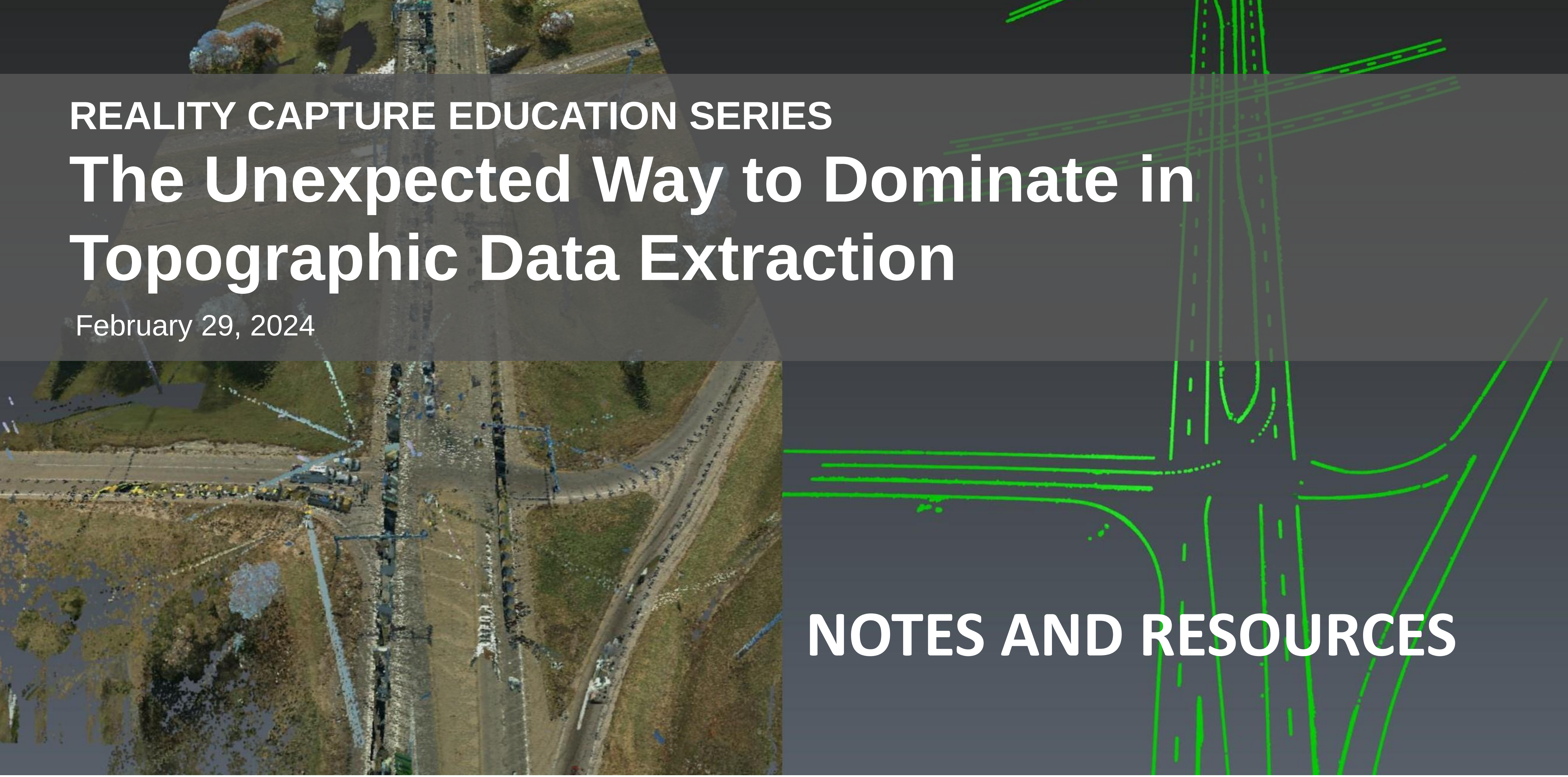
REALITY CAPTURE EDUCATION SERIES

# The Unexpected Way to Dominate in Topographic Data Extraction

February 29, 2024



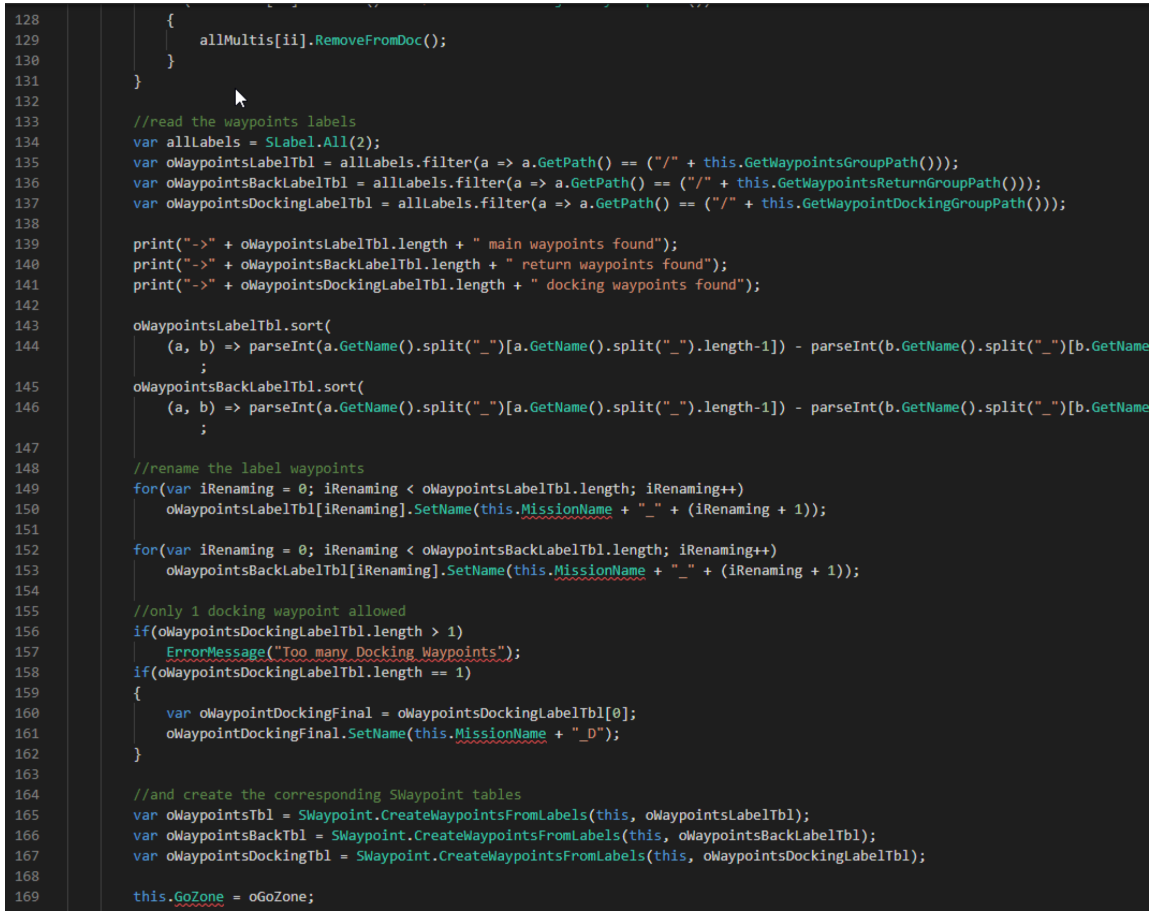
## NOTES AND RESOURCES

HEXAGON | Leica Geosystems

# Why Scripting?

## A Technical Perspective

- Unique workflows require unique solutions
- Why wait on official release?
- Build tools custom to individual projects
- Save button clicks on common workflows
- Automation is great but not always 100%
- It allows users to be flexible, fast, and efficient
- Document and examine workflows
    - It's just a set of instructions

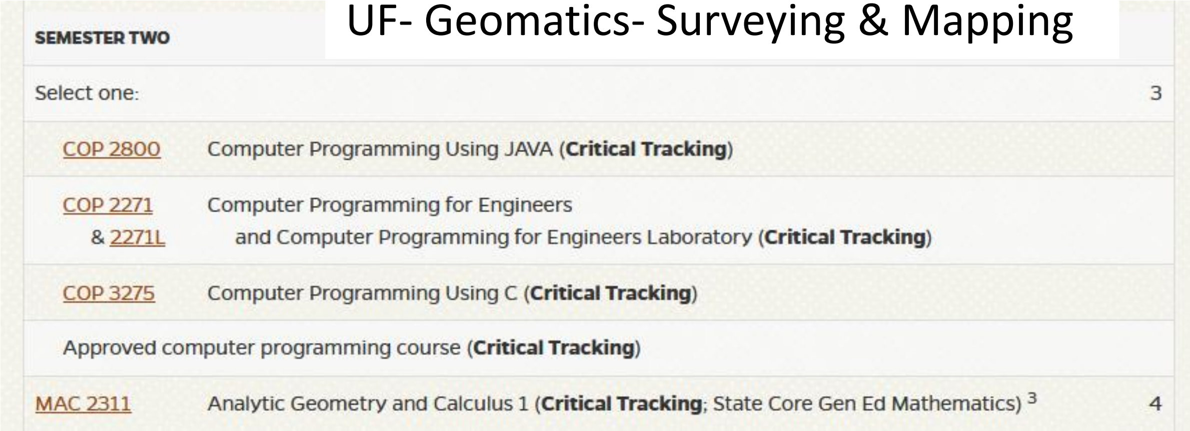


```
        {
            allMultis[ii].RemoveFromDoc();
        }
    }

    //read the waypoints labels
    var allLabels = SLabel.All(2);
    var oWaypointsLabelTbl = allLabels.filter(a => a.GetPath() == ("/" + this.GetWaypointsGroupPath()));
    var oWaypointsBackLabelTbl = allLabels.filter(a => a.GetPath() == ("/" + this.GetWaypointsReturnGroupPath()));
    var oWaypointsDockingLabelTbl = allLabels.filter(a => a.GetPath() == ("/" + this.GetWaypointDockingGroupPath()));

    print("->" + oWaypointsLabelTbl.length + " main waypoints found");
    print("->" + oWaypointsBackLabelTbl.length + " return waypoints found");
    print("->" + oWaypointsDockingLabelTbl.length + " docking waypoints found");

    oWaypointsLabelTbl.sort(
        (a, b) => parseInt(a.GetName().split("_")[a.GetName().split("_").length-1]) - parseInt(b.GetName().split("_")[b.GetName
            ;
    oWaypointsBackLabelTbl.sort(
        (a, b) => parseInt(a.GetName().split("_")[a.GetName().split("_").length-1]) - parseInt(b.GetName().split("_")[b.GetName
            ;

    //rename the label waypoints
    for(var iRenaming = 0; iRenaming < oWaypointsLabelTbl.length; iRenaming++)
        oWaypointsLabelTbl[iRenaming].SetName(this.MissionName + "_" + (iRenaming + 1));

    for(var iRenaming = 0; iRenaming < oWaypointsBackLabelTbl.length; iRenaming++)
        oWaypointsBackLabelTbl[iRenaming].SetName(this.MissionName + "_" + (iRenaming + 1));

    //only 1 docking waypoint allowed
    if(oWaypointsDockingLabelTbl.length > 1)
        ErrorMessage("Too many Docking Waypoints");
    if(oWaypointsDockingLabelTbl.length == 1)
    {
        var oWaypointDockingFinal = oWaypointsDockingLabelTbl[0];
        oWaypointDockingFinal.SetName(this.MissionName + "_D");
    }

    //and create the corresponding SWaypoint tables
    var oWaypointsTbl = SWaypoint.CreateWaypointsFromLabels(this, oWaypointsLabelTbl);
    var oWaypointsBackTbl = SWaypoint.CreateWaypointsFromLabels(this, oWaypointsBackLabelTbl);
    var oWaypointsDockingTbl = SWaypoint.CreateWaypointsFromLabels(this, oWaypointsDockingLabelTbl);

    this.GoZone = oGoZone;
```

# Why Scripting?

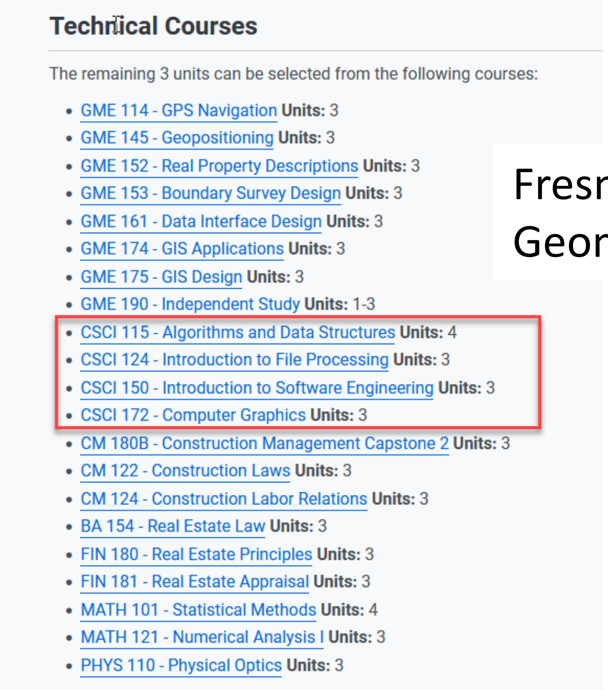
## A Management Perspective

- Human resources — make what you have more efficient
- Survey companies are tech companies now
- Survey teams have hidden resources on staff
- R&D is a rock-solid investment
- You're only as fast as your slowest workflow



UF- Geomatics- Surveying & Mapping

| SEMESTER TWO | | |
|---|---|---|
| Select one: | | 3 |
| COP 2800 | Computer Programming Using JAVA (**Critical Tracking**) | |
| COP 2271<br>& 2271L | Computer Programming for Engineers<br>and Computer Programming for Engineers Laboratory (**Critical Tracking**) | |
| COP 3275 | Computer Programming Using C (**Critical Tracking**) | |
| Approved computer programming course (**Critical Tracking**) | | |
| MAC 2311 | Analytic Geometry and Calculus 1 (**Critical Tracking**; State Core Gen Ed Mathematics) [3] | 4 |



### Technical Courses

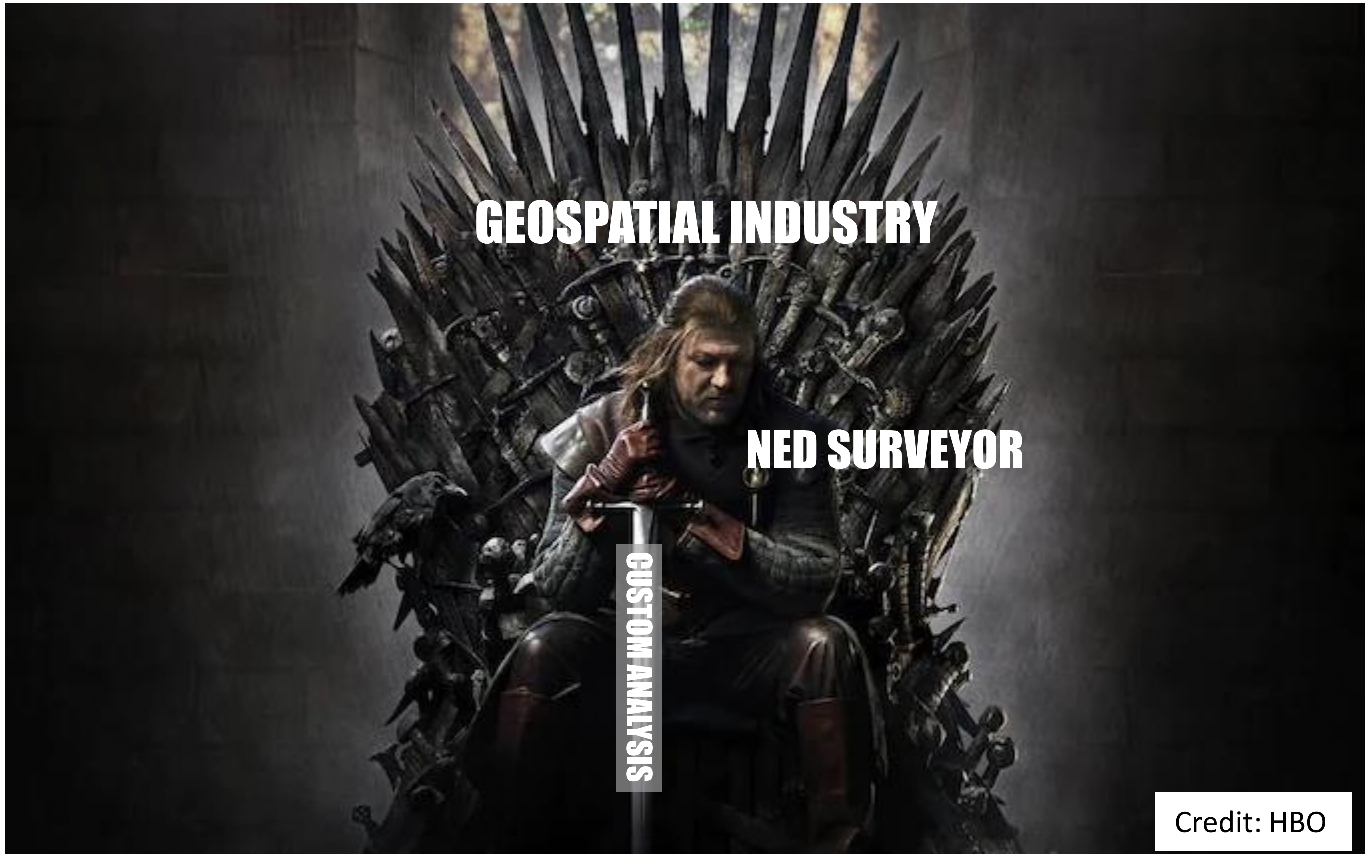
The remaining 3 units can be selected from the following courses:

- GME 114 - GPS Navigation **Units:** 3
- GME 145 - Geopositioning **Units:** 3
- GME 152 - Real Property Descriptions **Units:** 3
- GME 153 - Boundary Survey Design **Units:** 3
- GME 161 - Data Interface Design **Units:** 3
- GME 174 - GIS Applications **Units:** 3
- GME 175 - GIS Design **Units:** 3
- GME 190 - Independent Study **Units:** 1-3
- CSCI 115 - Algorithms and Data Structures **Units:** 4
- CSCI 124 - Introduction to File Processing **Units:** 3
- CSCI 150 - Introduction to Software Engineering **Units:** 3
- CSCI 172 - Computer Graphics **Units:** 3
- CM 180B - Construction Management Capstone 2 **Units:** 3
- CM 122 - Construction Laws **Units:** 3
- CM 124 - Construction Labor Relations **Units:** 3
- BA 154 - Real Estate Law **Units:** 3
- FIN 180 - Real Estate Principles **Units:** 3
- FIN 181 - Real Estate Appraisal **Units:** 3
- MATH 101 - Statistical Methods **Units:** 4
- MATH 121 - Numerical Analysis I **Units:** 3
- PHYS 110 - Physical Optics **Units:** 3

Fresno State Geomatics

HEXAGON | Leica Geosystems

# You’re only as strong as your weakest workflows

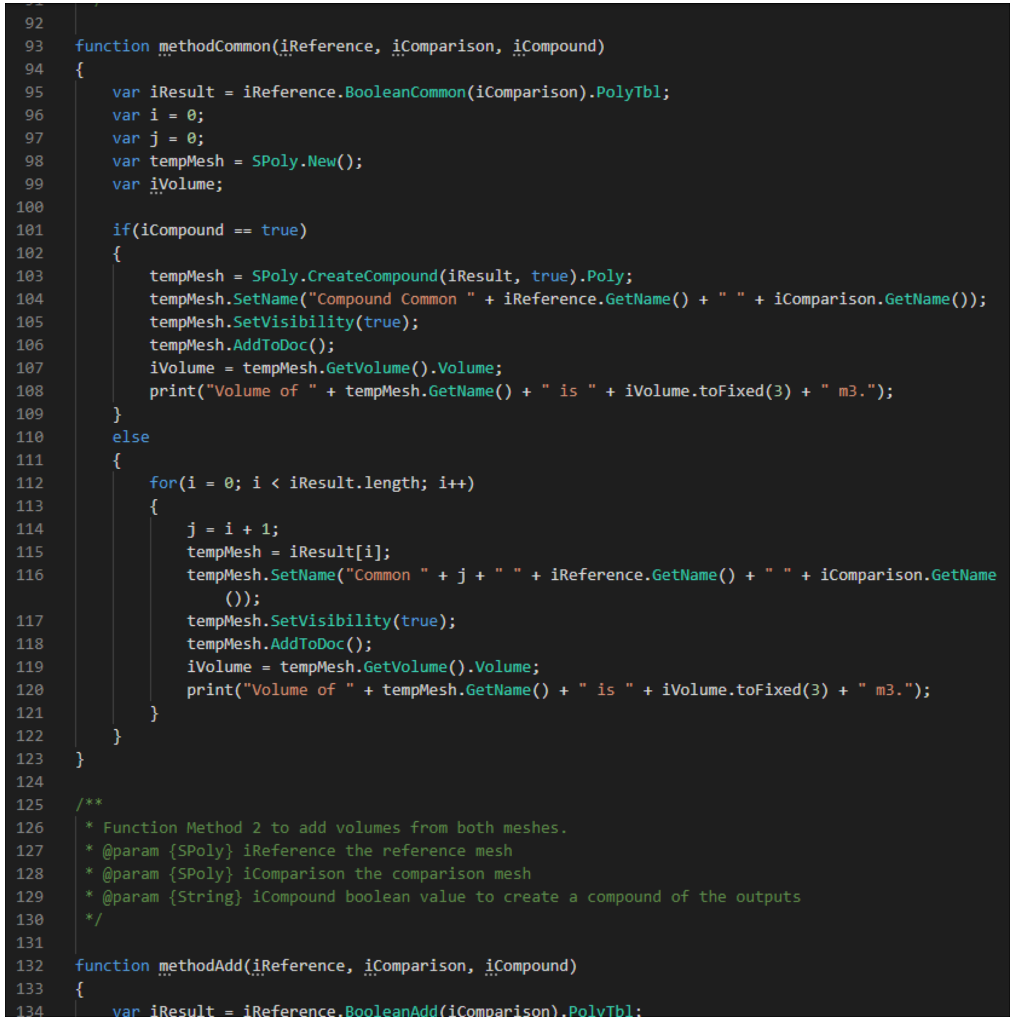


Credit: HBO

# Is Scripting Right for Me?

## It Depends…

- Does the tool exist already in the software?
  - Do you want to use it in combination with another tool?
- Do you have defined parameters you use every time?
- How often will this workflow be used in the future?
- Are you testing to see what works for a complex project?
- Do you want to hand a project off?

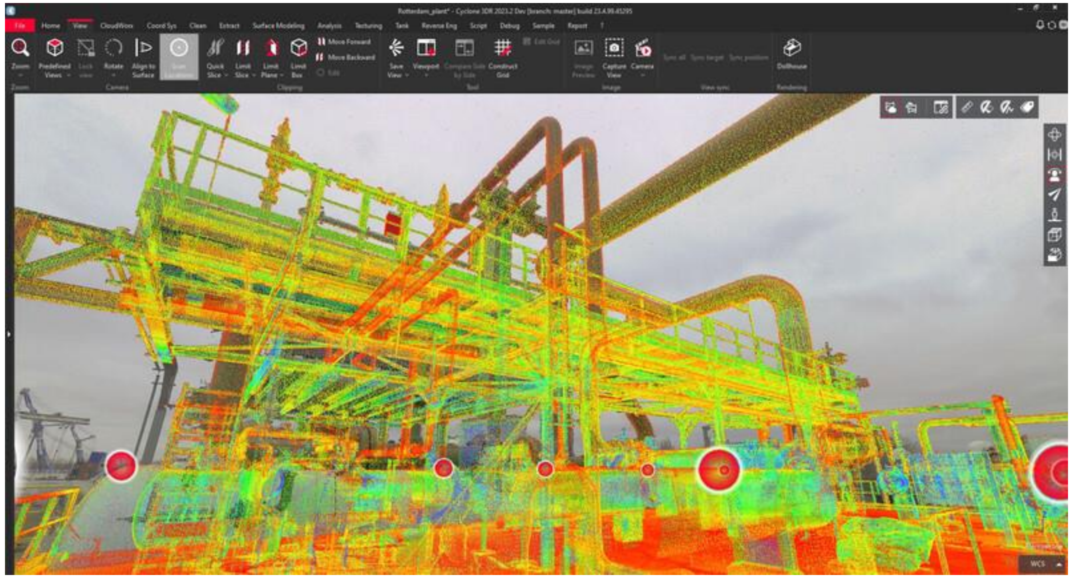


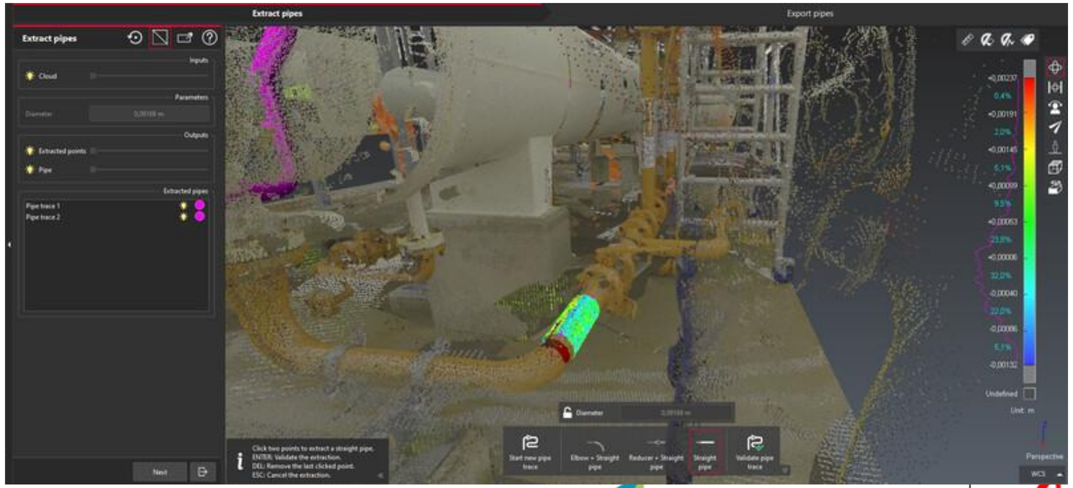
```
function methodCommon(iReference, iComparison, iCompound)
{
    var iResult = iReference.BooleanCommon(iComparison).PolyTbl;
    var i = 0;
    var j = 0;
    var tempMesh = SPoly.New();
    var iVolume;

    if(iCompound == true)
    {
        tempMesh = SPoly.CreateCompound(iResult, true).Poly;
        tempMesh.SetName("Compound Common " + iReference.GetName() + " " + iComparison.GetName());
        tempMesh.SetVisibility(true);
        tempMesh.AddToDoc();
        iVolume = tempMesh.GetVolume().Volume;
        print("Volume of " + tempMesh.GetName() + " is " + iVolume.toFixed(3) + " m3.");
    }
    else
    {
        for(i = 0; i < iResult.length; i++)
        {
            j = i + 1;
            tempMesh = iResult[i];
            tempMesh.SetName("Common " + j + " " + iReference.GetName() + " " + iComparison.GetName
                ());
            tempMesh.SetVisibility(true);
            tempMesh.AddToDoc();
            iVolume = tempMesh.GetVolume().Volume;
            print("Volume of " + tempMesh.GetName() + " is " + iVolume.toFixed(3) + " m3.");
        }
    }
}

/**
 * Function Method 2 to add volumes from both meshes.
 * @param {SPoly} iReference the reference mesh
 * @param {SPoly} iComparison the comparison mesh
 * @param {String} iCompound boolean value to create a compound of the outputs
 */

function methodAdd(iReference, iComparison, iCompound)
{
    var iResult = iReference.BooleanAdd(iComparison).PolyTbl;
```

# What is Cyclone 3DR?

## A toolbox for analyzing measured 3D data

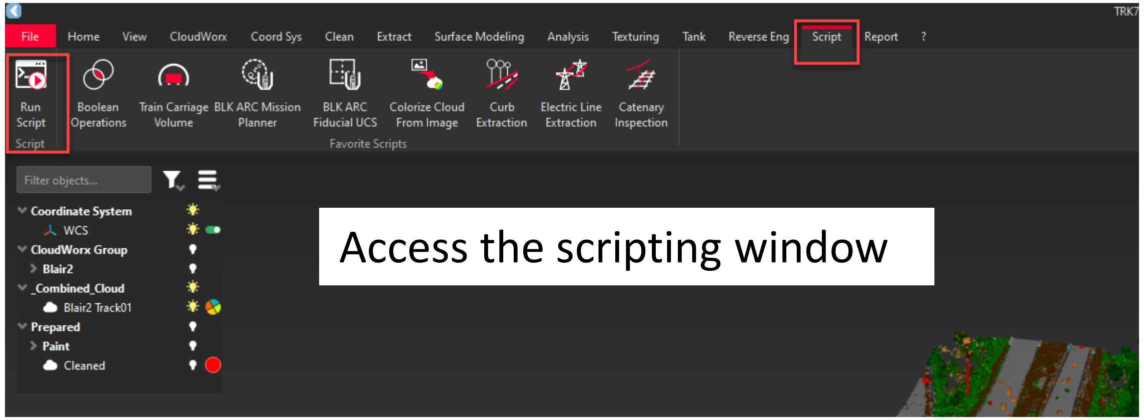
- Sensor agnostic
- Modular toolset with a powerful 3D viewing engine and useful connectivity
  - Built-in workflows
- Registration, meshing, image manipulation, reporting and extraction tools all in one place
- Well-documented help file with workflow guidance
- Powerful and well-document script API
  - Allows for specific and custom workflows to be built using the tools you find in the software.


Rotterdam_plant* - Cyclone 3DR 2023.2 Dev [branch: master] build 23.4.99.45295

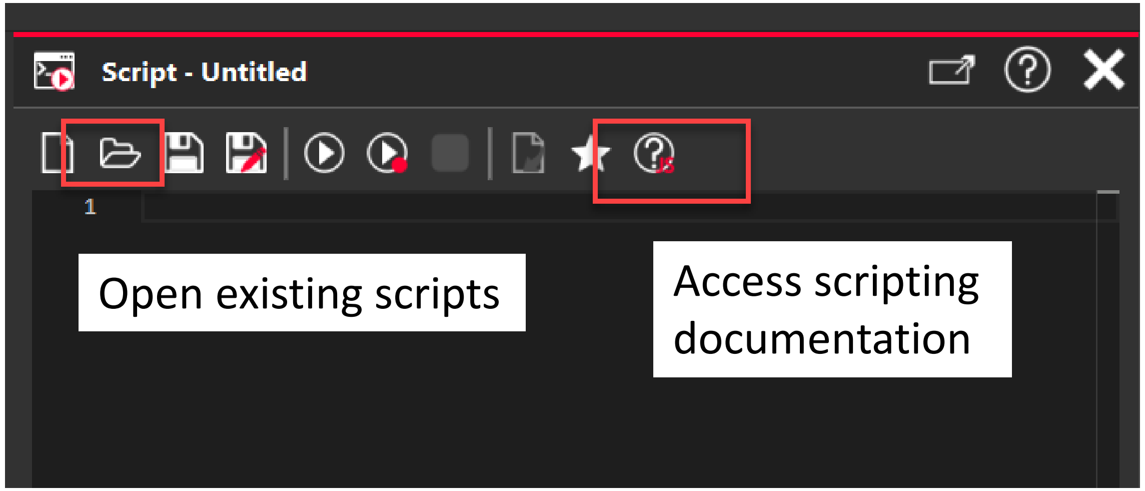


Click two points to extract a straight pipe.
ENTER: Validate the extraction.
DEL: Remove the last clicked point.
ESC: Cancel the extraction.

# Access the Scripting Window and Documentation



Access the scripting window

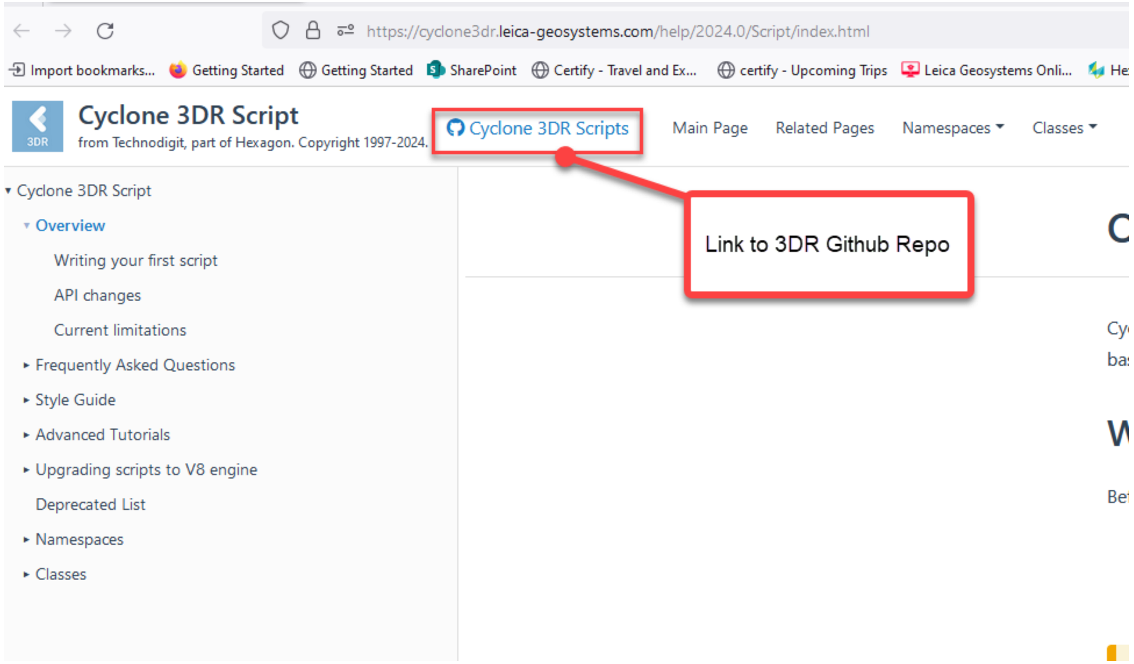


Open existing scripts

Access scripting documentation

# Help Files and Sample Scripts

## 3DR Script Documentation



https://cyclone3dr.leica-geosystems.com/help/2024.0/Script/index.html

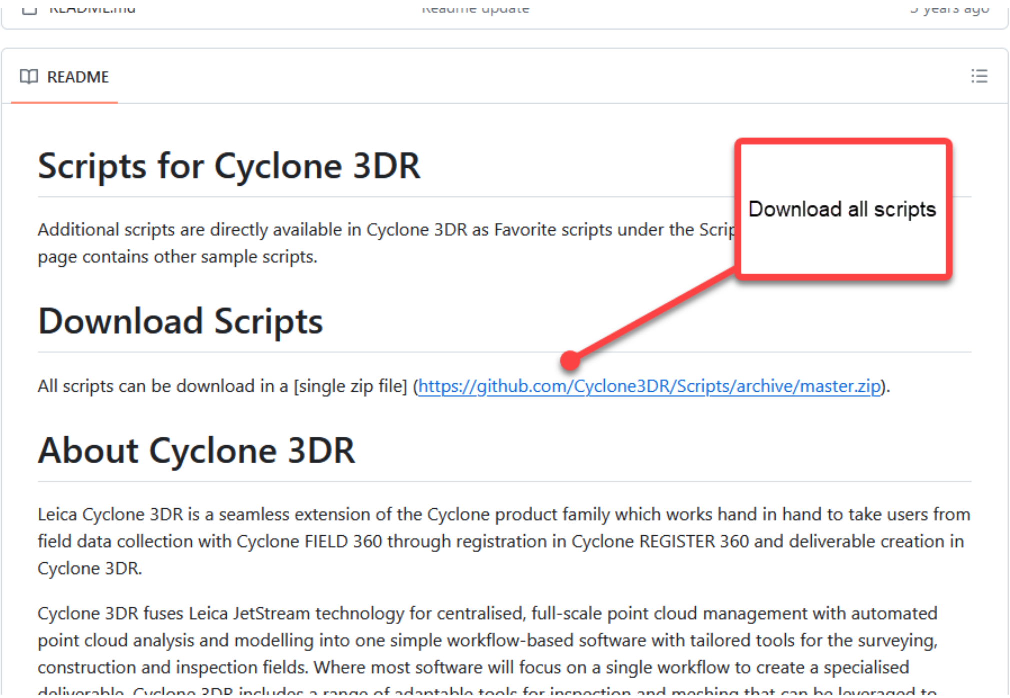
Cyclone 3DR Script

from Technodigit, part of Hexagon. Copyright 1997-2024.

- Cyclone 3DR Script
  - Overview
    - Writing your first script
    - API changes
    - Current limitations
  - Frequently Asked Questions
  - Style Guide
  - Advanced Tutorials
  - Upgrading scripts to V8 engine
  - Deprecated List
  - Namespaces
  - Classes

Link to 3DR Github Repo

## Github Scripts



### Scripts for Cyclone 3DR

Additional scripts are directly available in Cyclone 3DR as Favorite scripts under the Scrip page contains other sample scripts.

Download all scripts

### Download Scripts

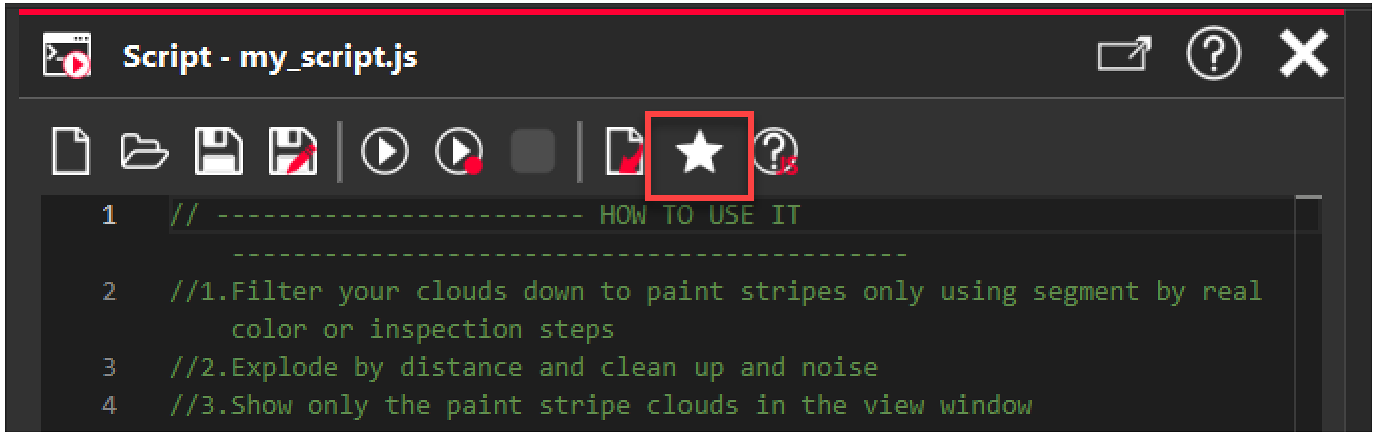
All scripts can be download in a [single zip file] (https://github.com/Cyclone3DR/Scripts/archive/master.zip).

### About Cyclone 3DR

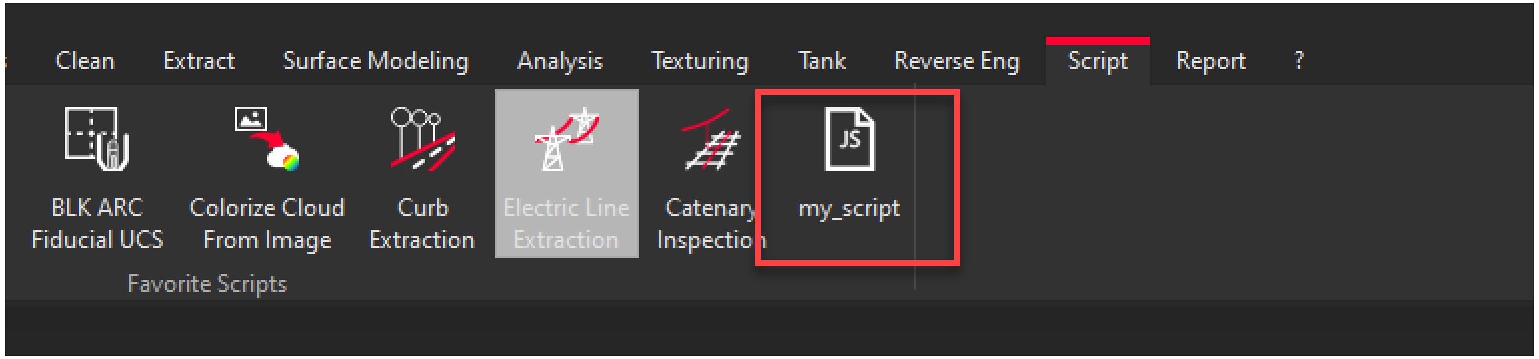
Leica Cyclone 3DR is a seamless extension of the Cyclone product family which works hand in hand to take users from field data collection with Cyclone FIELD 360 through registration in Cyclone REGISTER 360 and deliverable creation in Cyclone 3DR.

Cyclone 3DR fuses Leica JetStream technology for centralised, full-scale point cloud management with automated point cloud analysis and modelling into one simple workflow-based software with tailored tools for the surveying, construction and inspection fields. Where most software will focus on a single workflow to create a specialised deliverable. Cyclone 3DR includes a range of adaptable tools for inspection and meshing that can be leveraged to

# Saving a Script to Favorites



```
// ----------------------- HOW TO USE IT --------------------------------------------
//1.Filter your clouds down to paint stripes only using segment by real color or inspection steps
//2.Explode by distance and clean up and noise
//3.Show only the paint stripe clouds in the view window
```



# Planning for a Successful Script

1. Practice the workflow with the built-in tools.
    a. Write down everything, including the steps and parameters.
2. Evaluate how often the workflow will be used.
    a. If it’s only one or two times, just produce the results and get it done.
3. Write down in plain language what you want the script to do.
    a. i.e. “This script will take a group of clouds and color them different colors.”
    b. Then build out the logical framework in plain language using comments.
4. Look for the tools in the script documentation that are necessary.
    a. i.e. the script is working with clouds Classes > class list > SCloud and scan the available tools.
    b. Read and interpret the tool.
        a. Write a test script using that tool only to check the results.
5. Plug the necessary tools into your plain language framework and add the correct JavaScript syntax to the logic.
6. Test and troubleshoot.

# Important Terms

## Things to know before you get started…

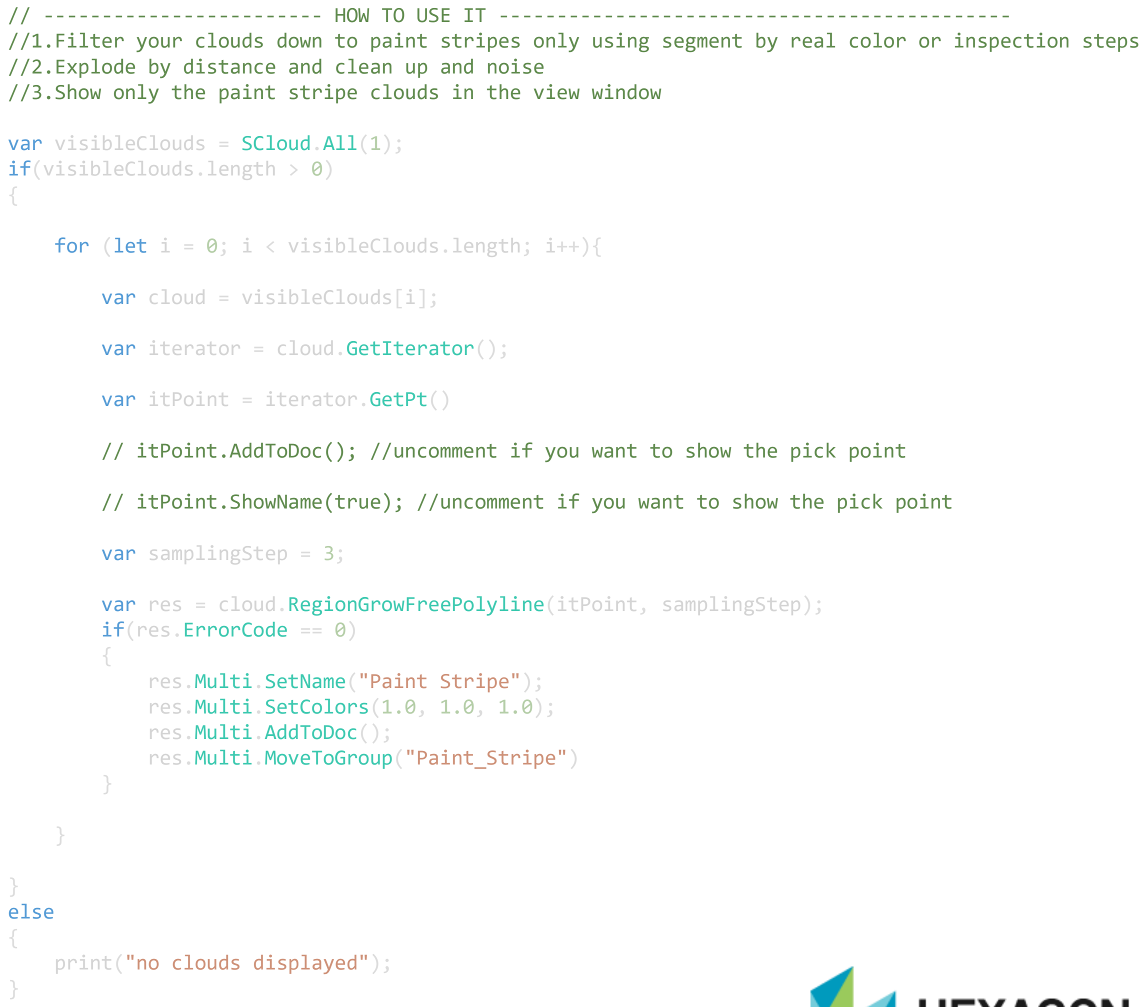
- JavaScript – the language that makes web pages interactive
- Variables – Where a value is stored for use later in the script
- Arrays – A list of things, variables, objects, other lists
- Dot notation – Syntax for accessing things on an object, of working down a tree of related tools
- If/else statements – A logical control that evaluates if a statement is true, then run some commands
- For loops – Logical control that runs the same commands a defined number of times
- Functions – A mini-program within a program.
- Objects – a way of storing many variables, other objects, functions together in a logical way
- Comments – anything with a // at the beginning will be ignored and can be used to make comments and notes

# Resources

## JavaScript and Script Examples

- JavaScript
  - Official documentation - Link
  - Video tutorial - Link
  - Text based tutorial - Link
- 3DR Script documentation - Link
- 3DR Script repository - Link

## Paint Stripe Script Example



```
// ----------------------- HOW TO USE IT -------------------------------------------
//1.Filter your clouds down to paint stripes only using segment by real color or inspection steps
//2.Explode by distance and clean up and noise
//3.Show only the paint stripe clouds in the view window

var visibleClouds = SCloud.All(1);
if(visibleClouds.length > 0)
{

    for (let i = 0; i < visibleClouds.length; i++){

        var cloud = visibleClouds[i];

        var iterator = cloud.GetIterator();

        var itPoint = iterator.GetPt()

        // itPoint.AddToDoc(); //uncomment if you want to show the pick point

        // itPoint.ShowName(true); //uncomment if you want to show the pick point

        var samplingStep = 3;

        var res = cloud.RegionGrowFreePolyline(itPoint, samplingStep);
        if(res.ErrorCode == 0)
        {
            res.Multi.SetName("Paint Stripe");
            res.Multi.SetColors(1.0, 1.0, 1.0);
            res.Multi.AddToDoc();
            res.Multi.MoveToGroup("Paint_Stripe")
        }

    }

}
else
{
    print("no clouds displayed");
}
```

## Learn more:

https://bit.ly/C3DRscriptingwebinar

## Contact us:

https://pure-surveying.com/contact/