

# VPython - symulacje fizyczne z grafiką 3D dla każdego

## wykład 9

Dr hab. Adam Bzdak, prof. AGH

# scena

```
vp_scena_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_9\vp_scena_1.py (2.7.12)
File Edit Format Run Options Window Help
from visual import *

scene = display()

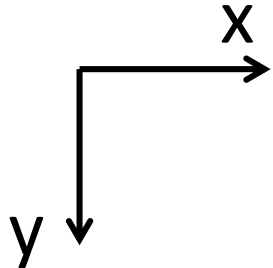
scene.title = 'my object'

scene.width = 550
scene.height = 580

scene.x = 150      pozycja w pikselach od górnego
scene.y = 50        lewego rogu ekranu

scene.range = 2

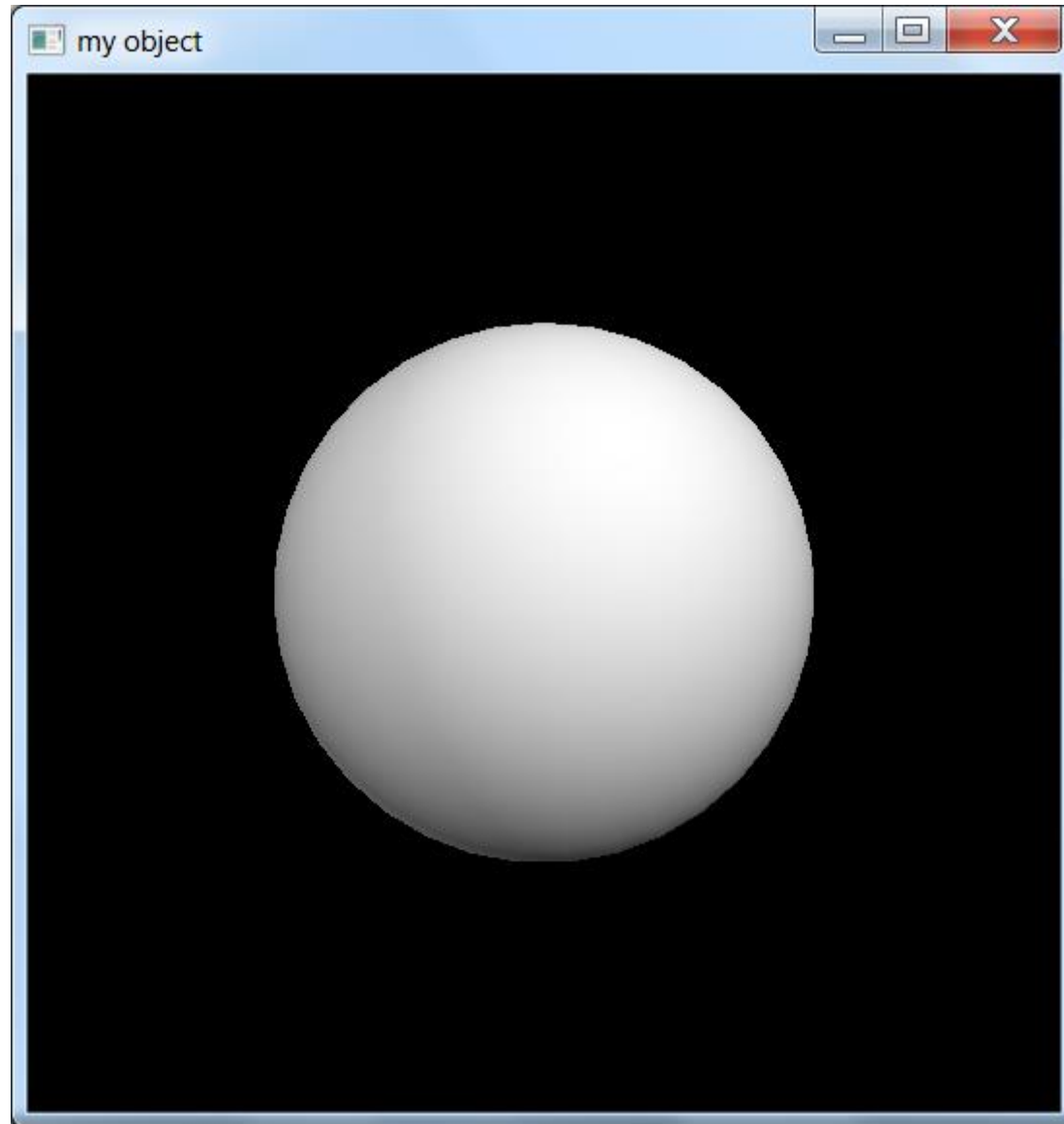
ball = sphere()
```



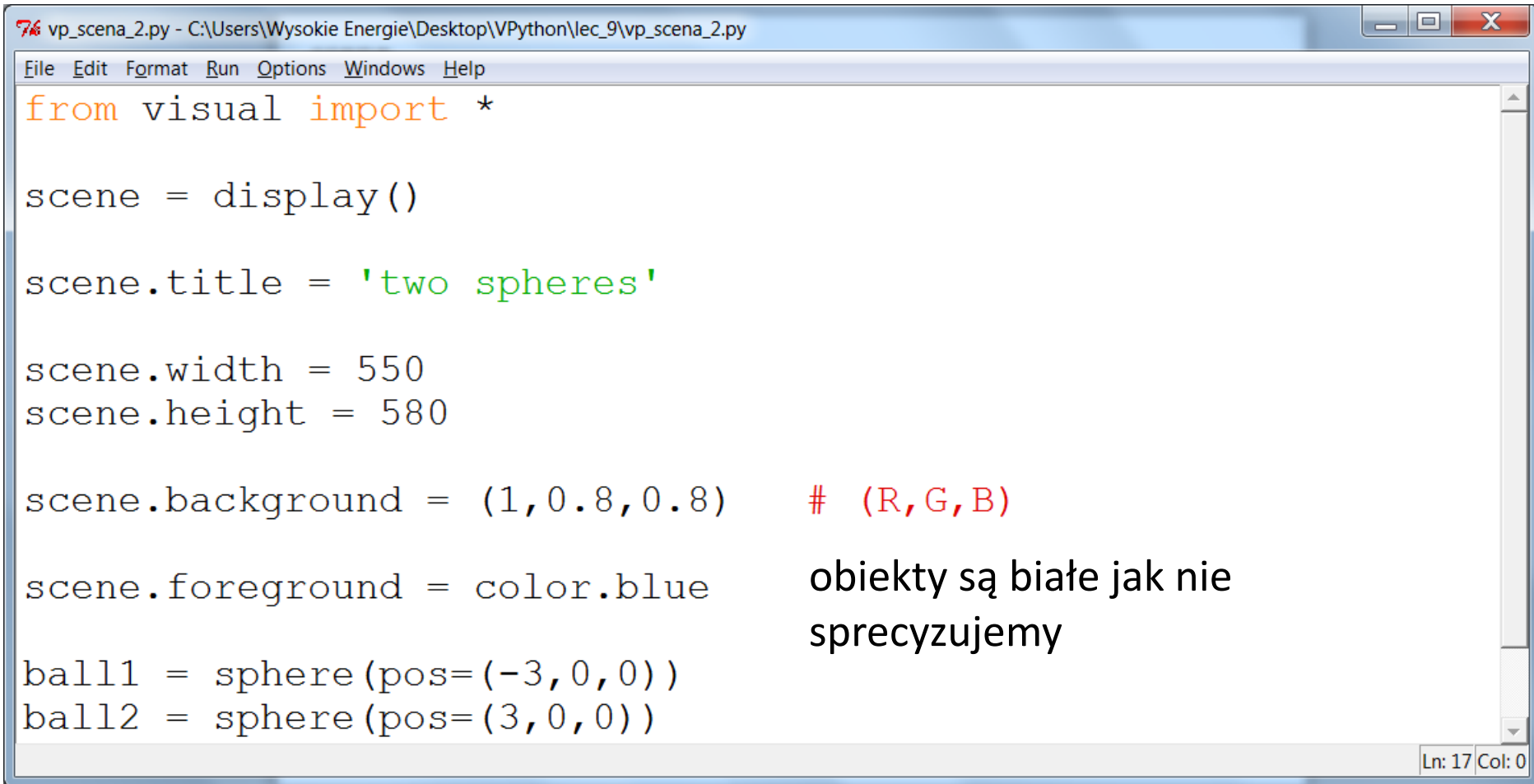
Ln: 16 Col: 0

Proszę pamiętać o różnicach pomiędzy VPython 6 i VPython 7.

tytuł



# scena



The image shows a screenshot of a Python IDE window titled "vp\_scena\_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_scena\_2.py". The window contains a Python script for creating a 3D scene using VPython. The script defines a scene, sets its title to "two spheres", and specifies its width and height. It also sets the background color to (1, 0.8, 0.8) and the foreground color to blue. Finally, it creates two spheres, ball1 and ball2, at positions (-3, 0, 0) and (3, 0, 0) respectively. A comment in the script indicates that the objects are white and will be specified later.

```
76 vp_scena_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_9\vp_scena_2.py
File Edit Format Run Options Windows Help
from visual import *

scene = display()

scene.title = 'two spheres'

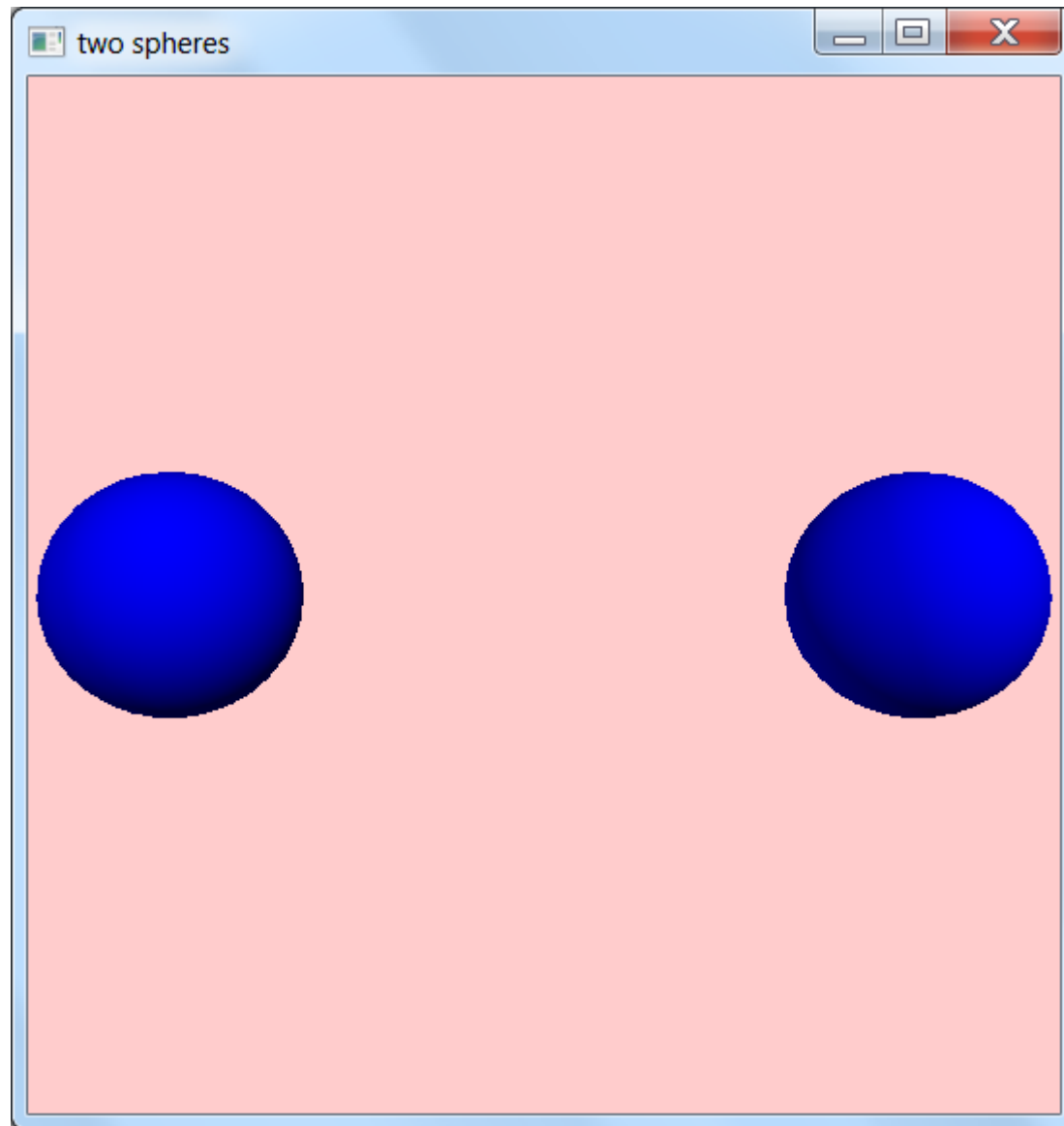
scene.width = 550
scene.height = 580

scene.background = (1,0.8,0.8)    # (R,G,B)

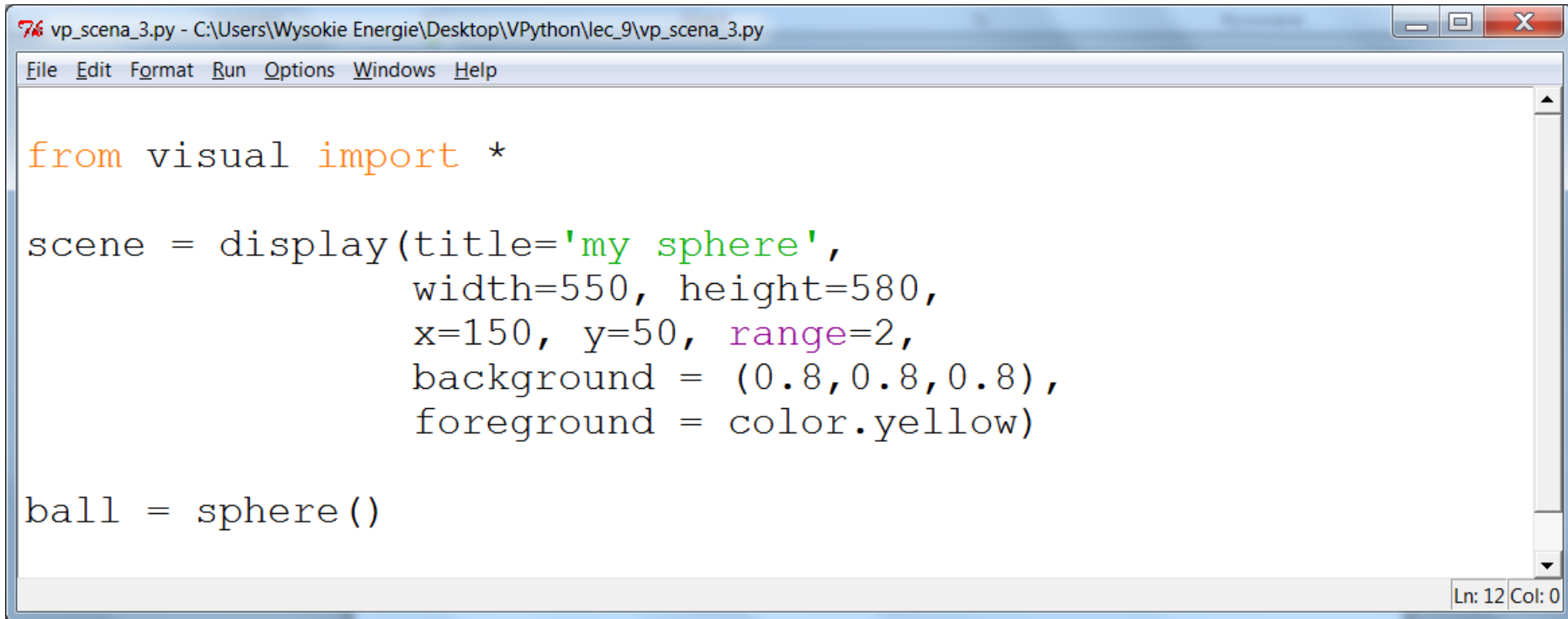
scene.foreground = color.blue     obiekty są białe jak nie
                                   sprecyzujemy

ball1 = sphere(pos=(-3,0,0))
ball2 = sphere(pos=(3,0,0))

Ln: 17 Col: 0
```



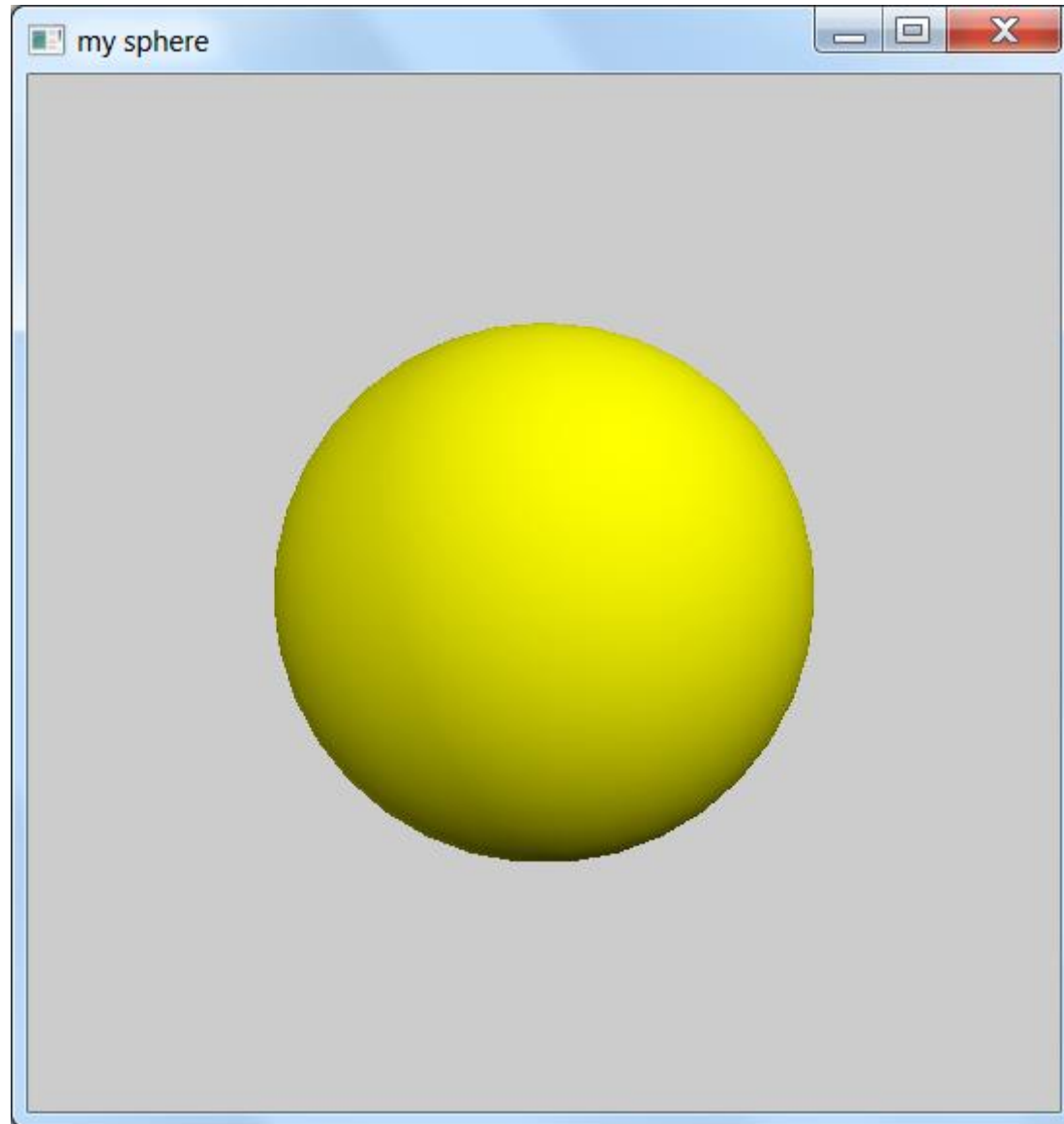
# scena



The image shows a screenshot of a Python IDE window. The title bar reads "vp\_scena\_3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_scena\_3.py". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following Python code:

```
from visual import *  
  
scene = display(title='my sphere',  
                width=550, height=580,  
                x=150, y=50, range=2,  
                background = (0.8,0.8,0.8),  
                foreground = color.yellow)  
  
ball = sphere()
```

The status bar at the bottom right indicates "Ln: 12 Col: 0".



## dwie sceny

```
74 vp_two_scenes.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_9\vp_two_scenes.py
File Edit Format Run Options Windows Help
from visual import *

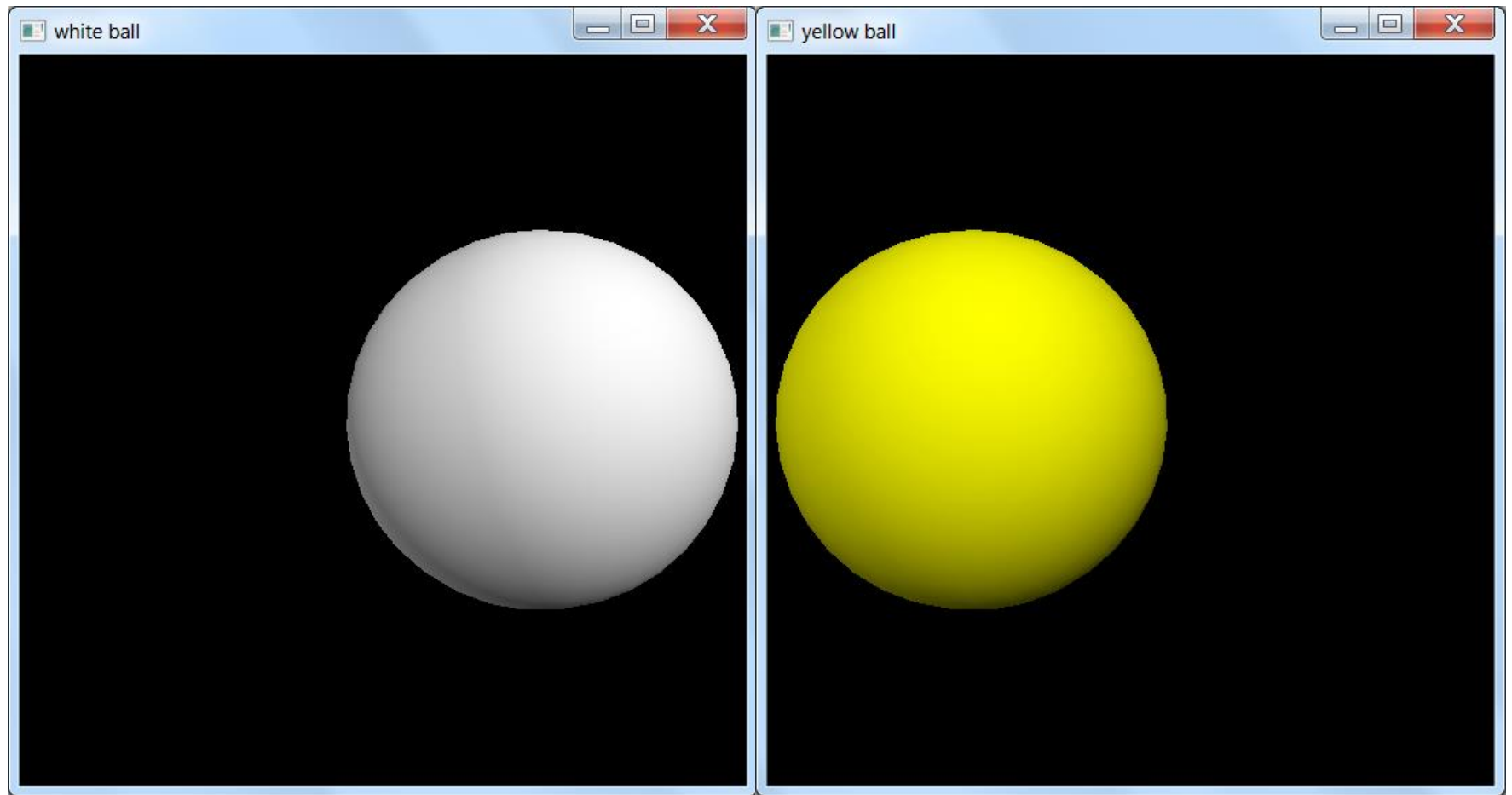
scene1 = display(title='white ball', width=550, height=580,
                  x=10, y=10, range=2)

scene2 = display(title='yellow ball', width=550, height=580,
                  x=561, y=10, range=2)

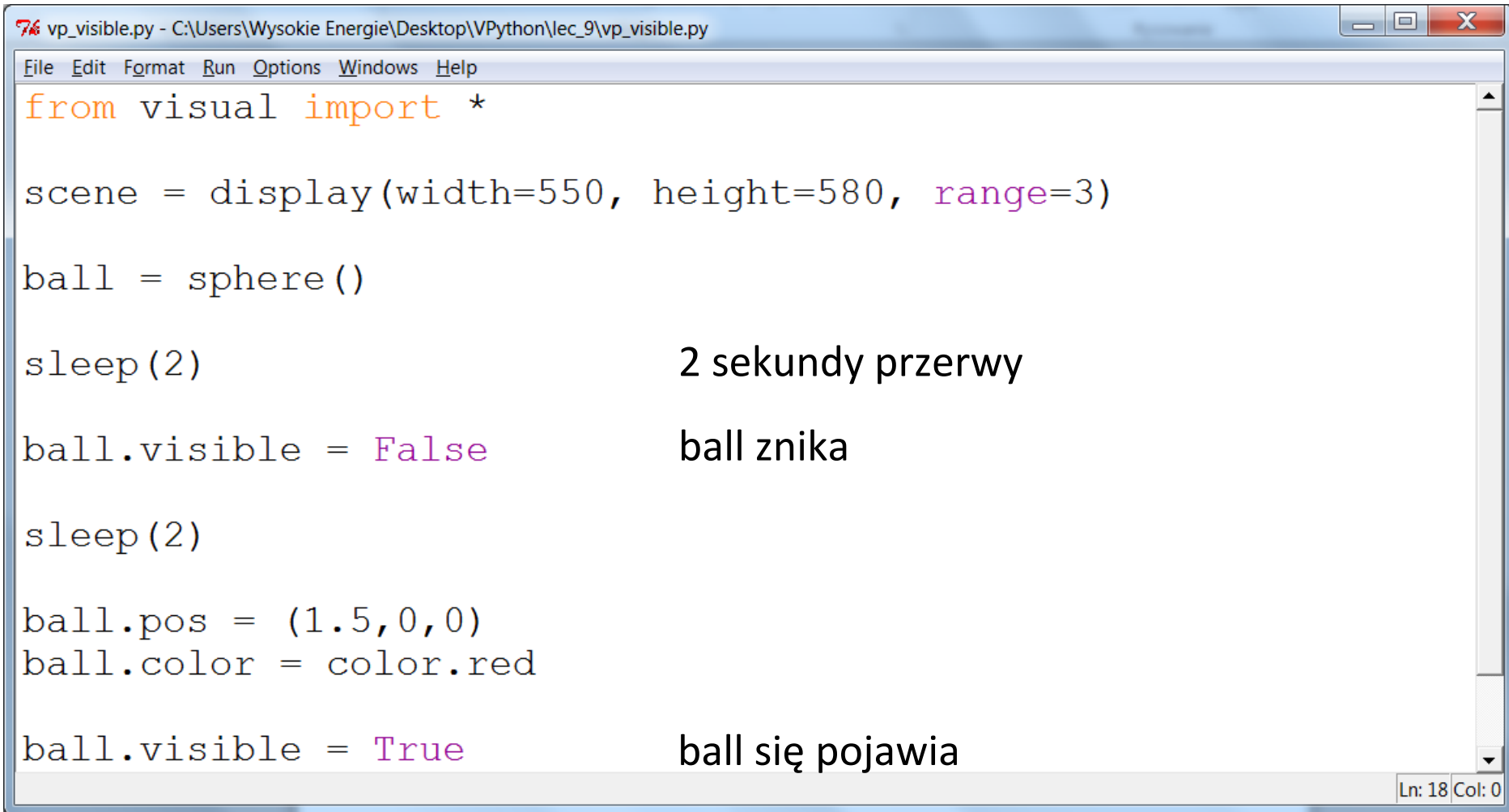
ball1 = sphere(display=scene1)
ball2 = sphere(display=scene2, color=color.yellow)

t = 0
dt = 0.01
while t<0.8:
    rate(100)
    ball1.pos += dt*vector(1,0,0)
    ball2.pos -= dt*vector(1,0,0)
    t += dt
Ln: 21 Col: 0
```





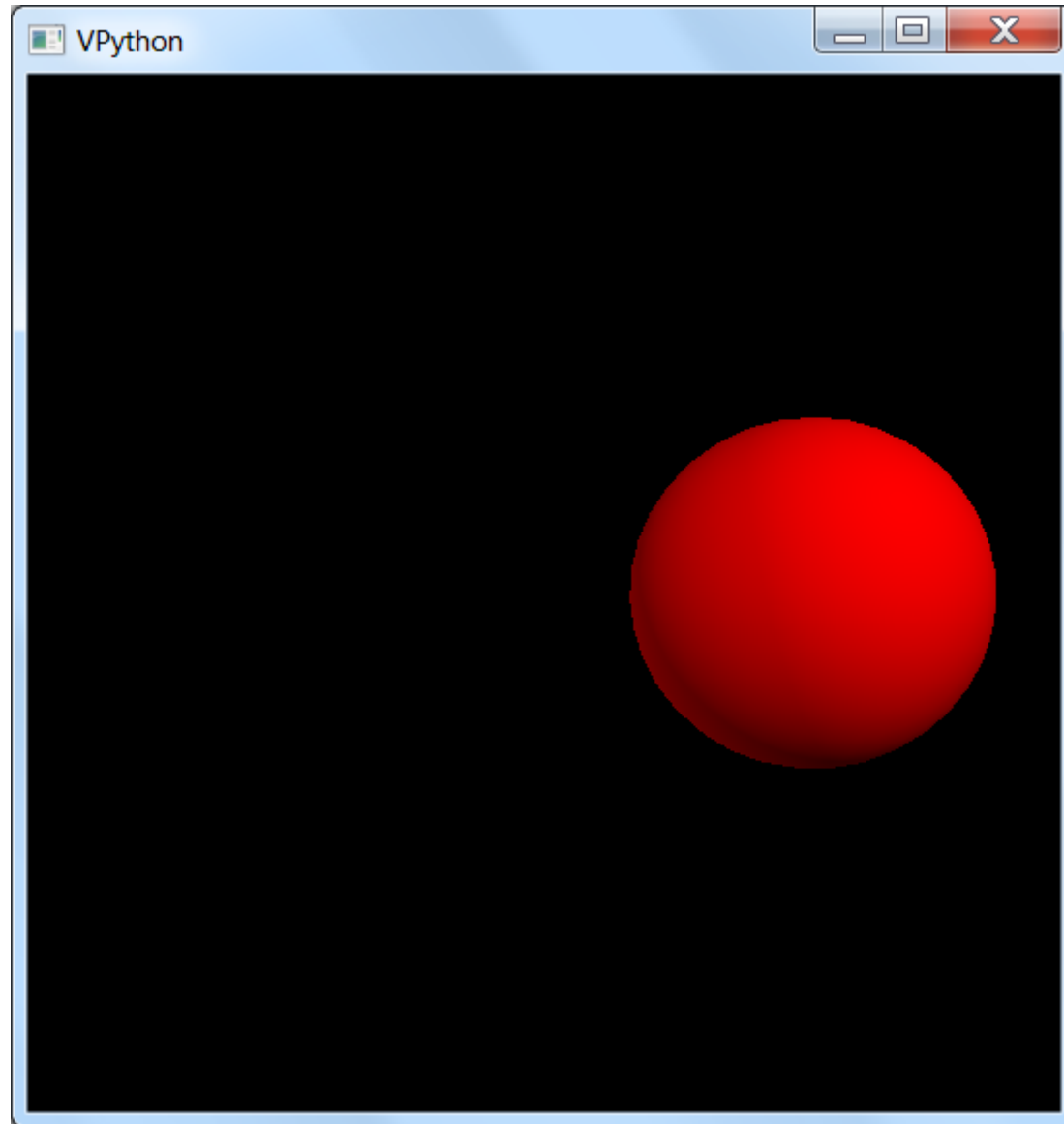
# widoczność, sleep()



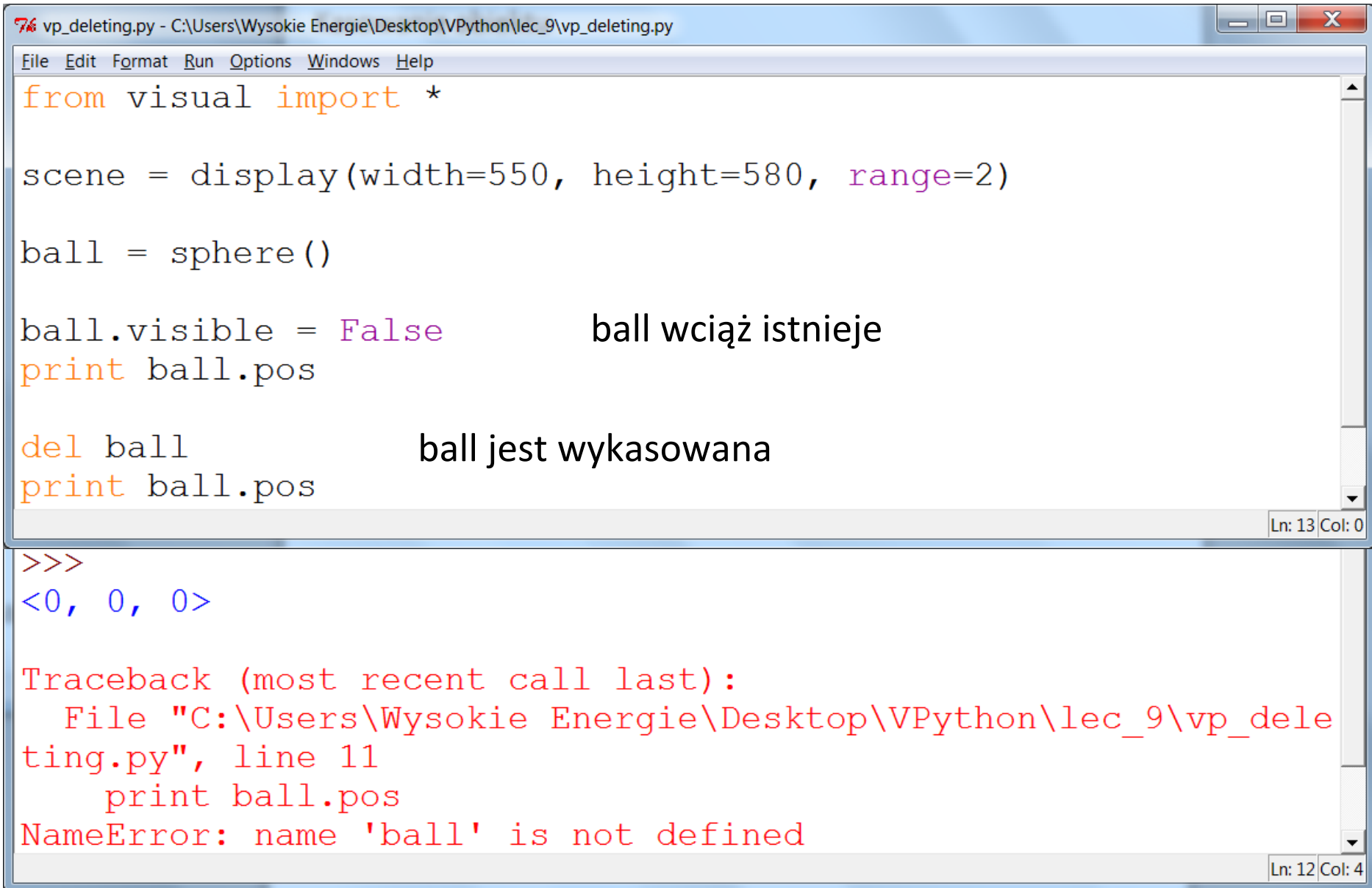
The image shows a screenshot of a Python IDE window titled "vp\_visible.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_visible.py". The window contains a Python script that creates a 3D scene and manipulates a ball's visibility. The script is as follows:

```
from visual import *  
  
scene = display(width=550, height=580, range=3)  
  
ball = sphere()  
  
sleep(2) 2 sekundy przerwy  
  
ball.visible = False ball znika  
  
sleep(2)  
  
ball.pos = (1.5, 0, 0)  
ball.color = color.red  
  
ball.visible = True ball się pojawia
```

The IDE window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The status bar at the bottom right shows "Ln: 18 Col: 0".



# Kasowanie obiektów



The image shows a screenshot of a Python IDE window titled "vp\_deleting.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_deleting.py". The window contains a Python script with the following code:

```
from visual import *  
  
scene = display(width=550, height=580, range=2)  
  
ball = sphere()  
  
ball.visible = False  
print ball.pos  
  
del ball  
print ball.pos
```

Annotations in the image explain the state of the object:

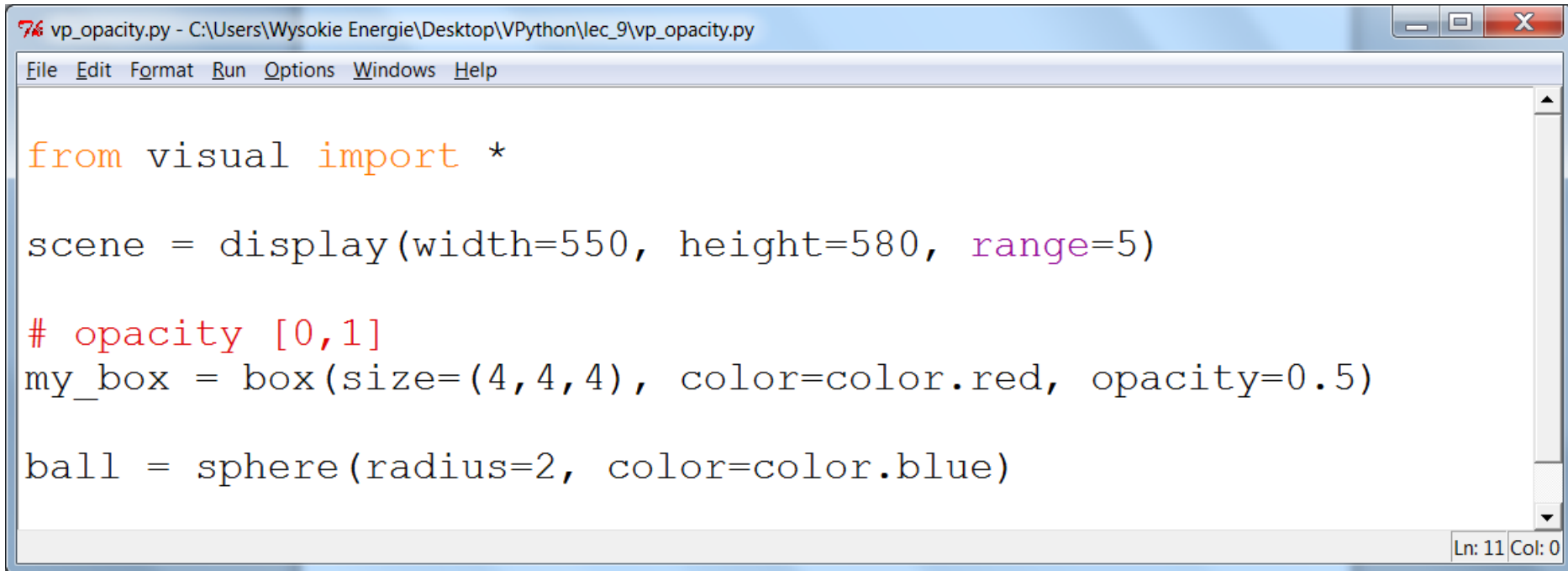
- "ball wciąż istnieje" (ball still exists) is placed next to the line `ball.visible = False`.
- "ball jest wykasowana" (ball is being deleted) is placed next to the line `del ball`.

Below the script, the output of the program is shown in a separate window. It displays the initial position of the ball and then a traceback error:

```
>>>  
<0, 0, 0>  
  
Traceback (most recent call last):  
  File "C:\Users\Wysokie Energie\Desktop\VPython\lec_9\vp_deleting.py", line 11  
    print ball.pos  
NameError: name 'ball' is not defined
```

The error message indicates that the variable `ball` is no longer defined after the `del` statement, which causes the `print ball.pos` statement to fail.

# Przeźroczystość (opacity)



```
76 vp_opacity.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_9\vp_opacity.py
File Edit Format Run Options Windows Help

from visual import *

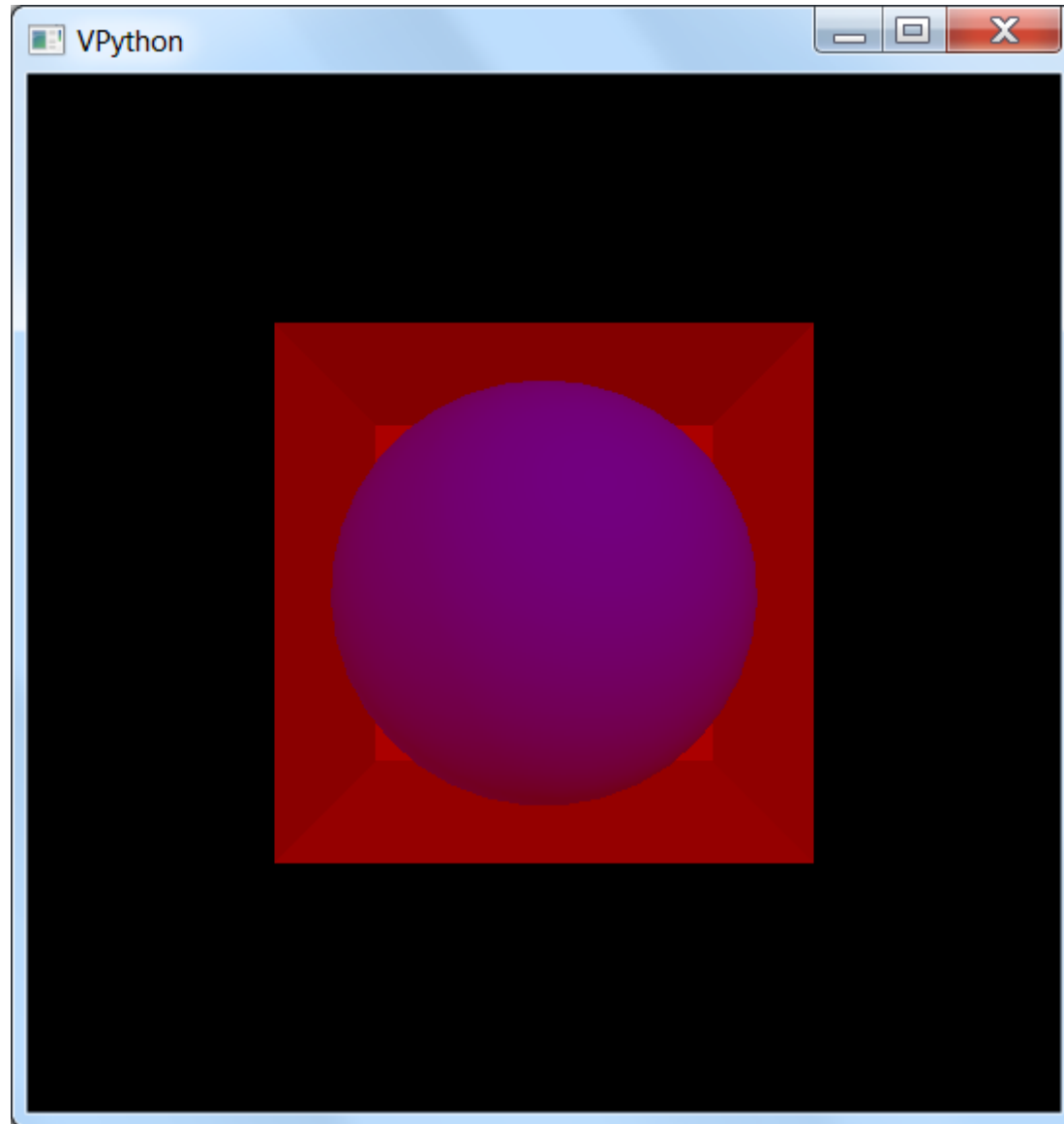
scene = display(width=550, height=580, range=5)

# opacity [0,1]
my_box = box(size=(4,4,4), color=color.red, opacity=0.5)

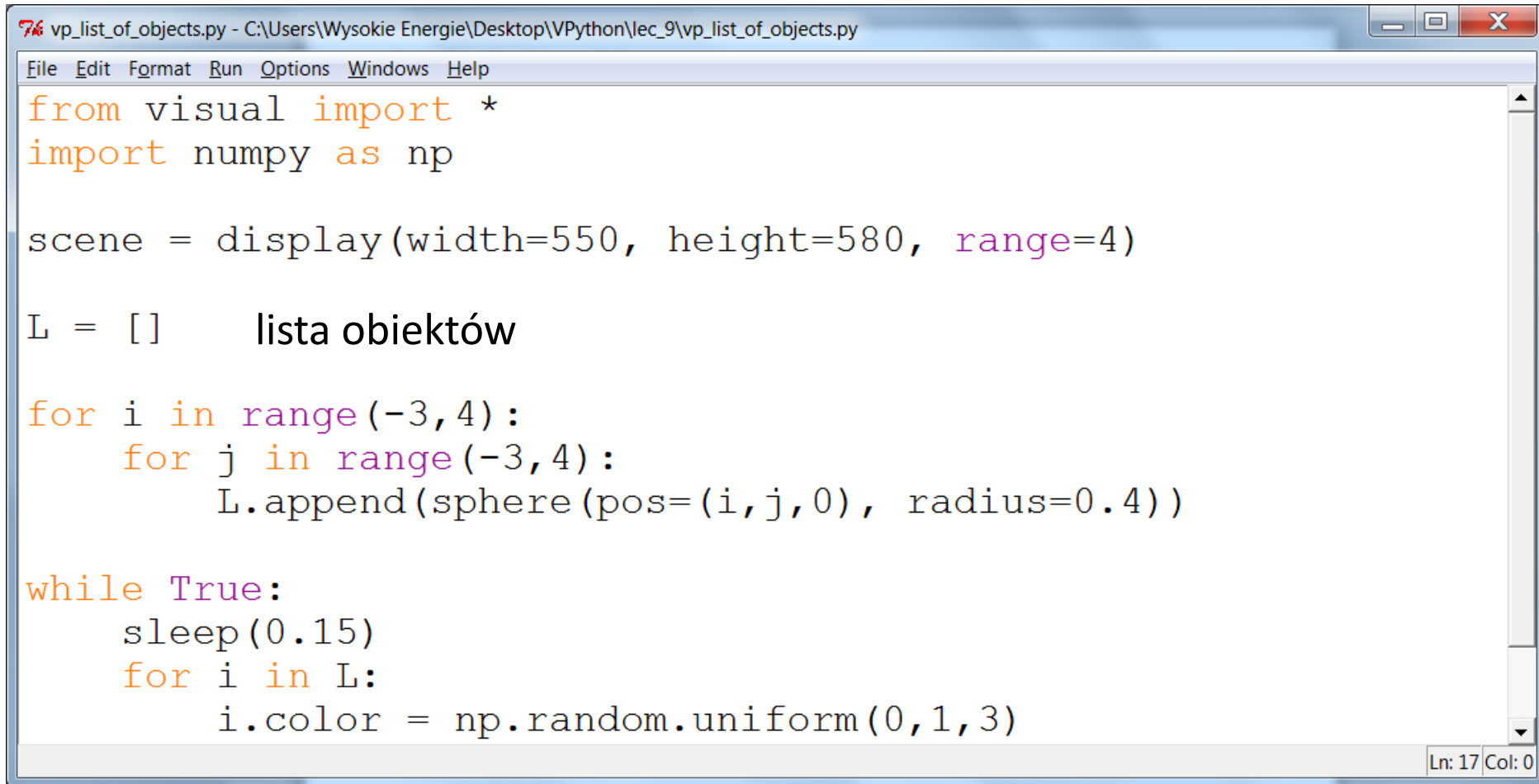
ball = sphere(radius=2, color=color.blue)
```

Ln: 11 Col: 0

Są problemy jak obiekty z  $\text{opacity} < 1$  się przekrywają



# Obiekty w liście



```
vp_list_of_objects.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_9\vp_list_of_objects.py
File Edit Format Run Options Windows Help

from visual import *
import numpy as np

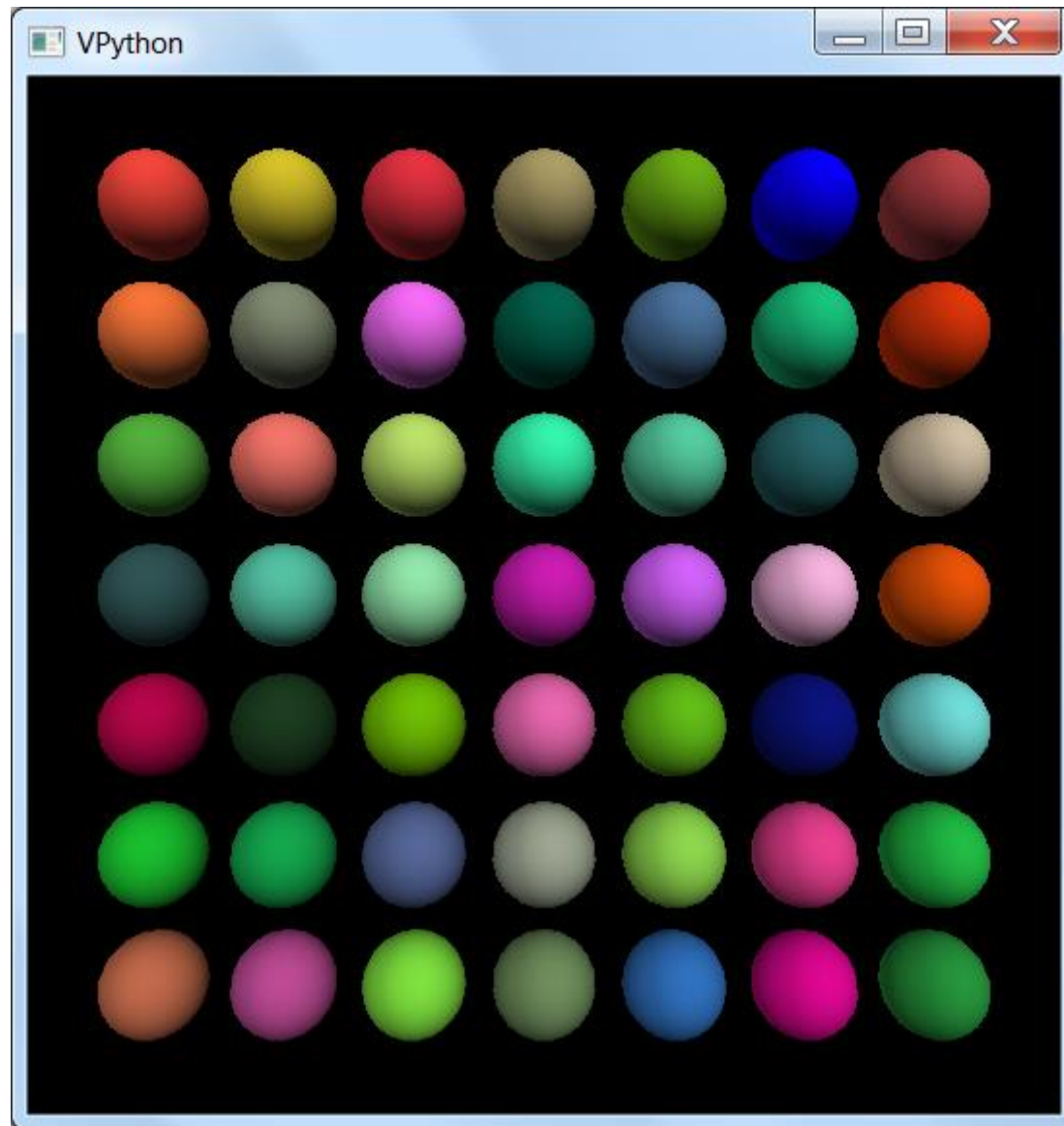
scene = display(width=550, height=580, range=4)

L = []      lista obiektów

for i in range(-3, 4):
    for j in range(-3, 4):
        L.append(sphere(pos=(i, j, 0), radius=0.4))

while True:
    sleep(0.15)
    for i in L:
        i.color = np.random.uniform(0, 1, 3)
```

Ln: 17 Col: 0





# wektory

vp\_vector\_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_vector\_1.py

File Edit Format Run Options Windows Help

```
from visual import *
```

```
v1 = vector(1,0,0)
```

```
v2 = vector(0,1,0)
```

```
v = v1 + v2
```

```
print v
```

```
print v.x, v.y, v.z
```

```
print mag(v)      # |v|
```

```
print mag2(v)     # |v|**2
```

```
print norm(v)     # v/|v|
```

Ln: 7 Col: 0

```
<1, 1, 0>
```

```
1.0 1.0 0.0
```

```
1.41421356237
```

```
2.0
```

```
<0.707107, 0.707107, 0>
```

Ln: 12 Col: 4

# wektory

vp\_vector\_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_vector\_2.py

File Edit Format Run Options Windows Help

```
from visual import *
```

```
v1 = vector(1,0,0)
```

```
v2 = vector(0,1,0)
```

```
print dot(v1,v2)
```

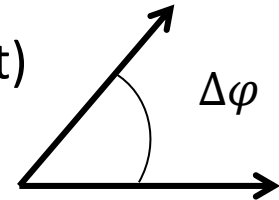
Iloczyn skalarny (dot product)

```
print cross(v1,v2)
```

Iloczyn wektorowy (cross product)

```
print diff_angle(v1,v2)
```

w radianach



Ln: 12 Col: 0

```
>>>
```

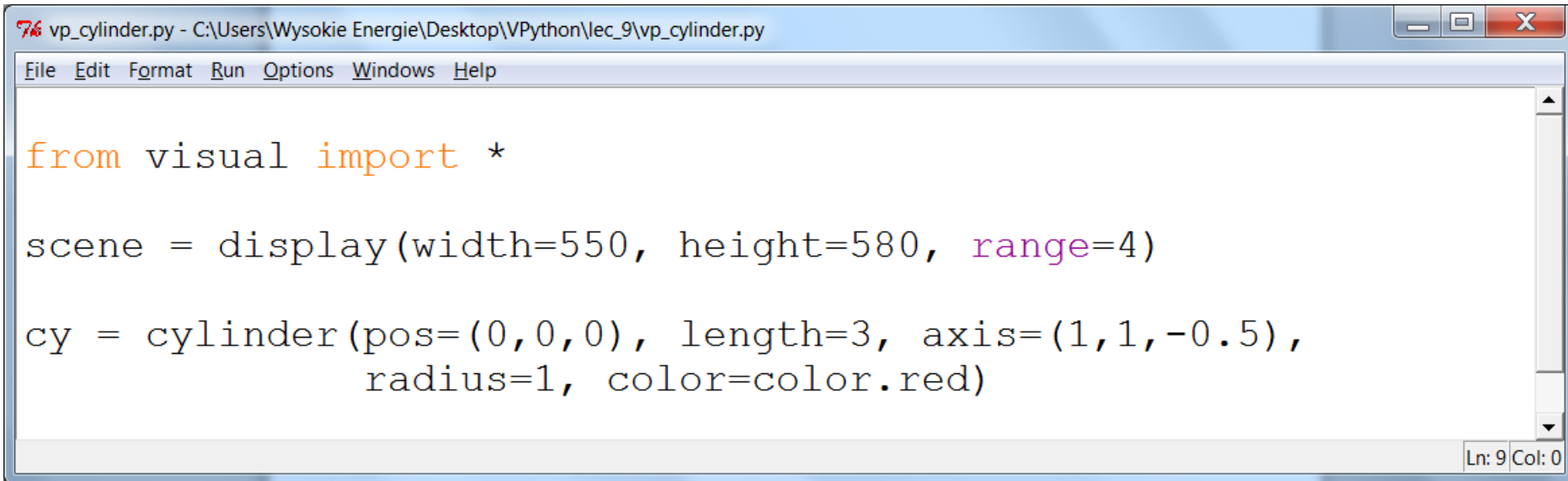
```
0.0
```

```
<0, 0, 1>
```

```
1.57079632679
```

Ln: 10 Col: 4

# cylinder

A screenshot of a Python IDE window titled 'vp\_cylinder.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_cylinder.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following Python code:

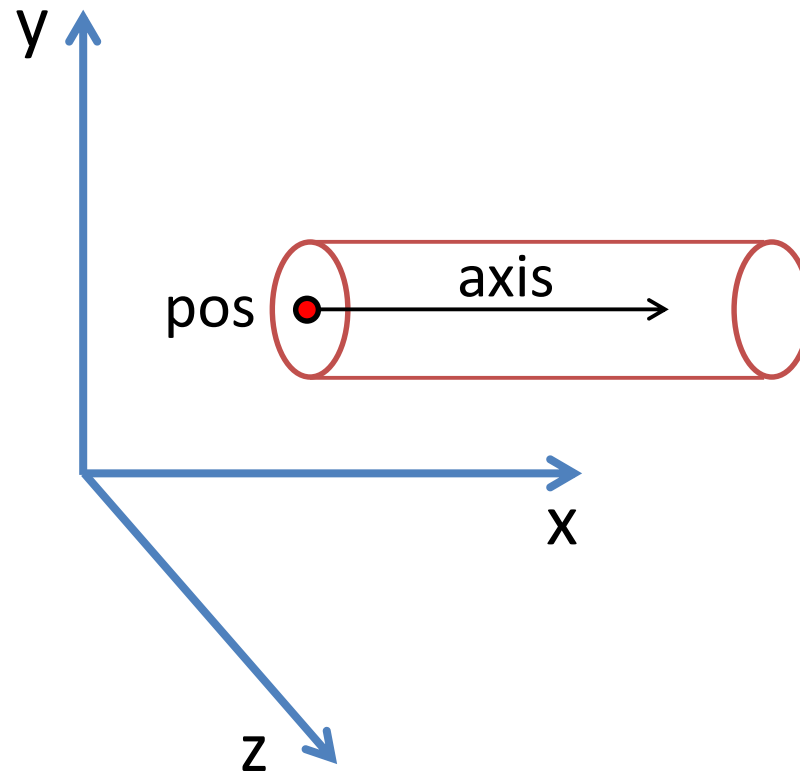
```
from visual import *  
  
scene = display(width=550, height=580, range=4)  
  
cy = cylinder(pos=(0,0,0), length=3, axis=(1,1,-0.5),  
              radius=1, color=color.red)
```

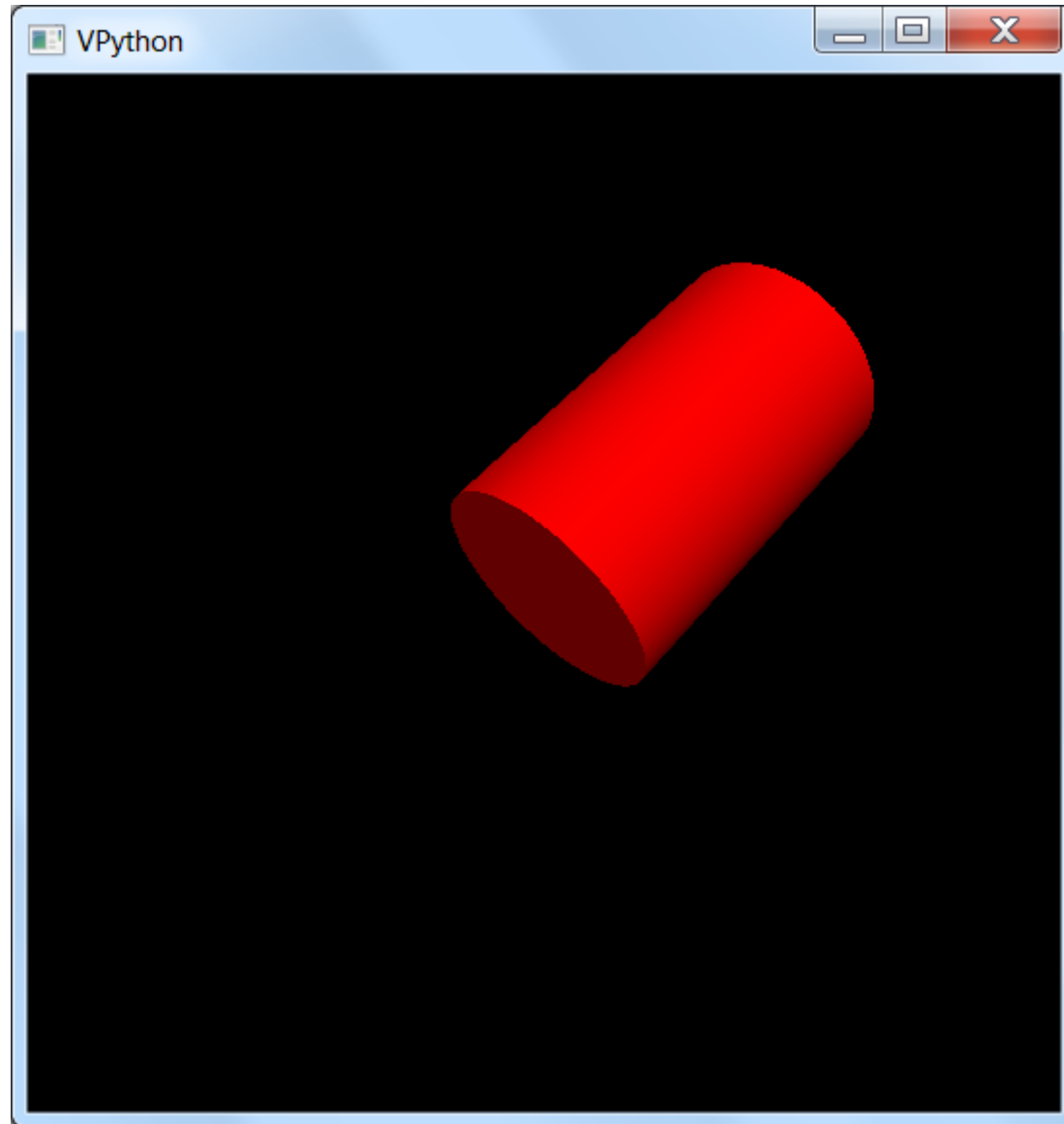
The status bar at the bottom right shows 'Ln: 9 Col: 0'.

jeśli length nie jest sprecyzowany to  $\text{length} = \text{mag}(\text{axis})$

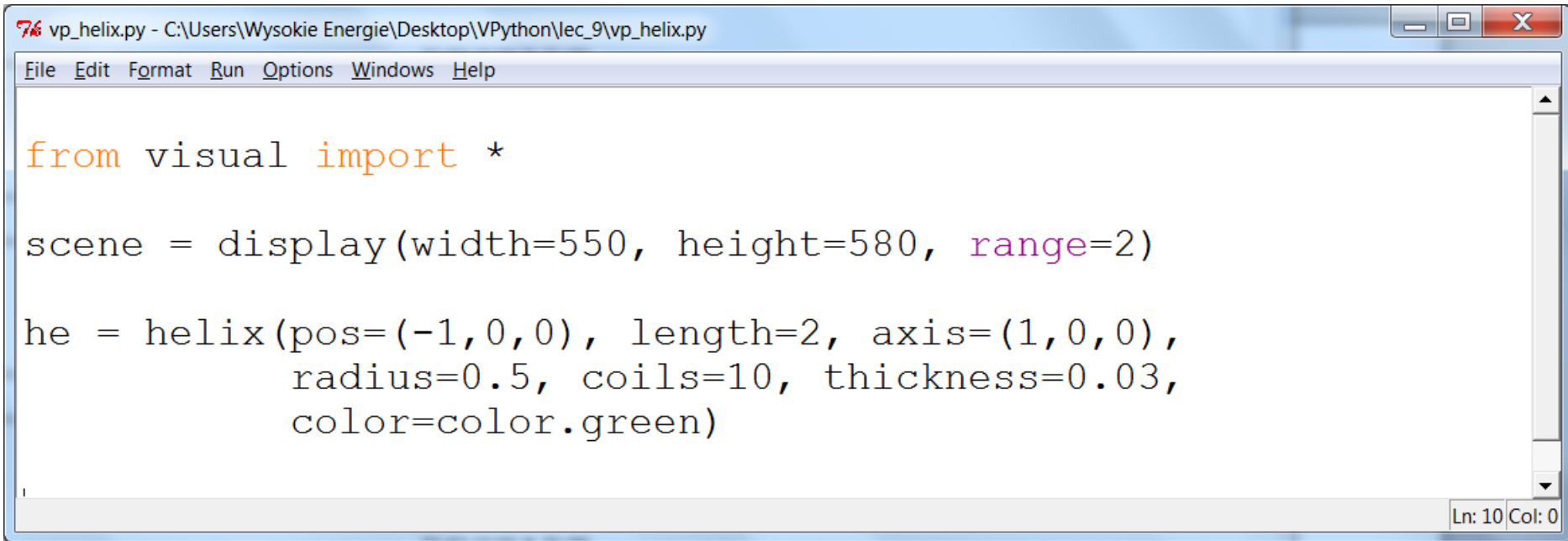
axis można też dodać do `box(...)`

cylinder, pos, axis





# Sprężynka (helix)

A screenshot of a Python IDE window titled 'vp\_helix.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_helix.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following Python code:

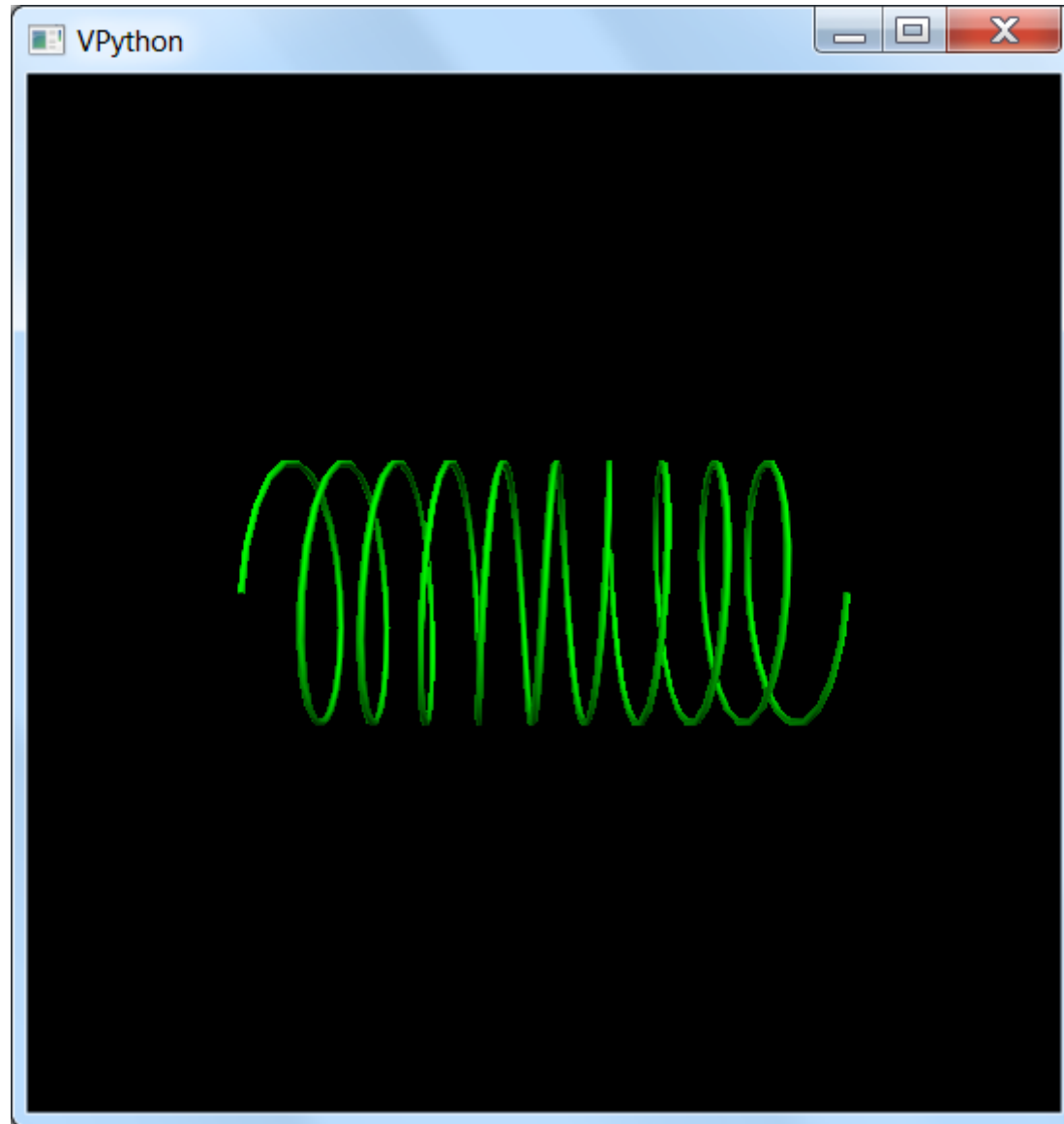
```
from visual import *  
  
scene = display(width=550, height=580, range=2)  
  
he = helix(pos=(-1,0,0), length=2, axis=(1,0,0),  
           radius=0.5, coils=10, thickness=0.03,  
           color=color.green)
```

The status bar at the bottom right shows 'Ln: 10 Col: 0'.

pos, length i axis jak dla cylindra

automatycznie jest coils = 5 i thickness = radius/20

nie można użyć opacity dla helix



# Symulacja kulka - sprężyna - kulka

76 vp\_helix\_sphere.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_9\vp\_helix\_sphere.py

File Edit Format Run Options Windows Help

```
from visual import *

scene = display(width=550, height=580, range=4)

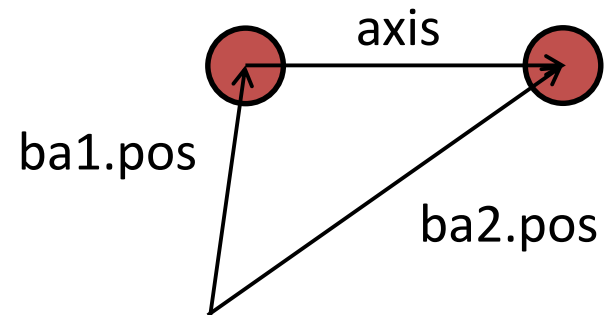
ba1 = sphere(pos=(-2,0,0), radius=0.5, color=color.blue)
ba2 = sphere(pos=(2,0,0), radius=0.5, color=color.blue)

he = helix(pos=ba1.pos, axis=ba2.pos-ba1.pos,
           radius=0.5, coils=10, thickness=0.05,
           color=color.green)

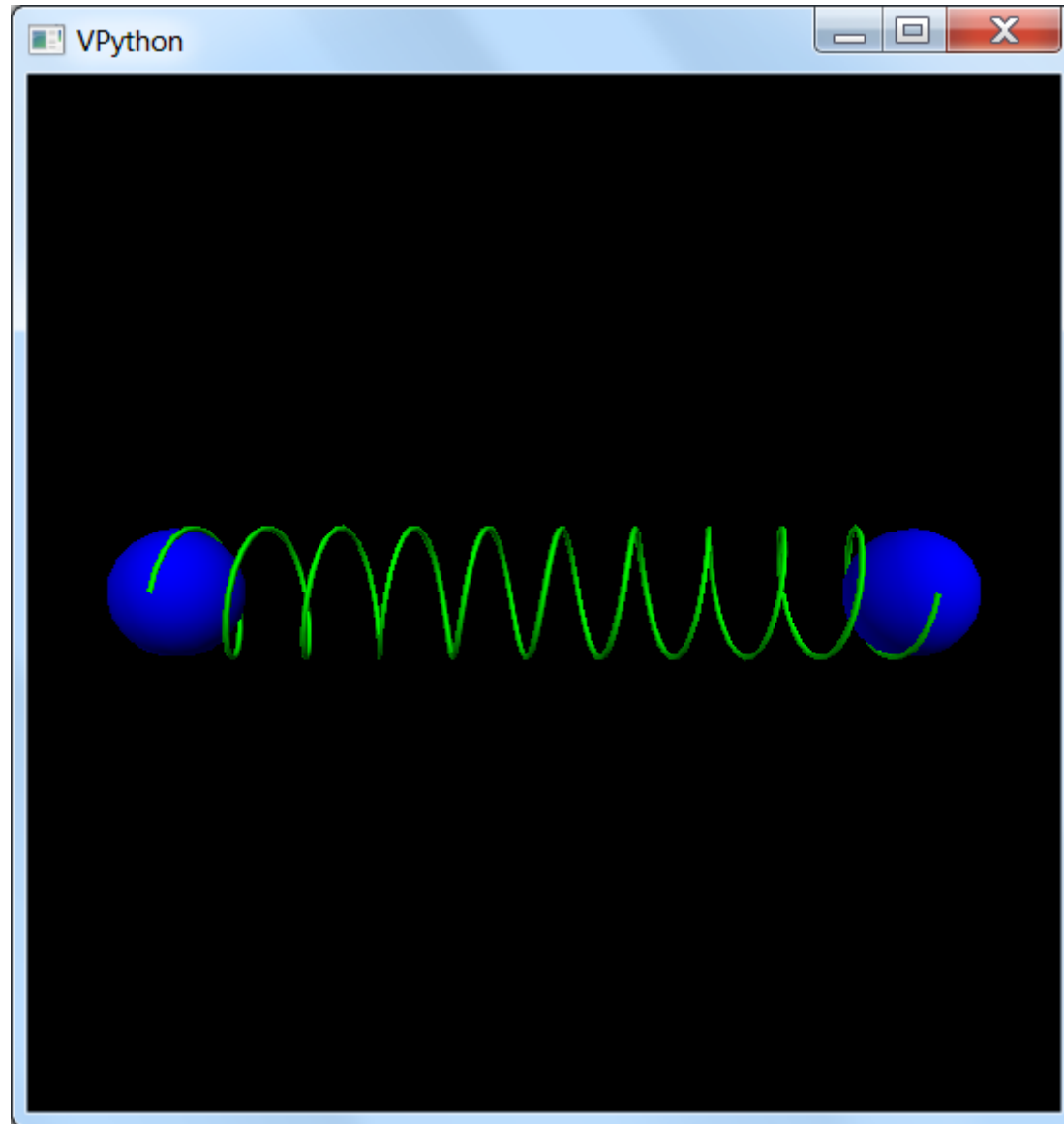
length = mag(axis)

sleep(2)

t = 0
while t<20:
    rate(300)
    ba1.pos.x = -2 - 0.9*sin(t)
    ba2.pos.x = 2 + 0.9*sin(t)
    he.pos = ba1.pos
    he.axis = ba2.pos - ba1.pos
    t += 0.01
```







# Symulacja z siłami

1) liczymy siły

$$\vec{F} = m \cdot \vec{a}$$

$$\vec{a} = \frac{\vec{v}_{\text{new}} - \vec{v}_{\text{old}}}{dt}$$

krok czasowy  $dt$   
musi być „mały”

$$\vec{v}_{\text{new}} = \vec{v}_{\text{old}} + \frac{\vec{F}}{m} \cdot dt$$

2) liczymy  $\vec{v}_{\text{new}}$


$$\vec{r}_{\text{new}} = \vec{r}_{\text{old}} + \vec{v}_{\text{new}} \cdot dt$$

3) liczymy  $\vec{r}_{\text{new}}$