google

GISTNORTH

Keyboard shortcuts Map data ©2025 Google 2 km Terms Report a map error

2. ใช้งานผ่าน Python API (Client Library)

- **รูปแบบ:** Python Library (earthengine-api)
- **วิธีการใช้งาน:** ติดตั้งผ่านคำสั่ง pip (pip install earthengine-api)
- **จุดเด่น:**
 - เหมาะกับงานวิจัยหรือโครงการที่ต้องการประมวลผลชั้นชั้นด้วยภาษา Python
 - สามารถใช้งานร่วมกับ Python libraries อื่นๆ เช่น NumPy, Pandas, GeoPandas
 - รองรับการใช้งานใน Jupyter Notebook, Colab และสคริปต์ Python อื่นๆ

ตัวอย่างการใช้งานผ่าน Python API

```
import ee  
ee.Initialize()  
  
image = ee.Image('COPERNICUS/S2/20240101T000000')  
url = image.getThumbURL({'min':0, 'max':3000, 'bands':['B4','B3','B2']})  
print(url)
```

3. ใช้งานผ่าน REST API (เชื่อมต่อโดยตรง)

- **รูปแบบ:** เชื่อมต่อโดยตรงกับเซิร์ฟเวอร์ GEE ผ่าน REST API
- **วิธีการใช้งาน:** ส่ง HTTP requests ตรงไปยัง GEE servers
- **จุดเด่น:**
 - เหมาะกับนักพัฒนาที่ต้องการสร้างแอปพลิเคชันหรือระบบที่ทำงานอัตโนมัติร่วมกับ GEE
 - มีความยืดหยุ่นสูงในการใช้งานร่วมกับระบบต่าง ๆ

ตัวอย่างการใช้งานผ่าน REST API

```
POST https://earthengine.googleapis.com/v1/projects/your-project-id/assets:exportImage
{
  "expression": {
    "image": {
      "assetId": "COPERNICUS/S2/20240101T000000"
    }
  },
  "fileFormat": "GEO_TIFF",
  "region": {...}
}
```

เปรียบเทียบการใช้งานทั้ง 3 วิธี

วิธีใช้งาน	ความง่าย	ความยืดหยุ่น	เหมาะสมกับ
GEE Code Editor	ง่ายที่สุด	ปานกลาง	เรียนรู้/ใช้งานทั่วไป
Python API	ปานกลาง	สูง	งานวิจัยและงานวิเคราะห์ข้อมูลสูง
REST API	ยากที่สุด	สูงสุด	การพัฒนาแอปพลิเคชันและระบบอัตโนมัติ

Google Cloud คืออะไร?



Google Cloud

- **Google Cloud** คือบริการคลาวด์คอมพิวติ้งของ Google ที่ให้บริการพื้นที่จัดเก็บข้อมูล, เครื่องมือในการวิเคราะห์ข้อมูล, และบริการประมวลผลผ่านอินเทอร์เน็ต
- ช่วยให้องค์กรและนักวิจัยสามารถใช้งานทรัพยากรด้านไอทีได้อย่างมีประสิทธิภาพสูง โดยไม่จำเป็นต้องลงทุนในฮาร์ดแวร์หรือซอฟต์แวร์ราคาแพง
- GEE ถูกสร้างขึ้นบนแพลตฟอร์ม Google Cloud
- การประมวลผลทั้งหมดของ GEE ดำเนินการบนเซิร์ฟเวอร์ Google Cloud
- ช่วยให้นักวิจัยสามารถเข้าถึงข้อมูลและประมวลผลได้ในเวลาอันรวดเร็ว

เหตุผลที่ต้องเปิดใช้งาน Google Cloud ก่อนใช้งาน GEE

- การใช้งาน GEE จำเป็นต้องเชื่อมโยงกับ Google Cloud Project เพื่อจัดการบัญชีผู้ใช้และจัดการทรัพยากรในการประมวลผล
- Google Cloud Project ทำหน้าที่ควบคุมสิทธิ์ในการใช้งาน GEE และช่วยให้ผู้ใช้งานสามารถจัดการข้อมูลส่วนตัว (Assets) ได้สะดวกยิ่งขึ้น
- ช่วยในการบริหารจัดการข้อมูลขนาดใหญ่ (เช่น การนำเข้า/ส่งออกข้อมูล, การจัดเก็บ, การสำรองข้อมูล, การเข้าถึงข้อมูลอย่างปลอดภัย)

GEE Code Editor (JavaScript) คืออะไร

- เป็นอินเทอร์เฟซหลักของ Google Earth Engine
- ใช้สำหรับเขียนโค้ด (ด้วย JavaScript) เพื่อวิเคราะห์ข้อมูลภูมิสารสนเทศบนระบบคลาวด์
- ทำงานผ่านเว็บเบราว์เซอร์ ไม่ต้องติดตั้งโปรแกรมเพิ่มเติม

วิธีเข้าใช้งาน

- เข้าไปที่เว็บไซต์ earthengine.google.com
- คลิก "Sign Up" หรือ "Get Started"
- ใช้บัญชี Google (เช่น Gmail) ในการลงทะเบียน
- ระบุวัตถุประสงค์ชัดเจน เช่น ใช้ในการศึกษา วิจัย หรือประโยชน์สาธารณะ

The screenshot shows the Google Earth Engine interface. At the top, there's a navigation bar with 'Google Earth Engine' and a search bar. Below it, a menu bar has 'Scripts' selected, followed by 'Docs', 'Assets', and various action buttons: 'Get Link', 'Save', 'Run', 'Reset', and 'Apps'. The main area is divided into several sections:

- Script Editor:** A code editor window showing a portion of a JavaScript script. The code defines a namespace 'app' and creates UI panels.
- Map View:** A map showing satellite imagery of a coastal area. Labels include '69E', '40D', 'Mc', and 'Re'. The map includes standard controls for zooming (+/-), panning (C), and switching between 'Layers', 'Map', and 'Satellite' modes.
- Visualization Controls:** Two dropdown menus labeled '2) Select an image' and '3) Select a visualization'. The first dropdown is set to 'LC08_026042_20170515'. The second dropdown is set to 'Atmospheric (B7/B6/B5)'. Below these, a descriptive text states: 'Coast lines and shores are well-defined. Vegetation appears blue.'

การเปิดใช้งาน Google Cloud Project

หลังสมัคร GEE ต้องเปิดใช้งาน Google Cloud

- สร้าง Project ใน Google Cloud และเลือก non-commercial use
- อธิบายเหตุผลในการใช้งานให้ชัดเจนเพื่อขออนุมัติการใช้งานจาก Google

หลังจากได้รับอนุมัติแล้ว สามารถเข้าที่เว็บไซต์:

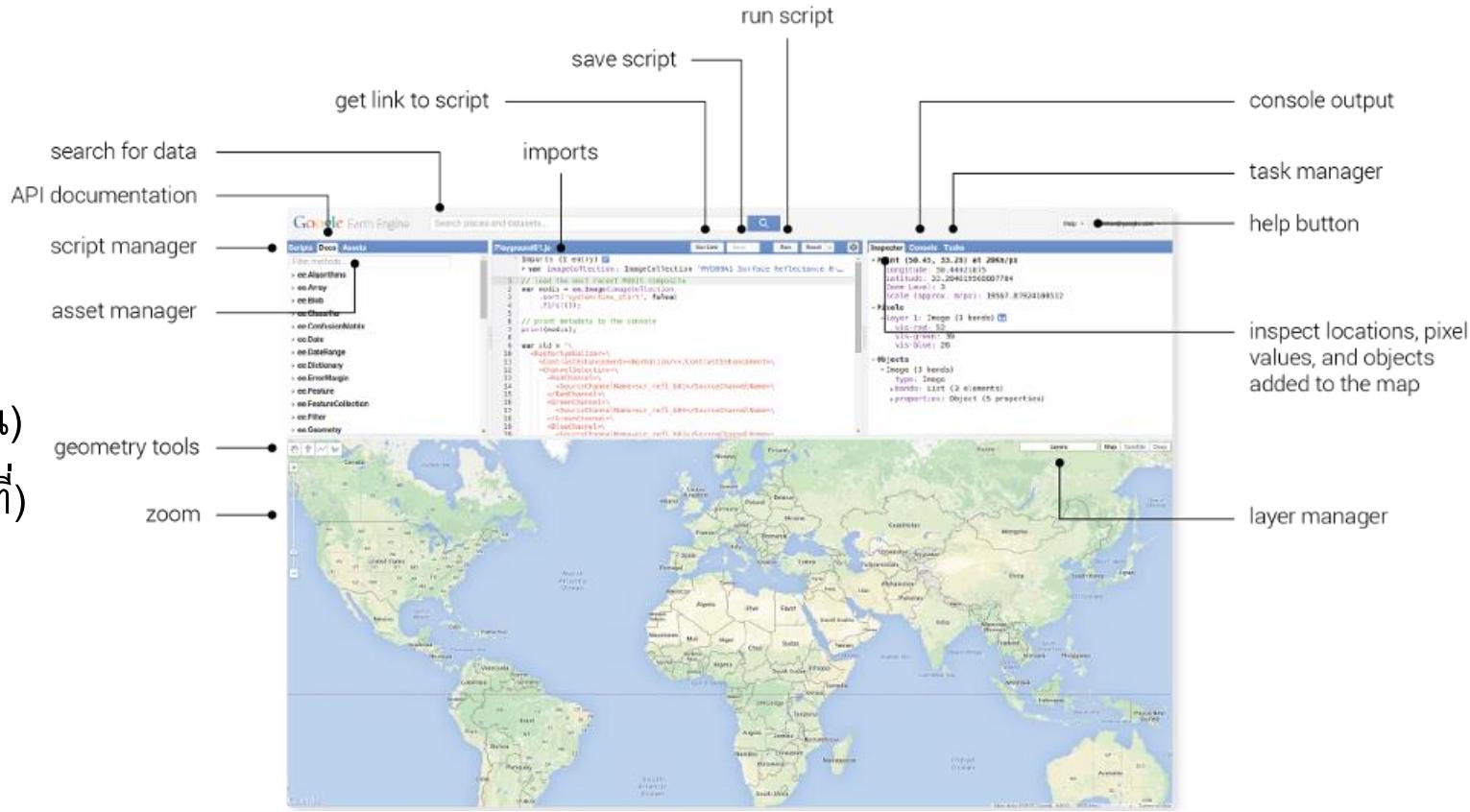
- URL: code.earthengine.google.com
- ล็อกอินด้วยบัญชี Google ที่อนุมัติ
- แนะนำการเปิดใช้งานครั้งแรกและการเลือก Cloud Project ที่สร้างไว้

- ลงทะเบียนการใช้งาน บทที่ 2

ส่วนประกอบของ GEE Code Editor

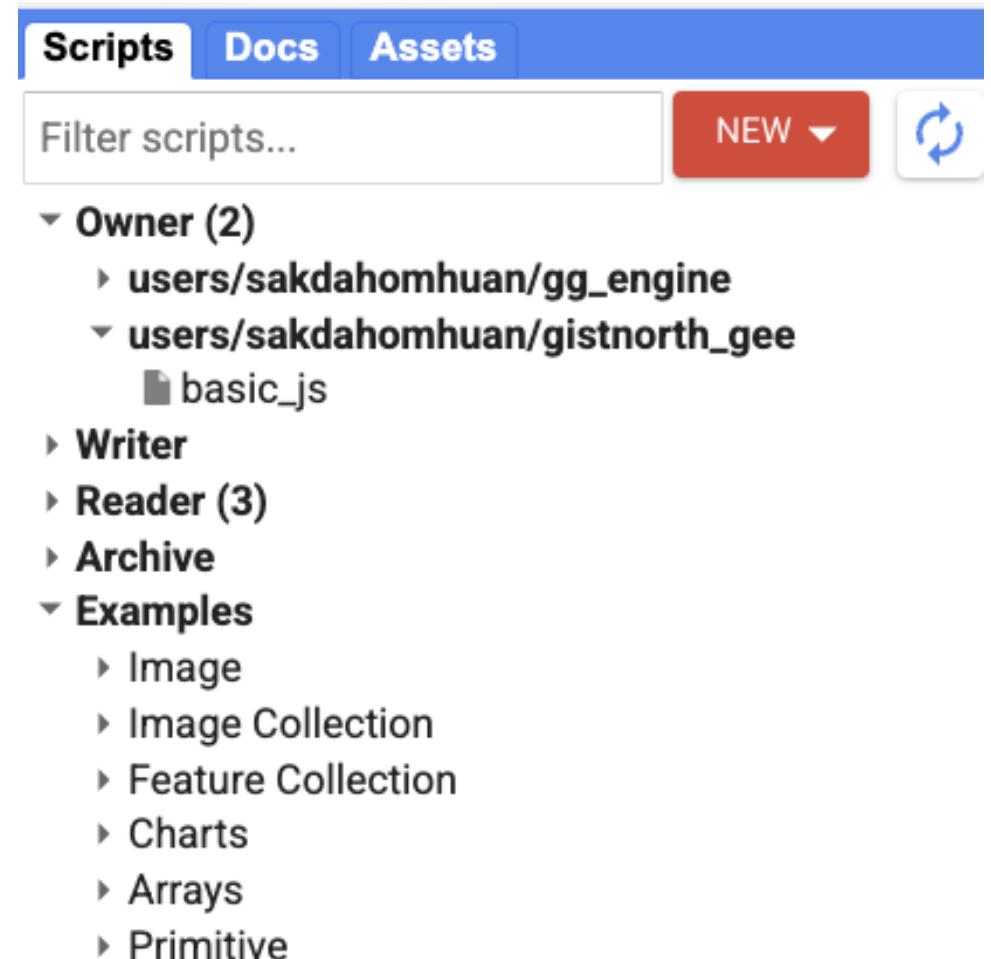
- Panel การใช้งาน

- **Scripts** (สคริปต์)
- **Docs** (เอกสารอ้างอิง)
- **Assets** (จัดการข้อมูลส่วนตัว)
- **Console** (แสดงผลลัพธ์และแจ้งเตือน)
- **Inspector** (ตรวจสอบข้อมูลบนแผนที่)
- **Tasks** (จัดการงานที่สั่งประมวลผล)



Scripts (สคริปต์)

- ใช้จัดการและเขียนไฟล์สคริปต์ JavaScript
- ประกอบด้วย:
 - Owner: สคริปต์ที่ผู้ใช้สร้างเอง
 - Writer/Reader: สคริปต์ที่คนอื่นแชร์ให้
 - Examples: ตัวอย่างสคริปต์จาก Google
- สร้างสคริปต์ใหม่ได้ที่ File → New → Script



Docs (เอกสารอ้างอิง)

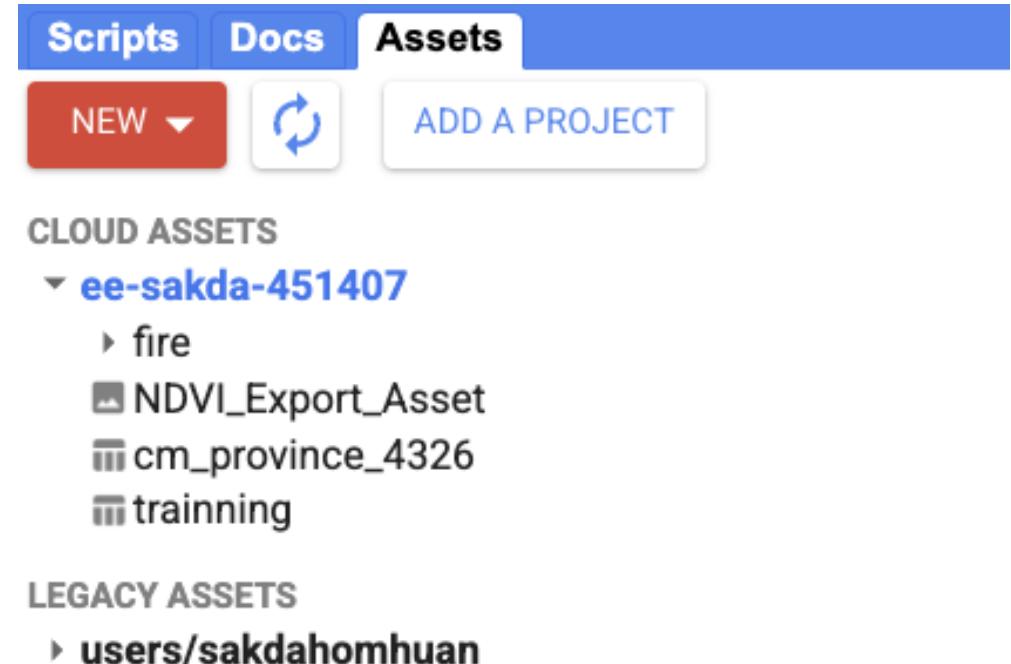
- แหล่งข้อมูลอ้างอิง (API documentation)
- ใช้ค้นหารายละเอียดการใช้งานฟังก์ชัน เช่น:
 - วิธีใช้พารามิเตอร์
 - ตัวอย่างโค้ดสั้น ๆ
- ค้นหาโดยการพิมพ์ชื่อฟังก์ชันหรือเลือกจากหมวดหมู่ที่มี

The screenshot shows a screenshot of a web-based API documentation interface. At the top, there is a blue header bar with three tabs: 'Scripts', 'Docs' (which is highlighted with a white background), and 'Assets'. Below the header is a search bar labeled 'Filter methods...'. A list of API classes is displayed below the search bar, each preceded by a right-pointing arrow:

- ee.Algorithms
- ee.Array
- ee.Blob
- ee.Classifier
- ee.Clusterer
- ee.ConfusionMatrix
- ee.Date
- ee.DateRange
- ee.Dictionary
- ee.ErrorMargin
- ee.Feature

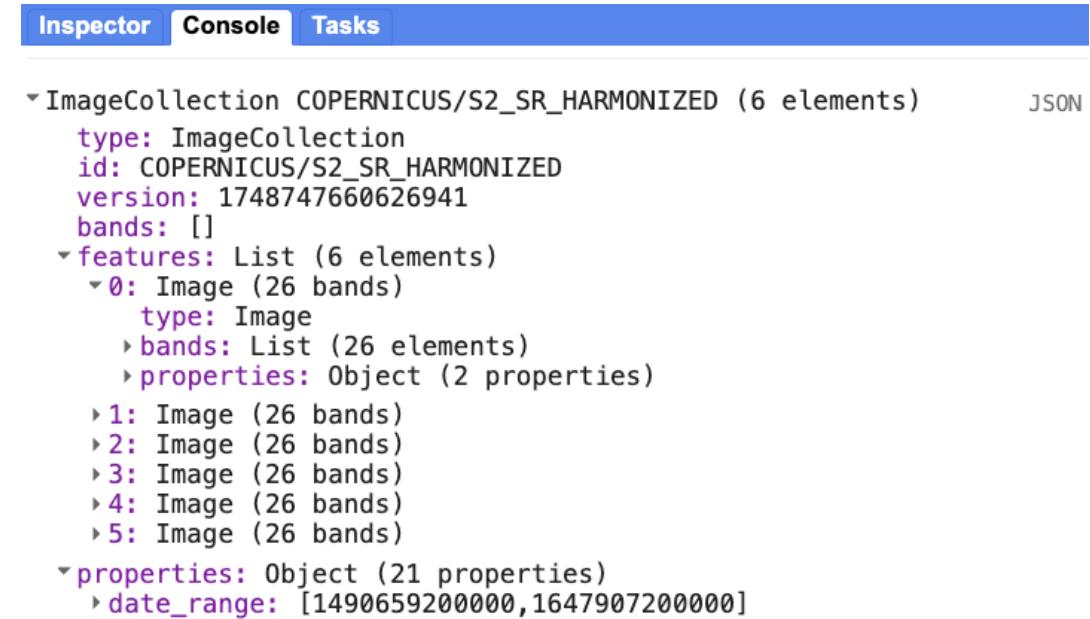
Assets (ข้อมูลส่วนตัว)

- ใช้อัปโหลดและจัดการข้อมูลเชิงพื้นที่ เช่น:
 - Shapefile
 - GeoTIFF
 - CSV
- ข้อมูลที่อัปโหลดสามารถเรียกใช้ได้โดยตรงในสคริปต์
- สามารถแชร์ Asset ให้ผู้อื่นเข้าถึงได้



Console (ผลลัพธ์และข้อผิดพลาด)

- แสดงผลลัพธ์จากสคริปต์ที่รัน เช่น ค่าเฉลี่ย NDVI
- แสดงข้อความแจ้งเตือนหรือข้อผิดพลาดที่เกิดขึ้น
- ตัวอย่างคำสั่ง `print('Hello GEE');`



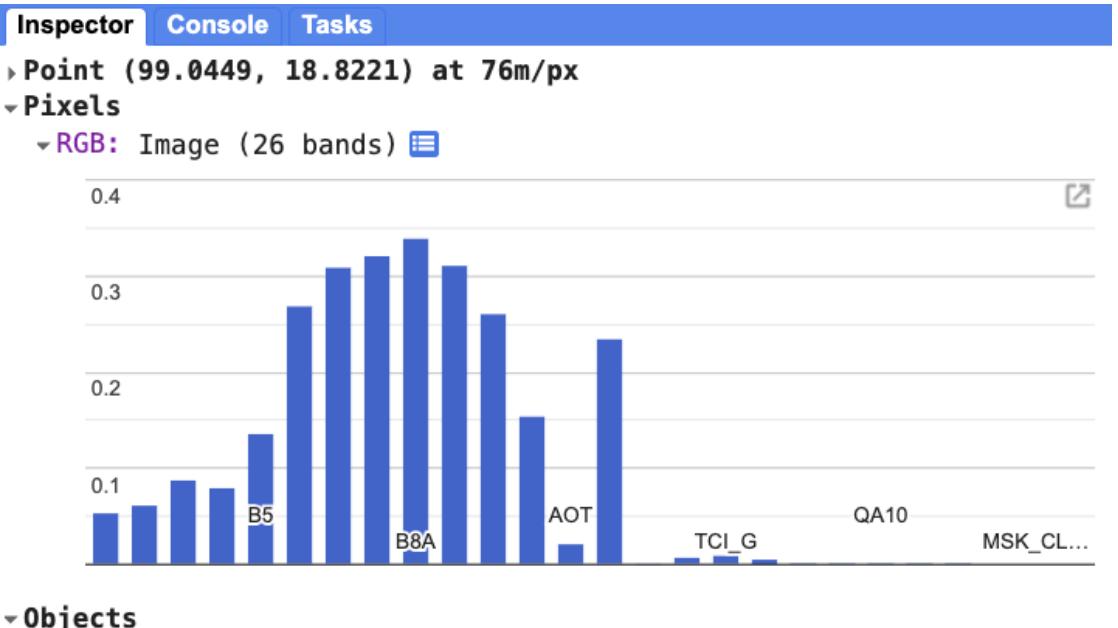
The screenshot shows the GEE Code Editor interface with the 'Console' tab selected. The output area displays the following JSON data:

```
Inspector Console Tasks

▼ ImageCollection COPERNICUS/S2_SR_HARMONIZED (6 elements) JSON
  type: ImageCollection
  id: COPERNICUS/S2_SR_HARMONIZED
  version: 1748747660626941
  bands: []
  ▼ features: List (6 elements)
    ▷ 0: Image (26 bands)
      type: Image
      ▷ bands: List (26 elements)
      ▷ properties: Object (2 properties)
        ▷ 1: Image (26 bands)
        ▷ 2: Image (26 bands)
        ▷ 3: Image (26 bands)
        ▷ 4: Image (26 bands)
        ▷ 5: Image (26 bands)
    ▷ properties: Object (21 properties)
    ▷ date_range: [1490659200000, 1647907200000]
```

Inspector (ตรวจสอบข้อมูลบนแผนที่)

- ใช้ตรวจสอบข้อมูลที่แสดงบนแผนที่
- คลิกบนแผนที่เพื่อแสดงข้อมูล:
 - ค่า Pixel ของภาพดาวเทียม
 - คุณสมบัติ (Properties) ของ Feature



Tasks (จัดการงานประมวลผล)

- ใช้สำหรับจัดการงานที่ส่งรันแบบเบื้องหลัง (เช่น Export ข้อมูล)
- สถานะของแต่ละ Task:
 - Submitted (ส่งงาน)
 - Running (กำลังทำงาน)
 - Completed (เสร็จสิ้น)
 - Failed (ล้มเหลว)
- ต้องกด Run เพื่อเริ่มงานที่ยังไม่ได้เริ่มทำ

The screenshot shows the 'Tasks' tab selected in the top navigation bar. Below it, a message encourages users to search or cancel multiple tasks in the Task Manager or try the Tasks Page in the Cloud Console. The main section, 'SUBMITTED TASKS', lists several tasks with their status and execution time:

Task Name	Status	Time
export_trainingFC	✓	<1m
export_trainingFC2	✓	<1m
export_landuse_zones	✓	<1m
NDVI_Export_Asset	✓	<1m
CM_Province_Export	✓	<1m
NDVI_Export	✓	<1m
Ingest table: "projects/ee-sakda-451407/assets/cm_province_4326"	↑	<1m

ข้อมูลใน GEE

Dataset ใน GEE คืออะไร?

- **Dataset (ชุดข้อมูล)** ใน Google Earth Engine (GEE) คือ กลุ่มข้อมูลเชิงพื้นที่ (**Geospatial Data**) ที่ถูกจัดเก็บและพร้อมใช้งานบนระบบคลาวด์ กลุ่มข้อมูลเชิงพื้นที่ (**Geospatial Data**) คือ:
 - ข้อมูลที่เกี่ยวข้องกับตำแหน่งหรือพื้นที่บนพื้นผิวโลก
 - โดยทั่วไปแสดงในรูปแบบแรสเตอร์ (Raster) เช่น ภาพถ่ายดาวเทียม, DEM หรือรูปแบบเวกเตอร์ (Vector) เช่น ขอบเขตพื้นที่, ถนน, แม่น้ำ
 - ข้อมูลเหล่านี้มีการอ้างอิงตำแหน่งที่ชัดเจนบนพื้นผิวโลกด้วยพิกัดทางภูมิศาสตร์ (Latitude และ Longitude)

ตัวอย่างชุดข้อมูลที่มีใน GEE:

- ภาพถ่ายดาวเทียม (เช่น Landsat, Sentinel, MODIS)
- ข้อมูลสภาพภูมิอากาศและอากาศ (Climate and Weather)
- ข้อมูลภูมิประเทศและระดับความสูง (Terrain & DEM)
- ข้อมูลการปักคุลพื้นที่และการใช้ที่ดิน (Land Cover & Land Use)
- ชุดข้อมูลเหล่านี้พร้อมใช้งานทันที โดยผู้ใช้ไม่จำเป็นต้องดาวน์โหลดหรือเก็บข้อมูลไว้ในเครื่อง
- สามารถนำไปใช้เคราะห์ผ่าน GEE Code Editor หรือ API ต่างๆ ได้ทันที เพื่อสนับสนุนงานวิจัย การศึกษา และการบริหารจัดการเชิงพื้นที่

Imagery Dataset (ข้อมูลภาพถ่ายดาวเทียม)

- คือชุดข้อมูลภาพถ่ายที่ได้จากการถ่ายดาวเทียมและเซนเซอร์ต่างๆ
- ตัวอย่างดาวเทียมสำคัญ:
 - Landsat:** ความละเอียด 30 เมตร ตั้งแต่ปี 1972 ถึงปัจจุบัน
 - Sentinel-2:** ความละเอียด 10-20 เมตร เหมาะสมกับการวิเคราะห์ NDVI
 - Sentinel-1:** ระบบเรดาร์ (SAR) ความละเอียด 10 เมตร ทะลุเมฆได้ดี
- ที่มา: developers.google.com



Landsat Collection

ชื่อดาวเทียม	เซ็นเซอร์หลัก	ข้อมูลที่ได้	ความละเอียดเชิงพื้นที่	ความถี่การผ่าน	ข้อมูลย้อนหลัง	ตัวอย่างการใช้งาน
Landsat 4	TM (Thematic Mapper)	Multispectral 7 แบนด์ (VIS, NIR, SWIR, Thermal)	30 ม. (TIR: 120 ม.)	16 วัน	ปี 1982–1993	การใช้ที่ดิน, NDVI, การเปลี่ยนแปลงพื้นที่ป่า
Landsat 5	TM	เหมือน Landsat 4	30 ม. (TIR: 120 ม.)	16 วัน	ปี 1984–2013	ป่าไม้, การเกษตร, วิเคราะห์แนวโน้มหลายศักราช
Landsat 7	ETM+ (Enhanced Thematic Mapper Plus)	Multispectral + Panchromatic + Thermal	30 ม. (Pan: 15 ม.)	16 วัน	ปี 1999–ปัจจุบัน	การจำแนกพื้นที่, ตรวจสอบเมือง, นำทั่วม
Landsat 8	OLI (Optical) + TIRS (Thermal Infrared Sensor)	11 แบนด์: VIS, NIR, SWIR, Thermal, Cirrus	30 ม. (Pan: 15 ม., TIR: 100 ม.)	16 วัน	ปี 2013–ปัจจุบัน	NDVI, LST, การวิเคราะห์พื้นที่เพาะปลูก, คุณภาพน้ำ
Landsat 9	OLI-2 + TIRS-2	เหมือน Landsat 8 แต่ปรับปรุงคุณภาพเชิงรังสี	30 ม. (Pan: 15 ม., TIR: 100 ม.)	16 วัน	ปี 2021–ปัจจุบัน	งานวิจัยสิ่งแวดล้อม, วางแผนพื้นที่เกษตร, เปรียบเทียบหล่ายช่วงเวลา

Collection 2

Landsat Collection 2, the second major reprocessing effort on the Landsat archive, resulted in several data product improvements that applied advancements in data processing and algorithm development.



Landsat 9 OLI-2/TIRS-2

2021–Present



Landsat 8 OLI/TIRS

2013–Present



Landsat 7 ETM+

1999–2021



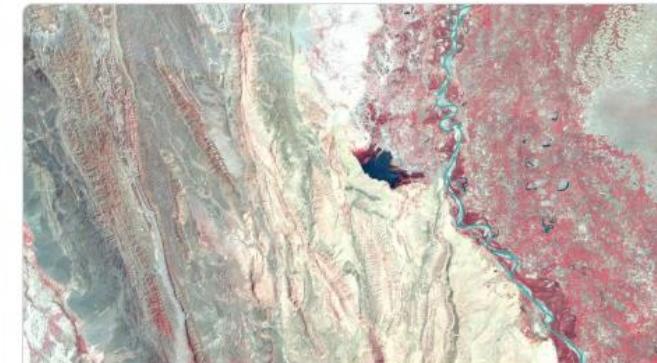
Landsat 5 TM

1984–2012



Landsat 4 TM

1982–1993



Landsat 1-5 MSS

1972–1999



Surface Reflectance

Landsat 9 OLI-2/TIRS-2 Collection 2 atmospherically corrected surface reflectance.

Dataset Availability: October 2021–Present

Tier 1

Tier 2



Top of Atmosphere

Landsat 9 OLI-2/TIRS-2 Collection 2 calibrated top-of-atmosphere (TOA) reflectance.

Dataset Availability: October 2021–Present

Tier 1

Tier 2



Raw Images

Landsat 9 OLI-2/TIRS-2 Collection 2 DN values, representing scaled, calibrated at-sensor radiance.

Dataset Availability: October 2021–Present

Tier 1

Tier 2

Data Tier

Tier	ความหมาย	ความแม่นยำเชิงตำแหน่ง	เหมาะสมสำหรับงาน
Tier 1	ผ่านการควบคุมคุณภาพตำแหน่งอย่างแม่นยำ	geolocation error < 12 เมตร	วิเคราะห์เชิงพื้นที่, time-series
Tier 2	ยังไม่ถึงมาตรฐาน Tier 1 หรือมีข้อจำกัดบางอย่าง	geolocation error > 12 เมตร	ดูภาพเบื้องต้น, งานวิเคราะห์ทั่วไป
RT	ข้อมูลล่าสุด เร่งด่วน ยังไม่ผ่านการจัดระดับ	ยังไม่ผ่านการตรวจสอบคุณภาพตำแหน่ง	งานภัยพิบัติ, สำรวจเบื้องต้น

Data Levels

ระดับ	ชื่อเต็ม	คำอธิบาย	การใช้งาน	หน่วย	ความแม่นยำ
Level-0 (L0)	Raw Data (ไม่เผยแพร่สาธารณะ)	ข้อมูลดิบจากเซ็นเซอร์ดาวเทียม โดยตรงแบบ bitstream ที่ยังไม่ผ่านการแปลงใด ๆ	ใช้ภายใน NASA/USGS เพื่อถอดรหัส	Bitstream	ต่ำมาก
Level-1 (L1)	Radiometrically and Geometrically Corrected	ข้อมูลที่แปลงจาก DN เป็น Radiance และผ่านการแก้ไขตำแหน่ง (orthorectification) แล้ว	ใช้สร้างภาพ RGB, ใช้เป็น input ต่อใน SR	Radiance หรือ DN	ปานกลางถึงสูง
Level-2 (L2)	Surface Reflectance / Surface Temperature	ข้อมูลที่ผ่านการแก้ชั้นบรรยากาศแล้ว (atmospheric correction) แสดงค่า reflectance และอุณหภูมิผิวโลกที่ถูกต้อง	วิเคราะห์ NDVI, LST, เปรียบเทียบข้ามเวลา	Reflectance (0–1 หรือ 0–10000), °C	สูงมาก

Surface Reflectance (SR), Top of Atmosphere (TOA) และ Raw Images

ประเภทข้อมูล	คำอธิบาย	การใช้งาน	หน่วย	Atmospheric Correction
Raw Image (DN)	ข้อมูลดิบที่ได้จากเซ็นเซอร์ดาวเทียม เป็นตัวเลขดิจิทัล (Digital Number) ที่ยังไม่ผ่านการประมวลผลใด ๆ	ใช้ในการสอบเทียบรังสีเบื้องต้น หรือประมวลผลเอง	ค่า DN (เช่น 0–255, 0–4096)	ไม่
TOA Reflectance	ค่าการสะท้อนแสงจากยอดชั้นบรรยากาศ ได้จากการแปลงค่า DN โดยคำนึงถึงระยะโลก-ดวงอาทิตย์ และมุมตากกระทบท่องแสงอาทิตย์	เหมาะสมกับงานวิเคราะห์ที่ไม่ต้องการความแม่นยำสูงมาก เช่น การดูสีพื้นผิว	ค่าสะท้อนกลับ (0–1 หรือ 0–100%)	ไม่
Surface Reflectance (SR)	ค่าการสะท้อนแสงที่ระดับผิวโลกจริง ผ่านการปรับแก้ผลกระทบจากชั้นบรรยากาศ เช่น ผุน ควัน ไอน้ำ และการกระเจิงของแสง	ใช้ในการวิเคราะห์เชิงวิทยาศาสตร์ เช่น NDVI, การจำแนกป่าไม้, ติดตามการเปลี่ยนแปลงระยะยาว	ค่าสะท้อนกลับ (0–1 หรือ 0–10000)	ใช่

USGS Landsat 9 Level 2, Collection 2, Tier 1



Dataset Availability

2021-10-31T00:00:00Z–2025-05-29T23:52:08.473000Z

Dataset Provider

USGS

Earth Engine Snippet

```
ee.ImageCollection("LANDSAT/LC09/C02/T1_L2")
```

Revisit Interval

16 Days

Tags

cfmask, cloud, fmask, global, l9sr, landsat, lasrc, lc09, lst, reflectance
satellite-imagery, sr, usgs

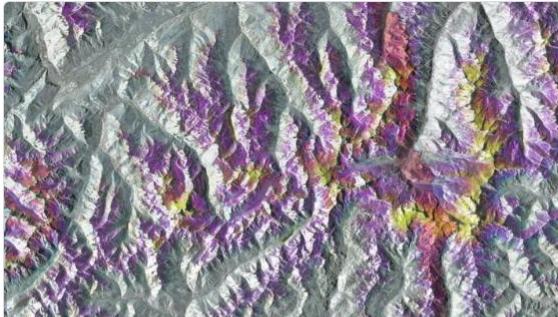
Description	Bands	Image Properties	Terms of Use
Pixel Size			
30 meters			
Bands			
Name	Units	Min	Max
SR_B1		1	65455
SR_B2		1	65455
SR_B3		1	65455
SR_B4		1	65455
SR_B5		1	65455
SR_B6		1	65455
SR_B7		1	65455
SR_QA_AEROSOL	Aerosol attributes		

Landsat collection structure

ID	Description
LANDSAT/LC08/C02/T1_RT_TOA	Landsat 8, Collection 2, Tier 1 + Real Time, TOA
LANDSAT/LC08/C02/T1_TOA	Landsat 8, Collection 2, Tier 1 only, TOA
LANDSAT/LC08/C02/T1_L2	Landsat 8, Collection 2, Tier 1 only, SR and LST
LANDSAT/LC08/C02/T2_TOA	Landsat 8, Collection 2, Tier 2 only, TOA

Sentinel Collections

ข้อมูลจาก ดาวเทียม Sentinel คือชุดข้อมูลจาก โครงการ Copernicus ของสหภาพยุโรป (EU) และ องค์การอวกาศยุโรป (ESA) ซึ่งออกแบบมาเพื่อการตรวจสอบโลก (Earth Observation) เพื่อสนับสนุนการจัดการทรัพยากรธรรมชาติ, ภัยพิบัติ, การเกษตร, ป่าไม้, การใช้ที่ดิน, คุณภาพอากาศ ฯลฯ



Sentinel-1 SAR GRD: C-band Synthetic Aperture Radar

Data availability: 2014 – Present

The Sentinel-1 mission provides data from a dual-polarization C-band Synthetic Aperture Radar (SAR) instrument. SAR instruments are capable of acquiring meaningful data in all weather conditions (even clouds) during daytime and nighttime. Sentinel-1 data is used across many domains, including maritime activity, sea-ice mapping, humanitarian aid, crisis response, and forest management.



Sentinel-2 MSI: Multispectral Instrument

Data availability: 2015 – Present

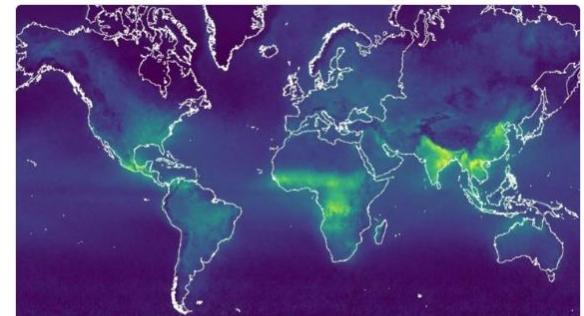
The Sentinel-2 mission collects high-resolution multispectral imagery useful for a broad range of applications, including monitoring of vegetation, soil and water cover, land cover change, as well as humanitarian and disaster risk.



Sentinel-3 OLCI EFR: Ocean and Land Color Instrument

Data availability: 2016 – Present

The Sentinel-3 instrument provides systematic measurements of the planet's oceans, land, ice, and atmosphere, including the temperature, color and height of the sea surface as well as the thickness of sea ice.



Sentinel-5 TROPOMI: TROPOspheric Monitoring Instrument

Data availability: 2018 – Present

The Sentinel-5 Precursor mission collects data useful for assessing air quality, including concentrations of: ozone, methane, formaldehyde, aerosol, carbon monoxide, nitrogen oxide, and sulphur dioxide.

ประเภทของดาวเทียม Sentinel

Sentinel satellites มีหลายดวงและแต่ละดวงมีภารกิจเฉพาะที่แตกต่างกัน

ชื่อดาวเทียม	เซ็นเซอร์หลัก	ข้อมูลที่ได้	ความละเอียด	ความถี่การผ่าน	การใช้งาน
Sentinel-1	Synthetic Aperture Radar (SAR-C)	เรดาร์ (VV, VH)	~10 เมตร	6–12 วัน	ตรวจน้ำท่วม, การเคลื่อนตัวของดิน
Sentinel-2	Multispectral Instrument (MSI)	ภาพถ่ายทาง ช่วงคลื่น (13 bands)	10–60 เมตร	5 วัน (S2A+S2B)	การเกษตร, ป่าไม้, NDVI
Sentinel-3	OLCI, SLSTR, SRAL	อุณหภูมิผิวน้ำ, ความสูงน้ำทะเล	300–1,200 เมตร	~1–2 วัน	มหาสมุทร, ภูมิอากาศ
Sentinel-5P	TROPOMI	มลพิษอากาศ (NO ₂ , O ₃ , SO ₂ , CO)	5.5 x 3.5 กม. Q3TNORTH	1 วัน	ติดตามคุณภาพ อากาศ

Sentinel-2

- **Sentinel-2** ชิ้งเป็นดาวเทียมบันทึกภาพหลายช่วงคลื่น (Multispectral)



Surface Reflectance

Level-2A orthorectified atmospherically corrected surface reflectance.

Dataset availability: 2017-03-28 – Present



Top-of-Atmosphere Reflectance

Level-1C orthorectified top-of-atmosphere reflectance.

Dataset availability: 2015-06-27 – Present

ข้อมูล Sentinel-2 (Processing Levels)

ระดับ	ชื่อเต็ม	คำอธิบาย	หน่วย	การใช้งาน
Level-0 (L0)	Raw Data	ข้อมูลดิบที่ยังไม่ผ่านการประมวลผลใด ๆ (ไม่เผยแพร่ต่อสาธารณะ)	Bitstream	ใช้ภายใน ESA เท่านั้น
Level-1C (L1C)	Top of Atmosphere Reflectance	ข้อมูลที่ผ่านการปรับเรขาคณิต (orthorectified) และปรับ radiometric แล้ว แต่ ยังไม่ปรับแก้ชั้นบรรยากาศ	Reflectance (0–1)	ใช้ดูภาพ, คำนวณเบื้องต้น, NDVI (แบบคร่าว ๆ)
Level-2A (L2A)	Surface Reflectance	ข้อมูลที่ผ่านการปรับชั้นบรรยากาศ แล้ว (atmospheric correction) แสดงค่าการสะท้อนกลับผิวโลก (surface)	Reflectance (0–1)	ใช้คำนวณ NDVI, NDWI, การจำแนกพื้นที่, time-series

Harmonized Sentinel-2 MSI: MultiSpectral Instrument, Level-2A (SR)



Dataset Availability

2017-03-28T00:00:00Z–2025-05-31T19:55:03.306000Z

Revisit Interval

5 Days

Dataset Provider

European Union/ESA/Copernicus

Tags

copernicus esa eu msi reflectance satellite-imagery sentinel sr

Earth Engine Snippet

```
ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
```

Description	Bands	Image Properties	Terms of Use
Bands			
Name			
B1	Units	Min	Max
		0.0001	60 meters
			443.9nm (S2A) / 442.3nm (S2B)
			Aerosols
B2			
		0.0001	10 meters
			496.6nm (S2A) / 492.1nm (S2B)
			Blue
B3			
		0.0001	10 meters
			560nm (S2A) / 559nm (S2B)
			Green
B4			
		0.0001	10 meters
			664.5nm (S2A) / 665nm (S2B)
			Red
B5			
		0.0001	20 meters
			703.9nm (S2A) / 703.8nm (S2B)
			Red Edge 1
B6			
		0.0001	20 meters
			740.2nm (S2A) / 739.1nm (S2B)
			Red Edge 2
B7			
		0.0001	20 meters
			782.5nm (S2A) / 779.7nm (S2B)
			Red Edge 3
B8			
		0.0001	10 meters
			835.1nm (S2A) / 833nm (S2B)
			NIR

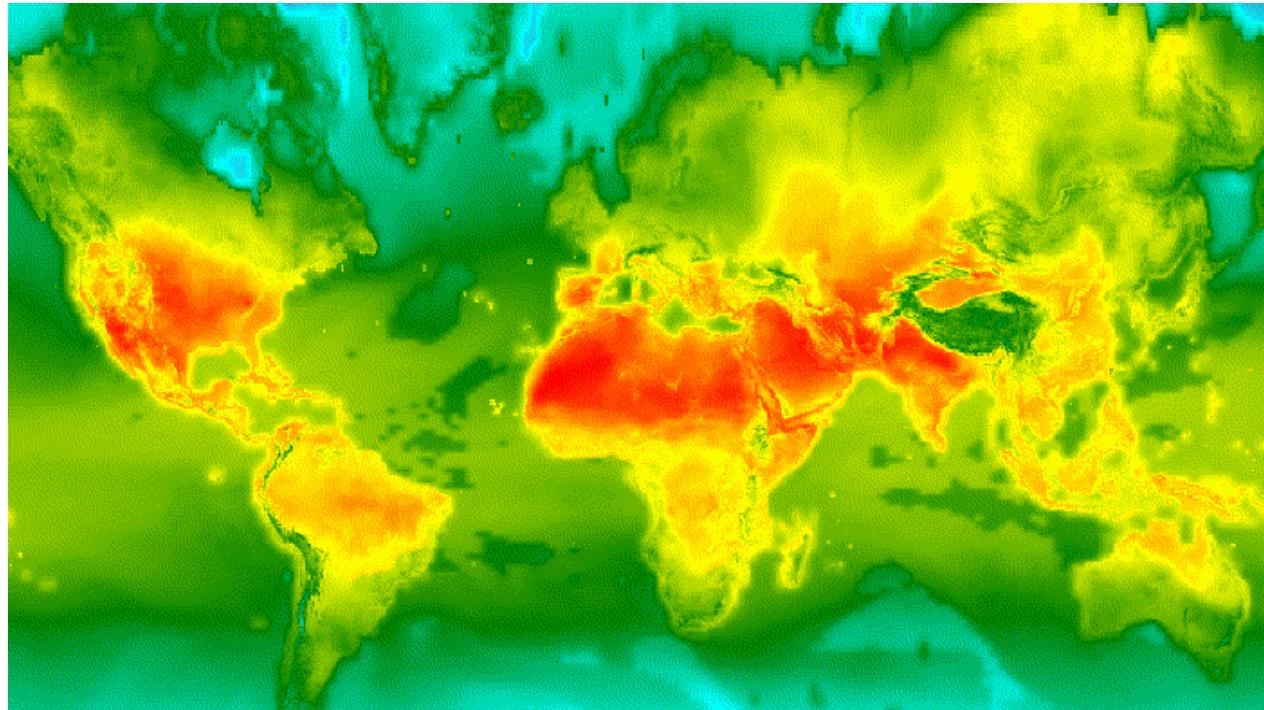
Terrain / Elevation Dataset (ข้อมูลระดับความสูง)

- ข้อมูลระดับความสูงและภูมิประเทศ (Digital Elevation Model: DEM)
- ตัวอย่างข้อมูล DEM ยอดนิยม:
 - **SRTM**: ความละเอียด 30 เมตร ระหว่าง -60° ถึง $+60^{\circ}$ ละติจูด
 - **ASTER GDEM**: ความละเอียด 30 เมตร ครอบคลุมทั่วโลก
- ประโยชน์ใช้: วิเคราะห์การไฟลของน้ำ, การประเมินความเสี่ยงดินถล่ม
- ที่มา: developers.google.com



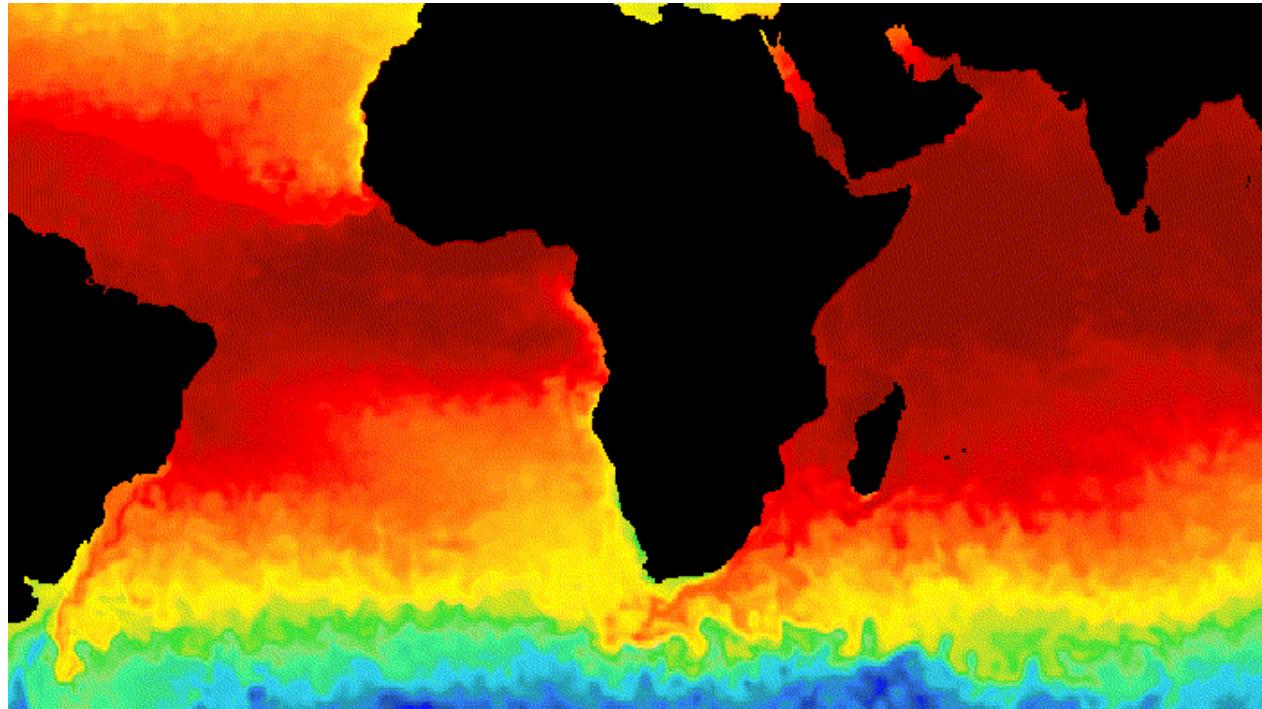
Climate Dataset (ข้อมูลภูมิอากาศระยะยาว)

- ข้อมูลสภาพภูมิอากาศย้อนหลังระยะยาว เช่น อุณหภูมิ ฝนตก
- ตัวอย่างชุดข้อมูล:
 - **ERA5-Land:** ความละเอียด 9 กม. ข้อมูลย้อนหลัง ตั้งแต่ปี 1981
 - **NCEP/NCAR:** ความละเอียด 2.5 องศา ตั้งแต่ปี 1948
- ประโยชน์ใช้: การวางแผนเกษตรกรรม, ประเมินผลผลกระทบการเปลี่ยนแปลงสภาพภูมิอากาศ
- ที่มา: developers.google.com



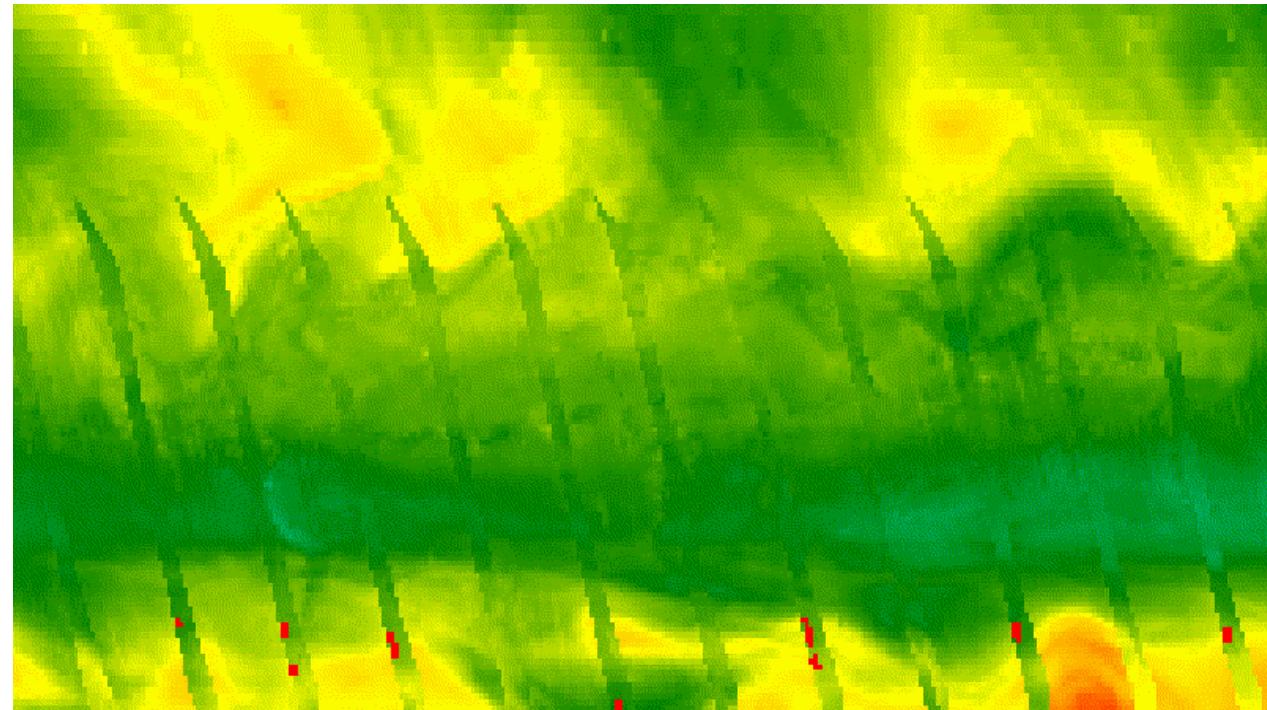
Weather Dataset (ข้อมูลสภาพอากาศรายวัน)

- ข้อมูลรายวันหรือรายชั่วโมงสำหรับพยากรณ์อากาศ
- ตัวอย่างชุดข้อมูล:
 - **CHIRPS:** ความละเอียด 0.05° ข้อมูลรายวันตั้งแต่ปี 1981
 - **GPM/TRMM:** ความละเอียด 0.1° ข้อมูลรายชั่วโมง
- ประโยชน์ใช้: เตือนภัยน้ำท่วม, สนับสนุนการตัดสินใจด้านการเกษตร
- ที่มา: developers.google.com



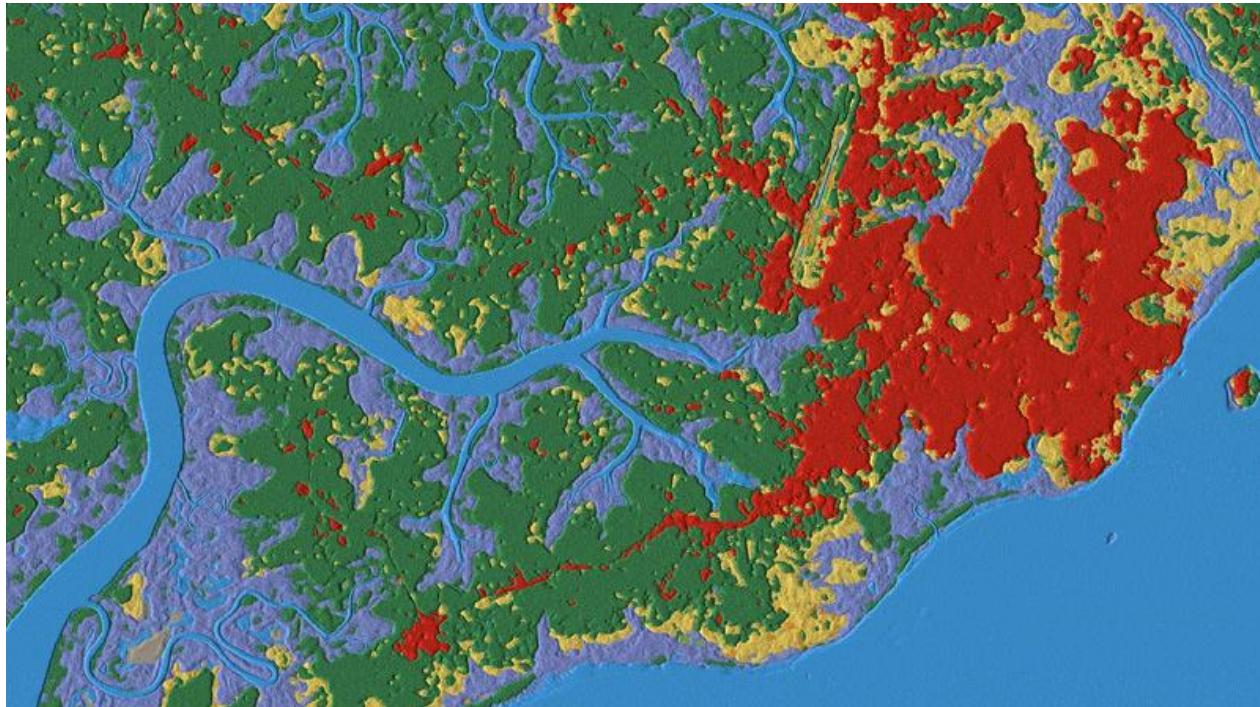
Atmospheric Dataset (ข้อมูลบรรยากาศ)

- ข้อมูลเกี่ยวกับองค์ประกอบของบรรยากาศ เช่น ฝุ่นละออง โอโซน
- ตัวอย่างชุดข้อมูล:
 - **MODIS Aerosol:** ค่าความหนาแน่นละอองลอย (AOD)
 - **OMI/TROPOMI:** ความเข้มข้นของโอโซนและ NO₂
- ประยุกต์ใช้: ประเมินคุณภาพอากาศ, วิจัยด้านมลพิษ
- ที่มา: developers.google.com



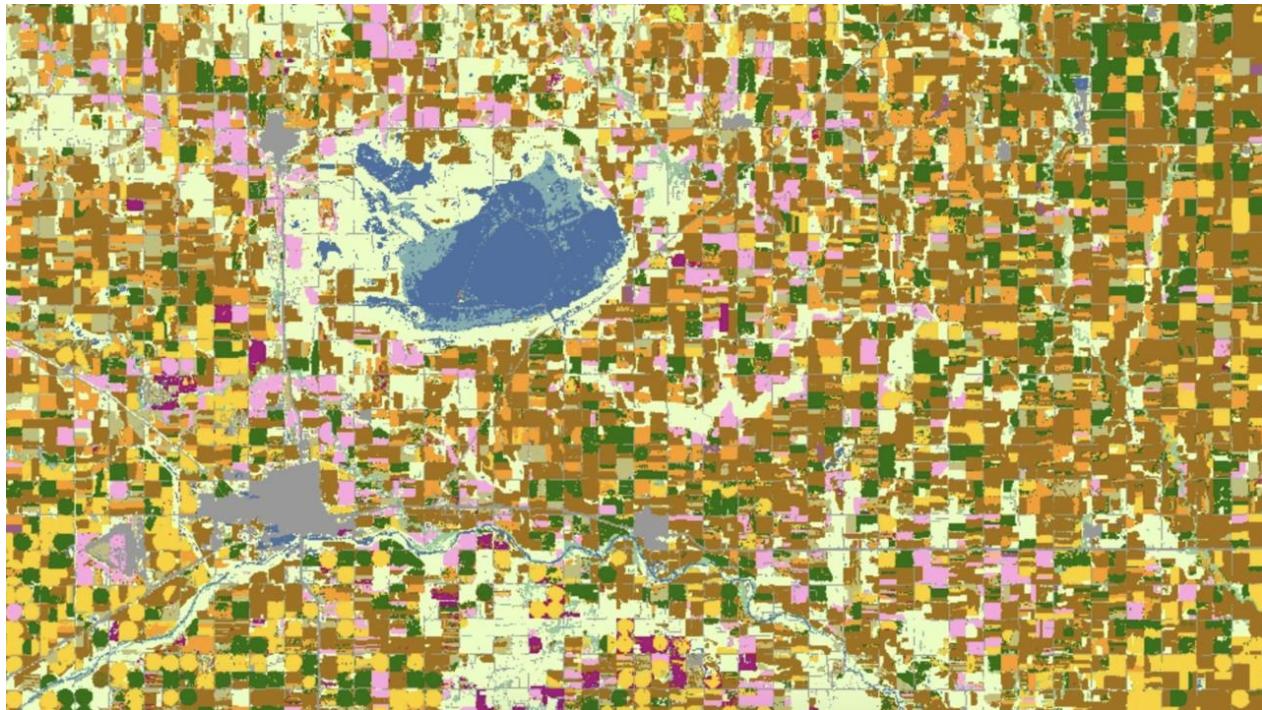
Land Cover/Land Use Dataset

- ข้อมูลการใช้ประโยชน์ที่ดินและการป่าคลุมพื้นที่
- ตัวอย่างชุดข้อมูล:
 - **MODIS Land Cover:** ความละเอียด 500 ม. รายปี
 - **Hansen Global Forest Change:** ความละเอียด 30 ม. รายปี
- ประยุกต์ใช้: ติดตามการเปลี่ยนแปลงพื้นที่ป่าไม้, วางแผนเมือง
- ที่มา: developers.google.com



Vegetation Indices Dataset

- ดัชนีสำหรับวิเคราะห์สุขภาพพืช เช่น NDVI, EVI
- ตัวอย่างชุดข้อมูล:
 - **MODIS NDVI/EVI:** ความละเอียด 250 ม. ทุก 16 วัน
- ประยุกต์ใช้: การเกษตรแม่นยำ, วิเคราะห์ภัยแล้ง
- ที่มา: developers.google.com



Hydrology Dataset (ข้อมูลอุทกวิทยา)

- ข้อมูลน้ำผิวดิน แม่น้ำ ทะเลสาบ
- ตัวอย่างชุดข้อมูล:
 - **JRC Global Surface Water:** ความละเอียด 30 ม.
- ประยุกต์ใช้: วิเคราะห์แหล่งน้ำ, การจัดการน้ำ, ประเมินพื้นที่เสี่ยงน้ำท่วม
- ที่มา: global-surface-water.appspot.com

Soils Dataset (ข้อมูลดิน)

- ข้อมูลดินสมบัติของดิน เช่น ค่าความเป็นกรดด่าง, ปริมาณคาร์บอน
- ตัวอย่างชุดข้อมูล:
 - **SoilGrids:** ความละเอียด 250 ม. หลายระดับความลึก
- ประโยชน์ใช้: ประเมินการเก็บกักคาร์บอน, วางแผนเกษตรกรรม
- ที่มา: soilgrids.org

Socioeconomic Dataset (ข้อมูลทางเศรษฐกิจและสังคม)

- ข้อมูลประชากร, GDP, การใช้พลังงาน
- ตัวอย่างชุดข้อมูล:
 - **WorldPop:** การกระจายประชากร (100 ม.)
- ประโยชน์ใช้: วางแผนนโยบาย, การจัดการภัยพิบัติ, วิเคราะห์เศรษฐกิจ
- ที่มา: worldpop.org

พื้นฐานภาษา JavaScript สำหรับ GEE

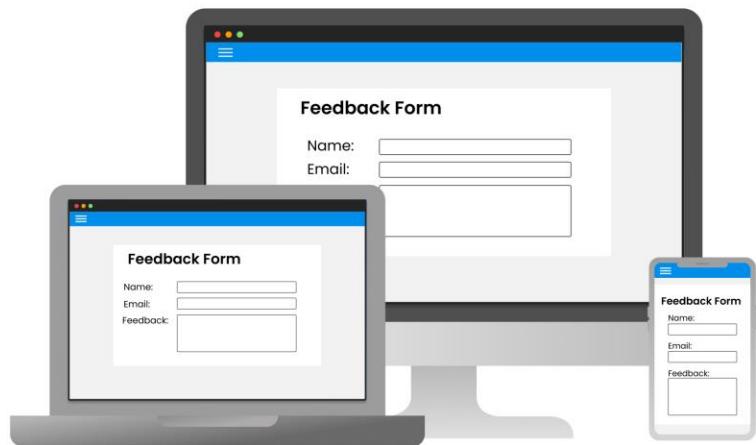
- GEE Code Editor รองรับภาษา JavaScript เป็นหลัก
- ใช้เขียนสคริปต์วิเคราะห์ข้อมูลภูมิสารสนเทศผ่าน Cloud
- เขียนคำสั่งเชิงวิเคราะห์ (NDVI, การกรองภาพ, การทำ mask) ได้ง่าย
- มีฟังก์ชันเฉพาะของ Earth Engine (เช่น ee.Image, ee.Geometry)

แนวคิด Client-side vs Server-side

- **Client-side:** ทำงานในเบราว์เซอร์ เช่น print(), Map.addLayer()
- **Server-side:** ประมวลผลข้อมูลขนาดใหญ่บนเซิร์ฟเวอร์ Google
- ตัวแบบ ee.Object จะทำงานผ่าน Server-side
- ต้องเข้าใจความแตกต่างเพื่อไม่ให้เกิด error จากการจัดการข้อมูลผิดฝั่ง

Client

Client-side code runs on the user's browser and is untrusted.



Responsible for:

- ✓ Displaying the UI
- ✓ Interactivity

Server

Server-side code runs on a server and is unseen by users.



Responsible for:

- ✓ Controlling central resources
- ✓ Permissions checks
- ✓ Handling secret information

ตัวอย่าง Server-side

```
var image = ee.Image('COPERNICUS/S2/20240101T000000');
```

ตัวอย่าง Client-side

```
Map.addLayer(image);  
print(image);
```

- การใช้ `print()` กับ `ee.Object` จะต้องให้เซิร์ฟเวอร์ดึงค่ากลับมาที่เบราว์เซอร์
- การพยายามใช้ค่าจาก `Server-side` มาประมวลผลใน `Client-side` โดยตรงจะทำให้เกิดข้อผิดพลาด เช่น `image.getInfo()` ต้องใช้กับข้อมูลขนาดเล็กเท่านั้น
- หลักการ: ควรเขียนสคริปต์ให้ทำงานอยู่ `Server-side` ให้มากที่สุดเพื่อความรวดเร็ว

การประการตัวแปร

- ใช้ var ในการประกาศตัวแปรใน JavaScript
- var city = 'Chiang Mai'; var value = 100;
- ตั้งชื่อตัวแปรตามหลัก Camel Case เช่น
ndviValue
- หลีกเลี่ยงการใช้คำส่วน (reserved words)

Google JavaScript Style Guide

Revision 2.93

Aaron Whyte
Bob Jervis
Dan Pupius
Erik Arvidsson
Fritz Schneider
Robby Walker

Each style point has a summary for which additional information is available by toggling the accompanying arrow button that looks this way: ▶. You may toggle all summaries with the big arrow button:

 Toggle all summaries

Table of Contents

JavaScript Language Rules	var Constants Semicolons Nested functions Function Declarations Within Blocks Exceptions Custom exceptions Standards features Wrapper objects for primitive types Multi-level prototype hierarchies Method and property definitions delete Closures eval() with() this for-in loop Associative Arrays Multiline string literals Array and Object literals Modifying prototypes of builtin objects Internet Explorer's Conditional Comments
JavaScript Style Rules	Naming Custom toString() methods Deferred initialization Explicit scope Code formatting Parentheses Strings Visibility (private and protected fields) JavaScript Types Comments Providing Dependencies With goog.provide Compiling Tips and Tricks

https://google.github.io/styleguide/javascriptguide.xml#JavaScript_Language_Rules

ประเภทข้อมูลพื้นฐาน

- Number: ตัวเลข เช่น 10, 3.14
- String: ข้อความ เช่น 'GEE', "เชียงใหม่"
- Boolean: true, false
- Array: กลุ่มข้อมูล เช่น [1, 2, 3]
- Object: เช่น {name: 'Thailand', area: 513000}

การใช้ Array และ Object

- **Object** ใช้เก็บข้อมูลในรูปแบบ key-value เหมาะสำหรับเก็บข้อมูลที่เกี่ยวข้องกัน

```
var province = {  
    name: "Chiang Mai",  
    population: 1700000,  
    area: 20107  
};  
print(province.name); // Chiang Mai
```

- **Array** ใช้เก็บชุดของข้อมูลที่มีลำดับ เช่น รายชื่อจังหวัด หรือค่าฝนในแต่ละวัน

```
var provinces = ["Chiang Mai", "Lamphun", "Chiang Rai"];  
print(provinces[0]); // Chiang Mai
```

ฟังก์ชัน (Function)

- ฟังก์ชันคือกลุ่มคำสั่งที่นำกลับมาใช้ซ้ำได้
- สามารถรับพารามิเตอร์และคืนค่าผลลัพธ์

```
function add(x, y) {  
    return x + y;  
}  
var sum = add(10, 5);  
print("Sum is", sum);
```

- ใน GEE นิยมใช้ฟังก์ชันกับการ map ข้อมูล หรือคำนวณค่าต่าง ๆ

```
var collection = ee.ImageCollection('COPERNICUS/S2')
  .filterDate('2024-01-01', '2024-01-31')
  .filterBounds(ee.Geometry.Point(100.5, 13.7));

var ndviCollection = collection.map(function(image) {
  var ndvi = image.normalizedDifference(['B8', 'B4'])
    .rename('NDVI');
  return image.addBands(ndvi);
});
```

การใช้ If-Else

- ใช้เพื่อตรวจสอบเงื่อนไขและตัดสินใจว่าจะทำคำสั่งใดต่อไป

```
var score = 85;
if (score >= 80) {
    print('A');
} else {
    print('B หรือ ต่ำกว่า');
}
```

- ใน GEE server-side ใช้เมธอดเปรียบเทียบเช่น .gt(), .lt() และ if-else ธรรมดा

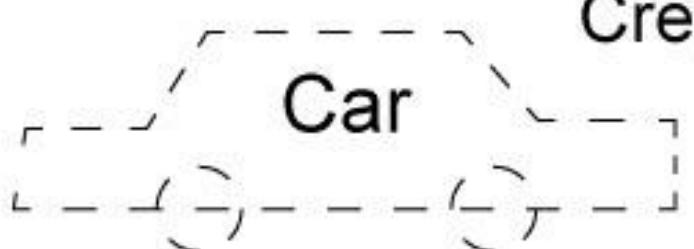
```
var mask = image.normalizedDifference(["B8", "B4"]).gt(0.5)
```

Object-Oriented Programming (OOP)

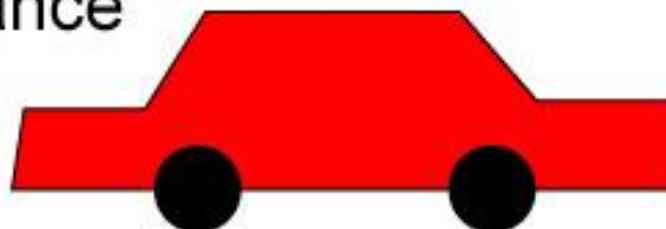
แนวคิดของ OOP

- เป็นการเขียนโปรแกรมโดยใช้แนวคิด “วัตถุ” (Object) ซึ่งรวมข้อมูล (properties) และฟังก์ชัน (methods) ไว้ภายในวัตถุเดียวกัน
- ส่งเสริมแนวทางการเขียนโค้ดที่สามารถนำกลับมาใช้ซ้ำ (reuse) ได้ง่าย และโครงสร้างโค้ดมีความชัดเจน องค์ประกอบของ Object
 - Properties (คุณสมบัติ)** เป็นข้อมูลที่เก็บอยู่ใน object
 - Methods (เมธอด)** เป็นฟังก์ชันที่ทำงานกับข้อมูลภายใน object นั้น

Class



Create an instance



Properties	Methods - behaviors
color	start()
price	backward()
km	forward()
model	stop()

Property values	Methods
color: red	start()
price: 23,000	backward()
km: 1,200	forward()
model: Audi	stop()

- ตัวอย่างการสร้าง Object และ Method

```
var landsat = {  
    satellite: "Landsat 8",  
    bands: ["B2", "B3", "B4", "B5"],  
    getNIR: function() {  
        return this.bands[3]; // คืนค่า NIR band  
    },  
    info: function() {  
        return this.satellite + " with bands: " + this.bands.join(", ");  
    }  
};  
  
print("NIR band:", landsat.getNIR());  
print("Satellite info:", landsat.info());
```

Method Chaining:

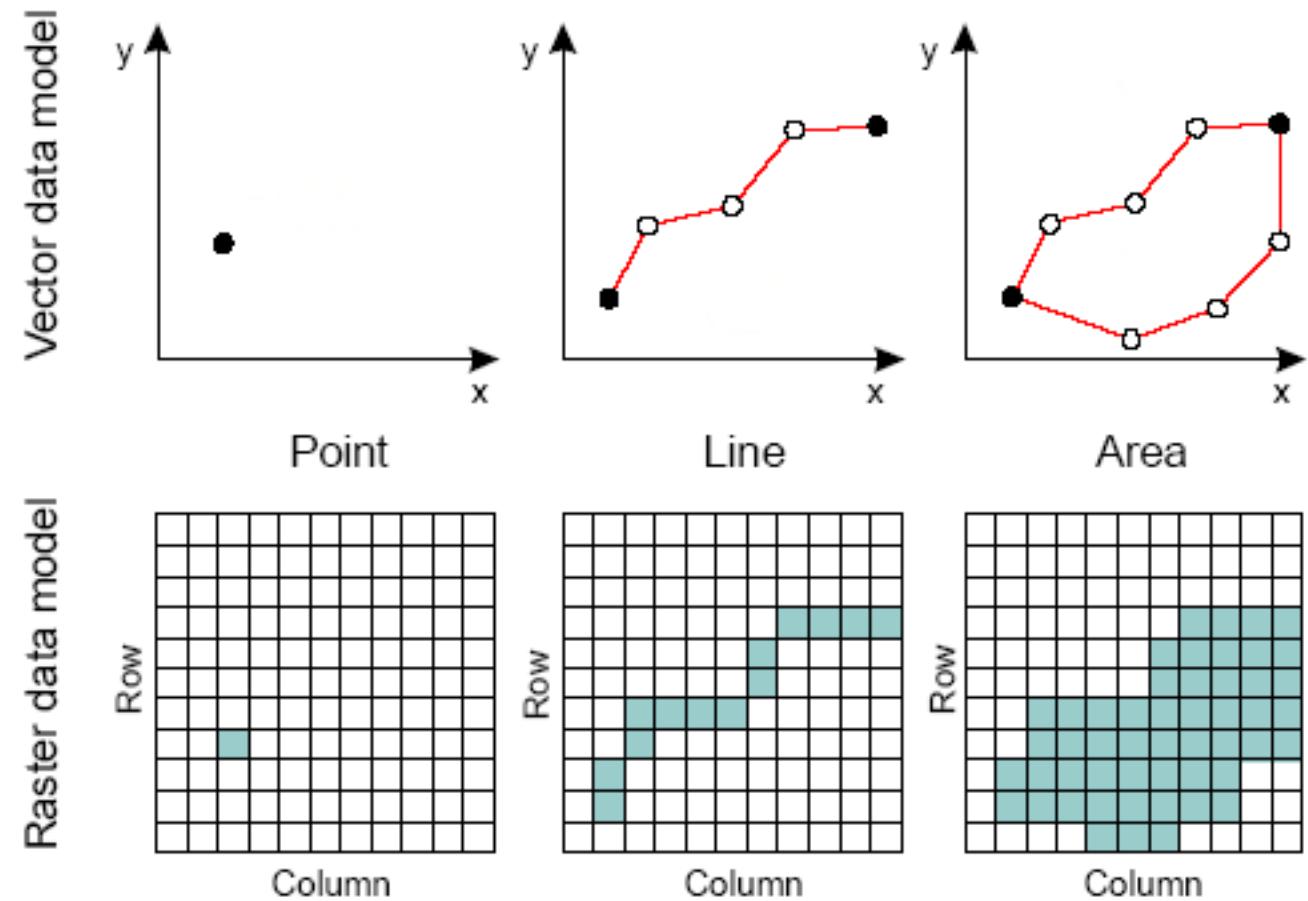
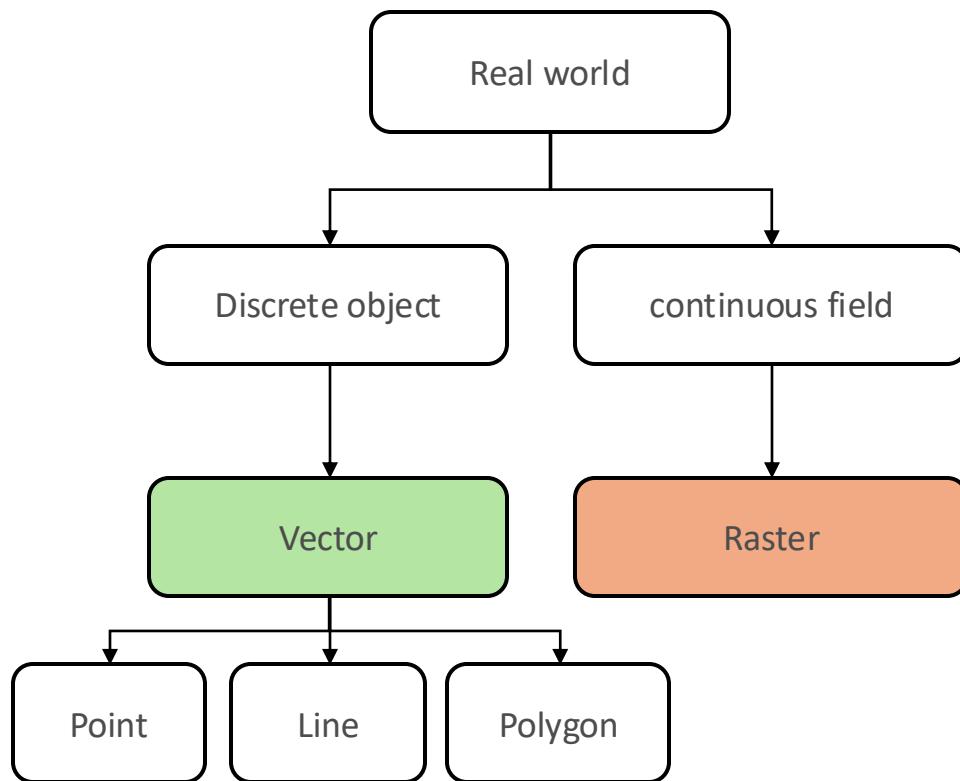
- เป็นการเรียกใช้ method หลายตัวต่อเนื่องจาก object เดียว
- ลดการสร้างตัวแปรชั่วคราวที่ไม่จำเป็น
- ช่วยแสดงลำดับของกระบวนการที่ได้ชัดเจน เช่น preprocessing -> calculation -> renaming

```
var ndvi = ee.Image('COPERNICUS/S2/20240101T000000')
  .clip(ee.Geometry.Point(100.5, 13.7).buffer(5000))
  .normalizedDifference(['B8', 'B4'])
  .rename('NDVI');
```

ประเภทข้อมูลหลักใน GEE

- ee.Image ข้อมูลภาพถ่ายจากดาวเทียม เช่น Landsat, Sentinel
- ee.ImageCollection กลุ่มของข้อมูลภาพถ่าย เช่น รายเดือน รายปี
- ee.Geometry รูปทรงเชิงพื้นที่ เช่น จุด เส้น พื้นที่
- ee.Feature ข้อมูลเชิงพื้นที่พร้อมคุณสมบัติ (Geometry + Attribute)
- ee.FeatureCollection กลุ่มของ feature

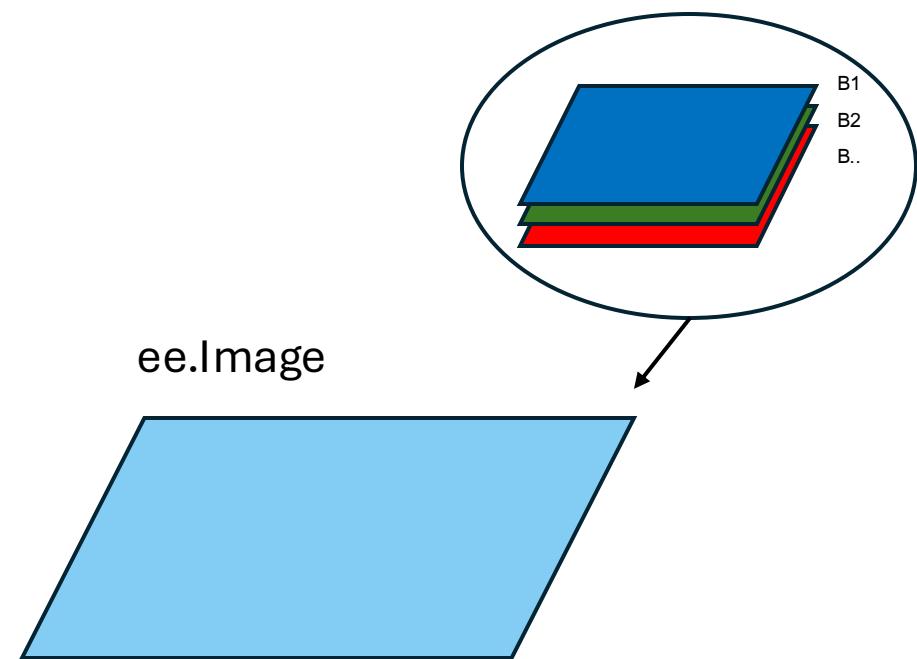
Spatial data transformation



ee.Image

ee.Image: ภาพเดี่ยว (Single Image)

- คือภาพ (Raster Image) ที่ประกอบด้วยข้อมูลแบบหลาย band ซึ่งแต่ละ band แทนข้อมูลจากคลื่นแม่เหล็กไฟฟ้าต่าง ๆ เช่น Red, Green, NIR เป็นต้น
- ใช้สำหรับการวิเคราะห์ข้อมูลแบบพิกเซล เช่น คำนวณ NDVI, การแยกแหล่งที่ดิน, การคำนวณดัชนีต่าง ๆ
- สามารถใช้ method เช่น .select(), .clip(), .addBands(), .normalizedDifference() กับ ee.Image



ตัวอย่าง Method สำหรับ ee.Image

เมธอด	คำอธิบาย	ตัวอย่าง
select()	เลือกแบนด์ (band) ที่ต้องการจากภาพ	image.select('B4')
addBands()	เพิ่มแบนด์ใหม่เข้าไปในภาพ	image.addBands(ndvi)
multiply(), divide(), add(), subtract()	คำนวณค่าพิกเซลแต่ละจุด (pixel-wise operation)	image.multiply(0.0001)
normalizedDifference()	คำนวณดัชนี normalized เช่น NDVI	image.normalizedDifference(['B5', 'B4'])
clip()	ตัดภาพตามขอบเขตที่กำหนด	image.clip(geometry)
mask()	กำหนดเพื่อแสดงเฉพาะพิกเซลที่ต้องการ	image.mask(cloudMask)
updateMask()	อัปเดต mask เดิม โดยซ้อนกับ mask ใหม่	image.updateMask(waterMask)
reduceRegion()	คำนวณค่าทางสถิติ เช่น mean, sum ภายใน geometry	image.reduceRegion({reducer: ee.Reducer.mean(), geometry: roi})
reduce()	ใช้กับ reducer เช่น mean, min, max สำหรับทุกแบนด์	image.reduce(ee.Reducer.mean())
rename()	เปลี่ยนชื่อแบนด์	image.rename(['Red', 'Green', 'Blue'])
resample()	ปรับวิธีการ resampling (เช่น bilinear, nearest)	image.resample('bilinear')
reproject()	กำหนดระบบพิกัดใหม่และความละเอียด	image.reproject('EPSG:4326', null, 30)
reduceConnectedComponents()	วิเคราะห์กลุ่มของพิกเซลติดกัน	ใช้ในการจำแนกวัตถุ
visualize()	เตรียมภาพให้แสดงผลได้ เช่น RGB	image.visualize({bands: ['B4','B3','B2'], min: 0, max: 3000})

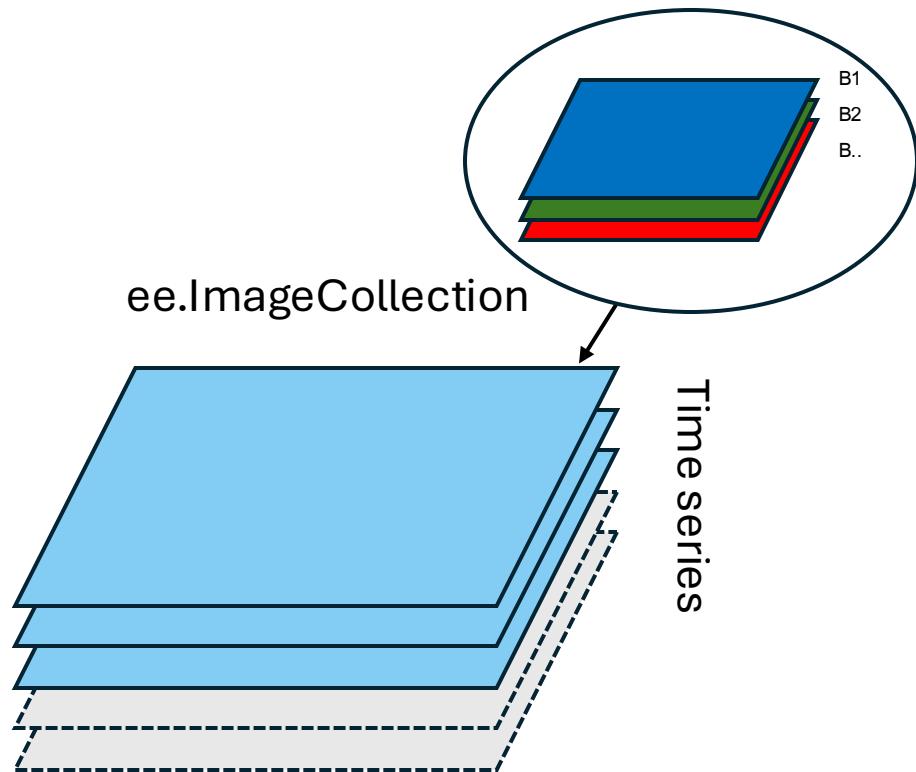
- ตัวอย่าง

```
// Image object
var image = ee.Image('LANDSAT/LC09/C02/T1_L2/LC09_129050_20231220')
    .multiply(0.0000275);
var band4 = image.select('SR_B4');
var band3 = image.select('SR_B3');
var band2 = image.select('SR_B2');
var rgb = band4.addBands(band3).addBands(band2);
Map.centerObject(image, 10);
Map.addLayer(rgb, {min: 0.2, max: 0.6, gamma: 2.0}, 'RGB');
```

ee.ImageCollection

ee.ImageCollection: กลุ่มภาพหลายภาพ (Image Set) ที่สามารถจัดการวิเคราะห์แบบชุดข้อมูลได้

- คือกลุ่มของ ee.Image หลาย ๆ ภาพที่จัดอยู่ในลักษณะเป็นคอลเลกชัน เช่น ภาพรายวัน รายเดือน รายปี หรือทุกภาพจากพื้นที่หนึ่ง
- ใช้สำหรับการวิเคราะห์ภาพในช่วงเวลาหรือหลายช่วงข้อมูล เช่น median, max, mean, หรือคำนวณแนวโน้ม (trend)
- มี method ที่สำคัญ เช่น .filterDate(), .filterBounds(), .map(), .median(), .mean()



ตัวอย่าง Method สำหรับ ee.ImageCollection

เมธอด	คำอธิบาย	ตัวอย่างการใช้งาน
filterDate(start, end)	กรองภาพตามช่วงวันที่	.filterDate('2023-01-01', '2023-12-31')
filterBounds(geometry)	กรองภาพตามขอบเขตพื้นที่	.filterBounds(roi)
filterMetadata()	กรองภาพจากค่าคุณสมบัติ (metadata) เช่น CLOUD_COVER	.filterMetadata('CLOUD_COVER', 'less_than', 10)
select(bands)	เลือกเฉพาะบางแบนด์	.select(['B4', 'B3', 'B2'])
map(function)	ใช้ฟังก์ชันกับทุกภาพใน collection	.map(function(img){ return img.normalizedDifference(['B8','B4']).rename('NDVI'); })
mean(), median(), min(), max(), sum(), stdDev()	คำนวณค่าสถิติของแต่ละพิกเซลข้าง collection	.median()
mosaic()	นำภาพทั้งหมดมาทับซ้อนกัน โดยใช้ค่าจากภาพที่อยู่บนสุด	.mosaic()
first(), limit(n)	เลือกภาพแรก, จำกัดจำนวนภาพ	.first() หรือ .limit(10)
sort(property, ascending)	เรียงลำดับภาพตามค่า metadata เช่น CLOUD_COVER	.sort('CLOUD_COVER')
size()	คืนจำนวนภาพใน collection	.size()
toList(n)	แปลงเป็น list ของภาพ	.toList(5)

- ตัวอย่าง

```
var geometry = ee.Geometry.Polygon(  
  [[[98.9171009716561, 18.815619476862654],  
   [98.9171009716561, 18.68557890893041],  
   [99.0873890575936, 18.68557890893041],  
   [99.0873890575936, 18.815619476862654]]]);  
  
var collection = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')  
  .filterDate('2021-01-01', '2021-01-31')  
  .filterBounds(geometry)  
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))  
  .select(['B4', 'B3', 'B2'])  
  .median()  
  .multiply(0.0001);  
  
Map.centerObject(geometry, 10);  
Map.addLayer(collection, {bands: ['B4', 'B3', 'B2'], min: 0.0, max: 0.3, gamma: 1.5}, 'Median Image')
```

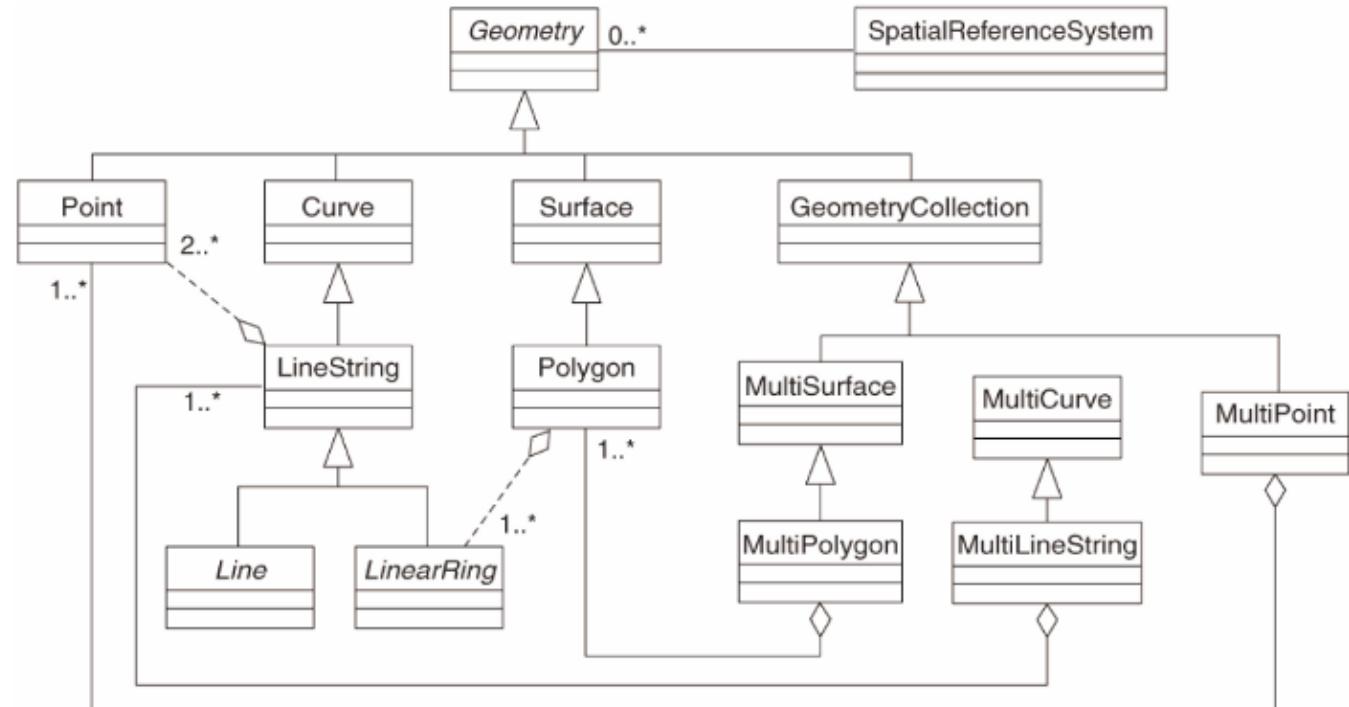
ee.Geometry, ee.Feature, และ ee.FeatureCollection

OGC SFA (Simple Feature Access)

เป็นมาตรฐานจาก **OGC (Open Geospatial Consortium)** สำหรับกำหนดรูปแบบ ข้อมูล เวกเตอร์ ที่ใช้ในระบบ GIS ได้แก่

- Geometry (จุด เส้น พื้นที่)
- Feature (วัตถุเชิงพื้นที่)
- FeatureCollection (กลุ่มของ Feature)

GEE รองรับแนวคิดนี้ ทั้งหมด เช่น
ee.Geometry, ee.Feature, และ
ee.FeatureCollection

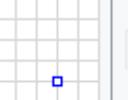
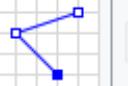
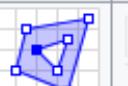


https://docs.qgis.org/3.40/en/docs/training_manual/spatial_database/s/simple_feature_model.html

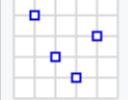
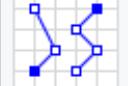
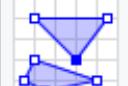
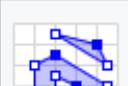
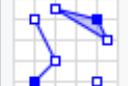
ee.Geometry

- เป็นองค์ประกอบเชิงพื้นที่พื้นฐาน เช่น จุด เส้น หรือพื้นที่ ซึ่งเป็นไปตาม OGC Geometry Types

Geometry primitives (2D)

Type	Examples	
Point		POINT (30 10)
LineString		LINESTRING (30 10, 10 30, 40 40)
Polygon		POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))
		POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))

Multipart geometries (2D)

Type	Examples	
MultiPoint		MULTIPOINT ((10 40), (40 30), (20 20), (30 10)) MULTIPOINT (10 40, 40 30, 20 20, 30 10)
MultiLineString		MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))
MultiPolygon		MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20), ((15 5, 40 10, 10 20, 5 10, 15 5))))
		MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40), (20 35, 10 30, 10 10, 30 5, 45 20, 20 35), (30 20, 20 15, 20 25, 30 20)))
GeometryCollection		GEOMETRYCOLLECTION (POINT (40 10), LINESTRING (10 10, 20 20, 10 40), POLYGON ((40 40, 20 45, 45 30, 40 40)))

- ตัวอย่าง

```
var point = ee.Geometry.Point([98.9171009716561, 18.815619476862654]);
var line = ee.Geometry.LineString([[98.9171009716561, 18.815619476862654], [99.0873890575936, 18.68557890893041]]);
var polygon = ee.Geometry.Polygon(
  [[[98.9171009716561, 18.815619476862654],
    [98.9171009716561, 18.68557890893041],
    [99.0873890575936, 18.68557890893041],
    [99.0873890575936, 18.815619476862654]]]);
```

ee.Feature

เป็นข้อมูลที่ประกอบด้วย

- **Geometry** (เป็นข้อมูลเชิงพื้นที่ เช่น point, line, polygon)
- **Properties** (เป็นข้อมูลไม่เชิงพื้นที่ เช่น name, id)

Feature: JSON
 Feature (Point, 2 properties) JSON
 type: Feature
 geometry: Point (98.92, 18.82)
 properties: Object (2 properties)

Feature: JSON
 Feature (Polygon, 2 properties) JSON
 type: Feature
 geometry: Polygon, 5 vertices
 properties: Object (2 properties)

- ตัวอย่าง

```
// Feature object
var point = ee.Geometry.Point([98.9171009716561, 18.815619476862654]);
var feature = ee.Feature(point, {name: 'Chiang Mai', population: 1000000});
print('Feature:', feature);

// Polygon feature
var polygon = ee.Geometry.Polygon(
  [[[98.9171009716561, 18.815619476862654],
    [98.9171009716561, 18.68557890893041],
    [99.0873890575936, 18.68557890893041],
    [99.0873890575936, 18.815619476862654]]]);
var feature = ee.Feature(polygon, {name: 'Chiang Mai', population: 1000000});
print('Feature:', feature);
// add to map
Map.centerObject(feature, 10);
Map.addLayer(feature, {color: 'red'}, 'Feature');
```

ee.FeatureCollection

- เป็น ชุดของ Feature

```
‐ FeatureCollection (2 elements, 3 columns)      JSON
    type: FeatureCollection
    ‐ columns: Object (3 properties)
    ‐ features: List (2 elements)
        ‐ 0: Feature 0 (Point, 2 properties)
            type: Feature
            id: 0
            ‐ geometry: Point (98.92, 18.82)
            ‐ properties: Object (2 properties)
        ‐ 1: Feature 1 (Point, 2 properties)
            type: Feature
            id: 1
            ‐ geometry: Point (99.09, 18.69)
            ‐ properties: Object (2 properties)
```

- ตัวอย่าง

```
var point1 = ee.Geometry.Point([98.9171009716561, 18.815619476862654]);
var point2 = ee.Geometry.Point([99.0873890575936, 18.68557890893041]);
var feature1 = ee.Feature(point1, {name: 'Chiang Mai', population: 1000000});
var feature2 = ee.Feature(point2, {name: 'Sarapee', population: 8000000});
var featureCollection = ee.FeatureCollection([feature1, feature2]);
print('Feature Collection:', featureCollection);
// add to map
Map.centerObject(featureCollection, 10);
Map.addLayer(featureCollection, {color: 'red'}, 'Feature Collection');
```

geojson

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "properties": {  
        "name": "point A"  
      },  
      "geometry": {  
        "coordinates": [  
          98.97561178794649,  
          18.793169794747826  
        ],  
        "type": "Point"  
      },  
      "id": 0  
    },  
    {  
      "type": "Feature",  
      "properties": {  
        "name": "point B"  
      },  
      "geometry": {  
        "coordinates": [  
          98.9603370670639,  
          18.74579268980837  
        ],  
        "type": "Point"  
      },  
      "id": 1  
    }  
  ]  
}
```

The diagram illustrates the hierarchical structure of a GeoJSON FeatureCollection. The root object is labeled 'ee.FeatureCollection'. It contains two 'ee.Feature' objects, each represented by a blue bracket. Each 'ee.Feature' object contains an 'ee.Geometry' object, also represented by a blue bracket. The 'ee.Geometry' objects contain 'coordinates' arrays.

Reduce

- **Reduce** คือการสรุปค่าข้อมูลจากหลาย ๆ พิกเซล หลายช่วงเวลา หรือหลาย band ให้เป็นค่าที่เรียบง่าย ขึ้น เช่น ค่าเฉลี่ย ค่าสูงสุด ค่าต่ำสุด หรือค่ามัธยฐาน
- ประเภทของ Reducers

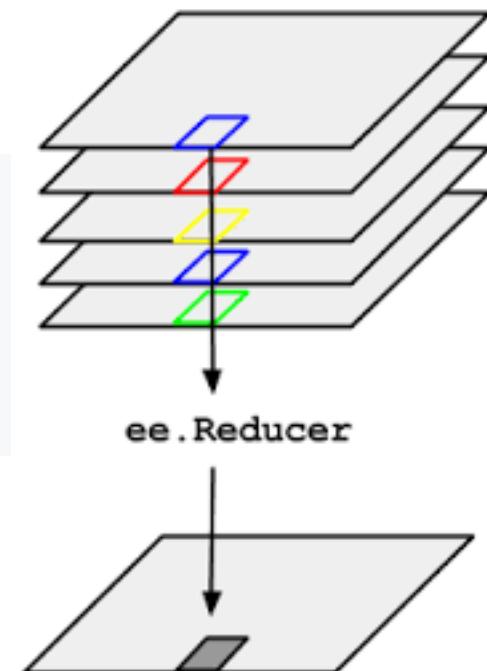
ประเภท	ใช้กับ	เป้าหมาย	ตัวอย่างเมธอด
Time Reducer	ImageCollection	รวมค่าตามเวลา	.reduce(ee.Reducer.mean())
Spatial Reducer	Image	รวมค่าตามพื้นที่ (geometry)	.reduceRegion(), .reduceRegions()
Band Reducer	Image หลาย band	รวมค่าระหว่าง band	.reduce(ee.Reducer.mean())

1. Time Reducers (การย่อข้อมูลตามเวลา)

- ใช้กับ ImageCollection เพื่อย่อหlaysภาพในช่วงเวลาหนึ่งให้เหลือเพียงภาพเดียว เช่น ใช้ mean(), median(), max(), min() เพื่อหาค่าเฉลี่ยรายปี/เดือน/วัน เป็นต้น

```
var s2 = ee.ImageCollection('COPERNICUS/S2')
    .filterDate('2021-01-01', '2021-12-31')
    .filterBounds(polygon);

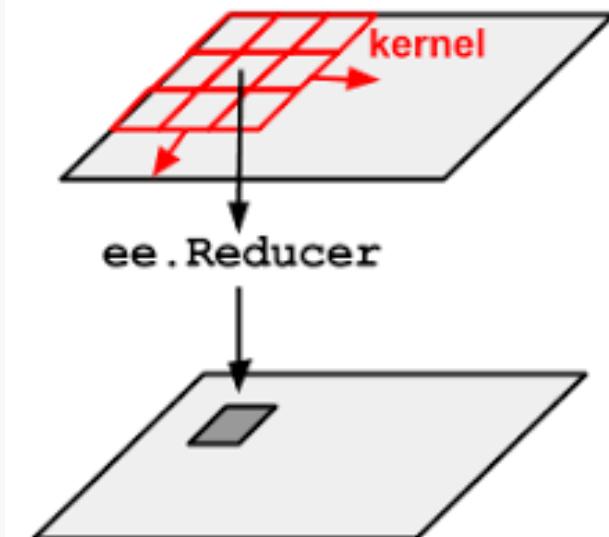
// Compute per-pixel mean across time
var meanTime = s2.reduce(ee.Reducer.mean());
```



2. Spatial Reducers (การย่อข้อมูลเชิงพื้นที่)

- ใช้ลดค่าพิกเซลภายในพื้นที่กำหนด เช่น พื้นที่ขอบเขตตำบล หรือ buffer
- ทำงานร่วมกับ .reduceRegion() หรือ .reduceRegions()
- เหมาะสมสำหรับการคำนวณค่าเฉลี่ย NDVI, LST, หรือข้อมูลอื่นๆ ภายใน polygon

```
var polygon = ee.Geometry.Polygon(  
  [[[98.9171009716561, 18.815619476862654],  
   [98.9171009716561, 18.68557890893041],  
   [99.0873890575936, 18.68557890893041],  
   [99.0873890575936, 18.815619476862654]]]);  
  
var s2 = ee.ImageCollection('COPERNICUS/S2')  
    .filterDate('2021-01-01','2021-03-31')  
    .filterBounds(polygon);  
  
var composite = s2.median();  
var bandStats = composite.reduceRegion({  
  reducer: ee.Reducer.mean(),  
  geometry: polygon,  
  scale: 100  
});
```



3. Band Reducers (การย่อข้อมูลระหว่าง band)

- ใช้กับ ee.Image ที่มีหลาย band และต้องการสรุปค่าจากหลาย band เป็นค่ารวมเดียว เช่น คำนวณค่าเฉลี่ยของ band ทุกอัน
- ได้ค่าผลรวมจากหลาย band เช่น ค่าความเข้มเฉลี่ยของ RGB หรือ spectral index หลายตัวรวมกัน

```
var s2 = ee.ImageCollection('COPERNICUS/S2')
    .filterDate('2021-01-01', '2021-12-31')
    .filterBounds(polygon)
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
    .select(['B8', 'B4']);

var ndviCollection = s2.map(function(img) {
  var ndvi = img.normalizedDifference(['B8', 'B4']).rename('NDVI');
  return ndvi.copyProperties(img, img.propertyNames());
});

var meanNDVI = ndviCollection.reduce(ee.Reducer.mean());
```

การแสดงผลภาพ (Map.addLayer())

```
Map.addLayer(image, visParams, 'ชื่อชั้นข้อมูล');
```

- image: ภาพหรือข้อมูลที่จะนำมาแสดง (เช่น NDVI, Sentinel-2)
- visParams: การตั้งค่าการแสดงผล เช่น band, min/max, palette
- ชื่อชั้นข้อมูล: ชื่อที่แสดงใน Layers Panel
- ตัวอย่าง

```
var vis = {bands: ['B4', 'B3', 'B2'], min: 0, max: 3000};  
Map.addLayer(image, vis, 'ภาพ RGB');
```

การแสดงผล: จากการเลือก Band

- การเลือก Band ที่ใช้ขึ้นกับวัตถุประสงค์ เช่น

ประเภทภาพ	ดาวเทียม	Band ที่ใช้	คำอธิบาย
True Color	Landsat 8/9	B4 (Red), B3 (Green), B2 (Blue)	เหมือนตามการมองเห็นของตาบุคคล
True Color	Sentinel-2	B4 (Red), B3 (Green), B2 (Blue)	ใช้ดูภูมิประเทศทั่วไป, เมือง, น้ำ
False Color (Vegetation)	Landsat 8/9	B5 (NIR), B4 (Red), B3 (Green)	พืชพรรณจะแสดงเป็นสีเขียวเข้ม (สุขภาพดี)
False Color (Vegetation)	Sentinel-2	B8 (NIR), B4 (Red), B3 (Green)	ใช้ตรวจสอบสุขภาพพืช และพื้นที่เกษตร
False Color (SWIR)	Landsat 8/9	B7 (SWIR2), B5 (NIR), B4 (Red)	เน้นความชื้น ดิน ป่าไม้ ไฟป่า
False Color (SWIR)	Sentinel-2	B12 (SWIR2), B8 (NIR), B4 (Red)	ใช้ในการติดตามน้ำท่วม ไฟป่า และดิน

- ตัวอย่างโค้ดการแสดงผลแบบ True Color (Sentinel-2)

```
var image = ee.Image('COPERNICUS/S2_SR/20220101T000000');
Map.addLayer(image, {
  bands: ['B4', 'B3', 'B2'],
  min: 0,
  max: 3000
}, 'True Color');
```

- ตัวอย่างโค้ดการแสดงผลแบบ False Color (Vegetation)

```
Map.addLayer(image, {
  bands: ['B8', 'B4', 'B3'],
  min: 0,
  max: 3000
}, 'False Color Vegetation');
```

การแสดงผล: ด้วย palette สำหรับข้อมูลเชิงค่าต่อเนื่อง (raster)

- พาเลตสีใช้ในการเน้นค่าข้อมูล เช่น NDVI, ความสูง, ความชื้น
- ใช้ร่วมกับคำสั่ง Map.addLayer() โดยกำหนด palette: ['สี1', 'สี2', ...]
- ตัวอย่าง แสดงผล NDVI ด้วยพาเลตสีเขียว

```
var ndviParams = {  
  min: 0,  
  max: 1,  
  palette: ['white', 'green']  
};  
Map.addLayer(ndvi, ndviParams, 'NDVI');
```

- ตัวอย่าง แสดงผลค่าความสูงจาก SRTM

```
var srtm = ee.Image('USGS/SRTMGL1_003');
var visElev = {
  min: 0,
  max: 3000,
  palette: ['blue', 'green', 'yellow', 'orange', 'red']
};
Map.addLayer(srtm, visElev, 'SRTM Elevation');
```

การแสดงผลแบบเวกเตอร์ (Vector Data)

- ตัวอย่าง: แสดงขอบเขตการปกรอง

```
var province = ee.FeatureCollection('TIGER/2018/States');
Map.addLayer(province, {
  color: 'blue'
}, 'ขอบเขตจังหวัด');
```

- ตัวอย่าง: การใส่สีพื้นที่ด้วย fillColor

```
var district = ee.FeatureCollection('FAO/GAUL_SIMPLIFIED_5
Map.addLayer(district.style({
  color: 'black',
  fillColor: 'yellow',
  width: 1
}), {}, 'เขตการปกรอง');
```

การสร้างส่วนติดต่อผู้ใช้ (UI) ใน Google Earth Engine

UI (User Interface) คือ การสร้างปฏิสัมพันธ์ระหว่างผู้ใช้กับโปรแกรม โดยใน GEE เราสามารถสร้างปุ่ม ตัวเลือก กล่องข้อความ และແຜງຄວບຄຸມເພື່ອໃຊ້ຈານຮ່ວມກັບແພນທີ

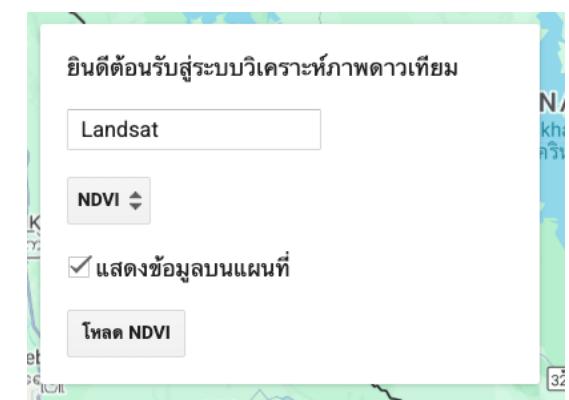
- ทำให้ผู้ใช้ຄວບຄຸມການທຳການຂອງແອປພລິເຄັ້ນໄດ້ຈໍາຍ
- ช່າຍສ້າງ dashboard ແລະ interactive map

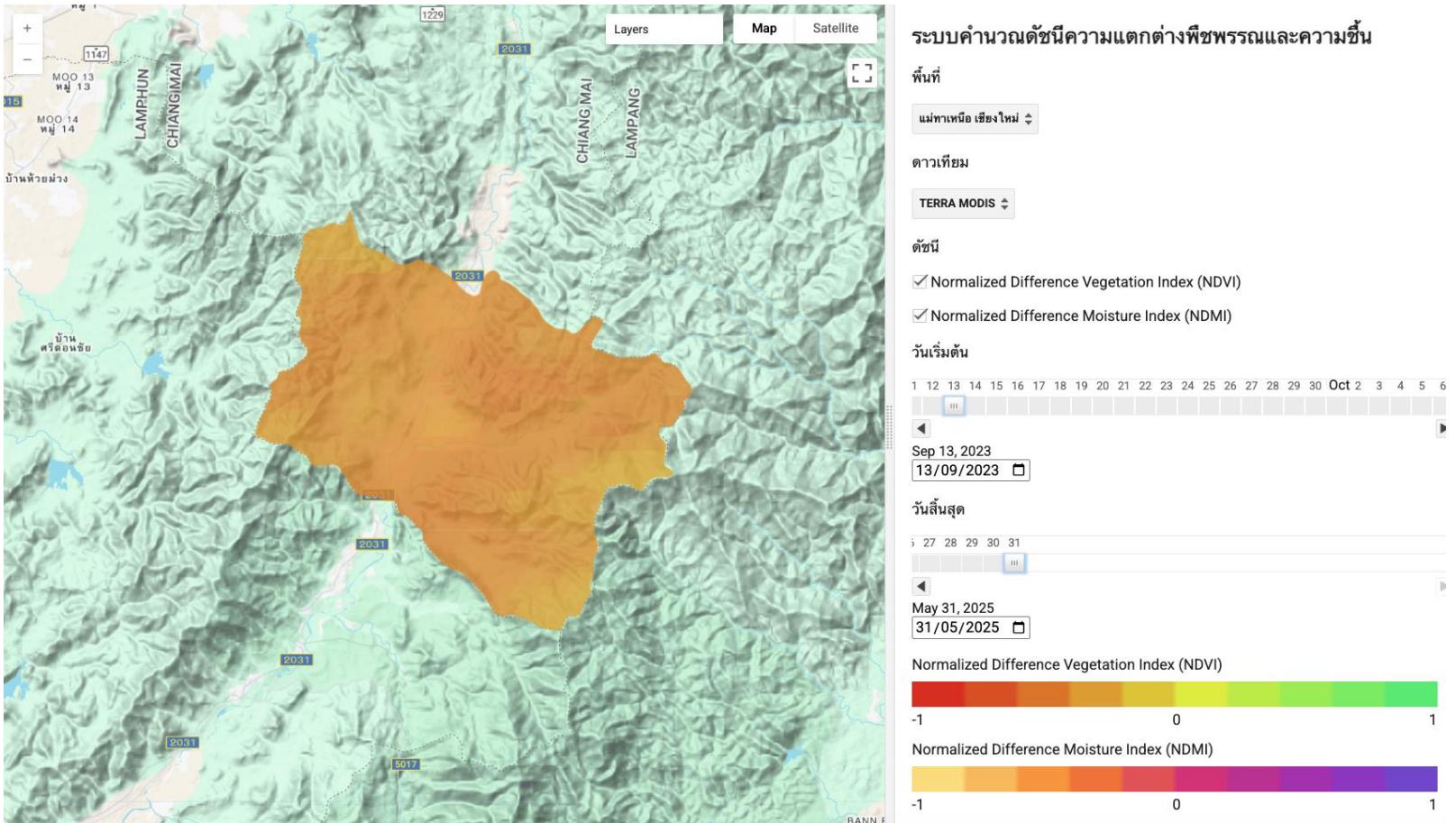
UI ประกอบด้วย

- Widgets (ເຄື່ອງມືອ):** ເຫັນ ປຸ່ມ (ui.Button), ຂ່ອງກອກຂໍ້ມູນ (ui.Textbox)
- Panel (ແຜງຄວບຄຸມ):** ສໍາຫຼັບຮ່ວມ widgets ມາຍໆ ອັນ
- Map (ແພນທີ):** ພື້ນທີ່ແສດງຜລຂໍ້ມູນເຊີງພື້ນທີ່
- Control & Event:** ກາຣໂຕຕອບເມື່ອຜູ້ໃຊ້ຄລິກຫຼືເລືອກຄ່າຕ່າງໆ

UI Widgets ใน GEE

Widget	คำอธิบาย
ui.Label	แสดงข้อความหรือผลลัพธ์
ui.Button	ปุ่มสำหรับสั่งงาน (เช่น คำนวณ, โหลดข้อมูล)
ui.Textbox	ช่องสำหรับกรอกข้อความหรือค่าต่าง ๆ
ui.Select	เมนูแบบเลือก (dropdown)
ui.Checkbox	กล่องติ๊กเลือกเปิด/ปิดเงื่อนไข
ui.Slider	แทบเลื่อนสำหรับค่าต่อเนื่อง (ตัวเลข)
ui.Panel	กล่องรวม widget หลายตัวเข้าเป็นชุด
ui.Chart	แสดงกราฟหรือข้อมูลเชิงสถิติ (ร่วมกับ ui.Panel)
ui.Map	ใช้เรียกและควบคุมแผนที่ (เช่น Map.addLayer(), Map.centerObject())
ui.SplitPanel	ใช้แบ่งพื้นที่แสดงผลแบบแนวตั้ง/แนวอน





ติดตามปริมาณเชื้อเพลิง

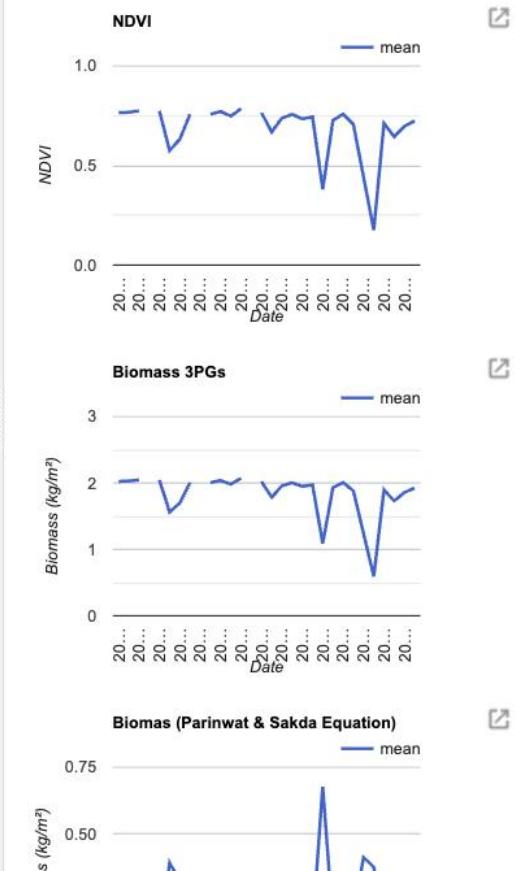
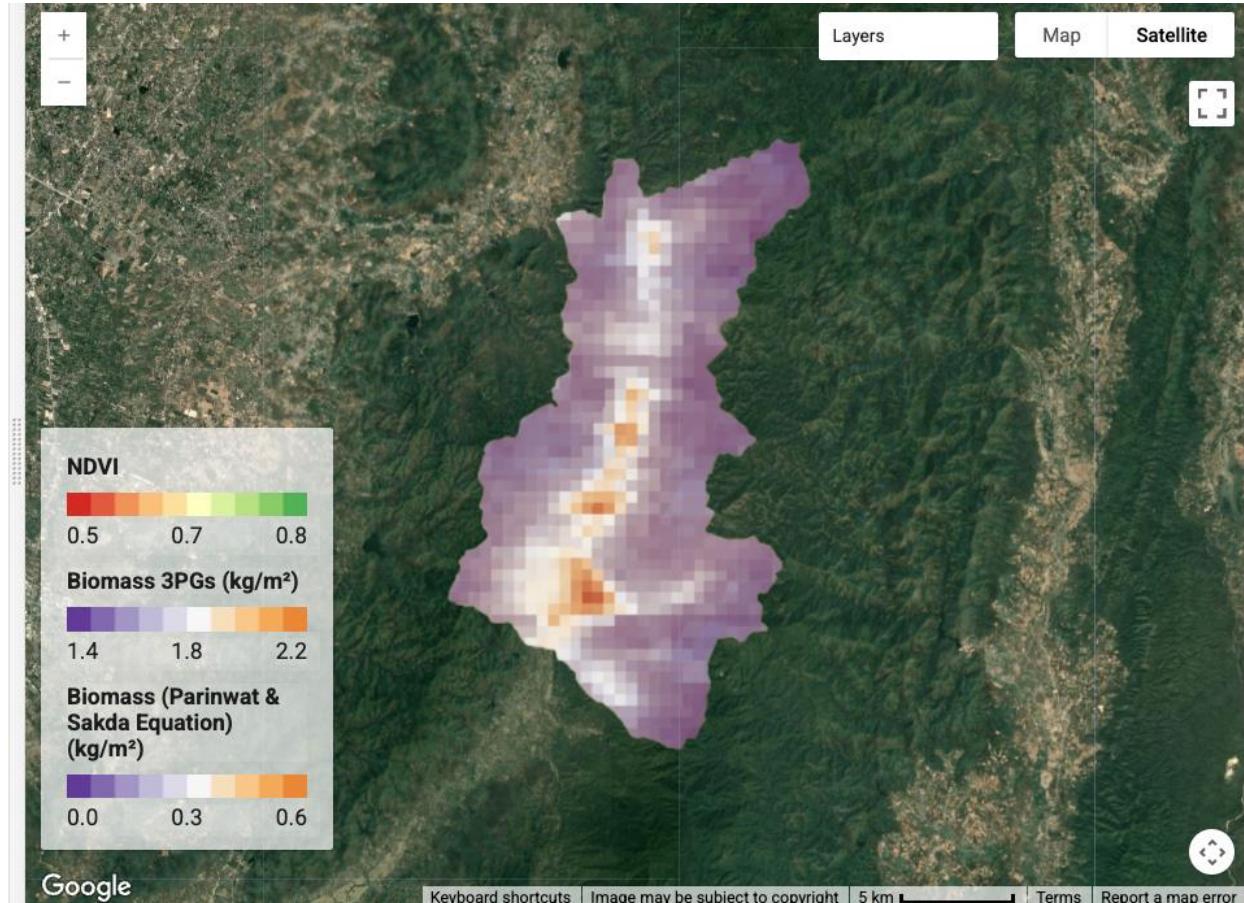
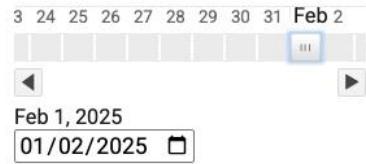
ค่านวณปริมาณเชื้อเพลิงด้วย:

1. จากวิธี 3PGs จากข้อมูลรายวันของดาวเทียม TERRA/AQUA MODIS
2. จากสมการที่พัฒนาขึ้นของโครงการ (Parinwat & Sakda Equation)

เลือกพื้นที่

แม่ทากเหนือ เชียงใหม่ ▾

เลือกวันที่



- Q&A