FreeCAD Scripting summary

This is not a tutorial. It is my private summary of what I have learned in relation to FreeCAD scripting.

jean-claude.feltes@education.lu

Much of this is taken from
https://www.freecadweb.org/wiki/index.php?title=Power_users_hub

Create a new empty document and set it as active document:

```
doc= App.newDocument("myDocument")
App.setActiveDocument("myDocument")
```

Create a 3D object:

```
box = doc.addObject("Part::Box","myBox")
doc.recompute()
```

The object is seen in the tree view as soon as it is created, but the doc.recompute() is necessary to see the result on the screen.

Alternative: use the Part module

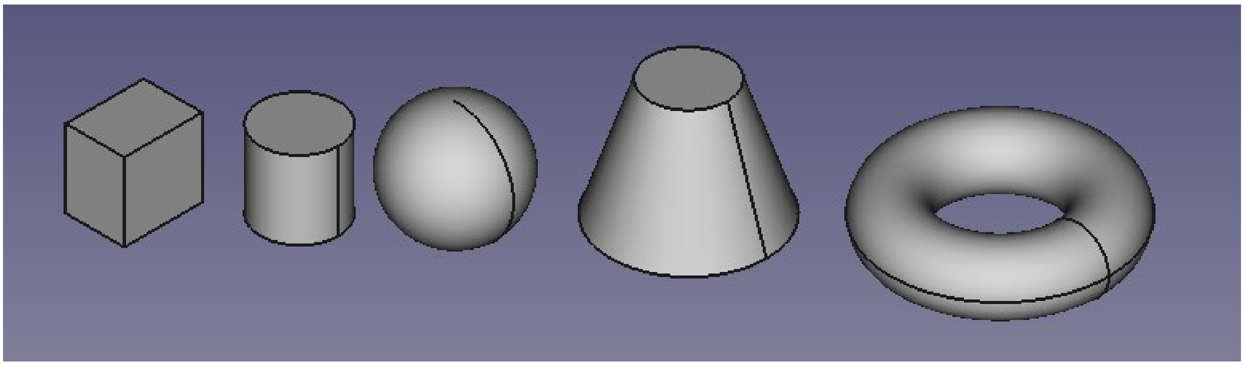
```
import Part
c = Part.makeCylinder(10,20)
Part.show(c)
```

Create 3D primitives

```
import Part

from FreeCAD import Vector
b1=Part.makeBox(15,20,20, Vector(0,0,0))
b2=Part.makeCylinder(10,20, Vector(30,30,0))
b3=Part.makeSphere(15, Vector(50,50,10))
b4=Part.makeCone(20,10,30, Vector(80,80,0))
b5=Part.makeTorus(20,8, Vector(120,120,0))


for i in [b1, b2, b3, b4, b5]:
      Part.show(i)
```



Here another syntax was used. Instead of

```
doc.addObject("Part::Box","myBox")
```

for example, the makeBox function of the Part module was used, and the object was positioned with a Vector. The Part.show() function corresponds to the recompute() in the above example.

Area, Volume etc.

Once an object is created, some interesting properties are available:

```
>>>import Part
>>> c = Part.makeCylinder(10,20)
>>> c.Area
1884.9555921538758
>>> c.BoundBox
BoundBox (-10.8239, -10.8239, 0, 10.8239, 10.8239, 20)
>>> c.CenterOfMass
Vector (1.4475057750305073e-16, 1.1132478570075483e-16, 9.999999999999998)
>>> c.Edges
[<Edge object at 0x371d120>, <Edge object at 0x401a150>, <Edge object at 0x40122b0>]
>>> c.Faces
[<Face object at 0x401b910>, <Face object at 0x401f020>, <Face object at 0x401b0e0>]
>>> c.Length
291.32741228718345
>>> c.Orientation
'Forward'
>>> c.Placement
Placement [Pos=(0,0,0), Yaw-Pitch-Roll=(0,0,0)]
```

Vectors

```
myvec = FreeCAD.Vector(2,0,0)
v1 = FreeCAD.Vector(2,3,0)
v2 = FreeCAD.Vector(3,4,1)
```

```
print v1+v2
Vector (5.0, 7.0, 1.0)
```

Placements

Placements contain the position (Base) and the orientation (Rotation) of the object.

Shift an object to a new place:

```
v1 = FreeCAD.Vector(2,3,0)
b.Placement.Base = v1
```

or

```
myobj.Placement.Base = FreeCAD.Vector(2,3,0)
```

A broader explanation is found here:

https://www.freecadweb.org/wiki/index.php?title=Placement

Accessing objects by code

Example:


```
>>> myobj = App.ActiveDocument.getObject("Cylinder")
>>> myobj.Name
'Cylinder'
>>> myobj.Label
u'Cylinder'
```

Note: this accesses the App part of the object, that means the geometric properties.

In contrast the Gui part of the object contains the properties in relation to it's representation on the screen, e.g. color, transparency etc.

For, example to change the color of the Cylinder to red:

```
myViewObject = Gui.ActiveDocument.getObject("Cylinder")
myViewObject.ShapeColor = (1.0,0.0,0.0)
```

Getting the selected object

To get the first selected object :

```
o=Gui.Selection.getSelection()[0]
print o.Name
```

Getting all selected objects:

```
sel=Gui.Selection.getSelection()
for obj in sel:
      print obj.Name
```

Which type of object is it?

```
>>> myobj.TypeId
'Part::Cylinder'
```

The difference between shape, wire, edge etc.

Here it is very well explained:

https://yorikvanhavre.gitbooks.io/a-freecad-manual/content/python_scripting/creating_and_manipulating_geometry.html

So to resume the whole diagram of Part Shapes:

- Everything starts with Vertices.
- With one or two vertices, you form an Edge (full circles have only one vertex).
- With one or more Edges, you form a Wire.
- With one or more closed Wires, you form a Face (the additional Wires become "holes" in the Face).
- With one or more Faces, you form a Shell.
- When a Shell is fully closed (watertight), you can form a Solid from it.
- And finally, you can join any number of Shapes of any types together, which is then called a Compound.

Analyse a selected object:

```
o=Gui.Selection.getSelection()[0]
print "selected object of type ", o.TypeId

print "-> to shape"
s= o.Shape
print "shape of type ", s.TypeId," ", s.ShapeType

print "Edges: "
print s.Edges

print "Faces: "
print s.Faces

print "Vertices: "
print s.Vertexes

print "Shells: "
print s.Shells
print "Solids: "
print s.Solids
```

Create 2D objects

```
import FreeCAD
import DraftTools
import Draft

faceflag= True

def place(x,y):
    pl=FreeCAD.Placement()
    #pl.Rotation.Q=(0.0,-0.0,-0.0,1.0)
    pl.Base=FreeCAD.Vector(x,y,0.0)
    return pl

def c(x,y,diameter):
    """circle"""
    radius = diameter/2
    pl=place(x,y)
    c = Draft.makeCircle(radius, placement=pl, face=faceflag)

def r(x,y,l,h):
    """rectangle       x,y = left bottom corner,  l = length,  h = height"""
    faceflag=True
    pl=place(x,y)
    r = Draft.makeRectangle(length=l,height=h,placement=pl, face=faceflag)

def p(x,y,diameter, nb_edges):
    """polygon"""
    radius = diameter/2
    pl=place(x,y)
    p= Draft.makePolygon(nb_edges,radius, inscribed=True,placement=pl, face=faceflag)

def l(x1, y1, x2, y2):
    """line"""
    p1=FreeCAD.Vector(x1,y1,0)
    p2=FreeCAD.Vector(x2,y2,0)
    l=Draft.makeLine(p1, p2)
```

Appendix

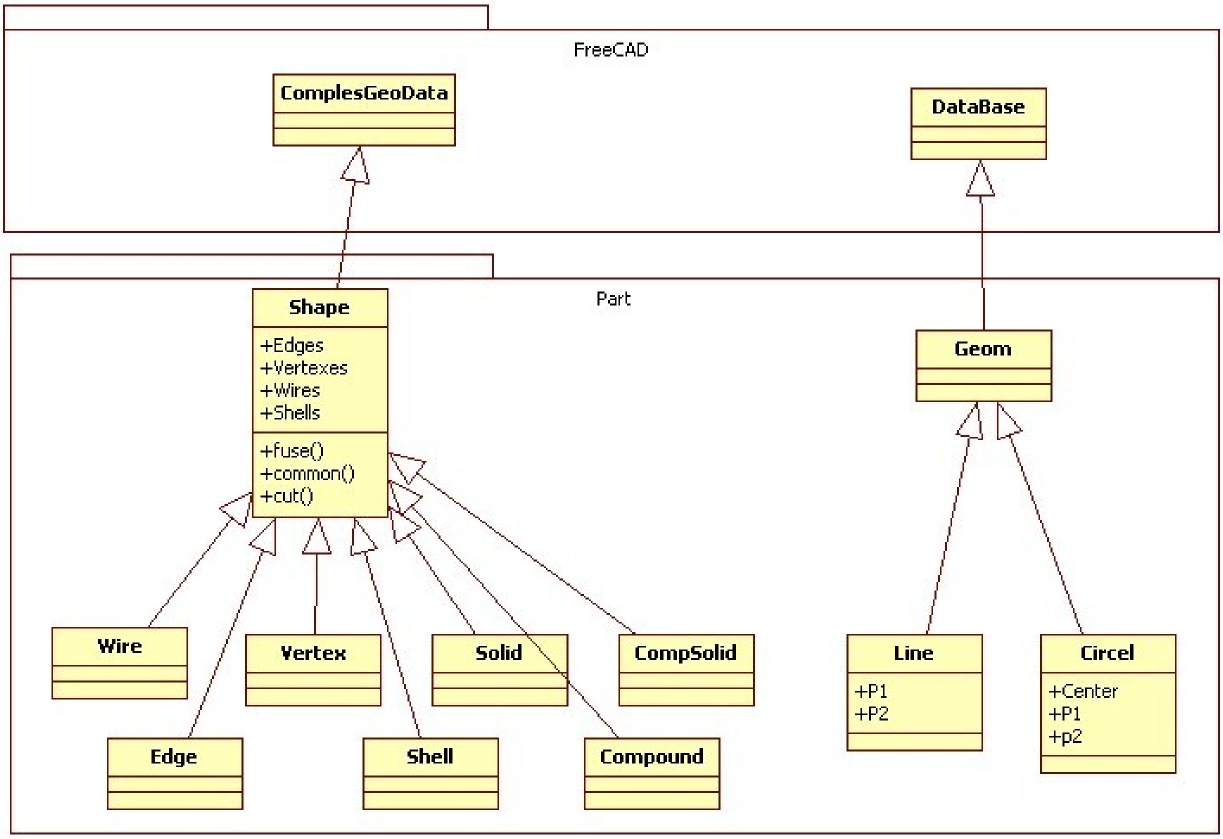
Getting the selected object   (other method)

```
s=Gui.Selection.getSelectionEx()[0]       # get first element of selection
myobj=s.Object
print myobj.Name
```

Note that there is a difference between the selection and the object itself! So it is necessary to convert with '.Object' !

Also be aware that the order of the elements we get is not necessarily the same as in the tree view!

Overview of Part classes



Geometry

The geometric objects are the building block of all topological objects:

- **Geom** Base class of the geometric objects
- **Line** A straight line in 3D, defined by starting point and end point
- **Circle** Circle or circle segment defined by a center point and start and end point
- **......** And soon some more

Topology

The following topological data types are available:

- **Compound** A group of any type of topological object.
- **Compsolid** A composite solid is a set of solids connected by their faces. It expands the notions of WIRE and SHELL to solids.
- **Solid** A part of space limited by shells. It is three dimensional.
- **Shell** A set of faces connected by their edges. A shell can be open or closed.
- **Face** In 2D it is part of a plane; in 3D it is part of a surface. Its geometry is constrained (trimmed) by contours. It is two dimensional.

- **Wire** A set of edges connected by their vertices. It can be an open or closed contour depending on whether the edges are linked or not.
- **Edge** A topological element corresponding to a restrained curve. An edge is generally limited by vertices. It has one dimension.
- **Vertex** A topological element corresponding to a point. It has zero dimension.
- **Shape** A generic term covering all of the above.