Introduction to Python for MicroStation

Dan Eng – MicroStation Product Expert

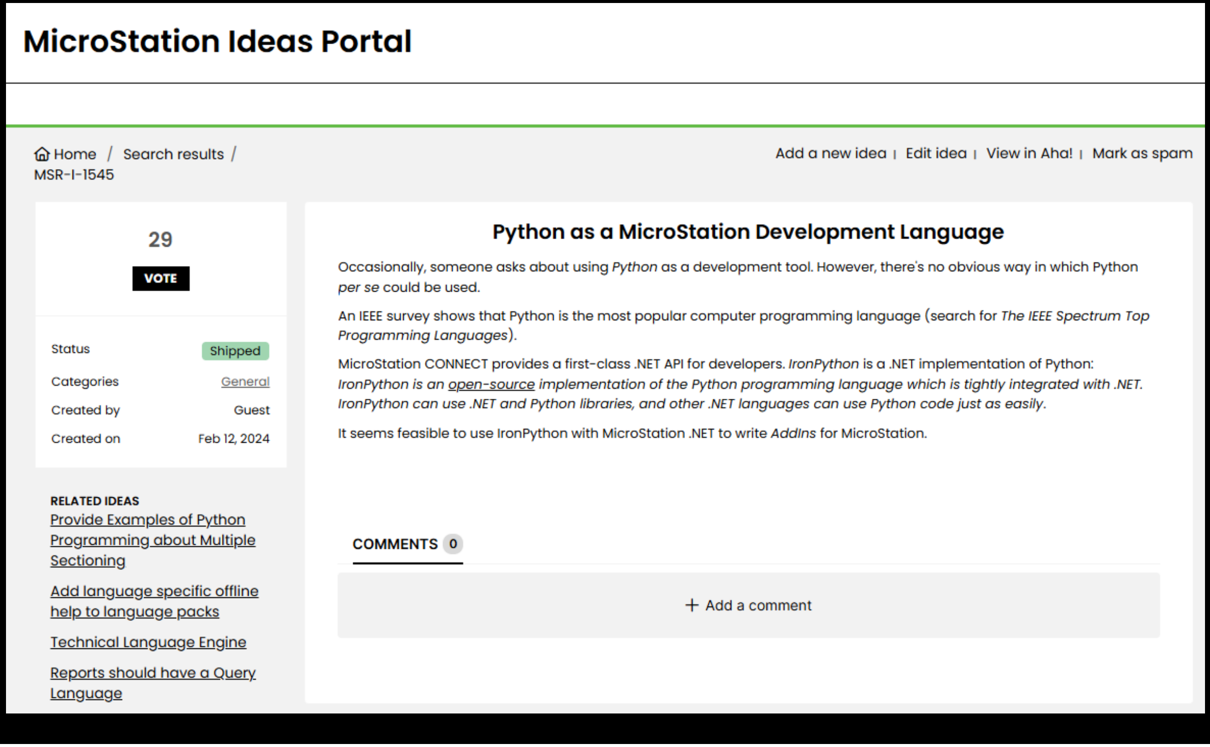
Bentley®

Presenters

Dan Eng
MicroStation Product Expert
dan.eng@bentley.com

Why Python?

- Easy to Learn and Use
- Mature and Supportive Community
- Thousands of Python Libraries
- Versatility, Efficiency, and Reliability
- Big data, Machine Learning and Cloud Computing
- First choice language for many programmers and students and professionals
- Its Open Source



MicroStation Ideas Portal

Home / Search results / MSR-I-1545
Add a new idea | Edit idea | View in Aha! | Mark as spam
29
VOTE

Status	Shipped
Categories	General
Created by	Guest
Created on	Feb 12, 2024

RELATED IDEAS

- Provide Examples of Python Programming about Multiple Sectioning
- Add language specific offline help to language packs
- Technical Language Engine
- Reports should have a Query Language

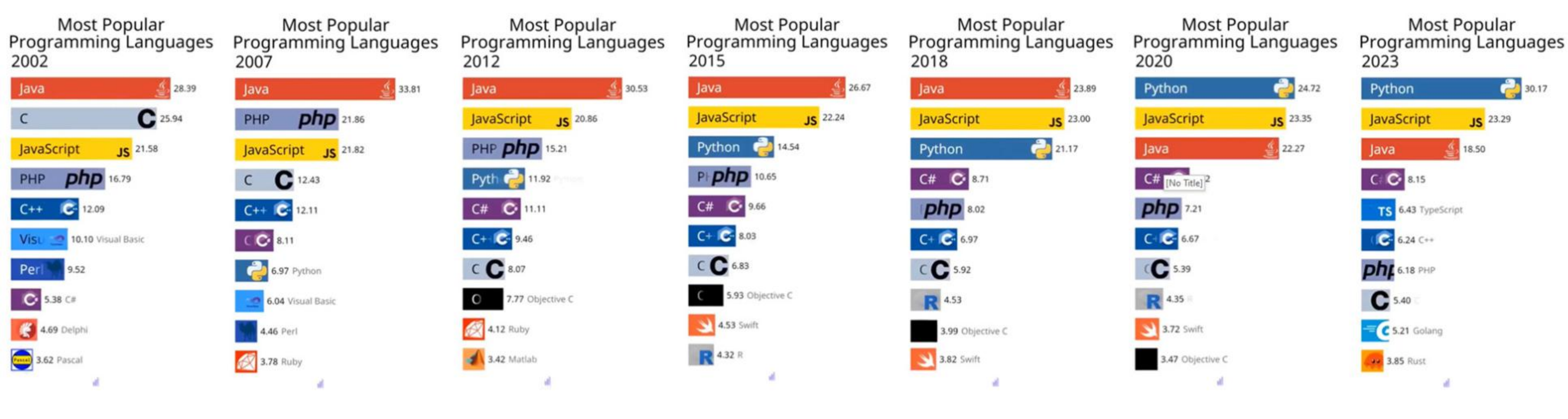
Python as a MicroStation Development Language

Occasionally, someone asks about using *Python* as a development tool. However, there's no obvious way in which Python *per se* could be used.

An IEEE survey shows that Python is the most popular computer programming language (search for *The IEEE Spectrum Top Programming Languages*).

MicroStation CONNECT provides a first-class .NET API for developers. *IronPython* is a .NET implementation of Python: *IronPython is an open-source implementation of the Python programming language which is tightly integrated with .NET. IronPython can use .NET and Python libraries, and other .NET languages can use Python code just as easily.*

It seems feasible to use IronPython with MicroStation .NET to write *AddIns* for MicroStation.



Most Popular Programming Languages 2002

Most Popular Programming Languages 2007

Most Popular Programming Languages 2012

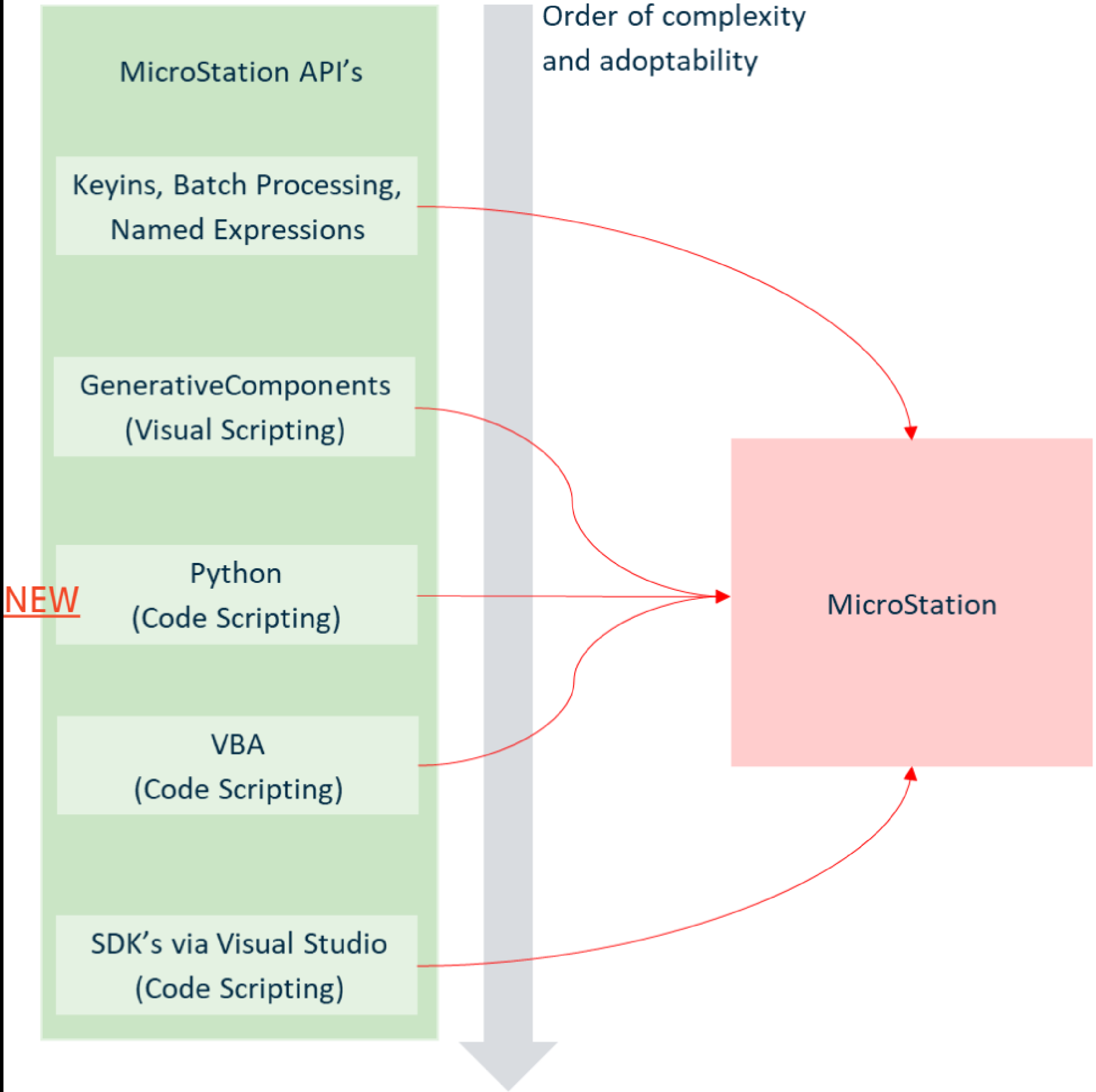
Most Popular Programming Languages 2015

Most Popular Programming Languages 2018

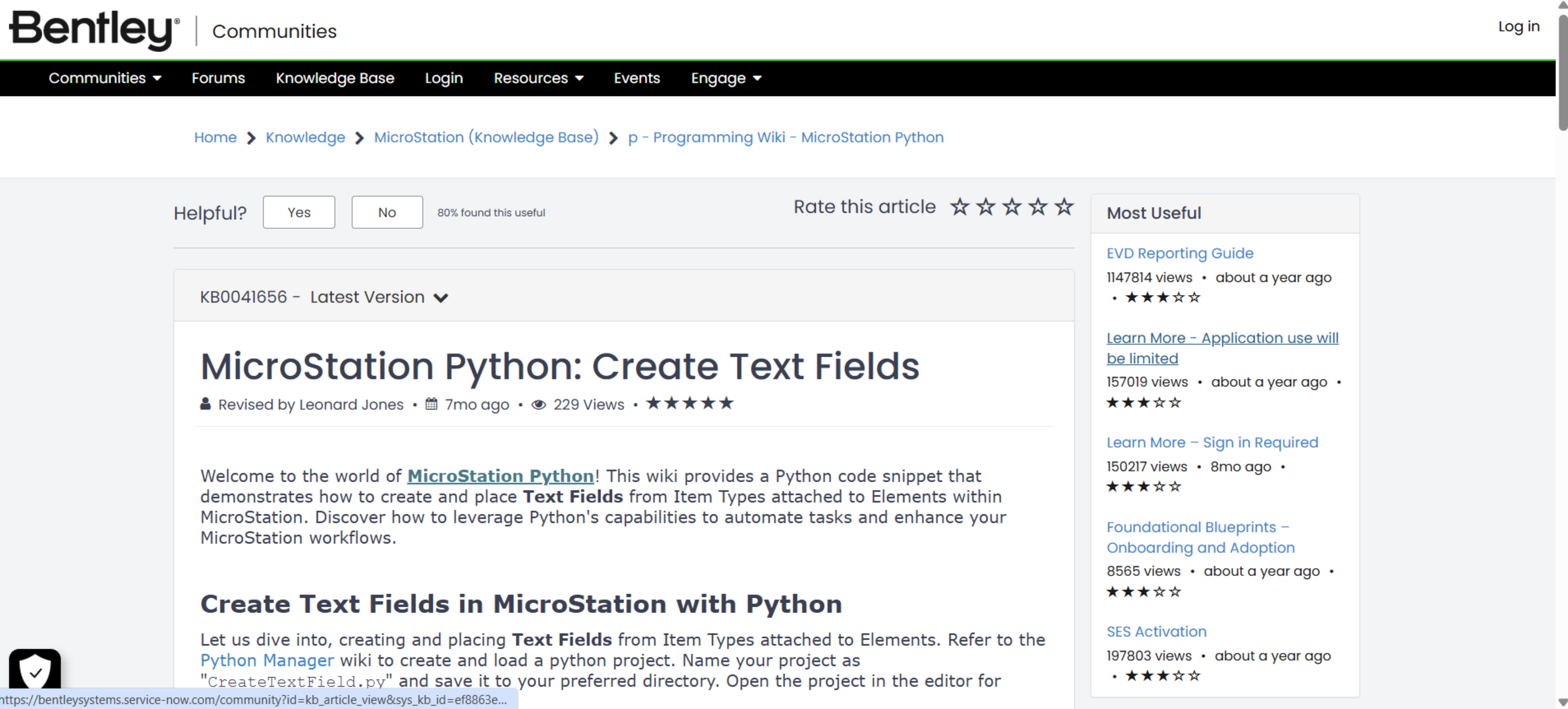
Most Popular Programming Languages 2020

Most Popular Programming Languages 2023

What can we use to Automate?



MicroStation API's
Keyins, Batch Processing, Named Expressions
GenerativeComponents (Visual Scripting)
NEW
Python (Code Scripting)
VBA (Code Scripting)
SDK's via Visual Studio (Code Scripting)
Order of complexity and adoptability
MicroStation



Home > Knowledge > MicroStation (Knowledge Base) > p - Programming Wiki - MicroStation Python

Helpful? Yes No 80% found this useful

Rate this article ☆☆☆☆☆

KB0041656 - Latest Version ⌄

MicroStation Python: Create Text Fields

Revised by Leonard Jones • 7mo ago • 229 Views • ★★★★★

Welcome to the world of **MicroStation Python**! This wiki provides a Python code snippet that demonstrates how to create and place **Text Fields** from Item Types attached to Elements within MicroStation. Discover how to leverage Python's capabilities to automate tasks and enhance your MicroStation workflows.

Create Text Fields in MicroStation with Python

Let us dive into, creating and placing **Text Fields** from Item Types attached to Elements. Refer to the Python Manager wiki to create and load a python project. Name your project as "`CreateTextField.py`" and save it to your preferred directory. Open the project in the editor for

Most Useful

EVD Reporting Guide
1147814 views • about a year ago • ★★★☆☆

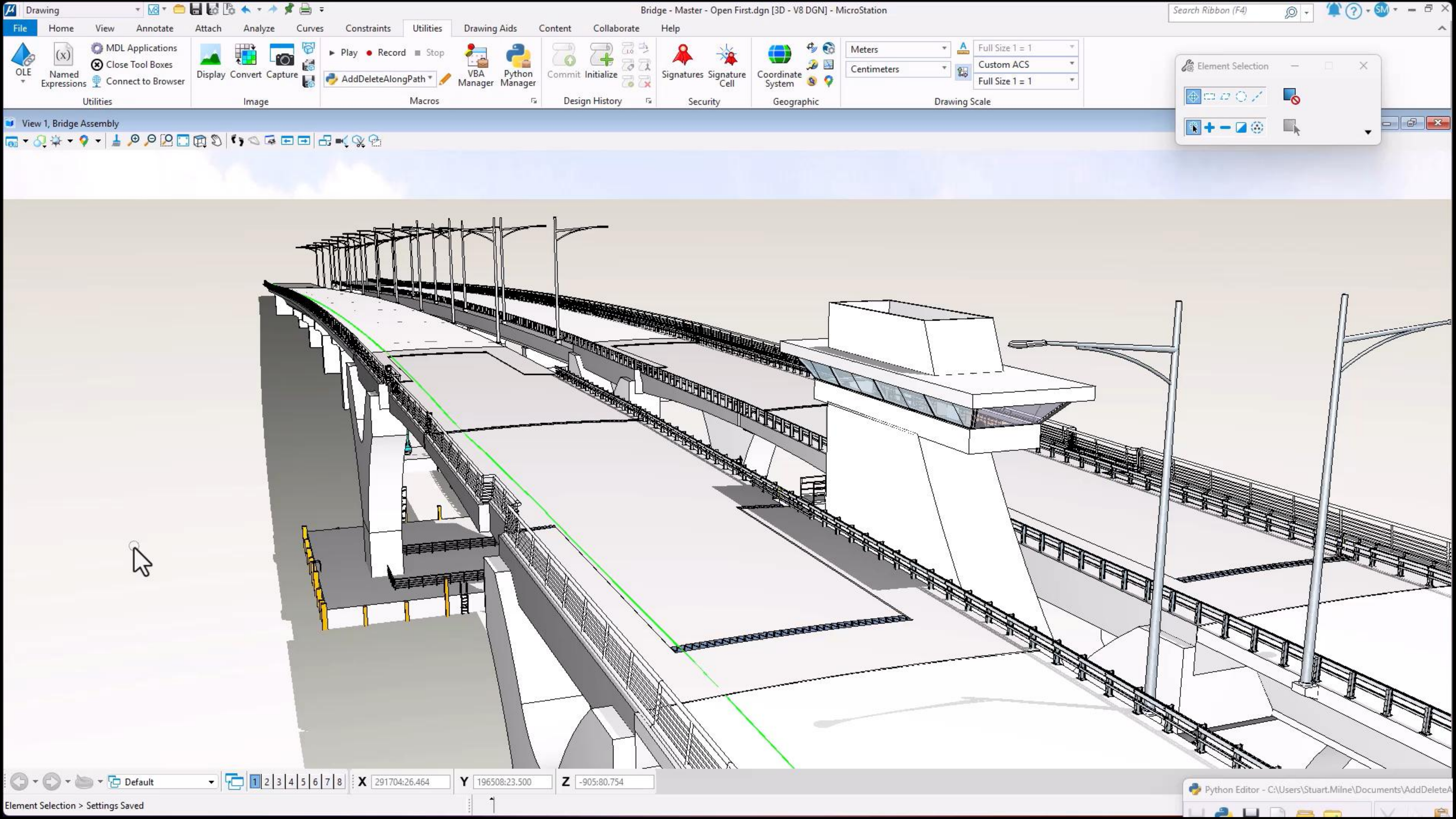
Learn More - Application use will be limited
157019 views • about a year ago • ★★★☆☆

Learn More – Sign in Required
150217 views • 8mo ago • ★★★☆☆

Foundational Blueprints – Onboarding and Adoption
8565 views • about a year ago • ★★★☆☆

SES Activation
197803 views • about a year ago • ★★★☆☆

https://bentleysystems.service-now.com/community?id=kb_article_view&sys_kb_id=ef8863e...



Bridge - Master - Open First.dgn [3D - V8 DGN] - MicroStation

Meters

Centimeters

Full Size 1 = 1

Custom ACS

Full Size 1 = 1

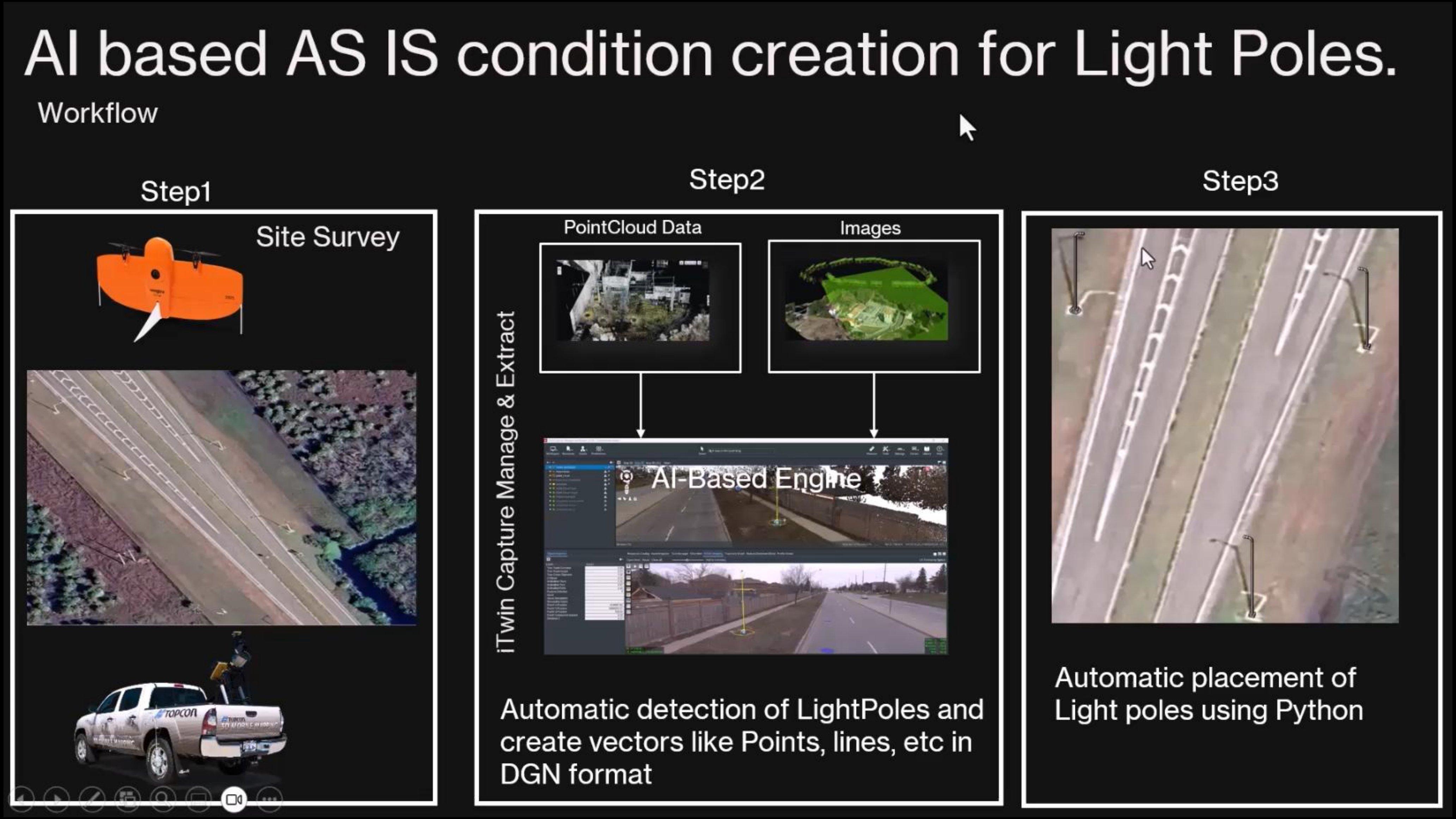
X 291704:26.464

Y 196508:23.500

Z -905:80.754

Element Selection > Settings Saved

Python Editor - C:\Users\Stuart.Milne\Documents\AddDeleteA



AI based AS IS condition creation for Light Poles.

Workflow

Step1

Site Survey

Step2

iTwin Capture Manage & Extract

Automatic detection of LightPoles and create vectors like Points, lines, etc in DGN format

Step3

Automatic placement of Light poles using Python

What about VBA?

- VBA is not going anywhere
- Python is an addition to VBA
- Updating VBA to Python?

What is the value of Python?

- Remove Roadblocks:
- Maximize adoptability:
- Improved Productivity:
- Win more work: .
- Increased visibility:
- Increased Footprint:
- Increased revenue:
- Innovation:
- Improved client engagement:
- Ecosystem development:

Python tools in MicroStation

Starting off with python code

Python Interpreter: Make sure you have Python installed on your computer. You can download it from the official Python website.

Text Editor or IDE: You can write your Python code in any text editor (like Notepad, Sublime Text, or VS Code) or an Integrated Development Environment (IDE) like PyCharm or Jupyter Notebook.

Basic Knowledge of Python Syntax: Understanding the basics of Python syntax, such as variables, data types, loops, and functions, is essential.

Libraries and Modules: Depending on what you want to achieve with your script, you might need to import certain libraries or modules (e.g., math, os, sys, requests).

Script File: Create a .py file where you'll write your code. For example, script.py.

```
Pydhon 3:10:12
(main: Fed 16 2024,
01:51:36]
[Clang 15:0 lang-
1500.0:40.1] on
wins2
```

```
def greet(name):
  return
   "'nglito, "
print(gret('brid")
```

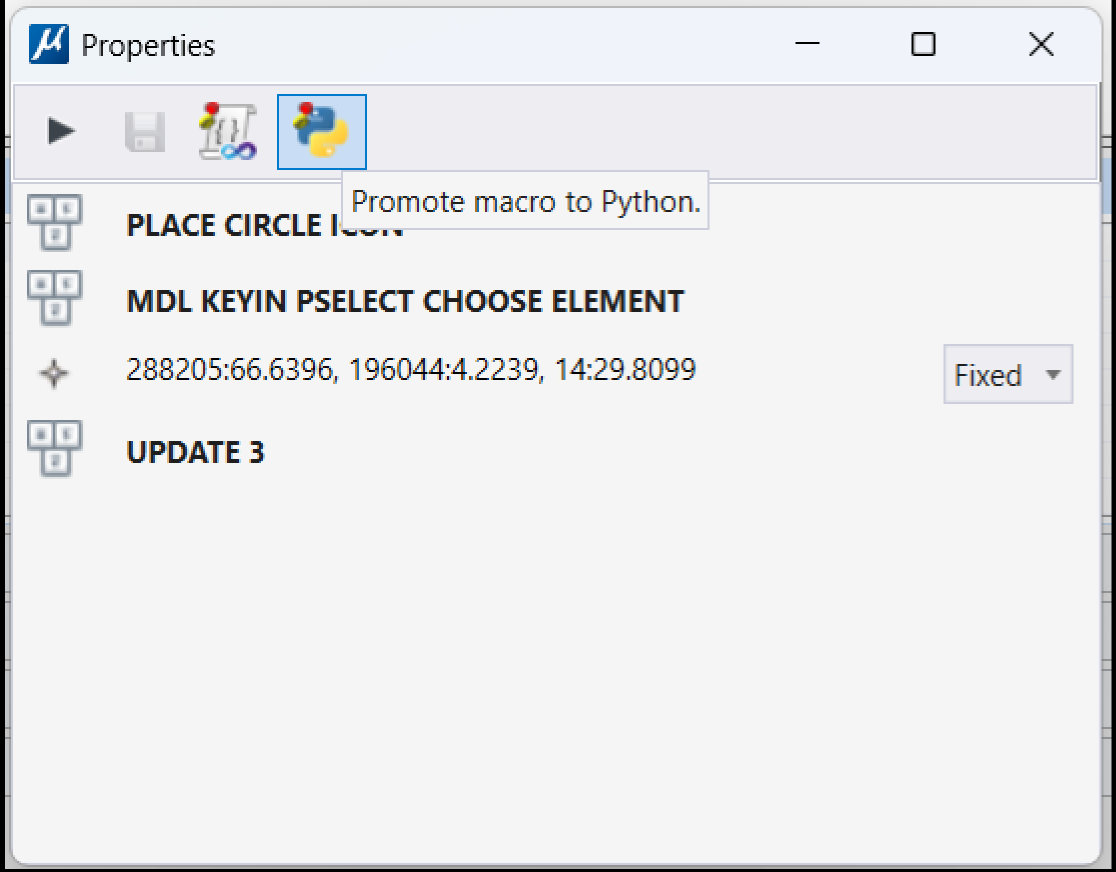
```
dsz:
  emv(a:
  print()
```

Ways to get started in MicroStation…

1. Python programming
2. Recording a macro – Promote to Python
3. Python Assistant

Ways to get started...

- Record a macro
- Promote macro to Python
- Review the code



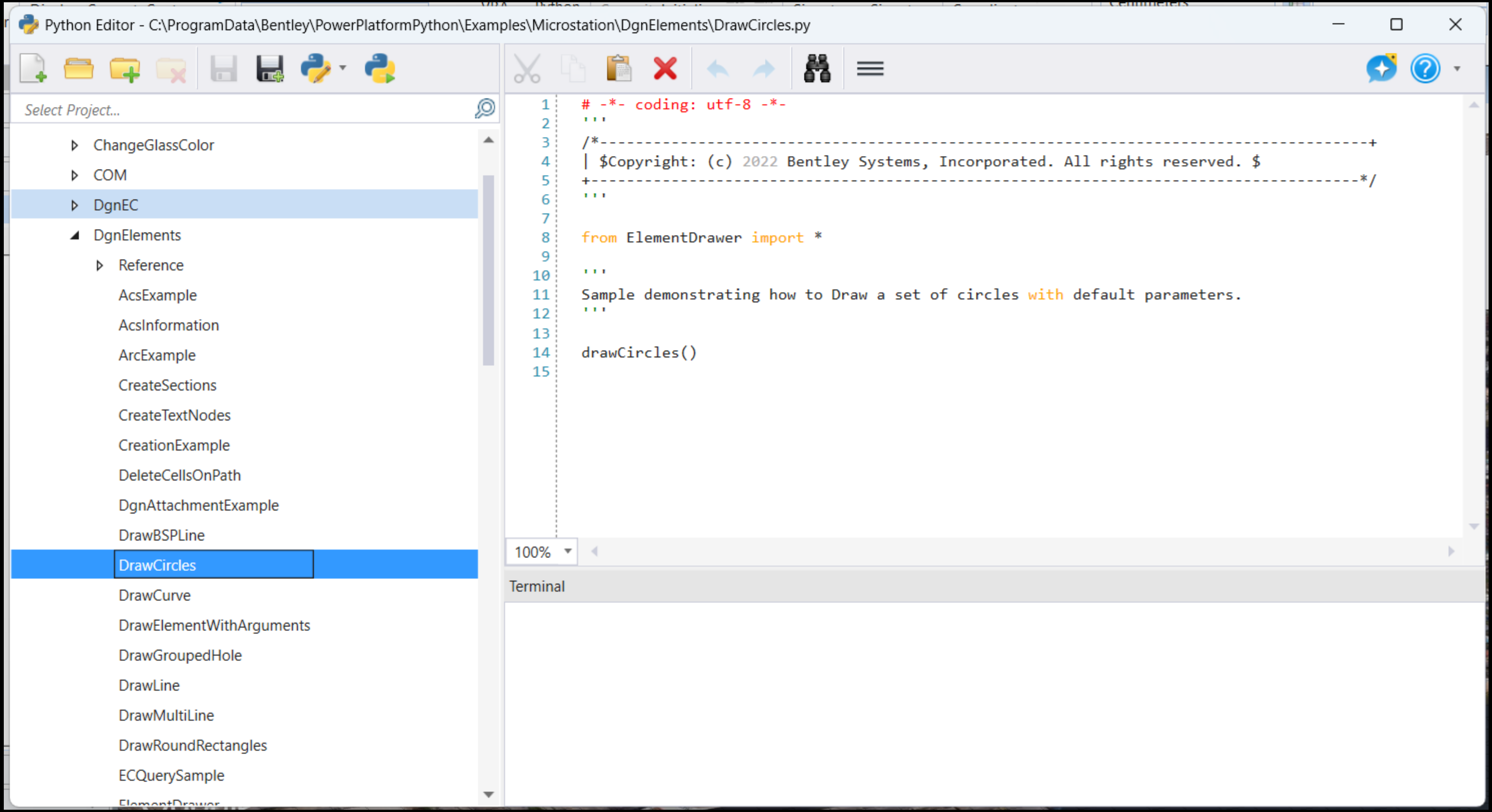
PLACE CIRCLE ICON

MDL KEYIN PSELECT CHOOSE ELEMENT

288205:66.6396, 196044:4.2239, 14:29.8099 Fixed

UPDATE 3

What if I’m not a programmer/developer?



Python Editor - C:\ProgramData\Bentley\PowerPlatformPython\Examples\Microstation\DgnElements\DrawCircles.py

```
# -*- coding: utf-8 -*-
'''
/*--------------------------------------------------------------------------------------+
| $Copyright: (c) 2022 Bentley Systems, Incorporated. All rights reserved. $
+--------------------------------------------------------------------------------------*/
'''

from ElementDrawer import *

'''
Sample demonstrating how to Draw a set of circles with default parameters.
'''

drawCircles()

```

Vibe Coding

"Vibe coding" is an approach to application development that involves heavily relying on a large language model (LLM) for generating code.

Coined by Andrej Karpathy in early 2025, vibe coding is about "coding by vibes"—you describe what you want in plain English (or another natural language), and the AI writes the code for you

⚙️ How It Works

Describe: You give a prompt like “Make a webpage that shows the weather for a city.”

Generate: The AI writes the code (HTML, JavaScript, etc.).

Test & Refine: You run the code, see what works, and ask the AI to fix or improve it.

Iterate: Repeat until satisfied.

✅ Benefits

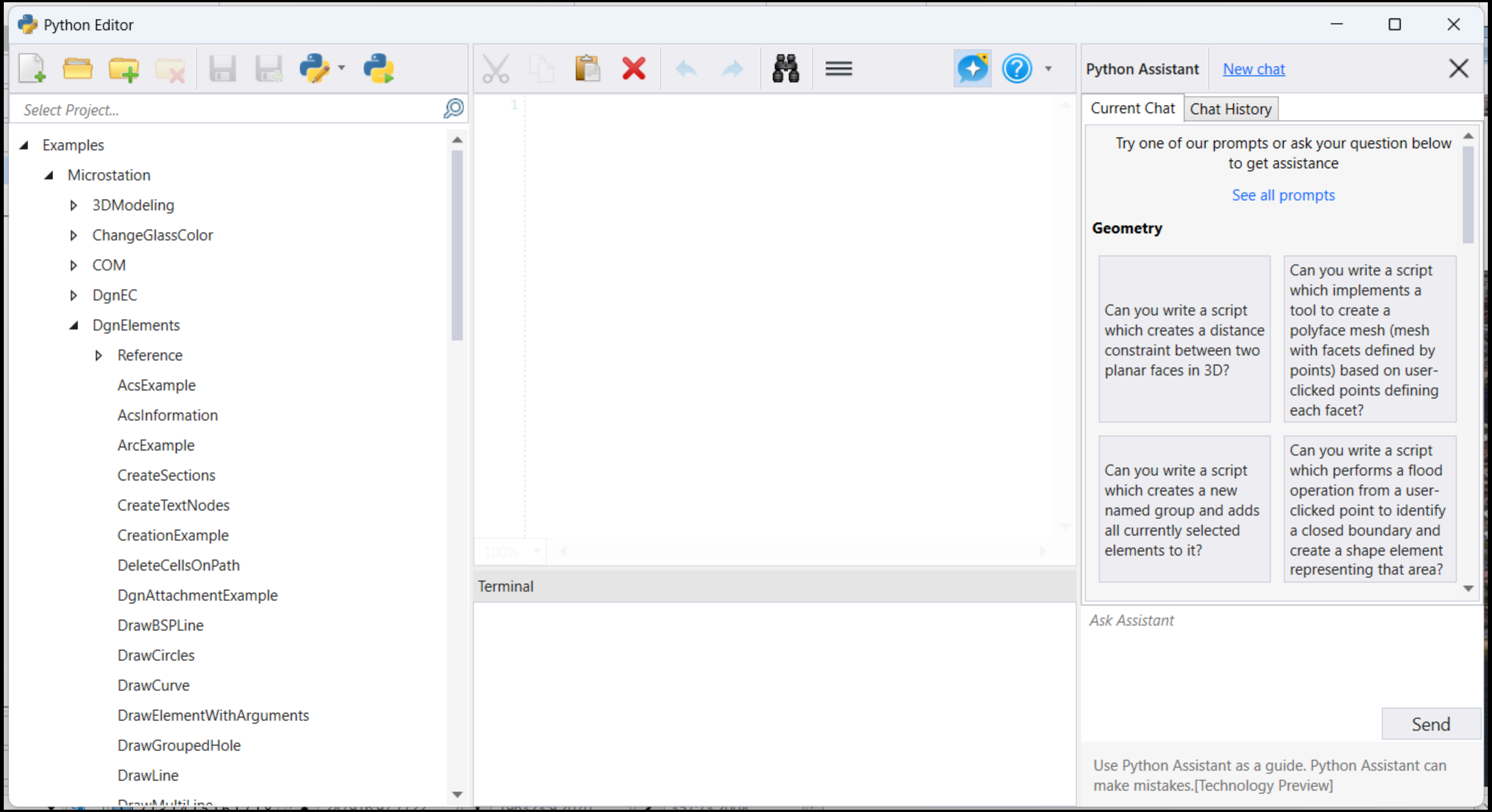
Accessibility: Great for non-programmers or beginners.

Speed: Rapid prototyping and iteration.

Creativity: Encourages artistic and expressive coding (e.g., music, visuals).

Fun: Makes coding feel more like play than work

Python Assistant – New in MicroStation 2025



Try one of our prompts or ask your question below to get assistance

Geometry

Can you write a script which creates a distance constraint between two planar faces in 3D?

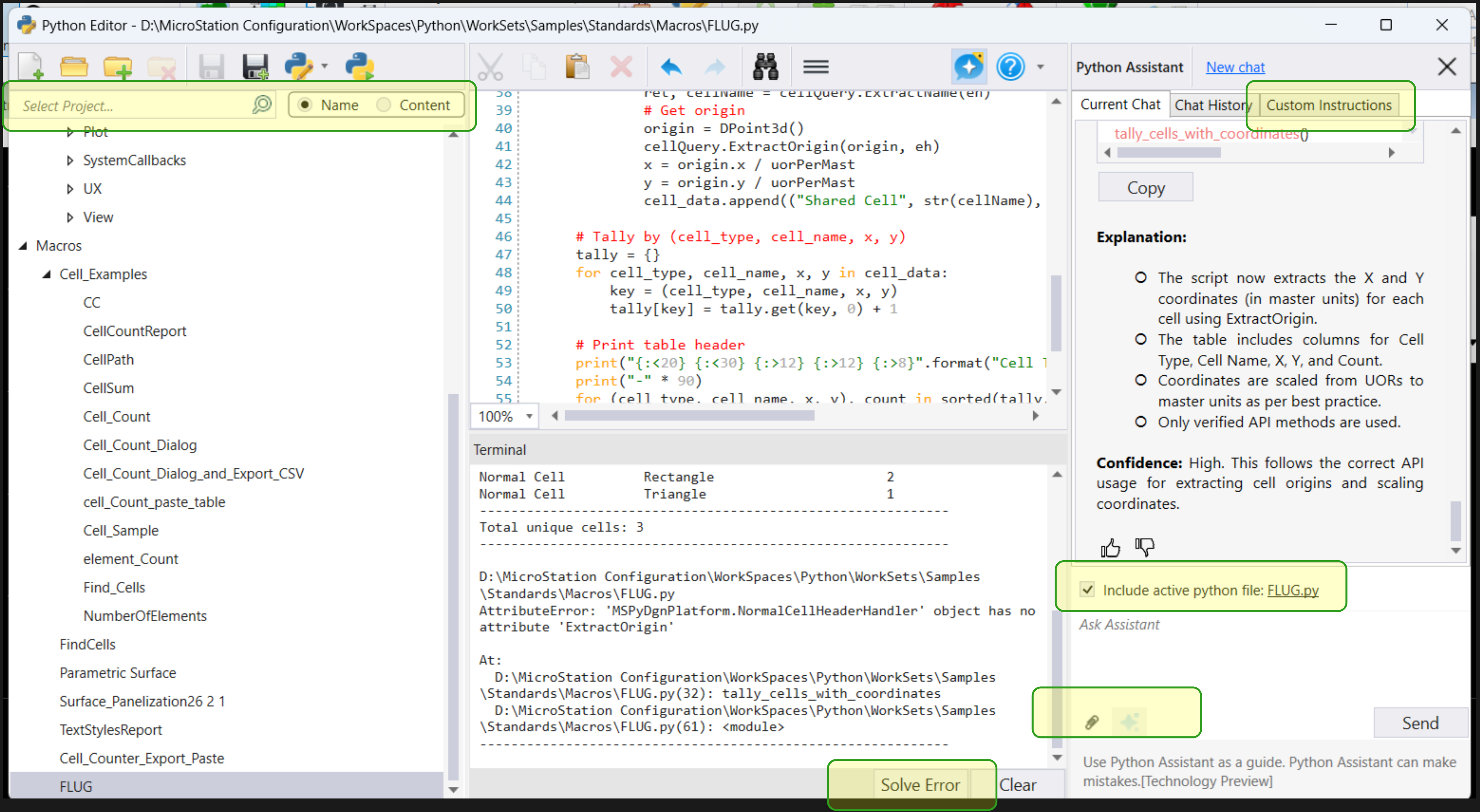
Can you write a script which implements a tool to create a polyface mesh (mesh with facets defined by points) based on user-clicked points defining each facet?

Can you write a script which creates a new named group and adds all currently selected elements to it?

Can you write a script which performs a flood operation from a user-clicked point to identify a closed boundary and create a shape element representing that area?

Use Python Assistant as a guide. Python Assistant can make mistakes.[Technology Preview]

MicroStation 2025.00.01.62



```
            ret, cellName = cellQuery.ExtractName(eh)
            # Get origin
            origin = DPoint3d()
            cellQuery.ExtractOrigin(origin, eh)
            x = origin.x / uorPerMast
            y = origin.y / uorPerMast
            cell_data.append(("Shared Cell", str(cellName),

    # Tally by (cell_type, cell_name, x, y)
    tally = {}
    for cell_type, cell_name, x, y in cell_data:
        key = (cell_type, cell_name, x, y)
        tally[key] = tally.get(key, 0) + 1

    # Print table header
    print("{:<20} {:<30} {:>12} {:>12} {:>8}".format("Cell T
    print("-" * 90)
    for (cell_type, cell_name, x, y), count in sorted(tally.
```

```
Normal Cell          Rectangle                    2
Normal Cell          Triangle                     1
------------------------------------------------------------
Total unique cells: 3
------------------------------------------------------------

D:\MicroStation Configuration\WorkSpaces\Python\WorkSets\Samples
\Standards\Macros\FLUG.py
AttributeError: 'MSPyDgnPlatform.NormalCellHeaderHandler' object has no
attribute 'ExtractOrigin'

At:
  D:\MicroStation Configuration\WorkSpaces\Python\WorkSets\Samples
\Standards\Macros\FLUG.py(32): tally_cells_with_coordinates
  D:\MicroStation Configuration\WorkSpaces\Python\WorkSets\Samples
\Standards\Macros\FLUG.py(61): <module>
------------------------------------------------------------
```

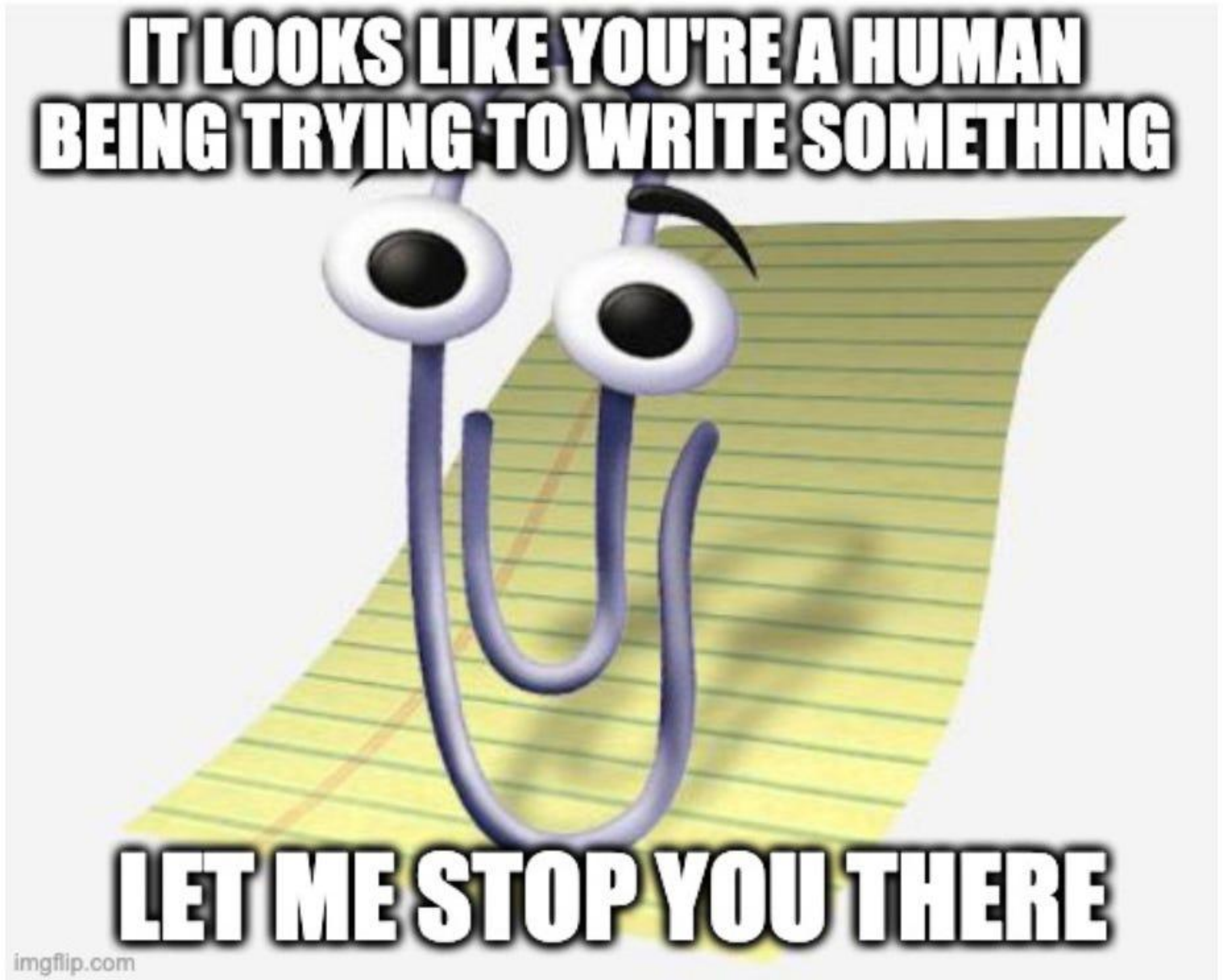
```
tally_cells_with_coordinates()
```

Explanation:

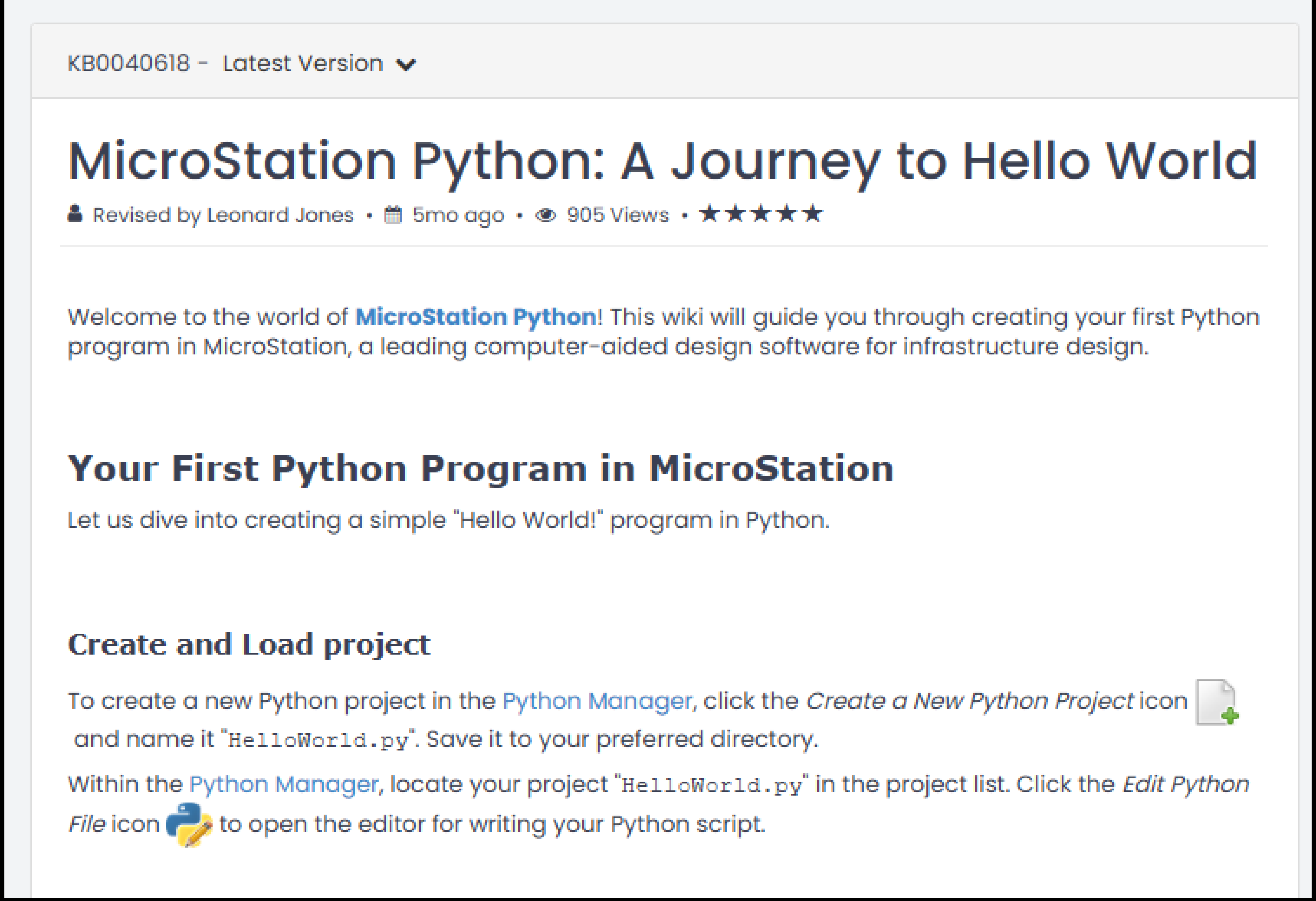
- The script now extracts the X and Y coordinates (in master units) for each cell using ExtractOrigin.
- The table includes columns for Cell Type, Cell Name, X, Y, and Count.
- Coordinates are scaled from UORs to master units as per best practice.
- Only verified API methods are used.

Confidence: High. This follows the correct API usage for extracting cell origins and scaling coordinates.

Use Python Assistant as a guide. Python Assistant can make mistakes.[Technology Preview]


IT LOOKS LIKE YOU'RE A HUMAN BEING TRYING TO WRITE SOMETHING
LET ME STOP YOU THERE
imgflip.com

Resources



KB0040618 - Latest Version ⌄

MicroStation Python: A Journey to Hello World

Revised by Leonard Jones • 5mo ago • 905 Views • ★★★★★

Welcome to the world of **MicroStation Python**! This wiki will guide you through creating your first Python program in MicroStation, a leading computer-aided design software for infrastructure design.

Your First Python Program in MicroStation

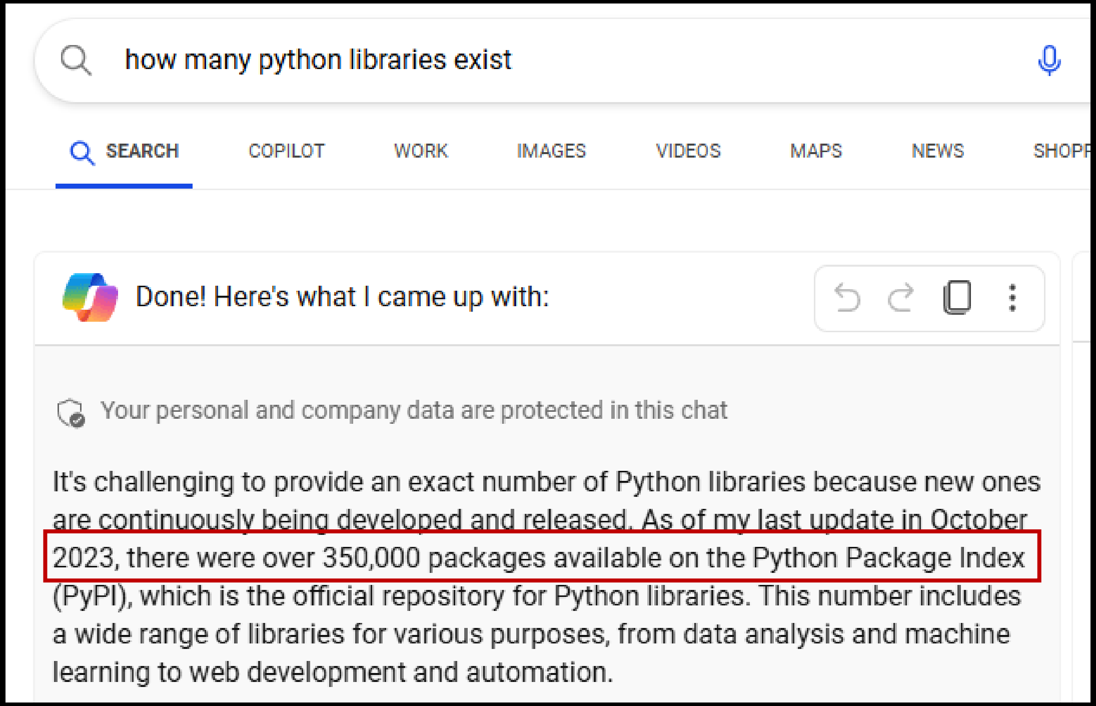
Let us dive into creating a simple "Hello World!" program in Python.

Create and Load project

To create a new Python project in the Python Manager, click the *Create a New Python Project* icon and name it "`HelloWorld.py`". Save it to your preferred directory.

Within the Python Manager, locate your project "`HelloWorld.py`" in the project list. Click the *Edit Python File* icon to open the editor for writing your Python script.

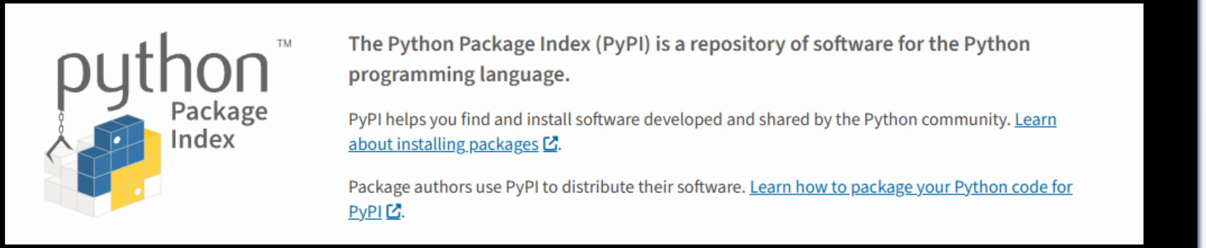
Python Libraries



Done! Here's what I came up with:

Your personal and company data are protected in this chat

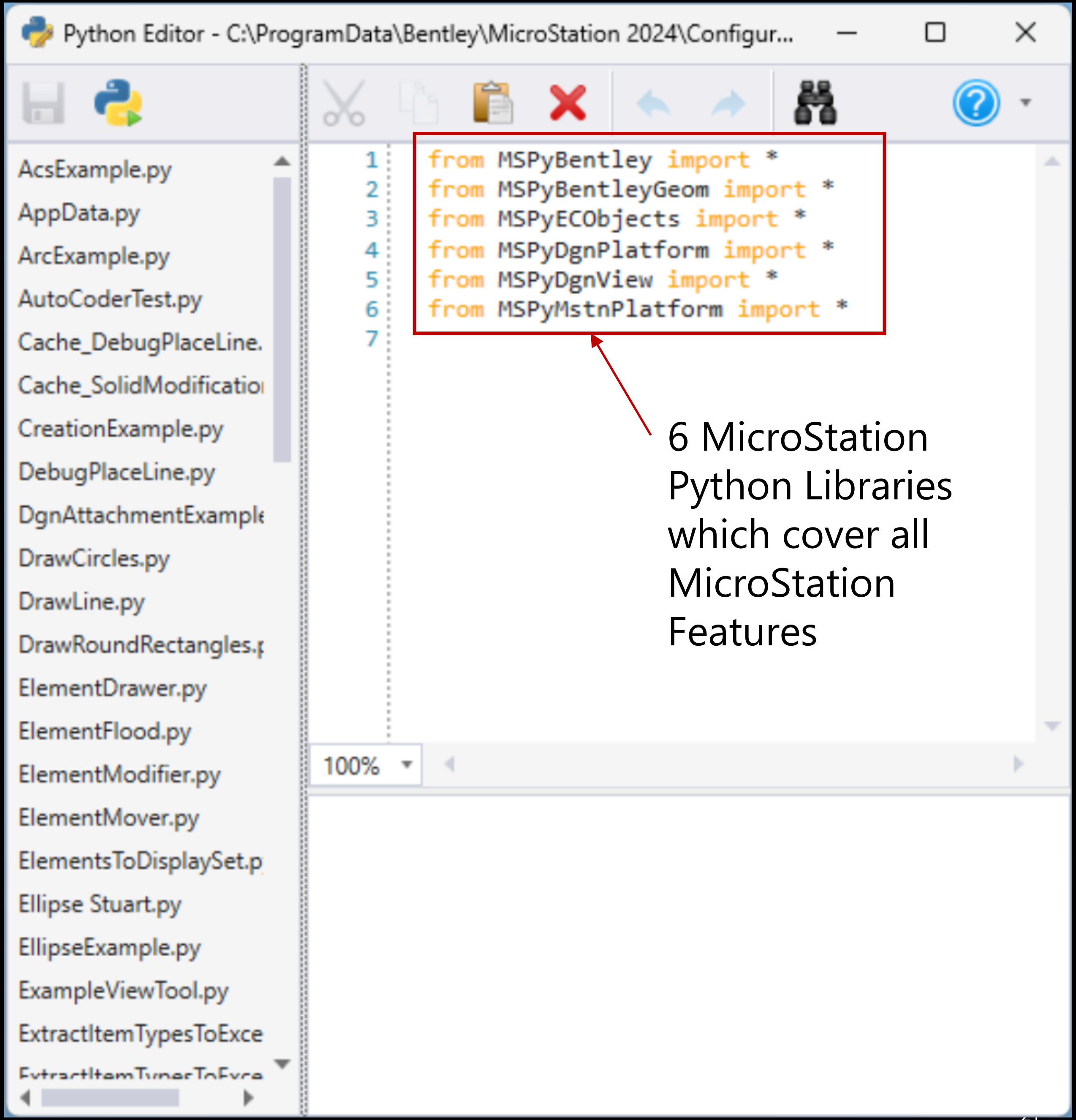
It's challenging to provide an exact number of Python libraries because new ones are continuously being developed and released. As of my last update in October 2023, there were over 350,000 packages available on the Python Package Index (PyPI), which is the official repository for Python libraries. This number includes a wide range of libraries for various purposes, from data analysis and machine learning to web development and automation.



The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.

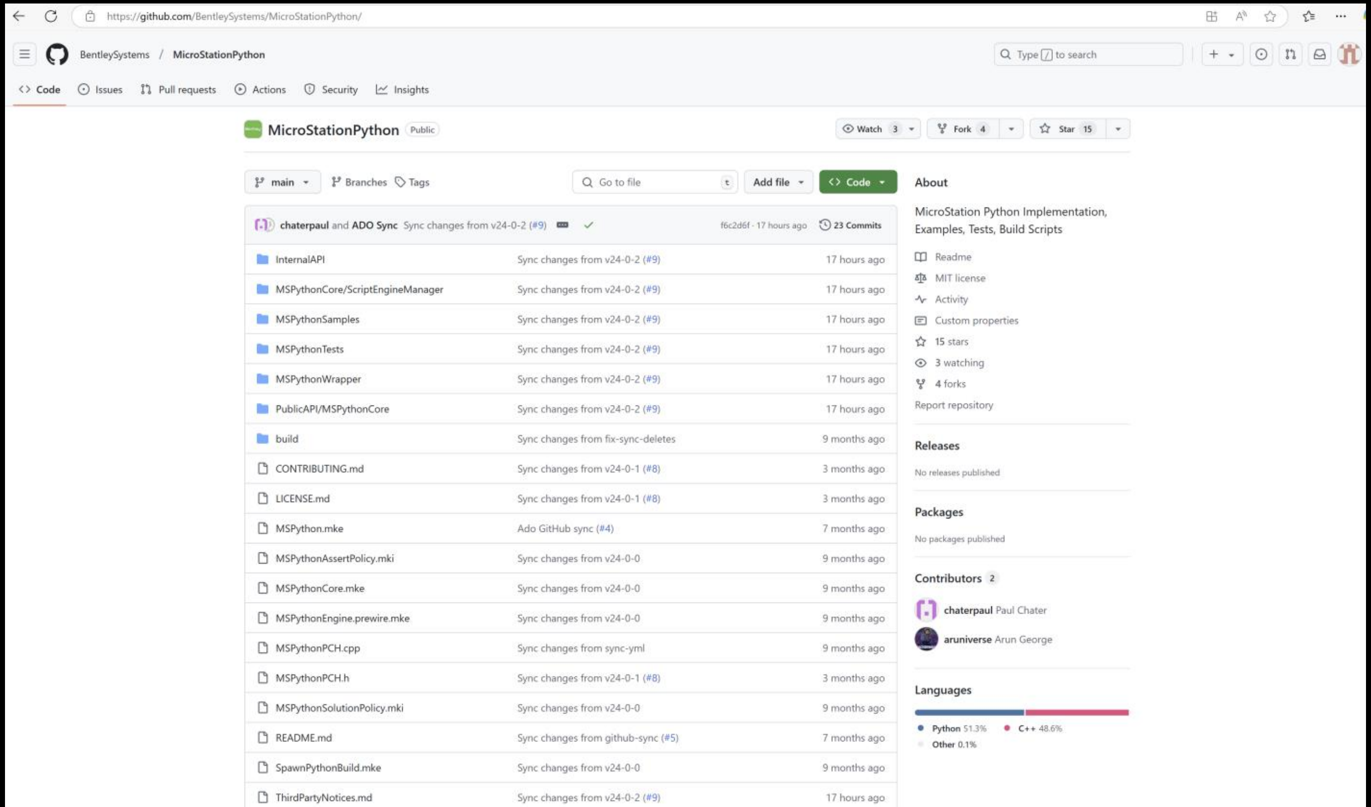


```python
from MSPyBentley import *
from MSPyBentleyGeom import *
from MSPyECObjects import *
from MSPyDgnPlatform import *
from MSPyDgnView import *
from MSPyMstnPlatform import *

```

6 MicroStation Python Libraries which cover all MicroStation Features

MicroStation Python GitHub Repository

- MicroStation Python is OpenSourced
- Create your own fork
- Create your own applications
- Contribute to the OpenSource development
- Stay up to date with latest Samples
- Download repository locally



chaterpaul and ADO Sync Sync changes from v24-0-2 (#9) f6c2d6f · 17 hours ago 23 Commits

InternalAPI	Sync changes from v24-0-2 (#9)	17 hours ago
MSPythonCore/ScriptEngineManager	Sync changes from v24-0-2 (#9)	17 hours ago
MSPythonSamples	Sync changes from v24-0-2 (#9)	17 hours ago
MSPythonTests	Sync changes from v24-0-2 (#9)	17 hours ago
MSPythonWrapper	Sync changes from v24-0-2 (#9)	17 hours ago
PublicAPI/MSPythonCore	Sync changes from v24-0-2 (#9)	17 hours ago
build	Sync changes from fix-sync-deletes	9 months ago
CONTRIBUTING.md	Sync changes from v24-0-1 (#8)	3 months ago
LICENSE.md	Sync changes from v24-0-1 (#8)	3 months ago
MSPython.mke	Ado GitHub sync (#4)	7 months ago
MSPythonAssertPolicy.mki	Sync changes from v24-0-0	9 months ago
MSPythonCore.mke	Sync changes from v24-0-0	9 months ago
MSPythonEngine.prewire.mke	Sync changes from v24-0-0	9 months ago
MSPythonPCH.cpp	Sync changes from sync-yml	9 months ago
MSPythonPCH.h	Sync changes from v24-0-1 (#8)	3 months ago
MSPythonSolutionPolicy.mki	Sync changes from v24-0-0	9 months ago
README.md	Sync changes from github-sync (#5)	7 months ago
SpawnPythonBuild.mke	Sync changes from v24-0-0	9 months ago
ThirdPartyNotices.md	Sync changes from v24-0-2 (#9)	17 hours ago

About

MicroStation Python Implementation, Examples, Tests, Build Scripts

Readme
MIT license
Activity
Custom properties
15 stars
3 watching
4 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 2

chaterpaul Paul Chater
aruniverse Arun George

Languages

● Python 51.3% ● C++ 48.6%
● Other 0.1%

Python Samples

- Python Samples
  - Solid Modifications
  - Library – Excel to Item Types
  - Display Sets
  - Arcs, Circles, Lines, Rectangles,Ellipse, LineStrings
  - Named Groups
  - Select By
  - Flood
  - Modifier and Mover
  - Item Types
  - Group Hole
  - Library - Clock
- 24.0 – 59 Python Samples
  - Referencing
  - Arguments
  - Raster
  - EC Data
  - Save to Image
  - Callbacks
  - PDF Printing

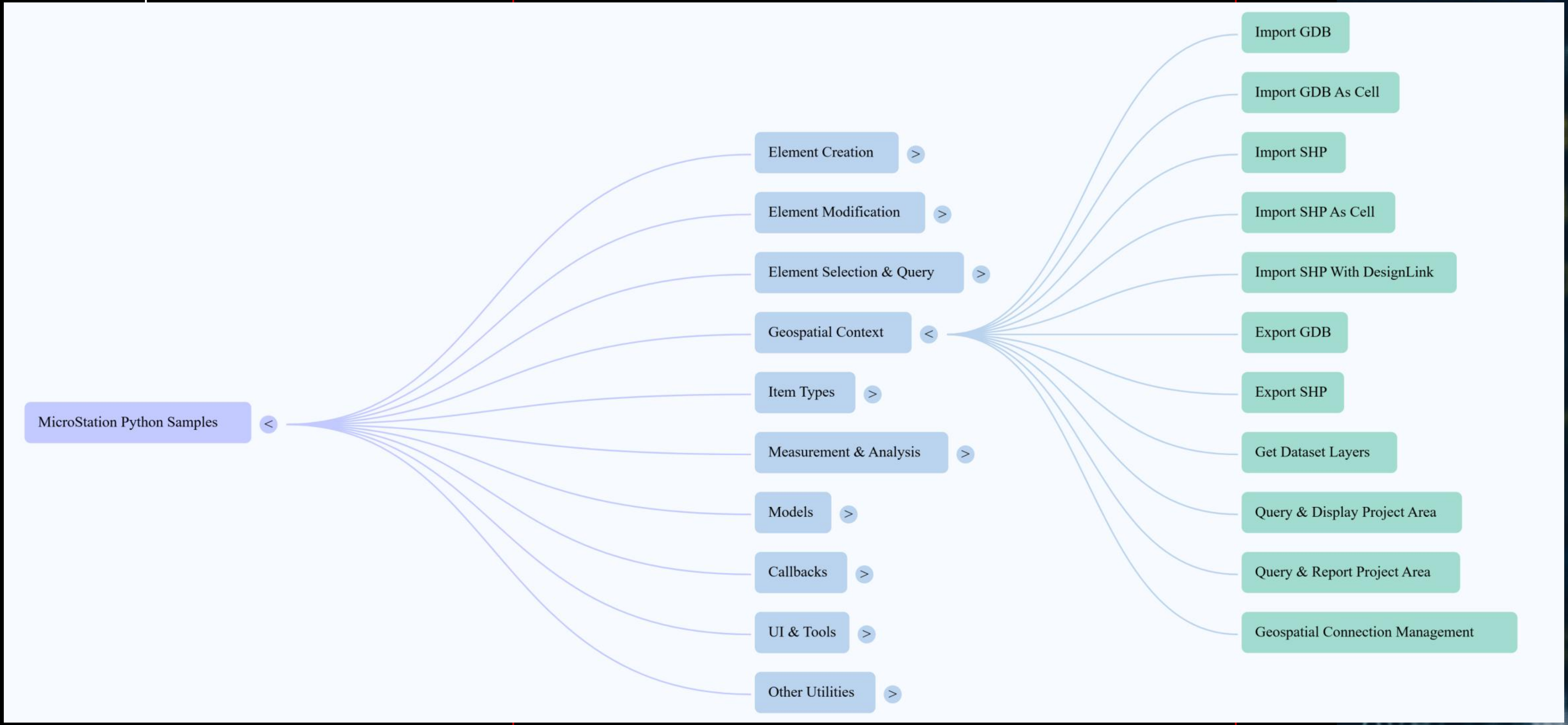
Added

- 24.1 – 69 Python Samples
  - Color Library
  - Create Sections
  - Constraints
  - Cells
  - Measure Angle and Distance
  - Text Nodes
  - Modify Model
  - Parametric Modelling
- 24.2 – 90 Python Samples
  - Bspline and Curve
  - Multi Line
  - Mesh
  - Polyface
  - Element Copy, Rotate and Scale
  - EC Tools
  - Geospatial
  - Viewing
- Currently – 108 Python Samples

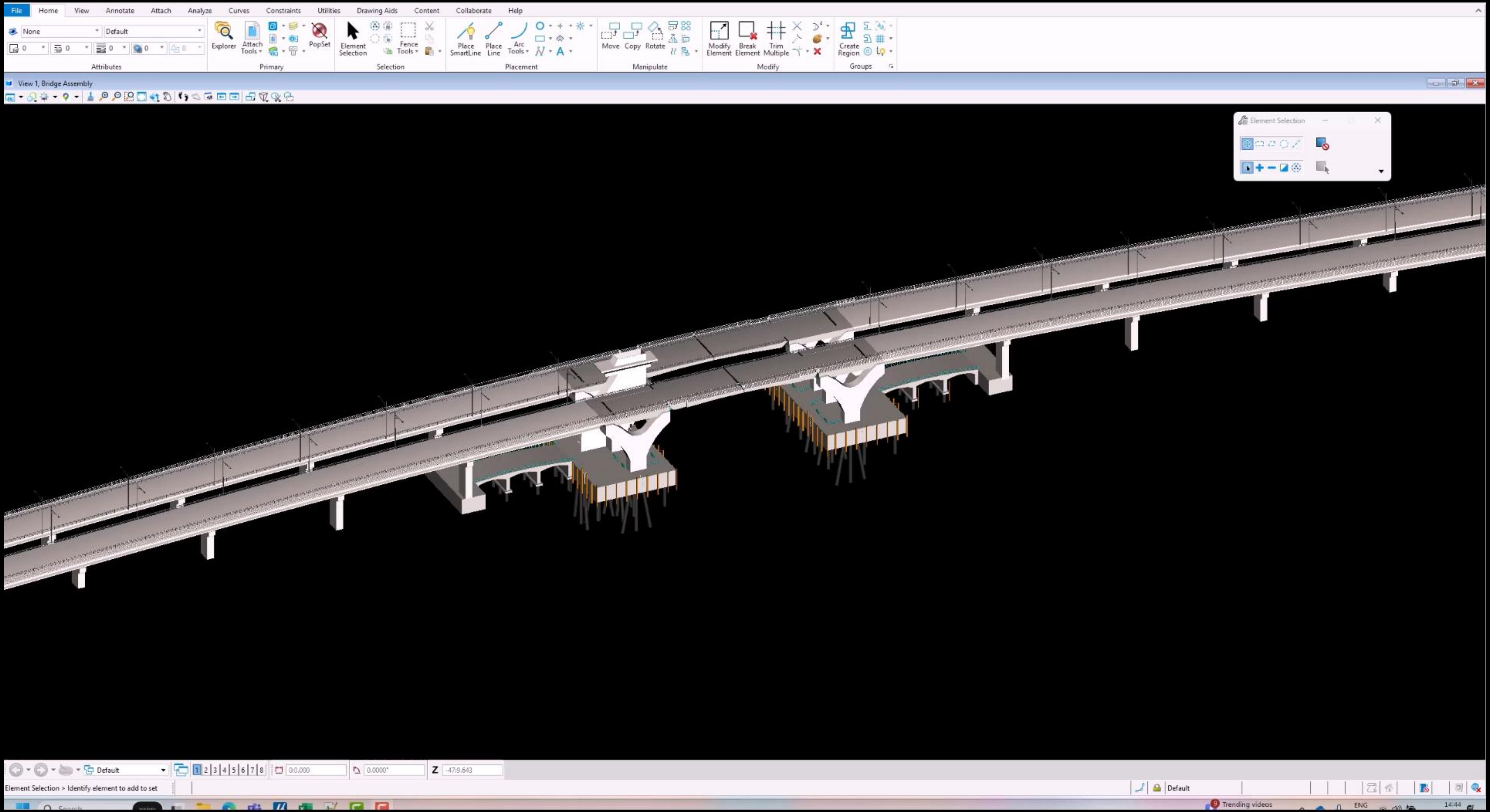
Added

Python Samples

- 25.0 - Python Samples (153)
  - Geospatial



Geospatial example with GCS change



Element Selection > Identify element to add to set

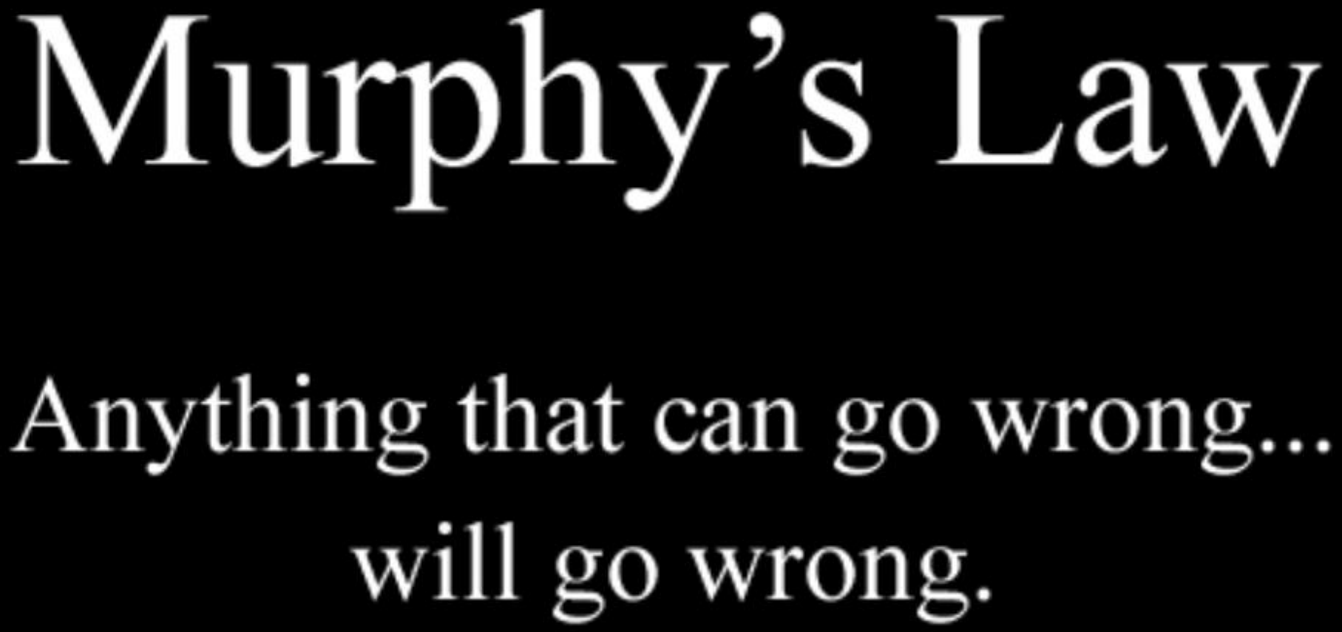
- Tips and tricks
- How to prompt Co-Pilot and ask it questions
- If it fails, iterate the issue into co-pilot to fix it

Be Aware...

- Consistency
  - Asking the Same Question twice, may result in different answers

Live Demo...

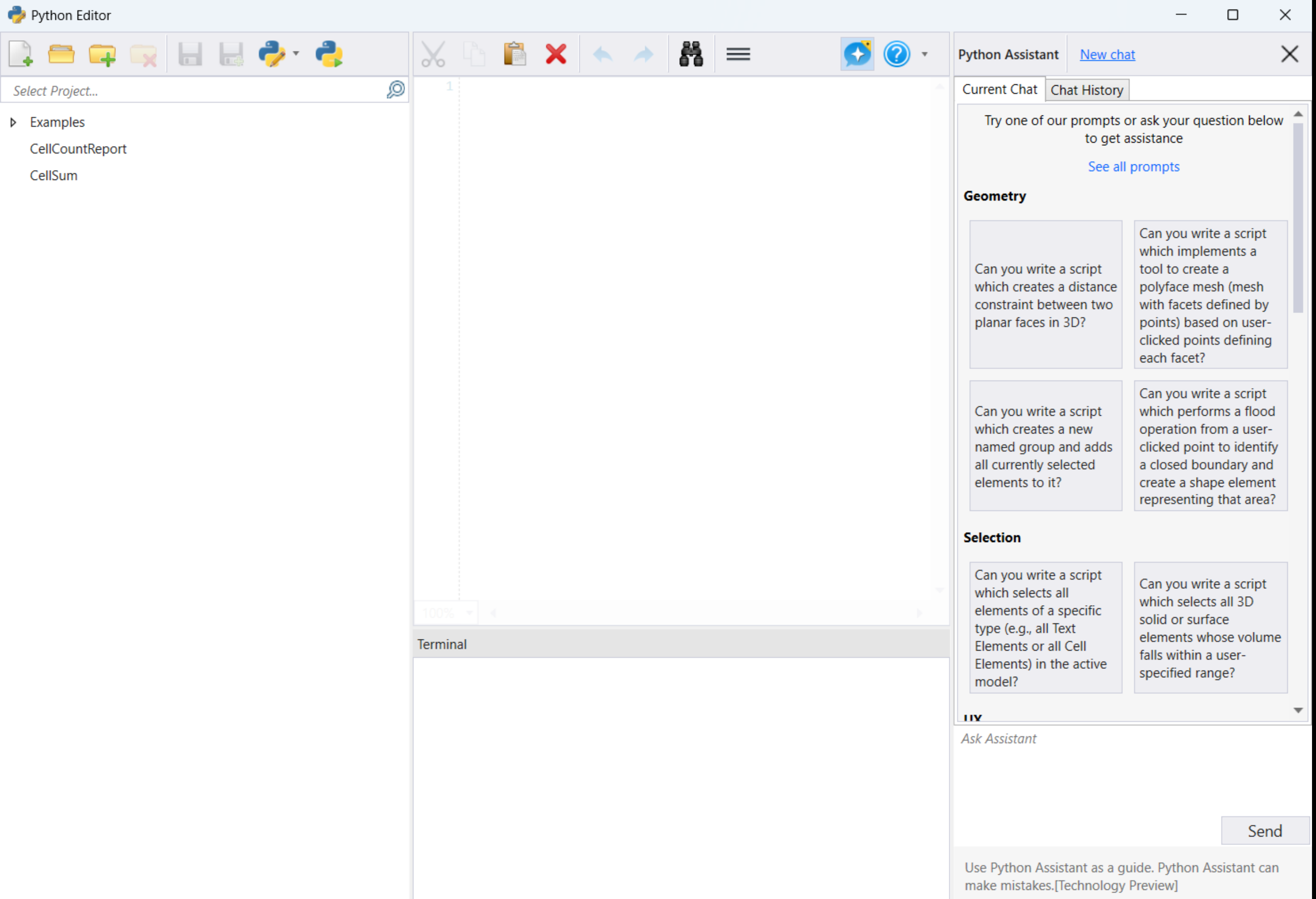
Tackle one of the common items that users have requested over the years...



Murphy's Law

Anything that can go wrong...
will go wrong.

Prompting Tips



Try one of our prompts or ask your question below to get assistance

See all prompts

Geometry

Can you write a script which creates a distance constraint between two planar faces in 3D?

Can you write a script which implements a tool to create a polyface mesh (mesh with facets defined by points) based on user-clicked points defining each facet?

Can you write a script which creates a new named group and adds all currently selected elements to it?

Can you write a script which performs a flood operation from a user-clicked point to identify a closed boundary and create a shape element representing that area?

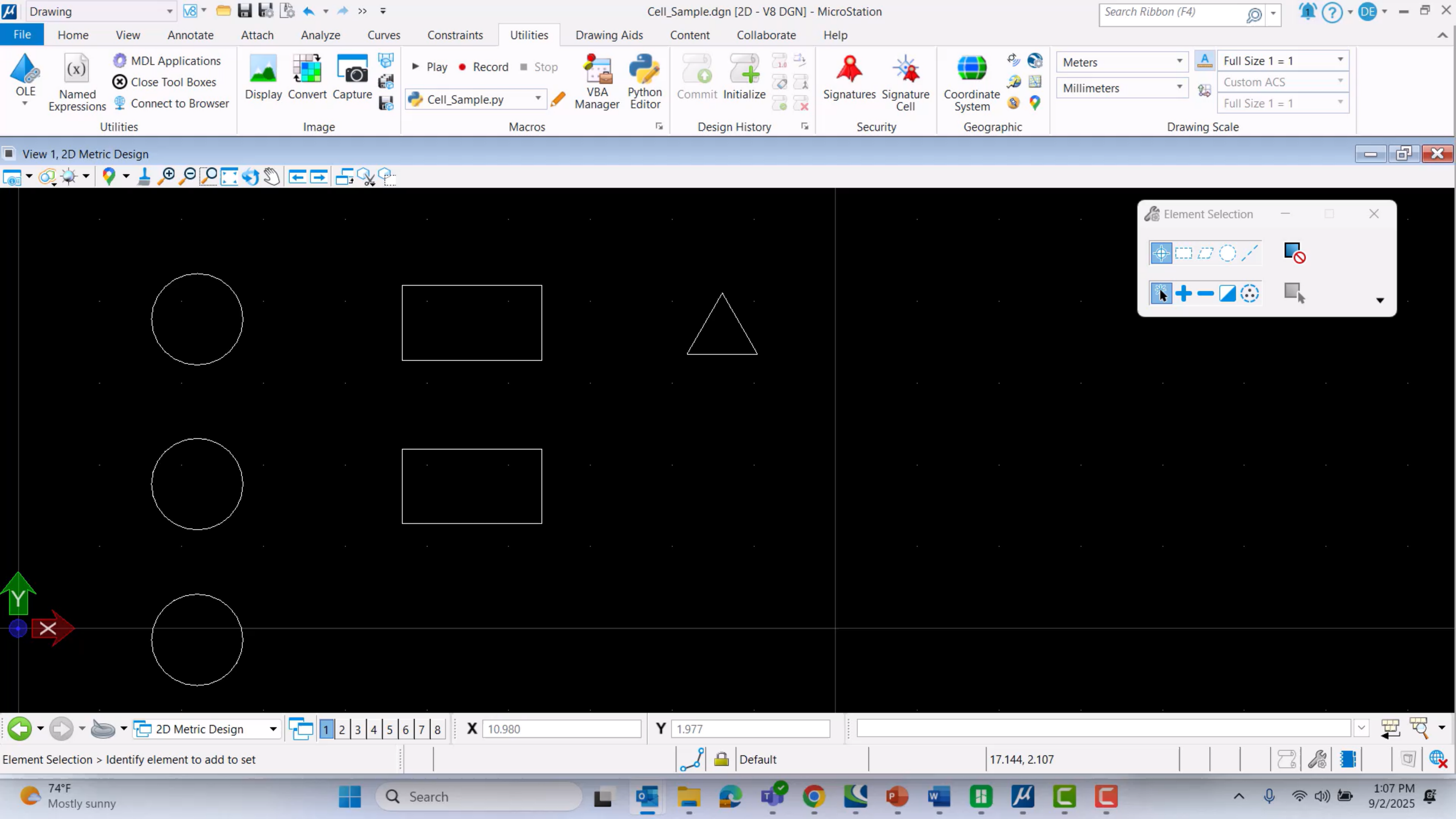
Selection

Can you write a script which selects all elements of a specific type (e.g., all Text Elements or all Cell Elements) in the active model?

Can you write a script which selects all 3D solid or surface elements whose volume falls within a user-specified range?

UX

Use Python Assistant as a guide. Python Assistant can make mistakes.[Technology Preview]

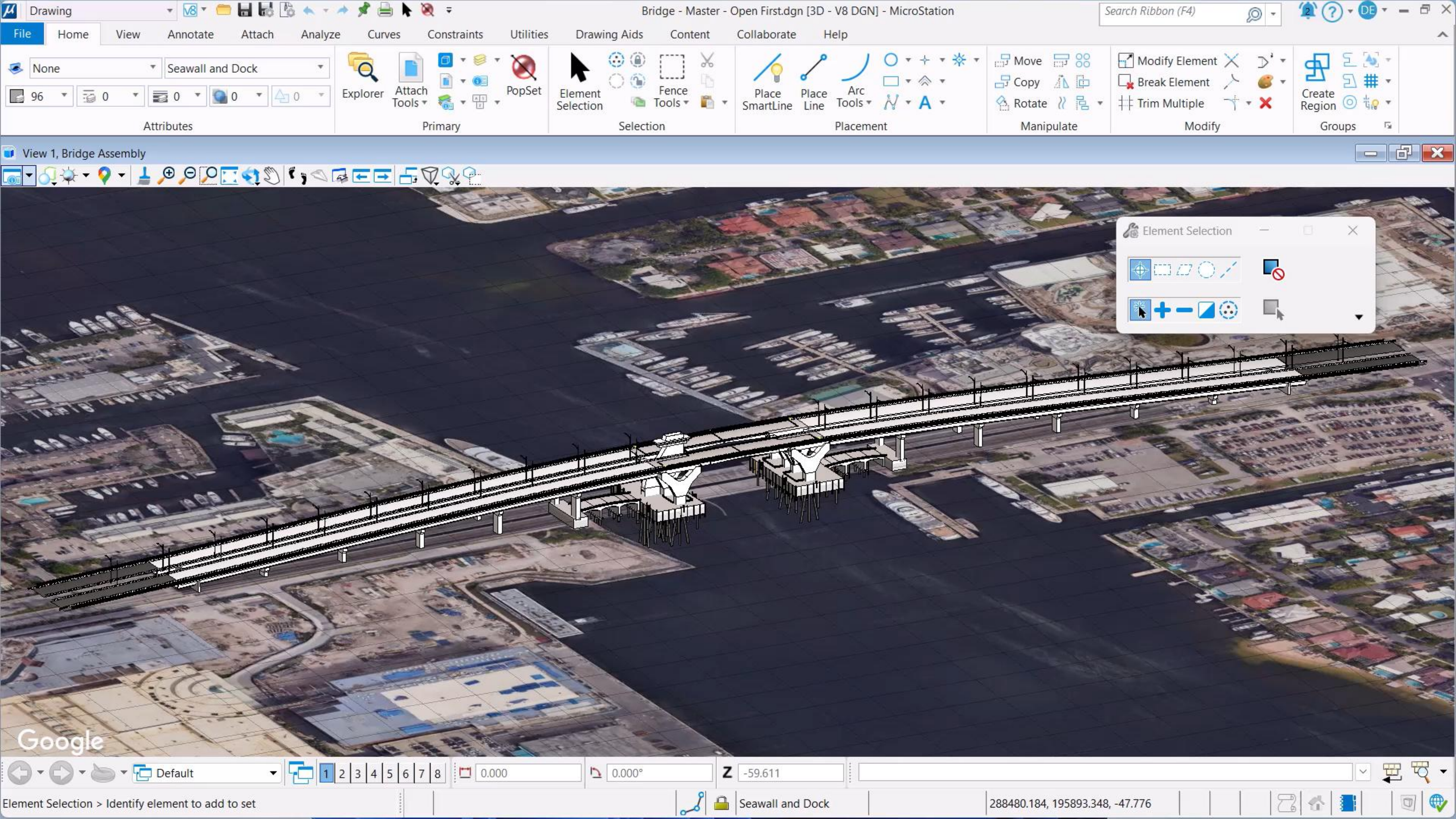


X 10.980

Y 1.977

Element Selection > Identify element to add to set

17.144, 2.107



Bridge - Master - Open First.dgn [3D - V8 DGN] - MicroStation

View 1, Bridge Assembly

None

Seawall and Dock

96

0

0

0

0

Default

0.000

0.000°

Z -59.611

Element Selection > Identify element to add to set

Seawall and Dock

288480.184, 195893.348, -47.776

Setting up some Rules May help:

1. *Always search and use the provided sample data as your primary source for code patterns, logic, and helper functions. Only use the official API as a secondary reference if the sample data does not cover the requirement.*

2. *When generating code, produce a complete, self-contained script. All functions referenced in your script (including any helper or utility functions) must be defined within the script itself, so that it runs without modification or missing dependencies.*

3. *If multiple relevant sample functions exist, combine them as needed to produce a working, ready-to-execute script.*

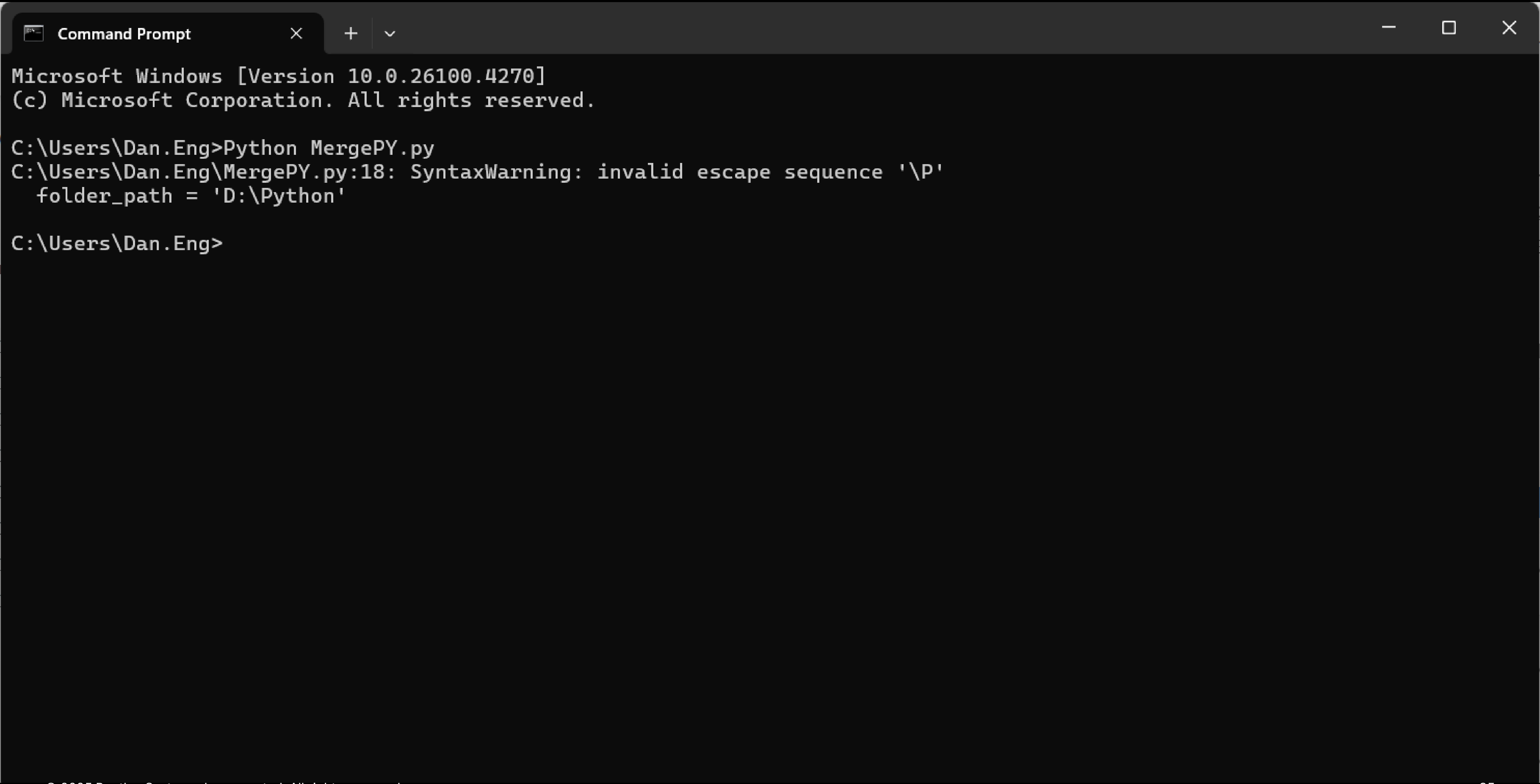
4. *Do not invent or fabricate any attributes, methods, or classes that are not present in the sample scripts or the official API documentation.*

5. *Ask Clarifying questions if you don't understand something or find something ambiguous.*

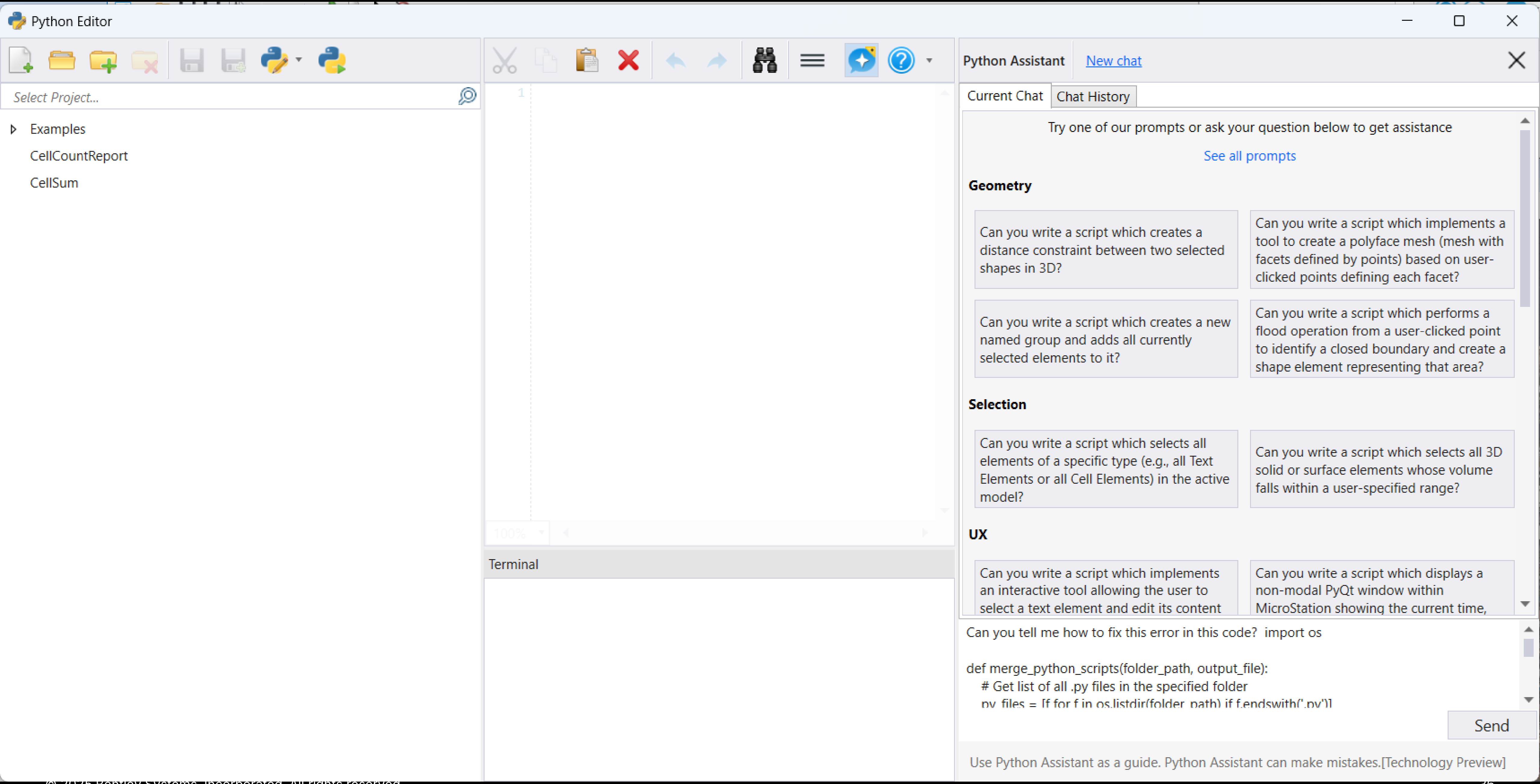
6. *Please document the code with annotations, comments and docstrings for future reference and code comprehension. These additions should be written to assist a novice programmer understand the code*

Errors

- Ask the Python Assistant for help on the error
- Tell the Python Assistant exactly what you want
- If something didn’t work, let the Python Assistant know



```
Microsoft Windows [Version 10.0.26100.4270]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dan.Eng>Python MergePY.py
C:\Users\Dan.Eng\MergePY.py:18: SyntaxWarning: invalid escape sequence '\P'
  folder_path = 'D:\Python'

C:\Users\Dan.Eng>
```



Try one of our prompts or ask your question below to get assistance

See all prompts

Geometry

Can you write a script which creates a distance constraint between two selected shapes in 3D?

Can you write a script which implements a tool to create a polyface mesh (mesh with facets defined by points) based on user-clicked points defining each facet?

Can you write a script which creates a new named group and adds all currently selected elements to it?

Can you write a script which performs a flood operation from a user-clicked point to identify a closed boundary and create a shape element representing that area?

Selection

Can you write a script which selects all elements of a specific type (e.g., all Text Elements or all Cell Elements) in the active model?

Can you write a script which selects all 3D solid or surface elements whose volume falls within a user-specified range?

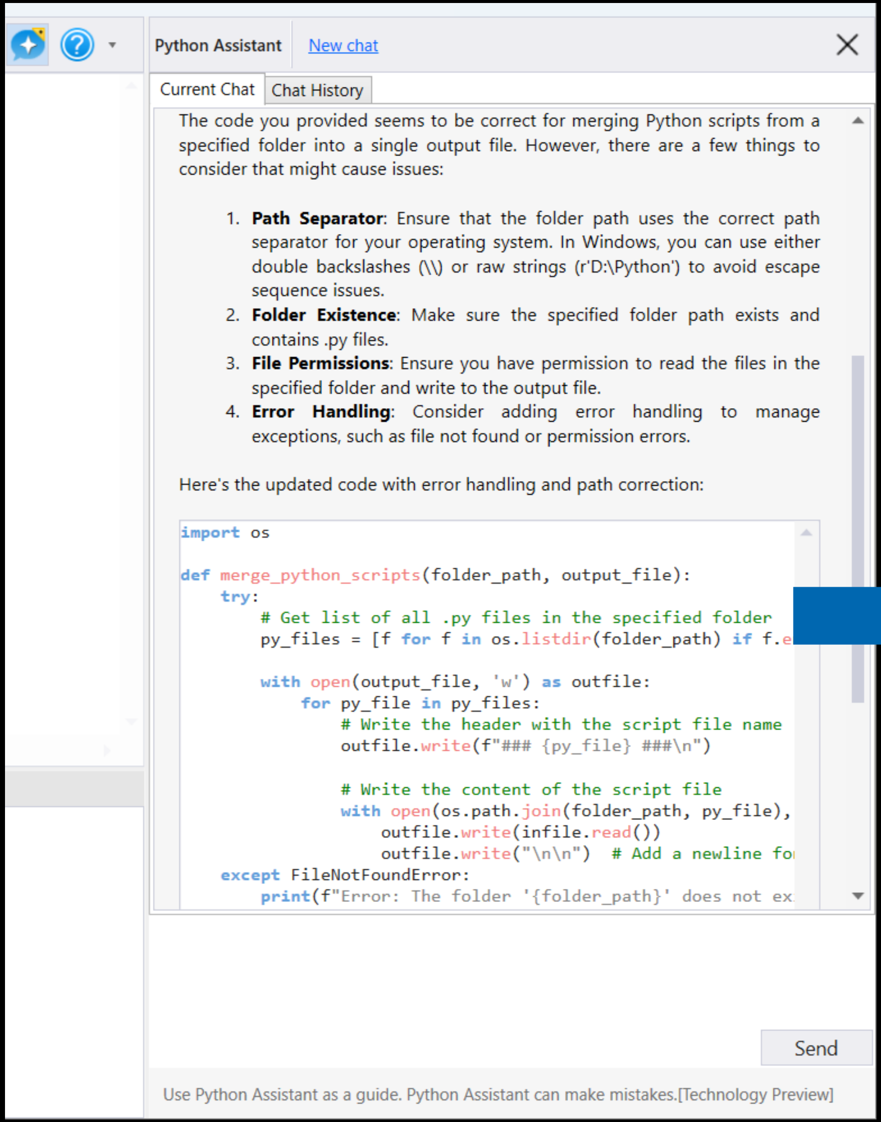
UX

Can you write a script which implements an interactive tool allowing the user to select a text element and edit its content

Can you write a script which displays a non-modal PyQt window within MicroStation showing the current time,

```
Can you tell me how to fix this error in this code?  import os

def merge_python_scripts(folder_path, output_file):
    # Get list of all .py files in the specified folder
    py_files = [f for f in os.listdir(folder_path) if f.endswith('.py')]
```

Use Python Assistant as a guide. Python Assistant can make mistakes.[Technology Preview]



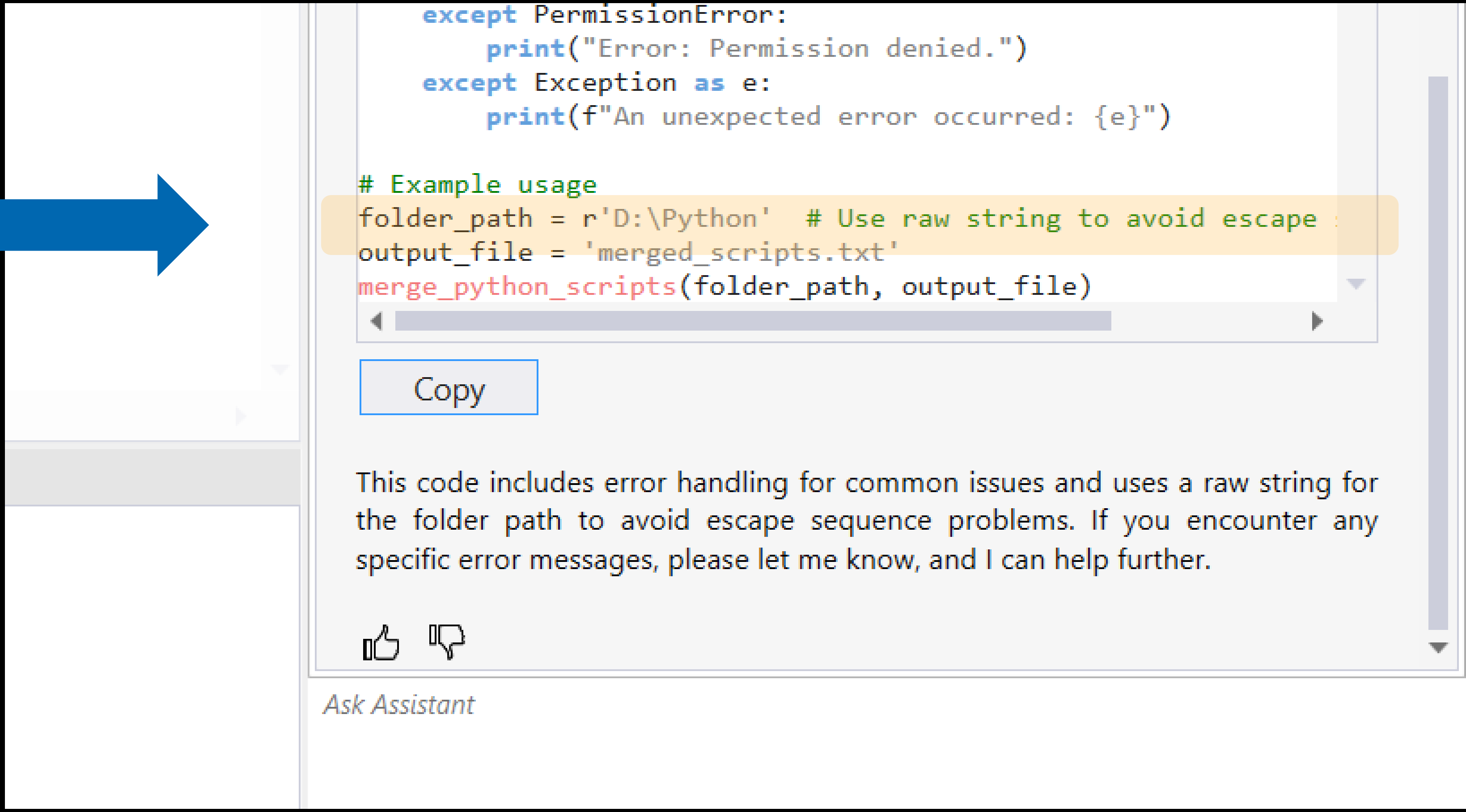
The code you provided seems to be correct for merging Python scripts from a specified folder into a single output file. However, there are a few things to consider that might cause issues:

1. **Path Separator**: Ensure that the folder path uses the correct path separator for your operating system. In Windows, you can use either double backslashes (\\) or raw strings (r'D:\Python') to avoid escape sequence issues.
2. **Folder Existence**: Make sure the specified folder path exists and contains .py files.
3. **File Permissions**: Ensure you have permission to read the files in the specified folder and write to the output file.
4. **Error Handling**: Consider adding error handling to manage exceptions, such as file not found or permission errors.

Here's the updated code with error handling and path correction:

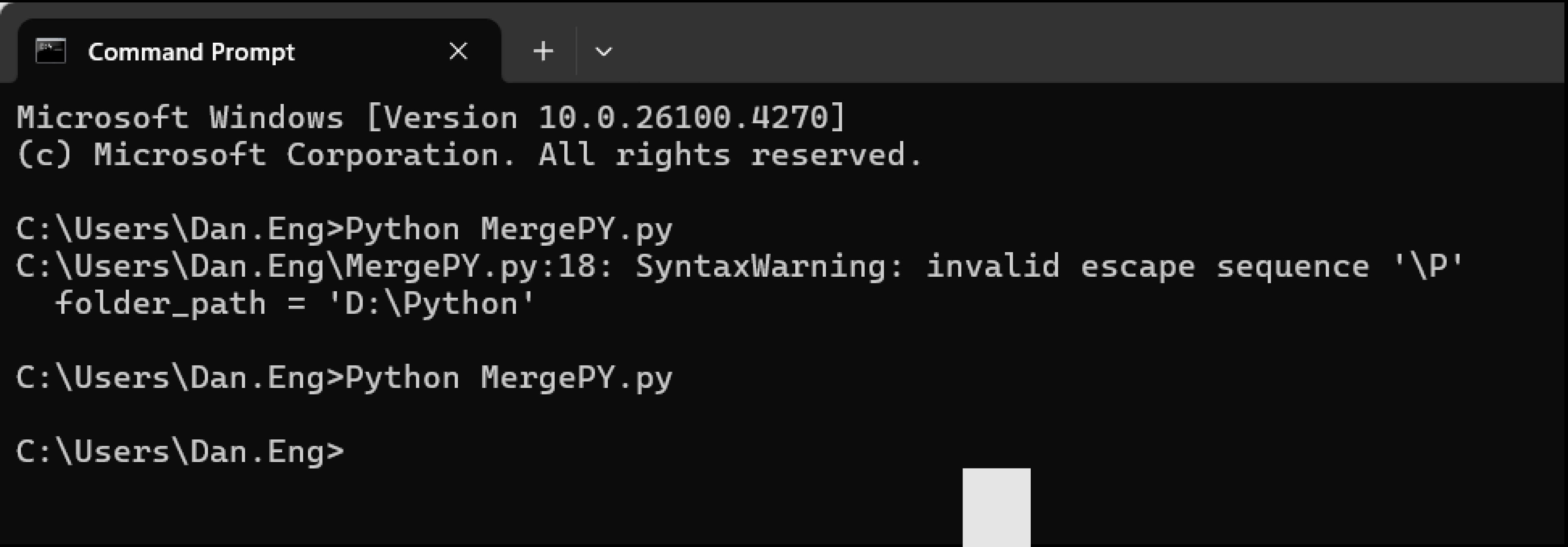
```
import os

def merge_python_scripts(folder_path, output_file):
    try:
        # Get list of all .py files in the specified folder
        py_files = [f for f in os.listdir(folder_path) if f.e

        with open(output_file, 'w') as outfile:
            for py_file in py_files:
                # Write the header with the script file name
                outfile.write(f"### {py_file} ###\n")

                # Write the content of the script file
                with open(os.path.join(folder_path, py_file),
                    outfile.write(infile.read())
                    outfile.write("\n\n")  # Add a newline fo
    except FileNotFoundError:
        print(f"Error: The folder '{folder_path}' does not ex
```

Use Python Assistant as a guide. Python Assistant can make mistakes.[Technology Preview]

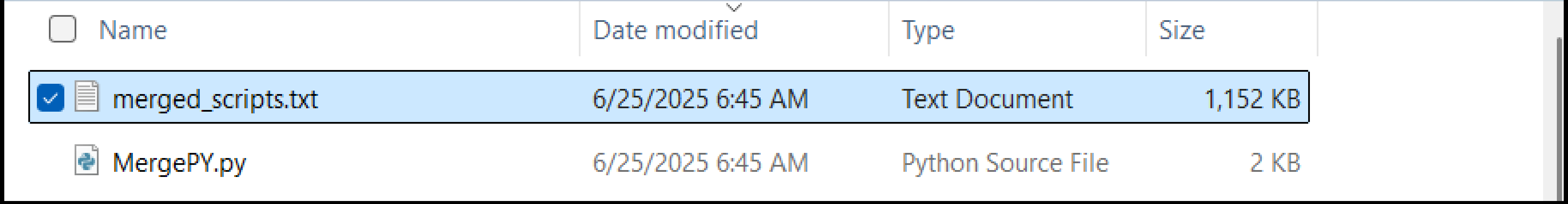


```
    except PermissionError:
        print("Error: Permission denied.")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")

# Example usage
folder_path = r'D:\Python'  # Use raw string to avoid escape
output_file = 'merged_scripts.txt'
merge_python_scripts(folder_path, output_file)
```

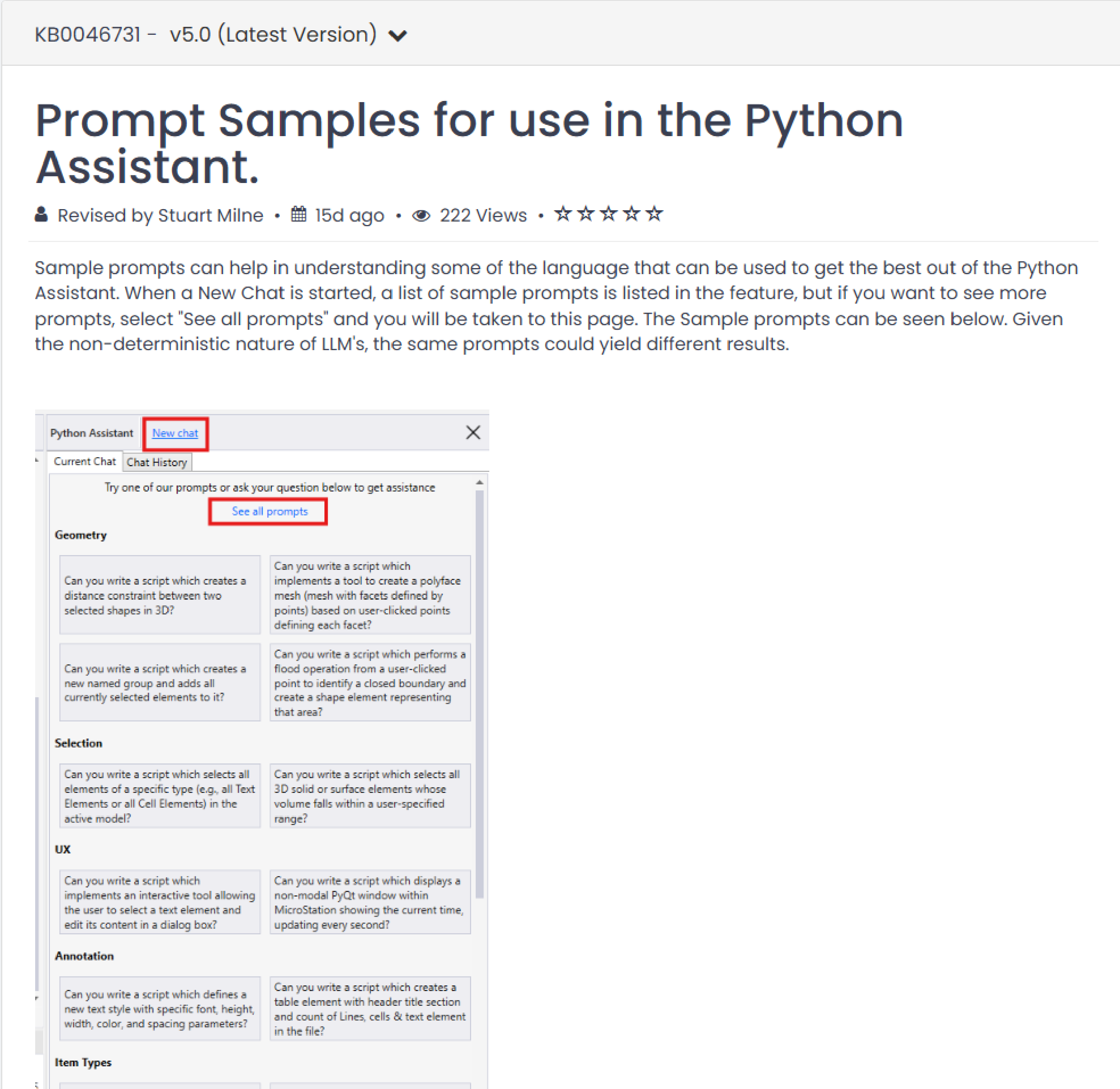
This code includes error handling for common issues and uses a raw string for the folder path to avoid escape sequence problems. If you encounter any specific error messages, please let me know, and I can help further.



```
Microsoft Windows [Version 10.0.26100.4270]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dan.Eng>Python MergePY.py
C:\Users\Dan.Eng\MergePY.py:18: SyntaxWarning: invalid escape sequence '\P'
  folder_path = 'D:\Python'

C:\Users\Dan.Eng>Python MergePY.py

C:\Users\Dan.Eng>
```



□ Name	Date modified	Type	Size
☑ merged_scripts.txt	6/25/2025 6:45 AM	Text Document	1,152 KB
MergePY.py	6/25/2025 6:45 AM	Python Source File	2 KB



KB0046731 - v5.0 (Latest Version)

Prompt Samples for use in the Python Assistant.

Revised by Stuart Milne • 15d ago • 222 Views • ☆☆☆☆☆

Sample prompts can help in understanding some of the language that can be used to get the best out of the Python Assistant. When a New Chat is started, a list of sample prompts is listed in the feature, but if you want to see more prompts, select "See all prompts" and you will be taken to this page. The Sample prompts can be seen below. Given the non-deterministic nature of LLM's, the same prompts could yield different results.

Try one of our prompts or ask your question below to get assistance

Geometry

Can you write a script which creates a distance constraint between two selected shapes in 3D?

Can you write a script which implements a tool to create a polyface mesh (mesh with facets defined by points) based on user-clicked points defining each facet?

Can you write a script which creates a new named group and adds all currently selected elements to it?

Can you write a script which performs a flood operation from a user-clicked point to identify a closed boundary and create a shape element representing that area?

Selection

Can you write a script which selects all elements of a specific type (e.g., all Text Elements or all Cell Elements) in the active model?

Can you write a script which selects all 3D solid or surface elements whose volume falls within a user-specified range?

UX

Can you write a script which implements an interactive tool allowing the user to select a text element and edit its content in a dialog box?

Can you write a script which displays a non-modal PyQt window within MicroStation showing the current time, updating every second?

Annotation

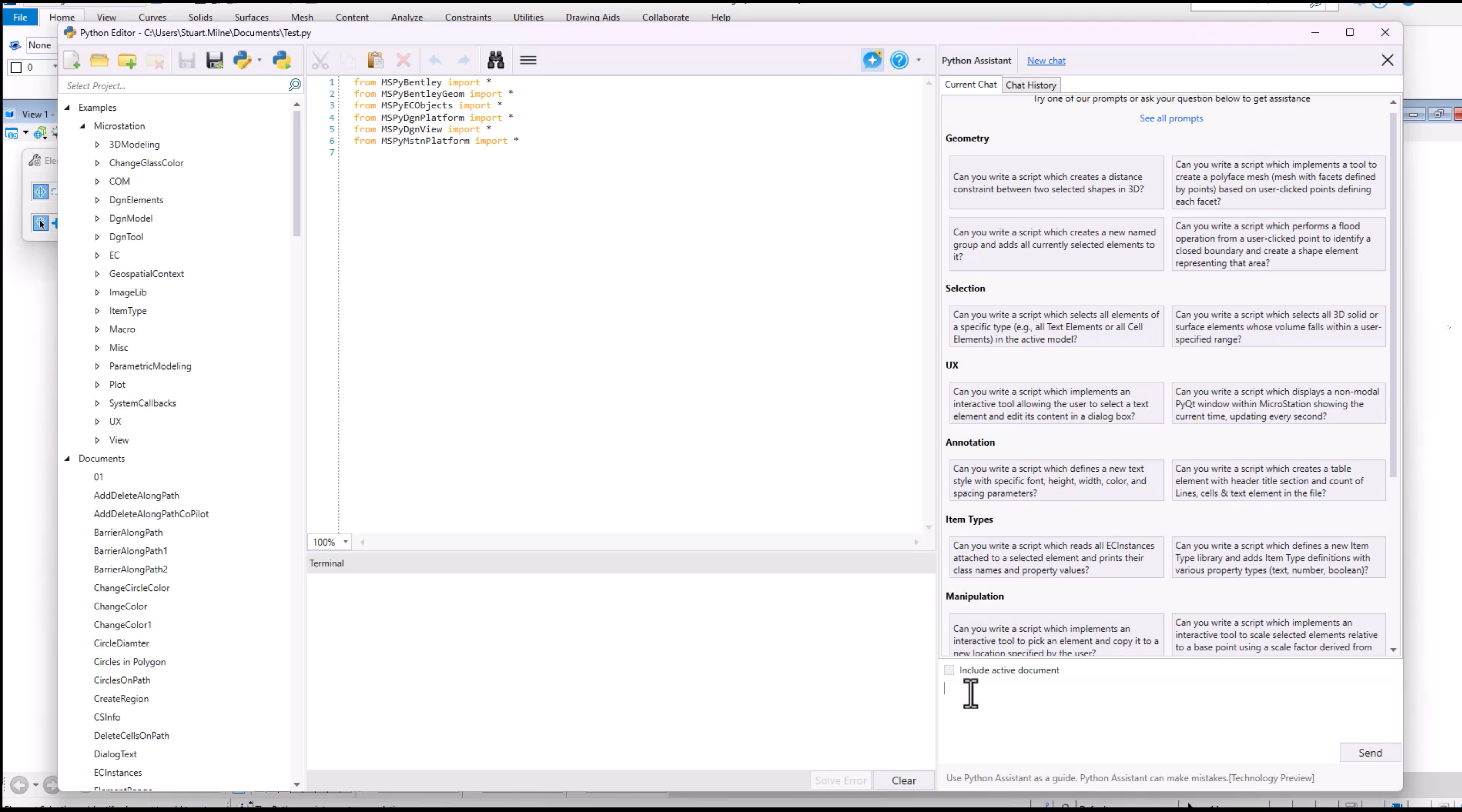
Can you write a script which defines a new text style with specific font, height, width, color, and spacing parameters?

Can you write a script which creates a table element with header title section and count of Lines, cells & text element in the file?

Item Types

Some more examples…

- Physics based Animations
- Geometry placement
- Patterns
- Solids Geometry manipulation
- Endless possibilities



```
from MSPyBentley import *
from MSPyBentleyGeom import *
from MSPyECObjects import *
from MSPyDgnPlatform import *
from MSPyDgnView import *
from MSPyMstnPlatform import *
```

Try one of our prompts or ask your question below to get assistance

Geometry

- Can you write a script which creates a distance constraint between two selected shapes in 3D?
- Can you write a script which implements a tool to create a polyface mesh (mesh with facets defined by points) based on user-clicked points defining each facet?
- Can you write a script which creates a new named group and adds all currently selected elements to it?
- Can you write a script which performs a flood operation from a user-clicked point to identify a closed boundary and create a shape element representing that area?

Selection

- Can you write a script which selects all elements of a specific type (e.g., all Text Elements or all Cell Elements) in the active model?
- Can you write a script which selects all 3D solid or surface elements whose volume falls within a user-specified range?

UX

- Can you write a script which implements an interactive tool allowing the user to select a text element and edit its content in a dialog box?
- Can you write a script which displays a non-modal PyQt window within MicroStation showing the current time, updating every second?

Annotation

- Can you write a script which defines a new text style with specific font, height, width, color, and spacing parameters?
- Can you write a script which creates a table element with header title section and count of Lines, cells & text element in the file?

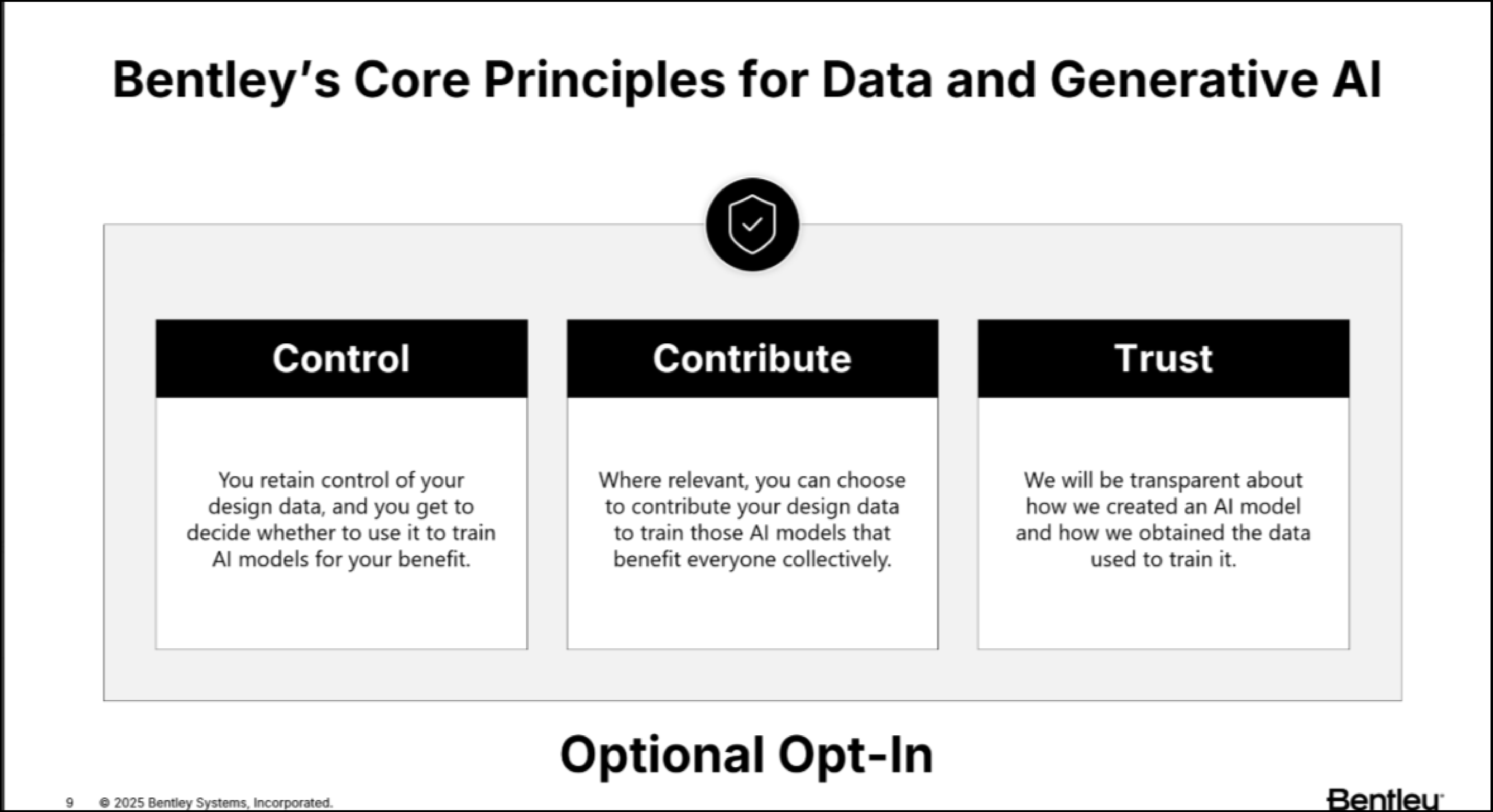
Item Types

- Can you write a script which reads all ECInstances attached to a selected element and prints their class names and property values?
- Can you write a script which defines a new Item Type library and adds Item Type definitions with various property types (text, number, boolean)?

Manipulation

- Can you write a script which implements an interactive tool to pick an element and copy it to a new location specified by the user?
- Can you write a script which implements an interactive tool to scale selected elements relative to a base point using a scale factor derived from

Use Python Assistant as a guide. Python Assistant can make mistakes.[Technology Preview]



Bentley’s Core Principles for Data and Generative AI

Control

You retain control of your design data, and you get to decide whether to use it to train AI models for your benefit.

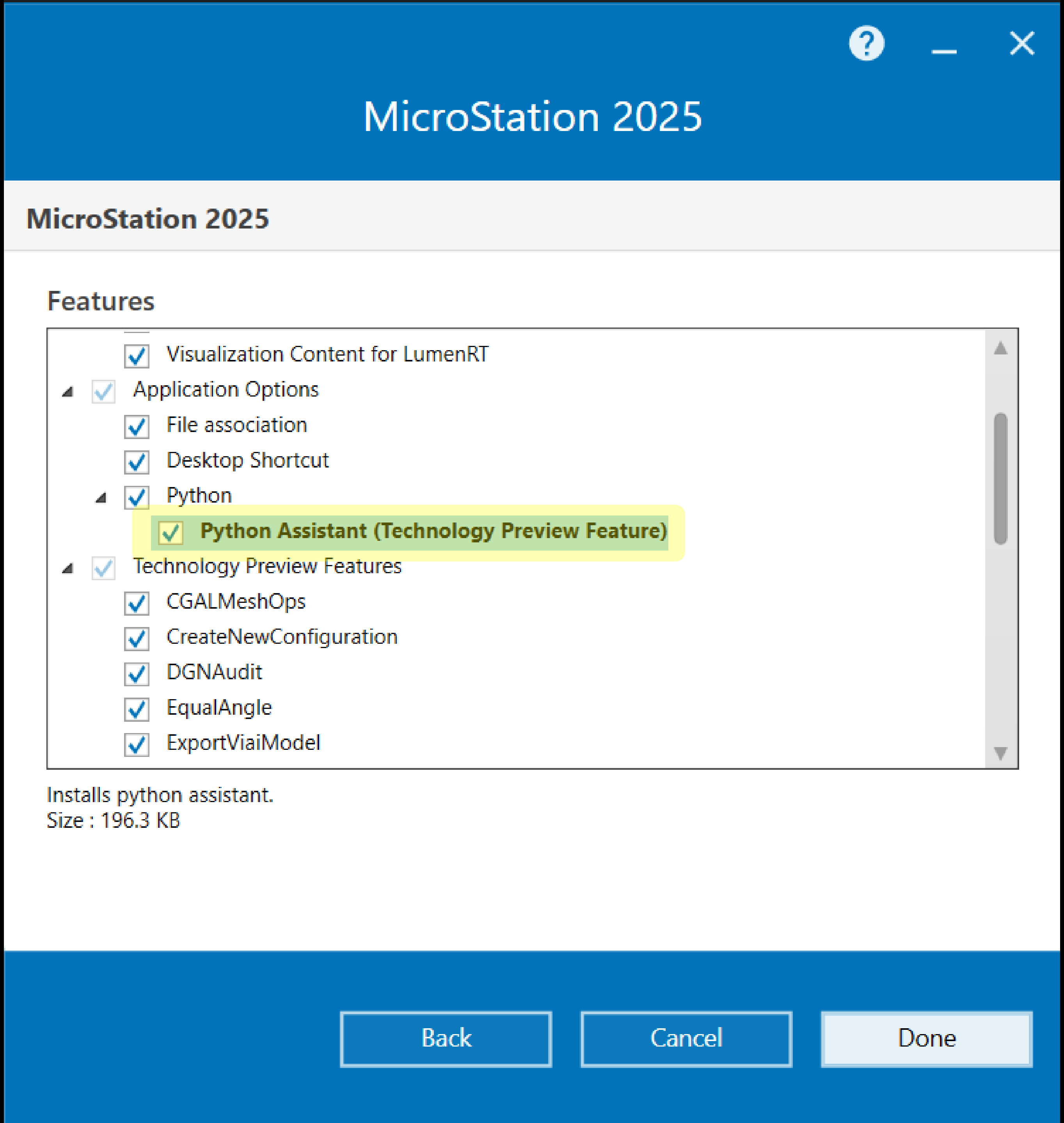
Contribute

Where relevant, you can choose to contribute your design data to train those AI models that benefit everyone collectively.

Trust

We will be transparent about how we created an AI model and how we obtained the data used to train it.

Optional Opt-In



MicroStation 2025

Features

- ☑ Visualization Content for LumenRT
- ☑ Application Options
  - ☑ File association
  - ☑ Desktop Shortcut
  - ☑ Python
    - ☑ **Python Assistant (Technology Preview Feature)**
- ☑ Technology Preview Features
  - ☑ CGALMeshOps
  - ☑ CreateNewConfiguration
  - ☑ DGNAudit
  - ☑ EqualAngle
  - ☑ ExportViaiModel

Installs python assistant.
Size : 196.3 KB

learn to code
learn to write prompt