



# McNeel Europe

European division of Robert McNeel & Associates

## Rhino.Python Training Program

For designers and engineers who want to lay the foundation for Rhino.Python scripting, as well as RhinoScript users who want to stay up to date, this workshop will convey necessary ideas and tools to successfully code geometry in Rhinoceros 5 on both Windows and Mac. McNeel's instructor is Giulio Piacentino.

As a participant, you will be guided through the details of this new, readable and friendly programming language within Rhino, including: automating commands at necessity with macros, performing calculations, making decisions after iterating code and manipulating several data structures. Additionally, we will study how to couple Python's iteration and recursion powers to extend Grasshopper.

### Structure

This course is divided in 6 units of 4 hours, for a total of three days. We will start by running scripts with the editor and then learn to write them by ourselves. We'll also learn tricks to debug and to optimize our code. We will cover all Rhino's geometrical base, building through vectors, points, curves, meshes, surfaces and polysurfaces in order to achieve an understanding of the potential of each of these elementary and first-level types.

We will use both the Rhinoscript syntax, that allows easy-to-learn scripting and rapid prototyping and the comprehensive RhinoCommon library, that contains methods that are used in-house daily within McNeel.

### Audience

This course is intended for existing Rhino users who have working knowledge of 3D modeling.

Examples of required previous knowledge are:

- Definition of line and circle
- Difference between NURBS surfaces and meshes to the naked eye
- How do you find the distance between two points? (Looking up formulas is allowed)

### Results

After this course, successful participants will know how to:

- Run, edit and compose Python code for Rhinoceros 5
- Identify and correct common bugs that can prevent code from executing
- Distinguish variables, functions and other procedural and object-oriented paradigms
- Find their way to access available resources both online and offline
- Express reasoned ideas in working pseudocode
- Given sufficient time, translate pseudocode to working Python code

### Software prerequisites

- Rhinoceros v5 (most recent version)
- Grasshopper (free - latest version from [www.grasshopper3d.com](http://www.grasshopper3d.com))
- GhPython (free - latest version from [www.food4rhino.com/project/GhPython](http://www.food4rhino.com/project/GhPython))

## Outline

This is the time plan and coverage of this three-day course. We will be anyways arranging the pace dynamically as we go through exercises, puzzles, questions and autonomous attempts.

### Day I, AM – Environment

The goal is to get acquainted with the scripting environment and get it to play by the rules

- The Python Script Editor
- Online resources:
  - RhinoScriptSyntax SDK Help location. RhinoCommon SDK Help location
- Running existing scripts
- The Python 'Welcome and good morning' script: your FIRST SCRIPT
- Variables: uses and name conventions. Conditionals and their use
- Common arithmetic operations

### Day I, PM – (2D+3D) Graphics from macro-style

Commanding Rhino and computer will be easier and also understanding what happens 'on the inside'

- Packaging: getting data from the user
- Expressions: evaluation
- Iteration: looping code several time: for loop and while loop. Construct autonomous textures
- Debug: break points, prints, call stack
- Points, vectors, lines, circles, cones. Containers: lists, tuples. Invariance
- Curves, surfaces, 3D space

### Day II, AM – Writing function

- Pseudocode for recursion-based fractals
- Splitting curves with parameterization: lines
- Applying trigonometric definitions to understand lengths of elements
- Quiz & open questions
- Brownian motion. Adding up particle movements
- Analyzing surfaces and curves. Modifying geometry to fit criteria
- Building geometrical types from data
- Loading and saving to files

### Day II, PM – Giving interfaces to your projects

- Books suggestions
- Last missing geometrical types: introductions, fundamentals. Duck typing (interfaces)
- Using library classes rather than RhinoScriptSyntax; why where and when
- Introduction to class instances: objects. Windows Forms user interface (UI)
- Animating the viewport and rendering
- Keeping the script responsive
- Profiling, variations and optimization

### Day III, AM – Your own enterprise

On third day we will focus on getting confident to start a scripting project or enterprise, even a very simple one, completely on our own. Please give it some thoughts beforehand, or just come to the class waiting to get some inspiration and willingness to continue one of the previous exercises.

- Rehearsal of the crucial parts of the previous days
- Deepening on student's questions (as time allows)
- Depicting of student's focus. Groups of interest
- Brainstorming on approaches and algorithms

### Day III, PM – Learn managing own projects

- Working on own assignment
- General discussions on projects
- Distribution of the certificates of completion