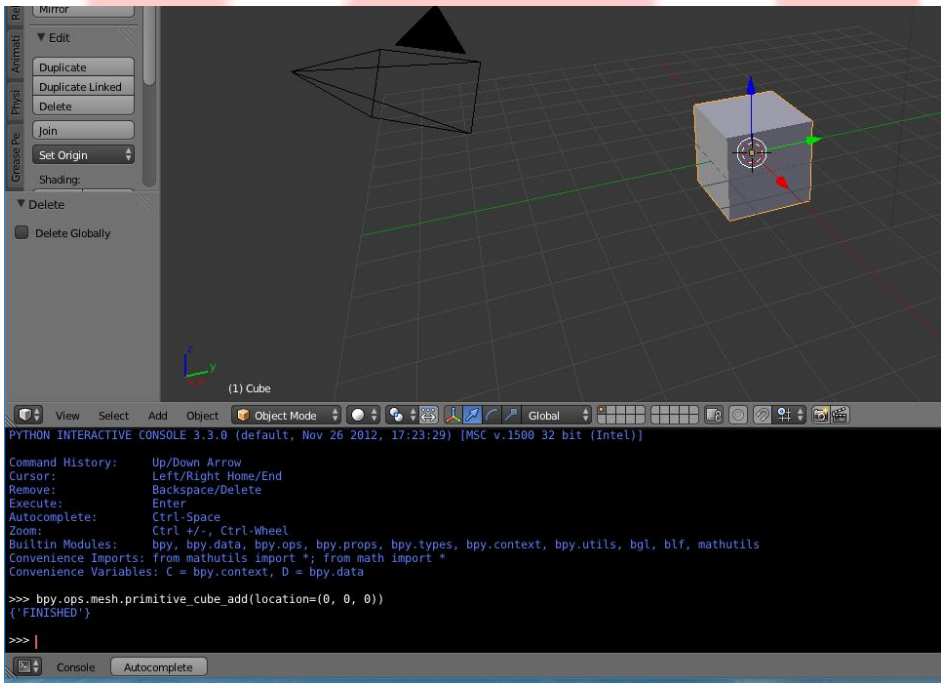
e-Yantra Robotics Competition (eYRC-2017)
Task 1 – Transporter Bot

We will now see the next sub module in BPY which is **bpy.ops.** The tool system is built around the concept of operators. Operators are typically executed from buttons or menus but can be called directly from Python too.

Operators don't return values as you might expect, instead they return a set() which is made up of: {'RUNNING_MODAL', 'CANCELLED', 'FINISHED', 'PASS_THROUGH'}. Common return values are {'FINISHED'} and {'CANCELLED'}.

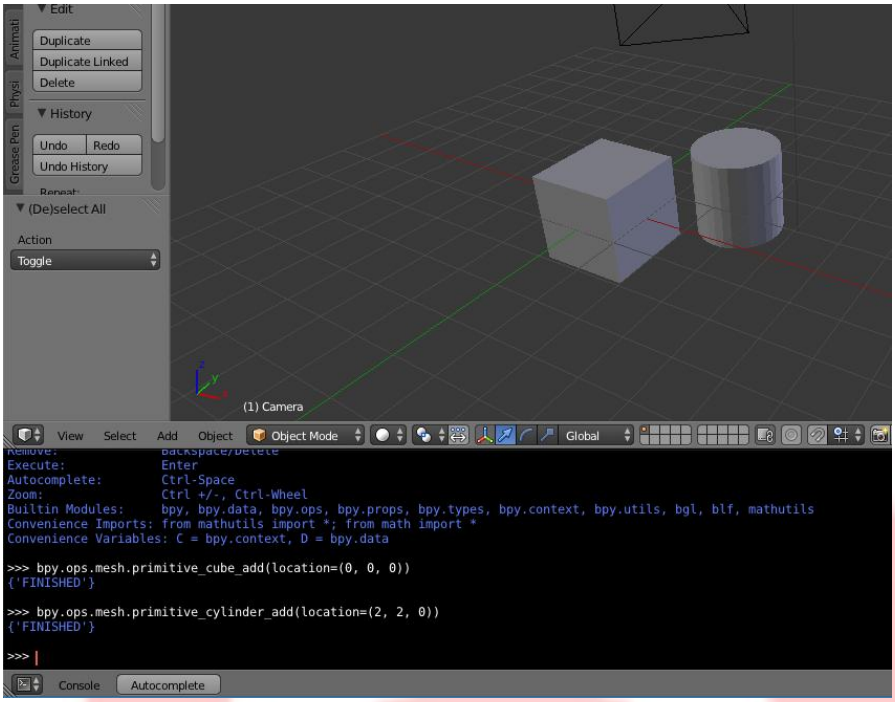
You can access, add or modify settings of Blender's data using Python script for:

1. **Add Mesh:**
   - Add Cube:
     bpy.ops.mesh.primitive_cube_add(location=(0, 0, 0))



   Where, the location = (0, 0, 0) is the x, y and z axis where you want the object to be added.

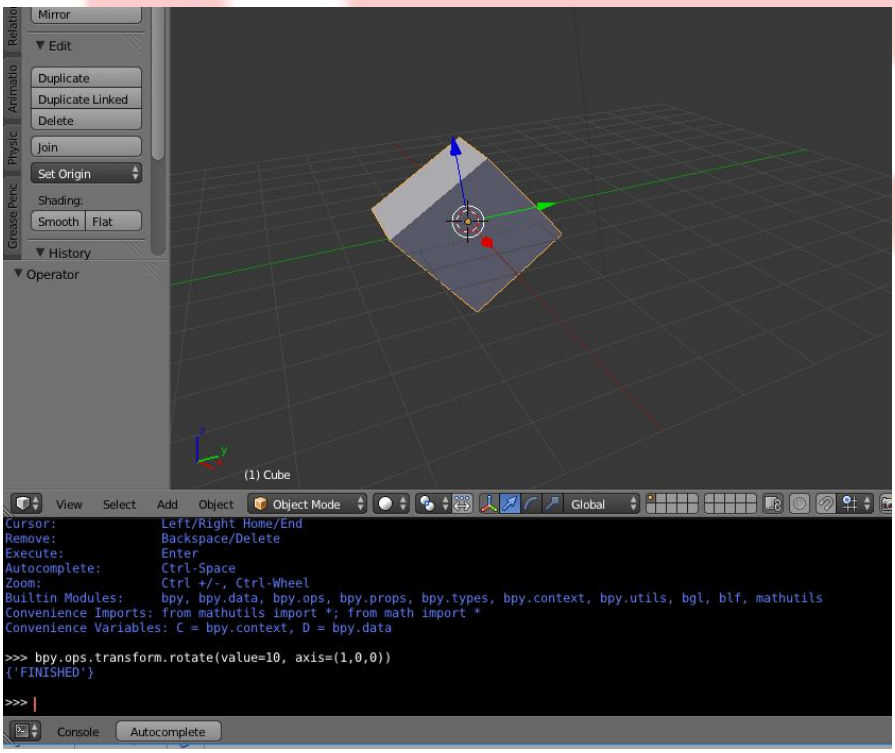
   - Add Cylinder:
     bpy.ops.mesh.primitive_cylinder_add(location=(0, 0, 0))



and similarly rest of the meshes. You can press “SHIFT+A” and hover on the object you want to add. The tooltip will show the Python attributes.

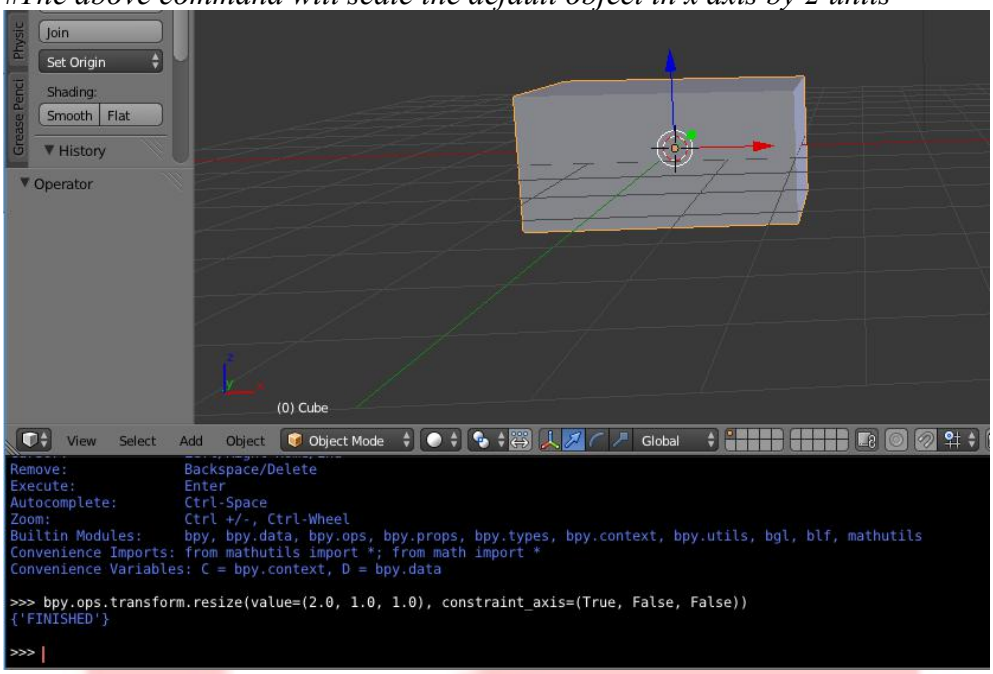
2. Basic Transformation:

- **Rotate:**
  bpy.ops.transform.rotate(value=10, axis=(1,0,0))



where,
value = 10, axis=(1,0,0) - rotate about global X by 10 degrees
value = 90, axis=(1,0,0) - rotate about global X by 90 degrees
value = 90, axis=(0,0,1) - rotate about global Z by 90 degrees

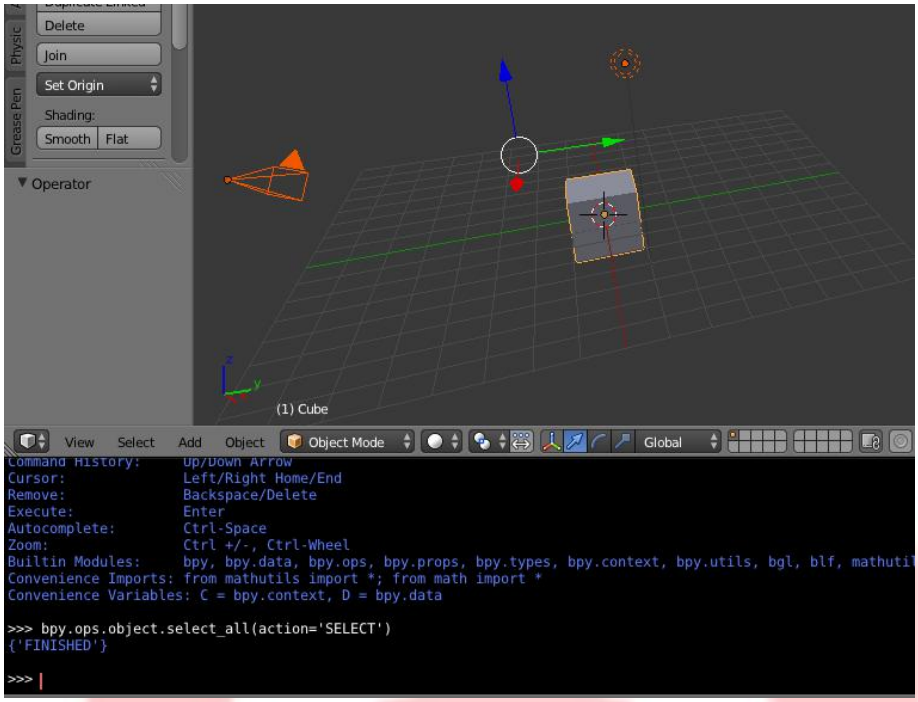
- **Scale:**
bpy.ops.transform.resize(value=(2.0, 1.0, 1.0), constraint_axis=(True, False, False))
#The above command will scale the default object in x axis by 2 units



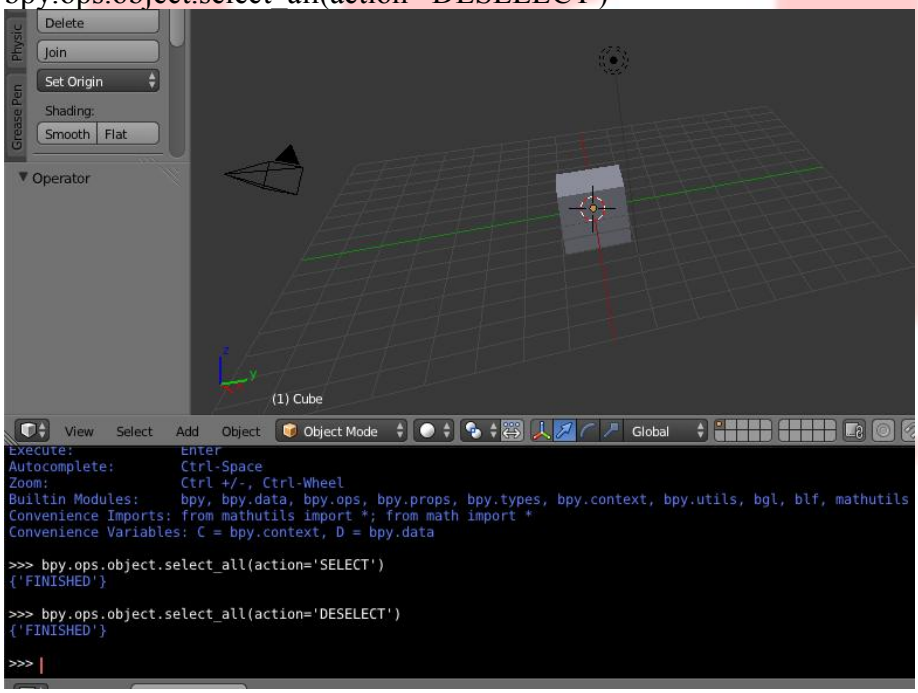
Similarly,
bpy.ops.transform.resize(value=(1.0, 2.0, 1.0), constraint_axis=(False, True, False))
#The above command will scale the default object in x axis by 2 units

bpy.ops.transform.resize(value=(1.0, 1.0, 2.0), constraint_axis=(False, False, True))
#The above command will scale the default object in x axis by 2 units

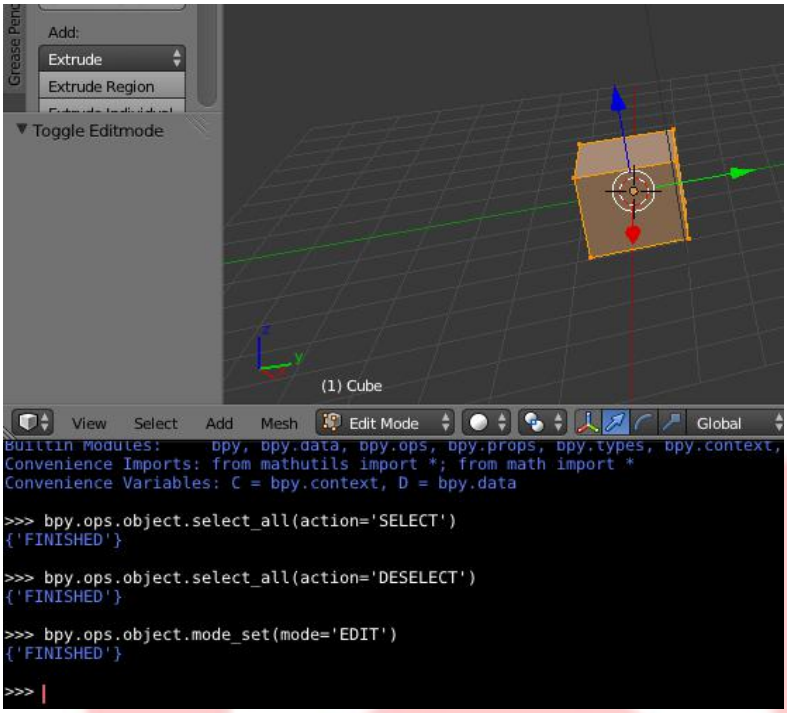
- **To select all objects present in the 3D window:**
bpy.ops.object.select_all(action='SELECT')



- **To deselect all:**
  bpy.ops.object.select_all(action='DESELECT')

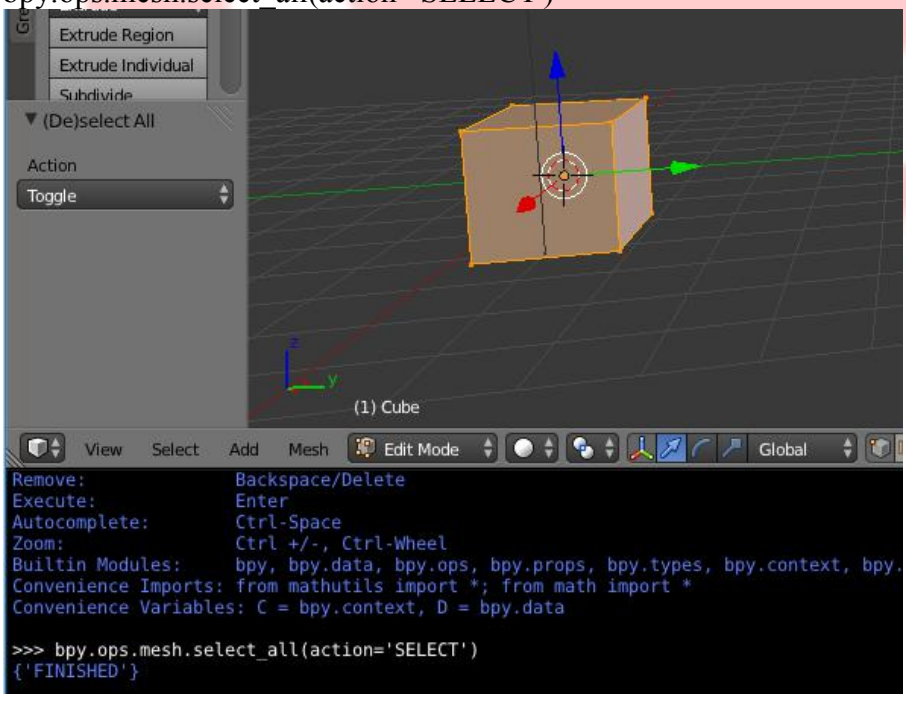


3. **Switch to Modes:**
   bpy.ops.object.mode_set(mode='OBJECT')
   #by default, objects are in object mode.
   bpy.ops.object.mode_set(mode='EDIT')



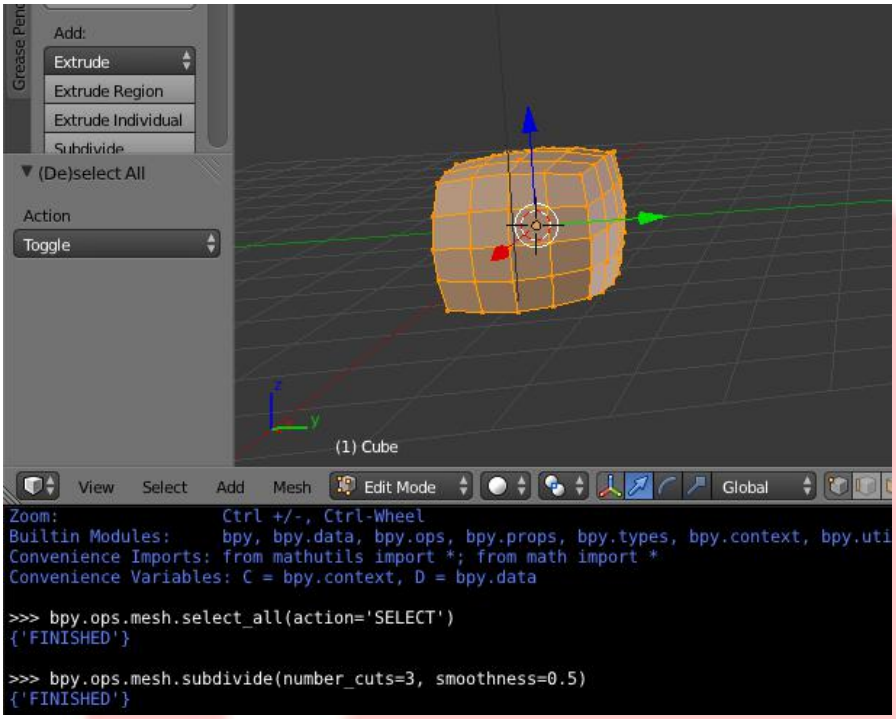
- Select types in edit mode:
  bpy.ops.mesh.select_mode(type="VERT")
  #by default the object when changed to edit mode, its type is Vertex
  bpy.ops.mesh.select_mode(type="EDGE")
  bpy.ops.mesh.select_mode(type="FACE")

- Select all vertices, edges or faces:
  bpy.ops.mesh.select_all(action='SELECT')



- Deselect:
  bpy.ops.mesh.select_all(action='DESELECT')
  #This command will deselect the selected object.

4. **Accessing other operators in edit mode:**
   - To subdivide the object:
     bpy.ops.mesh.subdivide(number_cuts=3, smoothness=0.5)



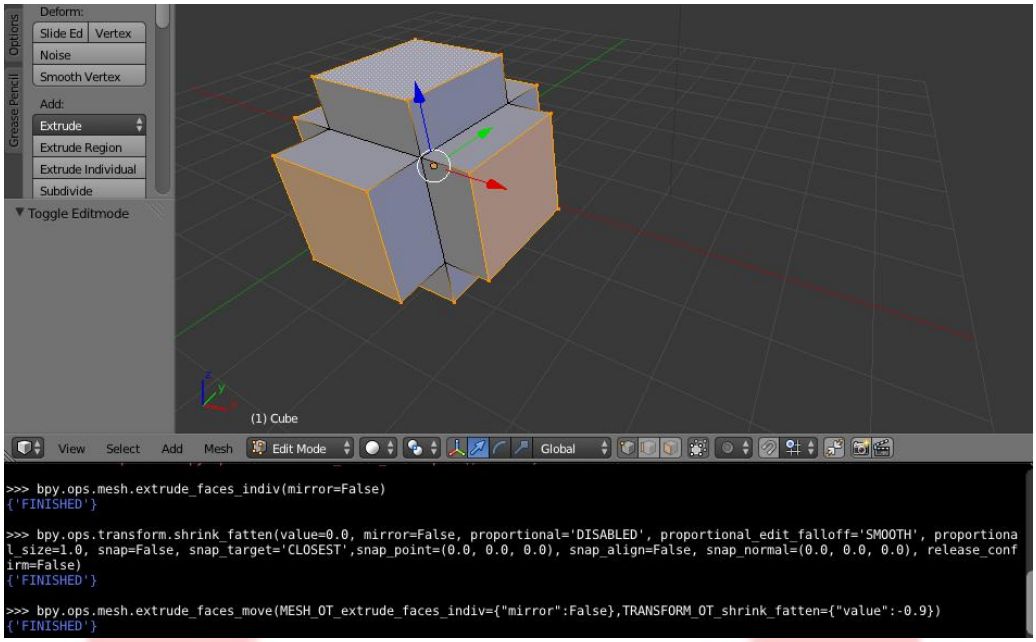
   - To extrude an object by its face:
     The extrude operators are macros. For the sub operators, you have to provide a dictionary with the properties you want to pass to them.

     bpy.ops.mesh.extrude_faces_move(**MESH_OT_extrude_faces_indiv**=None, **TRANSFORM_OT_shrink_fatten**=None)

     bpy.ops.mesh.extrude_faces_indiv(mirror=False)

     bpy.ops.transform.shrink_fatten(value=0.0, mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST',snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)

     bpy.ops.mesh.extrude_faces_move(MESH_OT_extrude_faces_indiv={"mirror":False}, TRANSFORM_OT_shrink_fatten={"value":-0.9})



Refer to the given link to know more on:

Blender Python API Documentation:
http://www.blender.org/api/blender_python_api_current/

Learn the basic BPY commands from the link given below:
https://www.blender.org/manual/editors/python_console.html