

# Python OCC

Uporabna literatura:

<http://trac.lecad.si/vaje/raw-attachment/wiki/PythonOcc/modalg.pdf>

## Struktura programa v Python OCC

Rabimo uvoziti uporabniški vmesnik, ki nam omogoči preprosto manipulacijo predstavljenega objekta (glava dokumenta):

```
from OCC.Display.SimpleGui import *
```

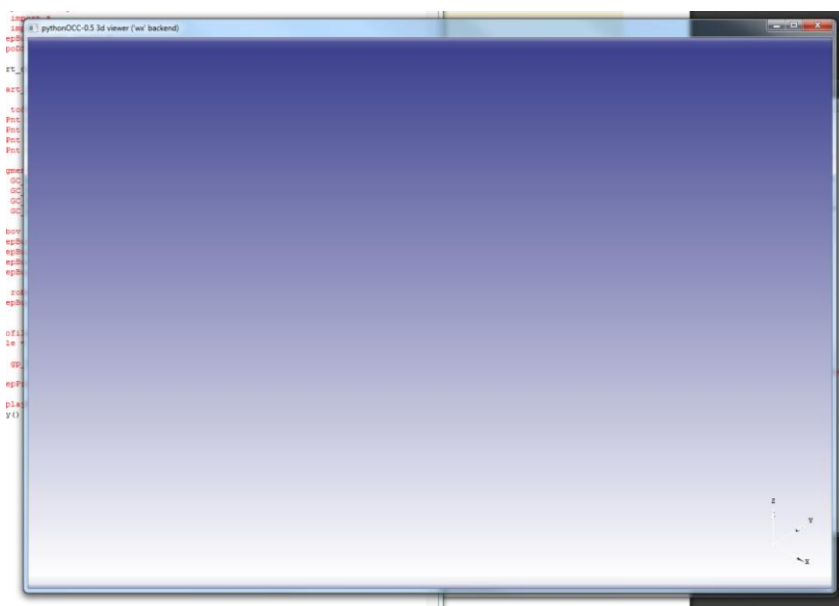
Uvedba funkcij za iniciacijo prikaza:

```
display, start_display, add_menu, add_function_to_menu = init_display()
```

Iniciacija prikaznega okna:

```
start_display()
```

Tako dobimo osnovno prikazno okno:



**Uporaba miške v uporabniškem vmesniku:**

- Rotacija objekta: levi gumb na miški
- Translacija objekta: srednji gumb na miški
- Povečava objekta: desni gumb na miški

**Uporaba tipkovnice v uporabniškem vmesniku:**

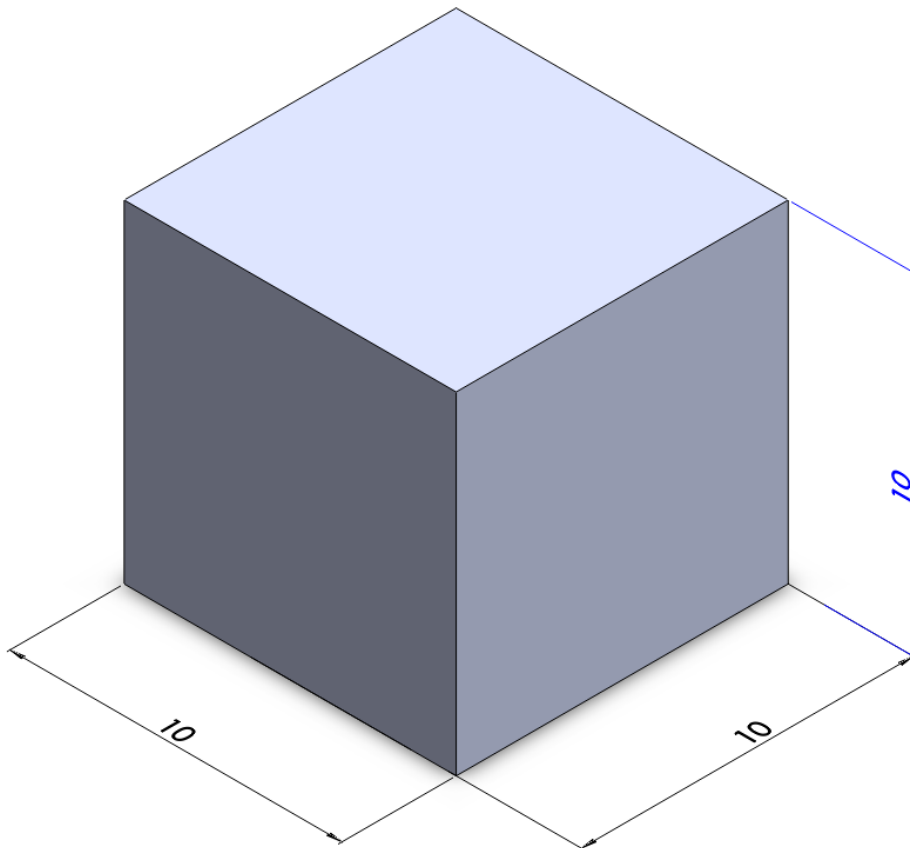
- Tipka 'w'- prikaz žičnega modela
- Tipka 'e'- prikaz vidnih robov
- Tipka 's'- prikaz volumnskega modela
- Tipka 'f'- prikaz celega objekta v prikaznem oknu
- Tipka 'q'- prikaz vidnih robov

**Pregled uporabljenih knjižnic:**

```
# -*- coding: utf-8 -*-
from OCC.Display.SimpleGui import * # Uporabniški vmesnik GUI
import math # Matematična knjižnica

# OpenCascade
from OCC.gp import * #točke
from OCC.BRepBuilderAPI import * #gradimo robove, segmente, mreže ...
from OCC.BRepPrimAPI import * #izdelava osnovnih geometrijskih primitivov
from OCC.BRepFilletAPI import * #izdelava zaokrožitvev
```

## Manualna definicija izvleka



Model kocke

Izdelava poljubne točke v prostoru:

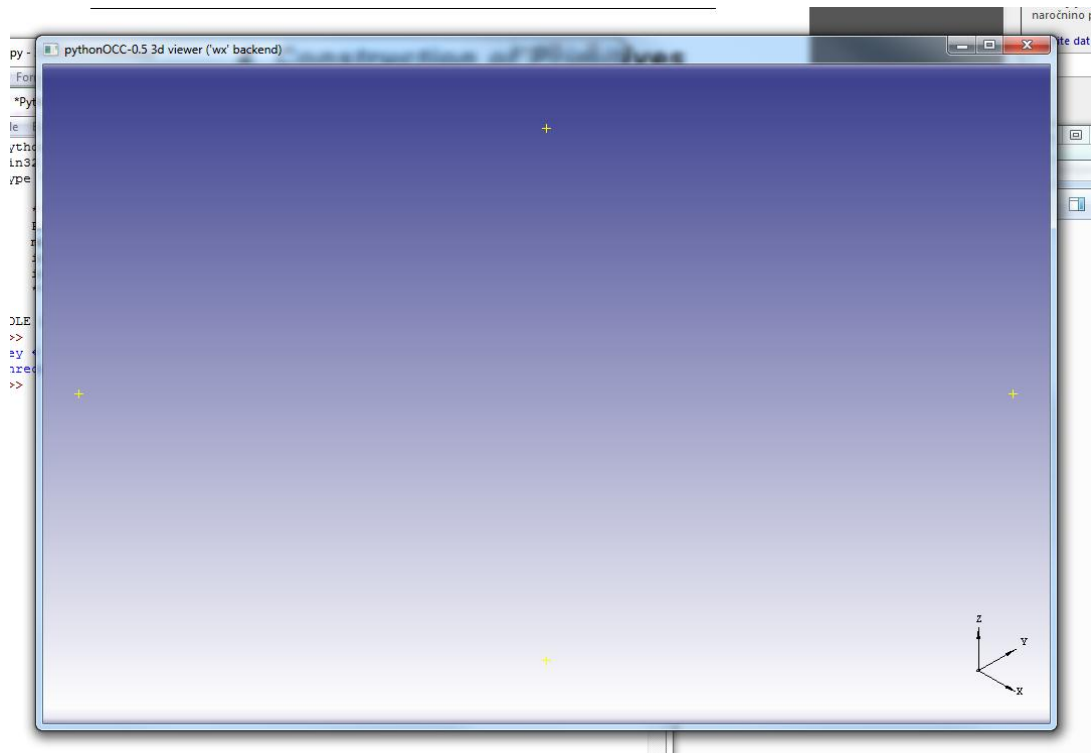
```
T1 = gp_Pnt(x_1 , y_1 , z_1)
```

Izdelava vozlišč na podlagi definiranih točk:

```
V1= BRepBuilderAPI_MakeVertex(T1)
```

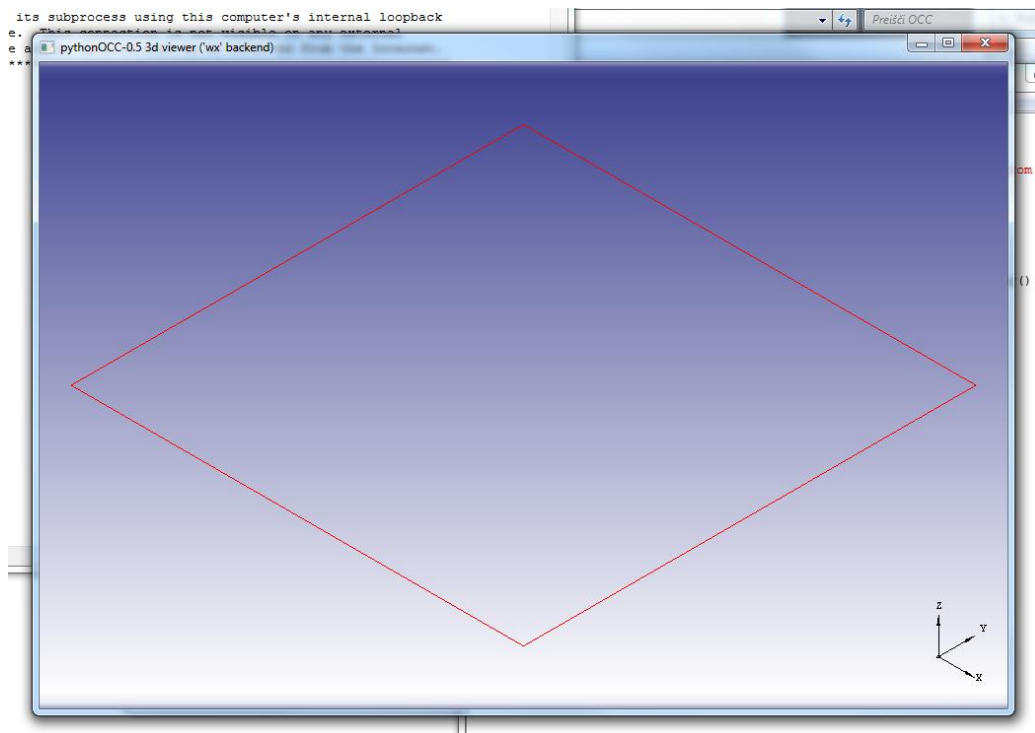
Prikaz vozlišč v prikaznem oknu:

```
display.DisplayShape(V1.Shape())
```



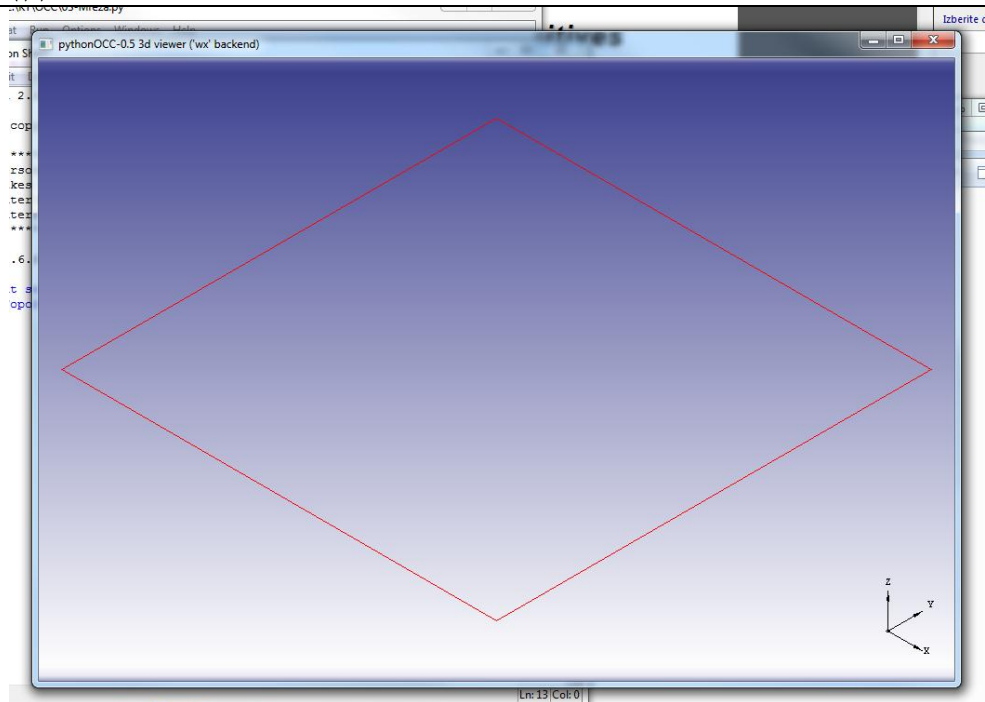
Izdelava robov iz definiranih točk:

```
E1 = BRepBuilderAPI_MakeEdge(T1, T2)
```



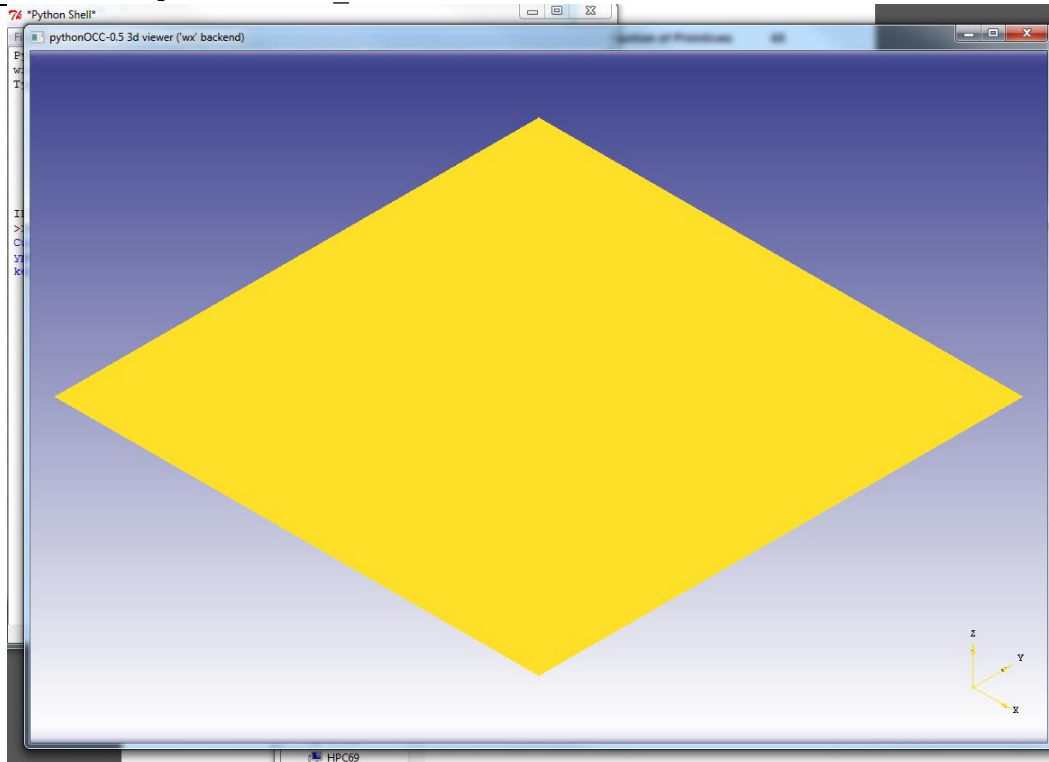
## Izdelava mreznega modela iz robov:

```
Mreza = BRepBuilderAPI_MakeWire(E1.Edge() , E2.Edge() ,E3.Edge() ,  
E4.Edge())
```



## Izdelava površine v prostoru:

```
Povrsina = BRepBuilderAPI_MakeFace(Mreza.Wire())
```

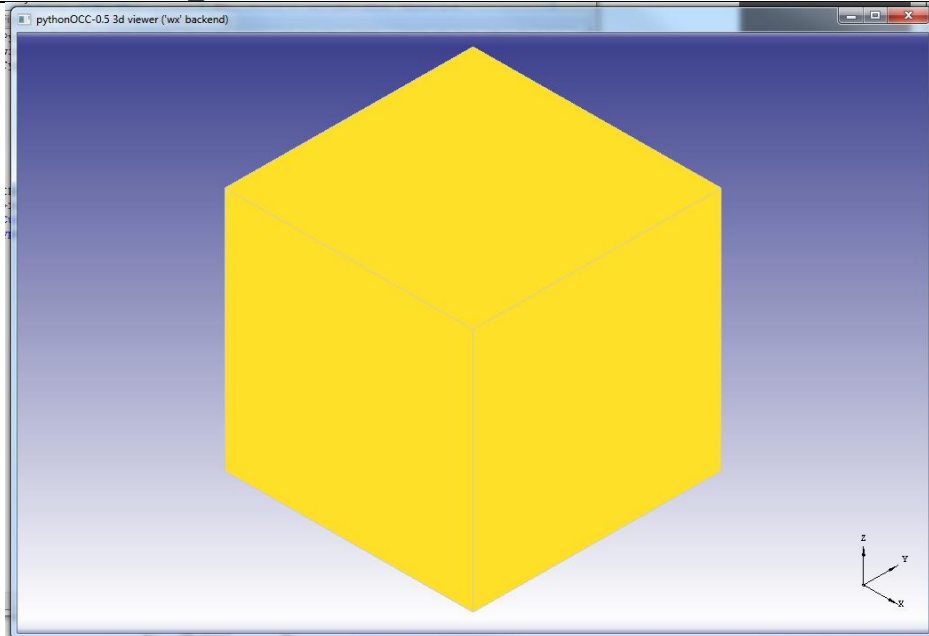


## Izdelava vektorja za izvlek v prostor

```
Vektor1 = gp_Vec(n_x , n_y, n_z)
```

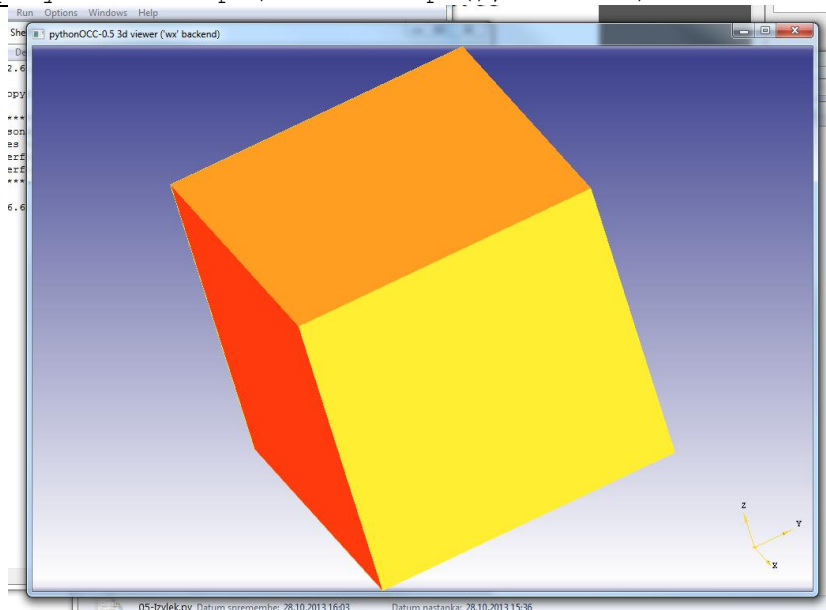
## Izdelava izvleka v prostor:

```
Izvlek = BRepPrimAPI_MakePrism(Povrsina.Face() , Vektor1)
```



## Sprememba barve modela:

```
display.DisplayColoredShape(OBLIKA.Shape(), 'BARVA')
```



## Izdelava geometrijskih primitivov

Za uporabo osnovnih geometrijskih primitivov znotraj PythonOCC-ja je potrebno najprej uvesti knjižnico:

```
from OCC.BRepPrimAPI import *
```

Znotraj knjižnice imamo na voljo več različnih geometrijskih primitivov

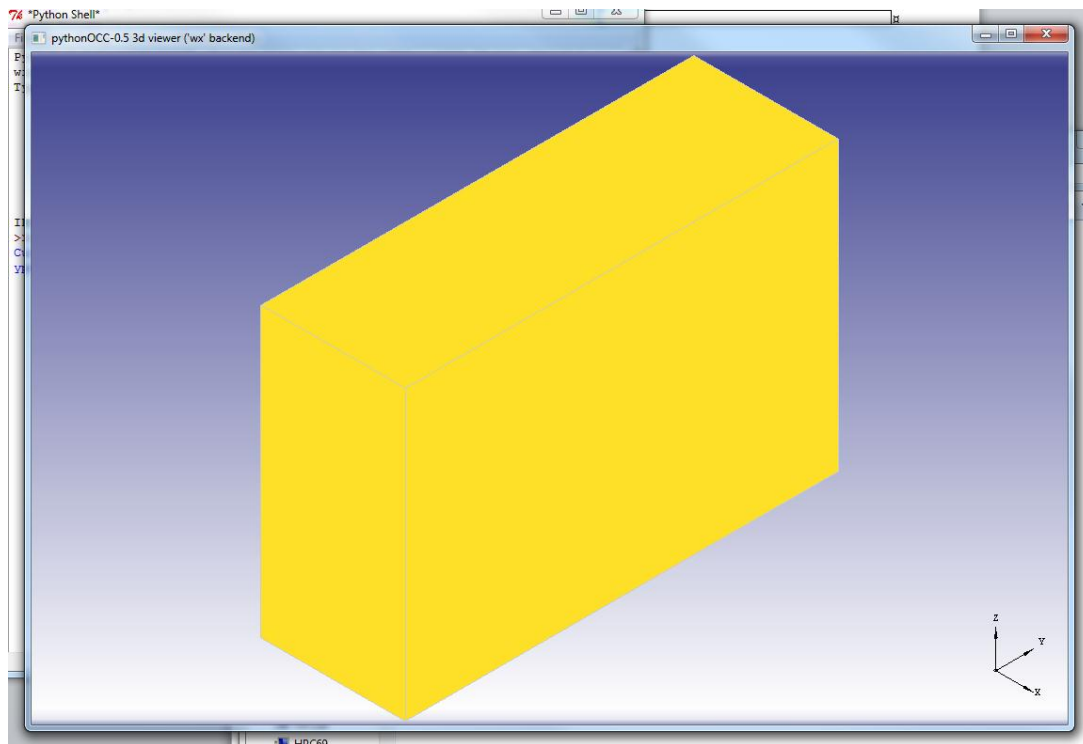
([http://opencascade.sourceforge.com/documentation/6.3.0.dfg.1/dir\\_975b4b36ea1331ec91cb6710c3f4e123.html](http://opencascade.sourceforge.com/documentation/6.3.0.dfg.1/dir_975b4b36ea1331ec91cb6710c3f4e123.html)):

- Izdelava izvleka (**BRepPrimAPI\_MakeBox**)

```
kvader = BRepPrimAPI_MakeBox(dolzina_X, dolzina_Y, dolzina_Z)  
display.DisplayShape(kvader.Shape())
```

```
kvader = BRepPrimAPI_MakeBox(Tocka1, dolzina_X, dolzina_Y, dolzina_Z)  
display.DisplayShape(kvader.Shape())
```

```
kvader = BRepPrimAPI_MakeBox(Tocka1, Tocka2)  
display.DisplayShape(kvader.Shape())
```



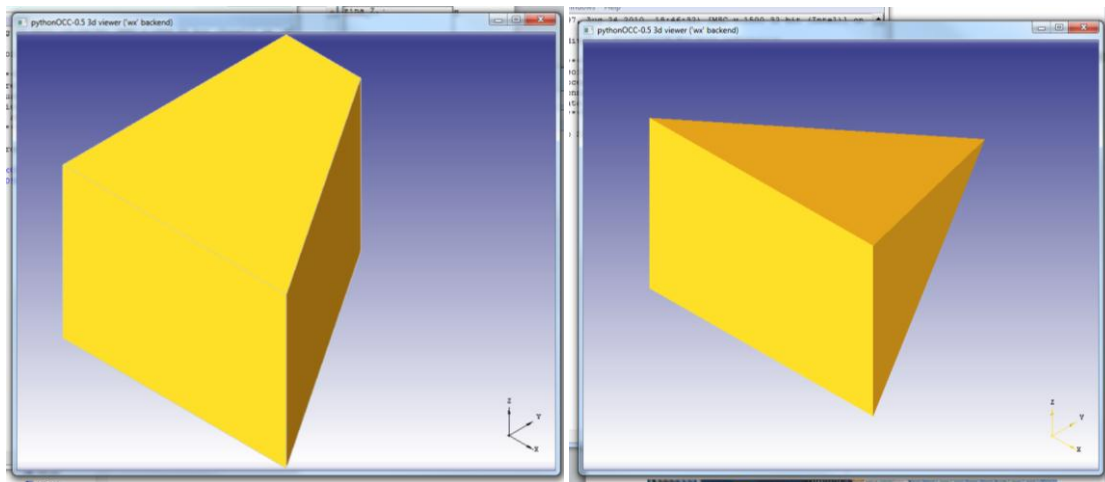
- Izdelava izvleka pod kotom nagiba (**BRepPrimAPI\_MakeWedge**)

```
kvader= BRepPrimAPI_MakeWedge(dolzina_X, dolzina_Y, dolzina_Z, dolzina_X2)
display.DisplayShape(kvader.Shape())
```

```
kvader= BRepPrimAPI_MakeWedge(dolzina_X, dolzina_Y, dolzina_Z,
dolzina_Xmin, dolzina_Zmin, dolzina_Xmax, dolzina_Zmax)
display.DisplayShape(kvader.Shape())
```

```
kvader= BRepPrimAPI_MakeWedge(dolzina_X, dolzina_Y, dolzina_Z,
dolzina_X/2, 0, dolzina_X/2, dolzina_Z)
display.DisplayShape(kvader.Shape())
```

```
kvader= BRepPrimAPI_MakeWedge(dolzina_X, dolzina_Y, dolzina_Z,
dolzina_X/2, dolzina_Z/2, dolzina_X/2, dolzina_Z/2)
display.DisplayShape(kvader.Shape())
```



- Izdelava valjastega objekta (**BRepPrimAPI\_MakeCylinder**)

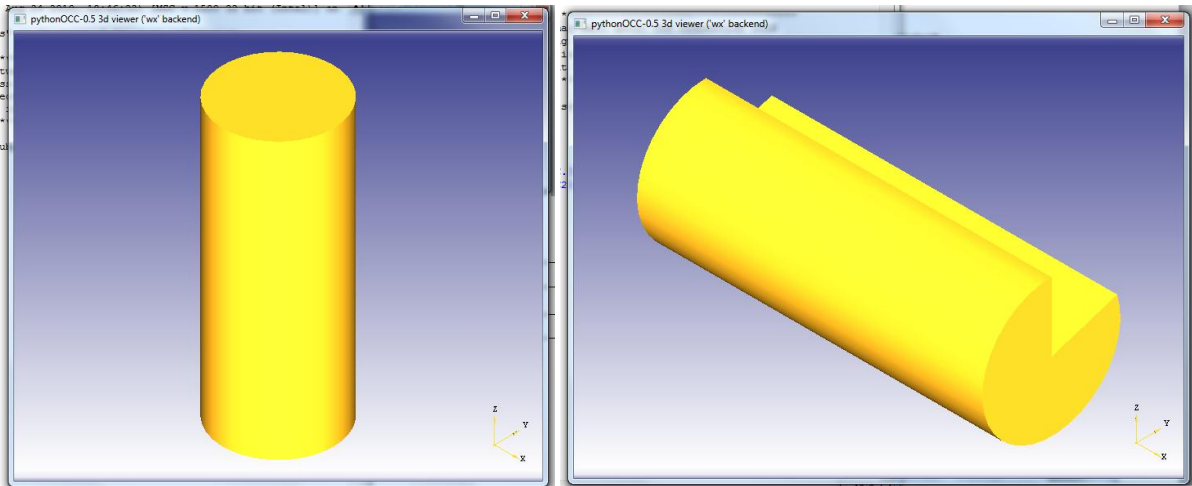
```
valj= BRepPrimAPI_MakeCylinder(Radij, Visina, Kot_radiani)
display.DisplayShape(valj.Shape())
```

```
valj= BRepPrimAPI_MakeCylinder(Os, Radij, Visina, Kot_radiani)
display.DisplayShape(valj.Shape())
```

Potrebno je definirati os vrtenja:

```
Tocka = gp_Pnt(X, Y, Z)          #definicija tocke
normala = gp_DX()                #smer normale
os = gp_Ax2(Tocka, normala)     #definicija osi
```

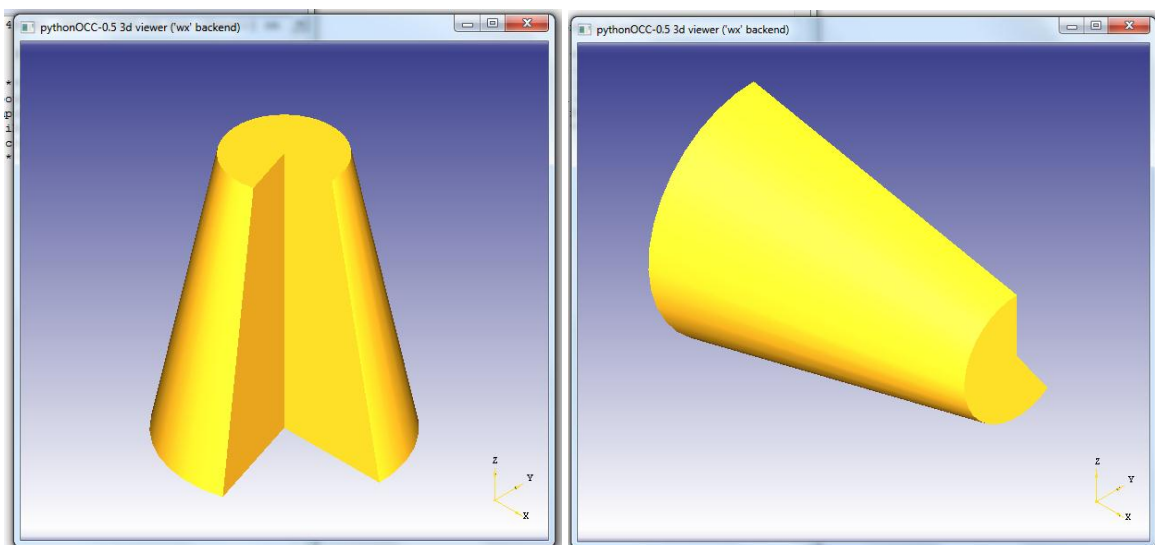




- Izdelava stožčastega objekta (**BRepPrimAPI\_MakeCone**)

```
stozec= BRepPrimAPI_MakeCone(Radij_1, Radij_2, Visina, Kot_radiani)
display.DisplayShape(stozec.Shape())
```

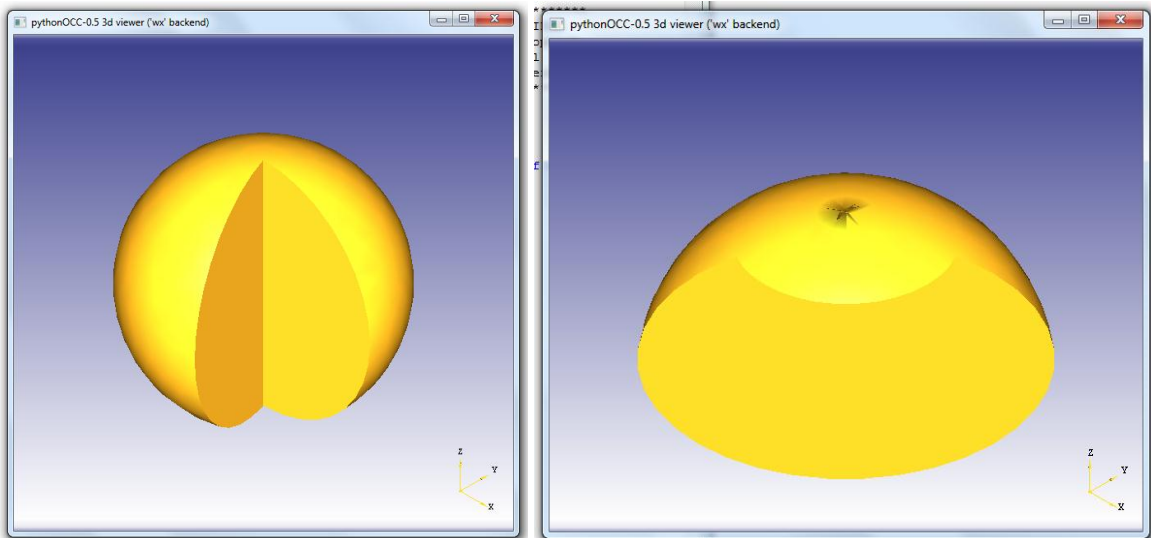
```
stozec= BRepPrimAPI_MakeCone(Os, Radij_1, Radij_2, Visina, Kot_radiani)
display.DisplayShape(stozec.Shape())
```



- Izdelava sfere (**BRepPrimAPI\_MakeSphere**)

```
sphere= BRepPrimAPI_MakeSphere(Radij, Kot_radiani)
display.DisplayShape(sphere.Shape())
```

```
sphere= BRepPrimAPI_MakeSphere(Radij, Kot_a1_rad, Kot_a2_rad)
display.DisplayShape(sphere.Shape())
```

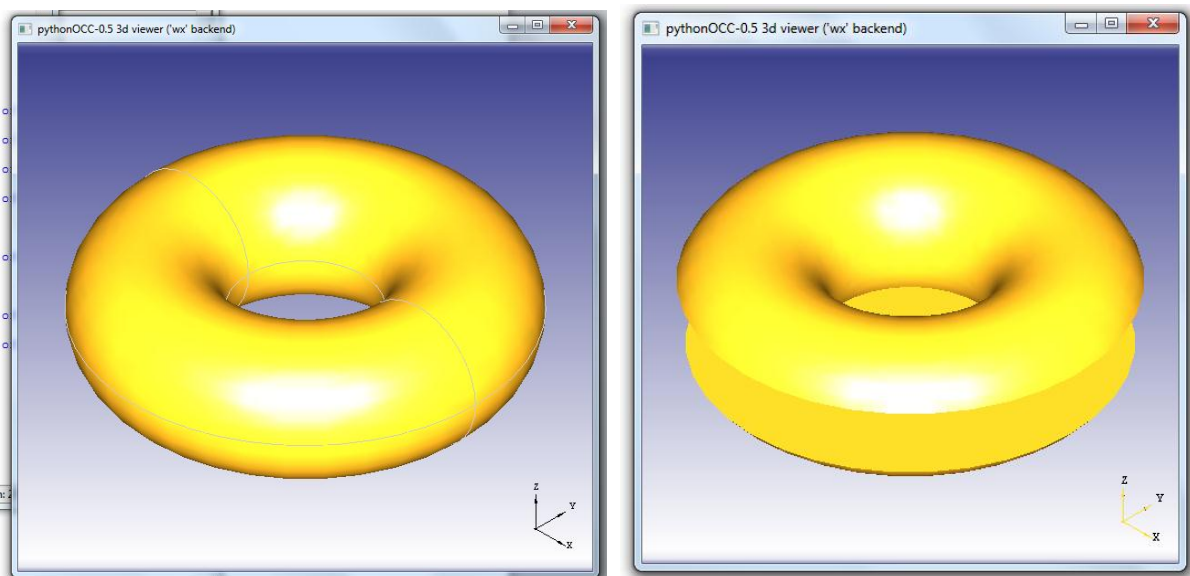


- Izdelava torusa (**BRepPrimAPI\_MakeTorus**)

```
torus= BRepPrimAPI_MakeTorus(Radij_torusa, Radij_prereza)
display.DisplayShape(torus.Shape())
```

```
torus= BRepPrimAPI_MakeTorus(Radij_torusa, Radij_prereza, Kot_radiani)
display.DisplayShape(torus.Shape())
```

```
torus= BRepPrimAPI_MakeTorus(Radij_torusa, Radij_prereza, Kot_a1_rad,
Kot_a2_rad)
display.DisplayShape(torus.Shape())
```



- Izdelava izvleka (**BRepPrimAPI\_MakePrism**)

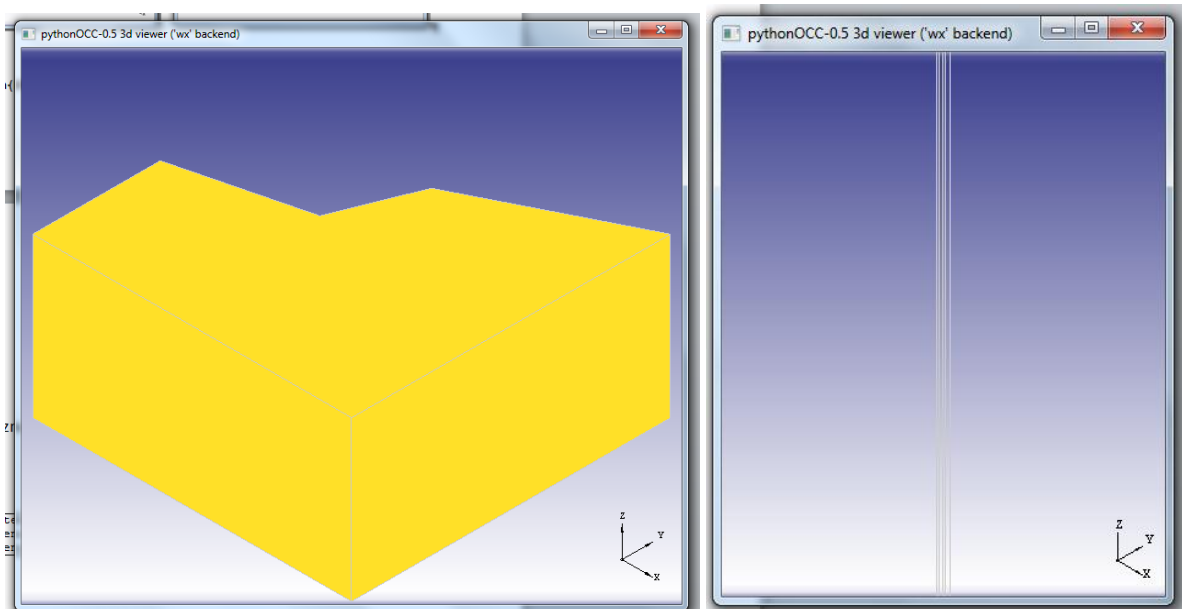
```
Izvlek = BRepPrimAPI_MakePrism(Povrsina.Face() , Vektor1)
display.DisplayShape(Izvlek.Shape())
```

```
Izvlek = BRepPrimAPI_MakePrism(Povrsina.Face() , Smer*)
display.DisplayShape(Izvlek.Shape())
```

\*neskončno visoki izvlek

Rabimo:

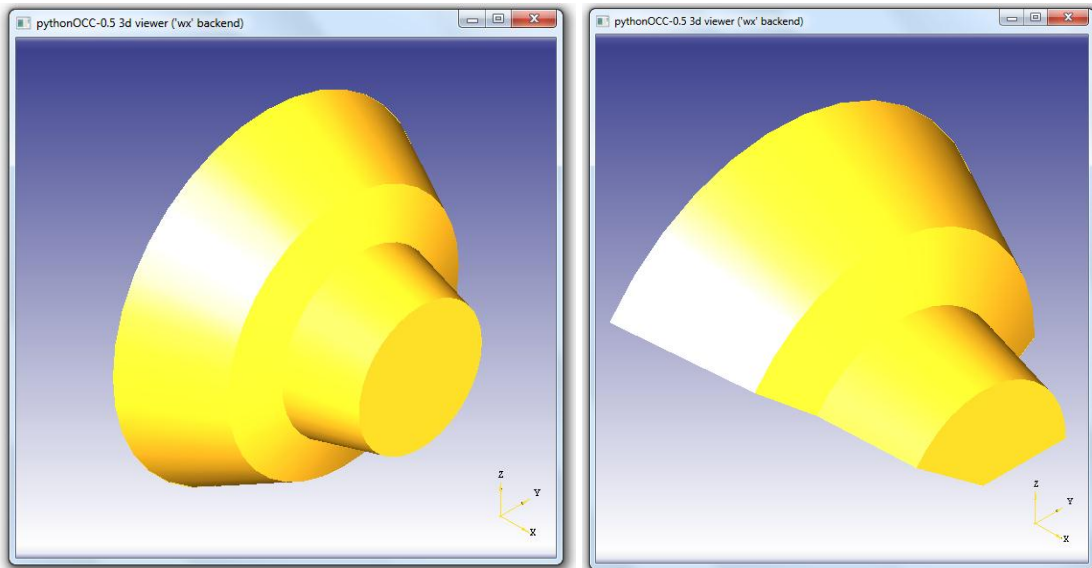
- Površino
- Smerni vektor (in njegovo dolžino)



- Izdelava kroznega izvleka(vrtenina) (**BRepPrimAPI\_MakeRevol**)

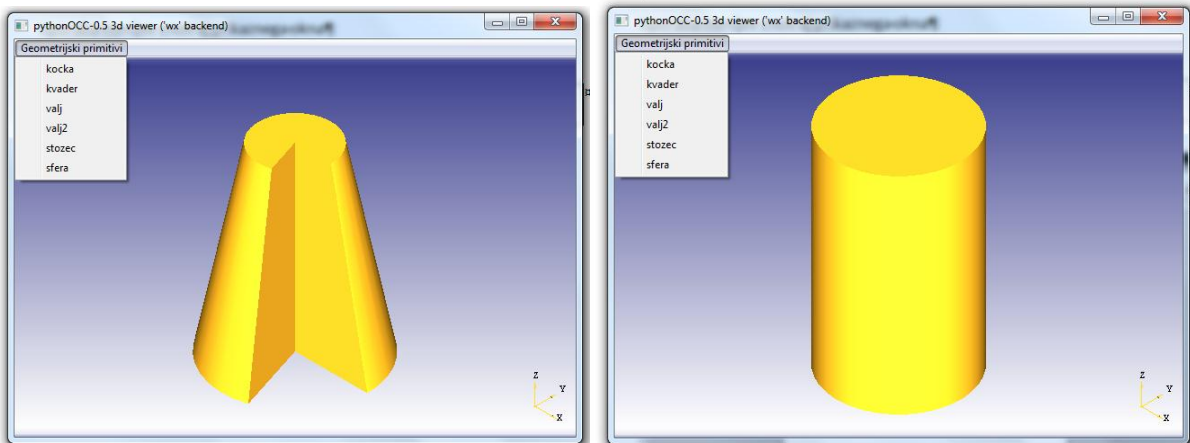
```
Vrtenina = BRepPrimAPI_MakeRevol(Povrsina.Face() , Os)
display.DisplayShape(Vrtenina.Shape())
```

```
Vrtenina = BRepPrimAPI_MakeRevol(Povrsina.Face() , Os, Kot_radiani)
display.DisplayShape(Vrtenina.Shape())
```



Uporaba menijev znotraj prikaznega okna

```
add_menu('Ime_menija')  
add_function_to_menu('Ime_menija',Funkcija1)  
add_function_to_menu('Ime_menija',Funkcija2)
```



## Naslednjič: Boolova Algebra

Pripravil: as. Pavel Tomšič