VPython

Introduction

- VPython is based on the Python programming language with **3D** displays and animation (called Visual).

- Why VPython:
  - Python is a good first programming language
  - 3D Interactive Modeling
  - Python –background language
  - The simplicity of VPython made it a great tool for demonstrating simple physics concepts.

Resources (and many of these slides come from here)

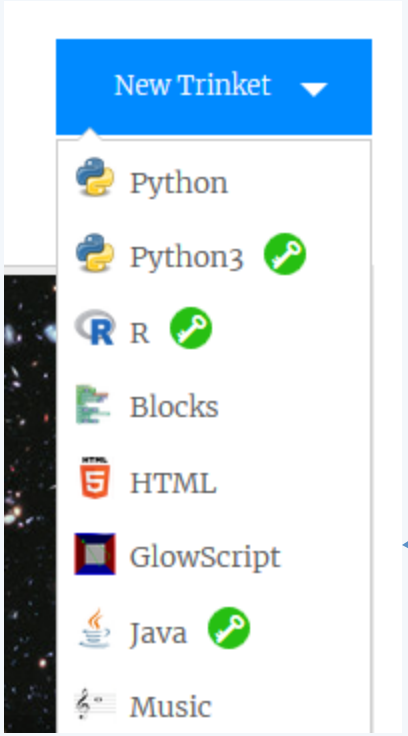
- http://www.glowscript.org/docs/VPythonDocs/index.html
- http://www.glowscript.org/docs/VPythonDocs/VisualIntro.html
- http://www.glowscript.org/docs/VPythonDocs/primitives.html
- https://www.youtube.com/playlist?list=PLdCdV2GBGyXOnMaPS1BgO7IOU_00ApuMo

Trinket

- Vpython = Python + Visual
- GlowScript a more current implementation of VPython that allows you to create a 3D environment in a web browser.
- Trinket lets you run and write code in a browser in a very user-friendly manner. Examples:
  - GlowScript
  - Python
  - Java
  - HTML
  - …

Trinket

- https://trinket.io/home
- Login with an account.



Select GlowScript

Let's Learn a Little Python

```
print (”hello, world”)
```

Totally intolerant of typing mistakes: this will not work

```
prind (”hello, world”)
```

Case sensitive: this won't work either

```
Print (”hello, world”)
```

Variables

A variable named **name**

is given the value

Rachel

```
name = "Rachel"
print ("hello, ", name)
```

What gets printed?

```
hello,  Rachel
```

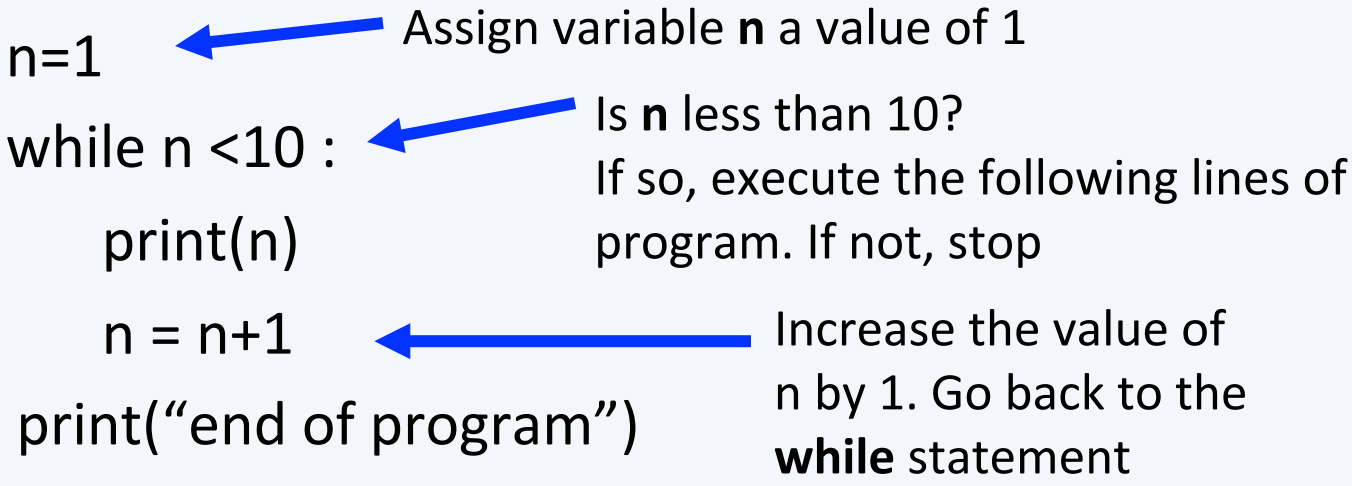
First Python executes the first line of the program

Next Python executes the second line of the program: it prints **hello,** followed by the value of the variable name

Python Loops

Often we wish to have a program execute the same lines over and over

Loops do this, Example:



Will 10 get printed?

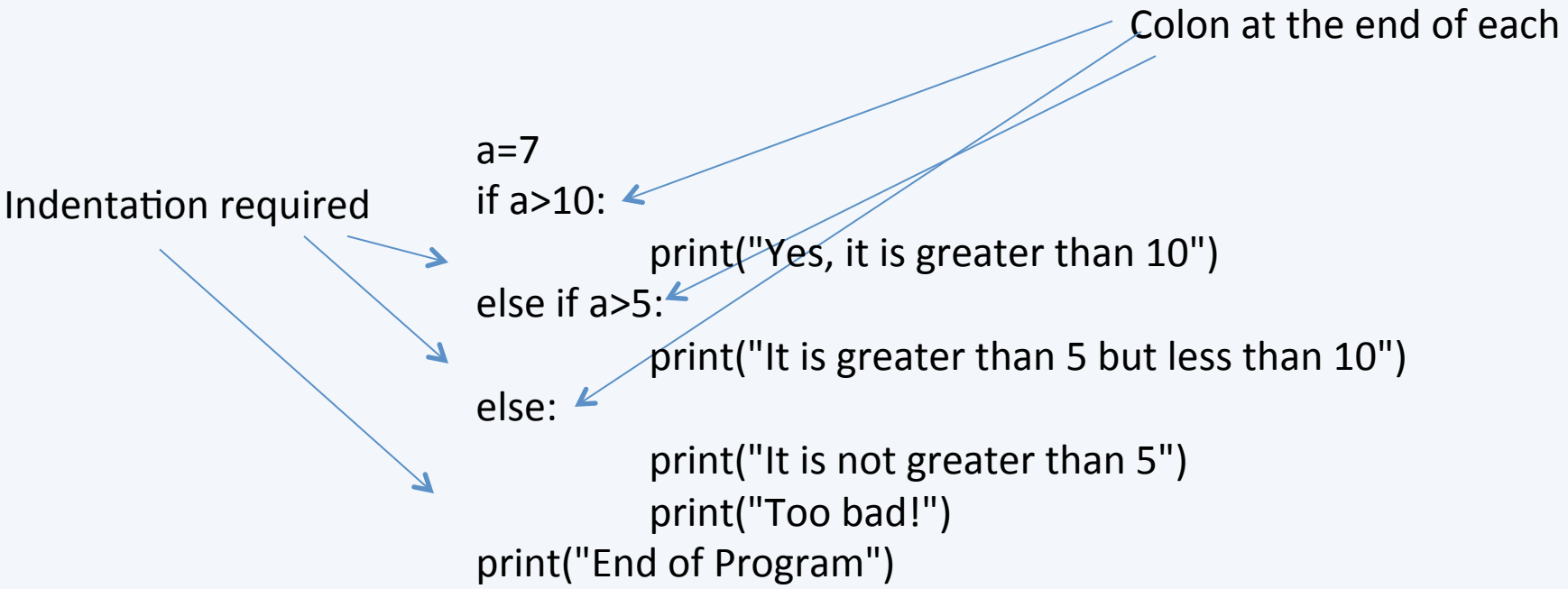
While Loop

```
while(condition):
    ....
    ....
```

() optional

Hit enter – then it knows what's in the loop because it's indented

Python Conditionals




```
It is greater than 5 but less than 10
End of Program
```

Python Conditionals, cont.

You can also use elif for else if

```
3
if a>10:
        print("Yes, it is greater than 10")
elif a>5:
        print("It is greater than 5 but less than 10")
else:
        print("It is not greater than 5")
        print("Too bad!")
print("End of Program")
```

What will print?

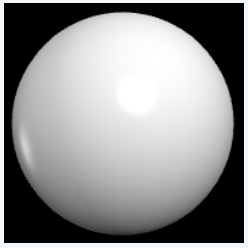
```
It is not greater than 5
Too bad!
End of Program
```

3D Visual

- To create an object type the object with ()

- Example:

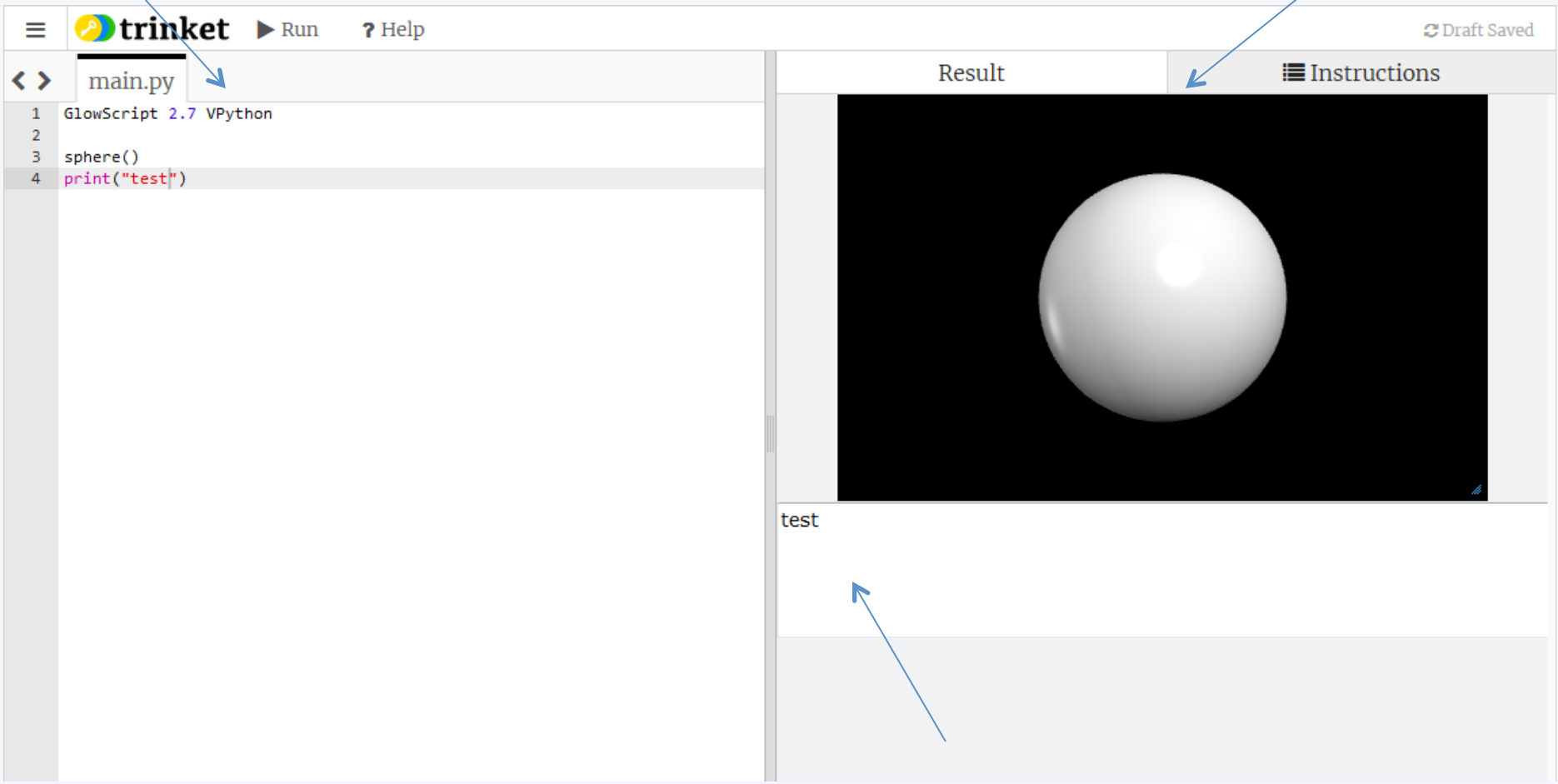
sphere()



VPython Environment

Code Window

Display Window



Output Window

Navigation

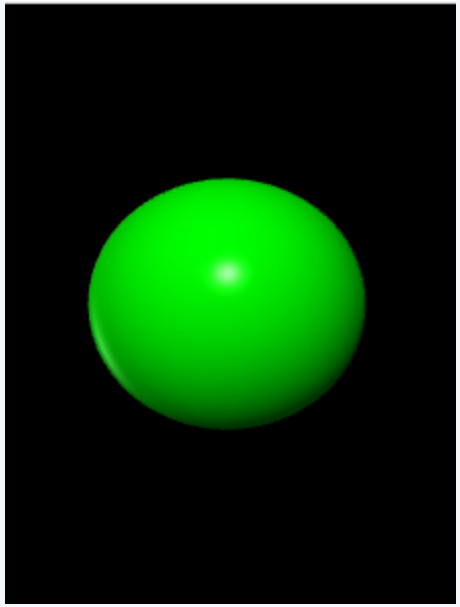
- Zoom
  - Hold middle button and move mouse
  - Or, hold both left and right buttons
- Rotate
  - Hold right button

Attributes

- Sphere has attributes:
  - Radius
  - Color
- They are given default values – let's change them!

Modify Attributes

- Modify Radius
  - sphere(radius=.5, color=color.green)

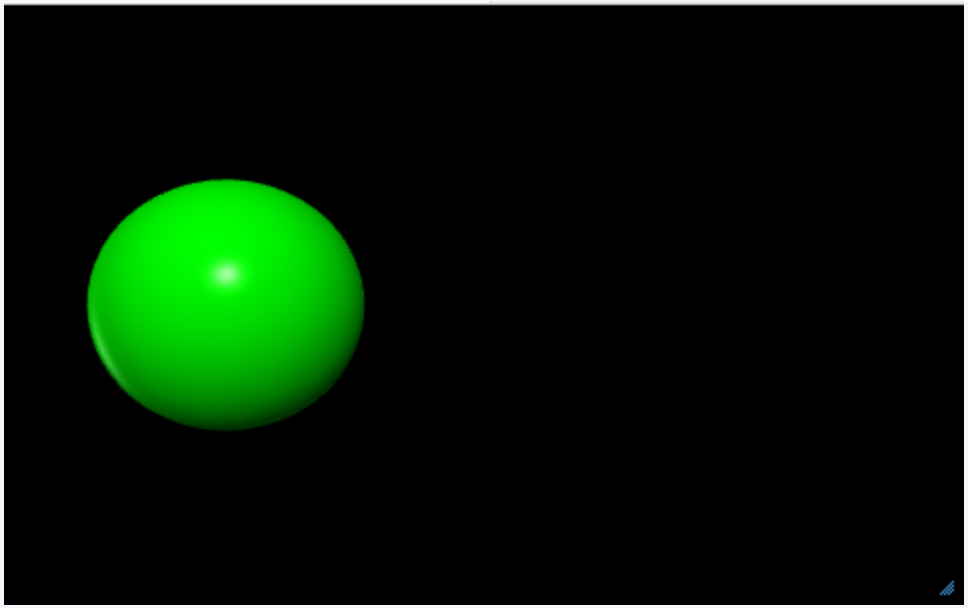


Modify Pos Attribute

- Modify Position

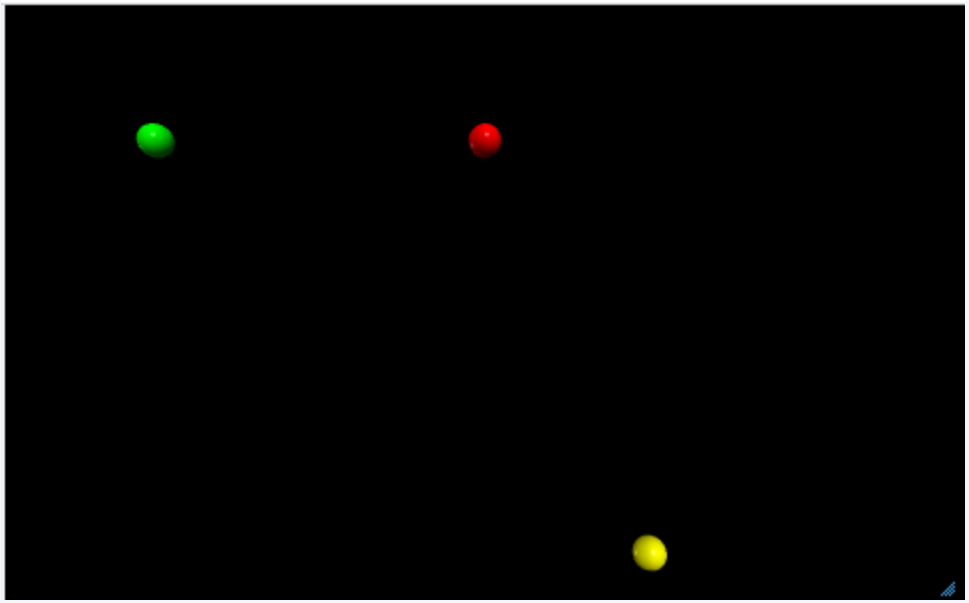
  -1 – moves 1 to the left

  sphere(pos=vector(-1,0,0), radius=.5, color=color.green)



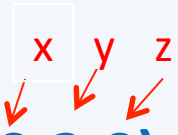
Balls

- ball1 = sphere(pos=vector(0,1,0),color=color.red, radius=0.1)
- ball2 = sphere(pos=vector(-2,1,0),color=color.green, radius=0.1)
- ball3 = sphere(pos=vector(1,-1.5,0),color=color.yellow, radius=0.1)



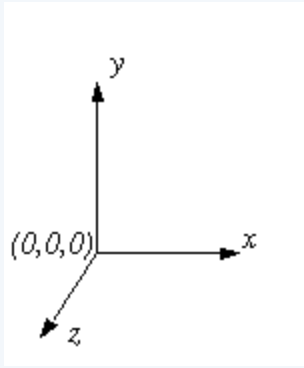
Changing Sphere Attributes

- Color
  - **sphere(color=color.red)**
- Radius
  - **sphere(radius=0.5,color=color.red)**
- Name
  - **ball = sphere(radius=0.5,color=color.red)**
- Position
  - **ball = sphere(pos=vector(0,2,0),radius=0.5,color=color.red)**

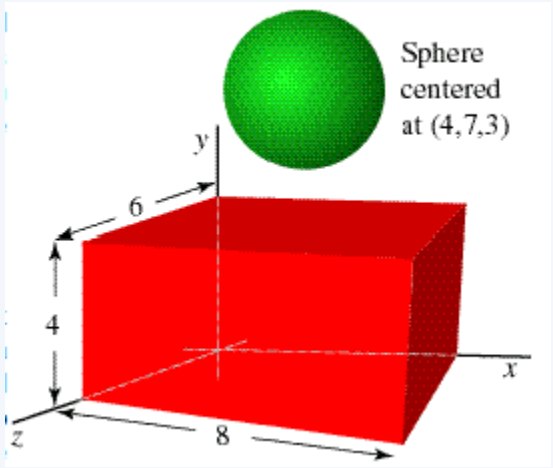


- Change position
  - **ball.pos = vector(1,2,3)**

VPython Display Window



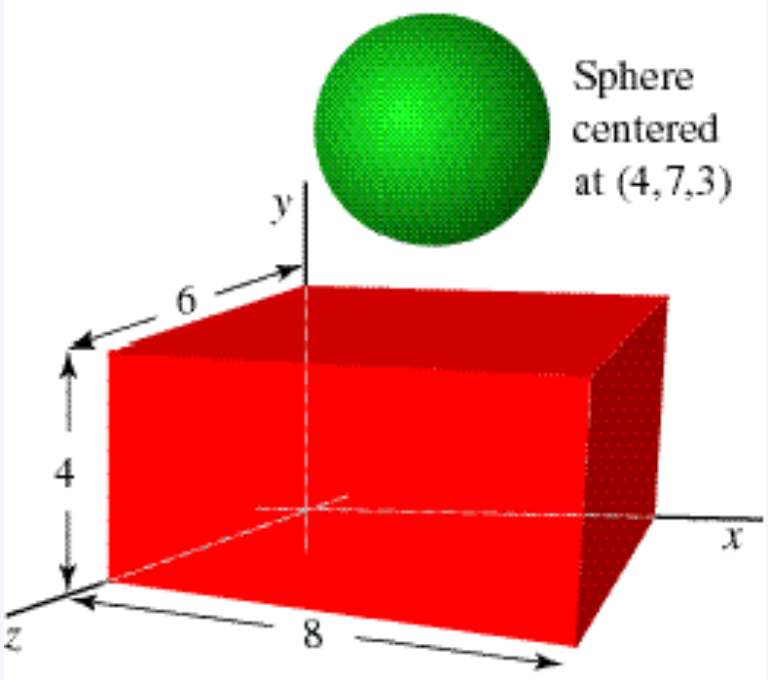
- When using Vpython, the display window shows objects in 3D.
- (0,0,0) is in the center of the display window . The +x axis runs to the right, the +y axis runs up, and the +z axis points out of the screen, toward you.



https://www.glowscript.org/docs/VPythonDocs/VisualIntro.html

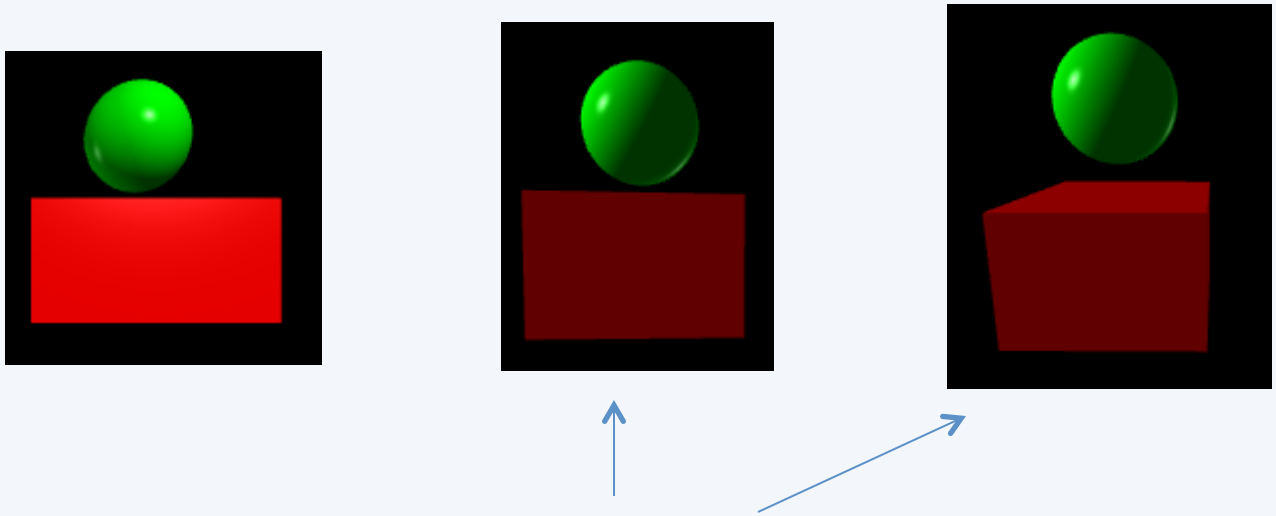
Another example

redbox=box(pos=vector(4,2,3), size=vector(8,4,6),color=color.red)
ball=sphere(pos=vector(4,7,3),radius=2,color=color.green)



https://www.glowscript.org/docs/VPythonDocs/VisualIntro.html

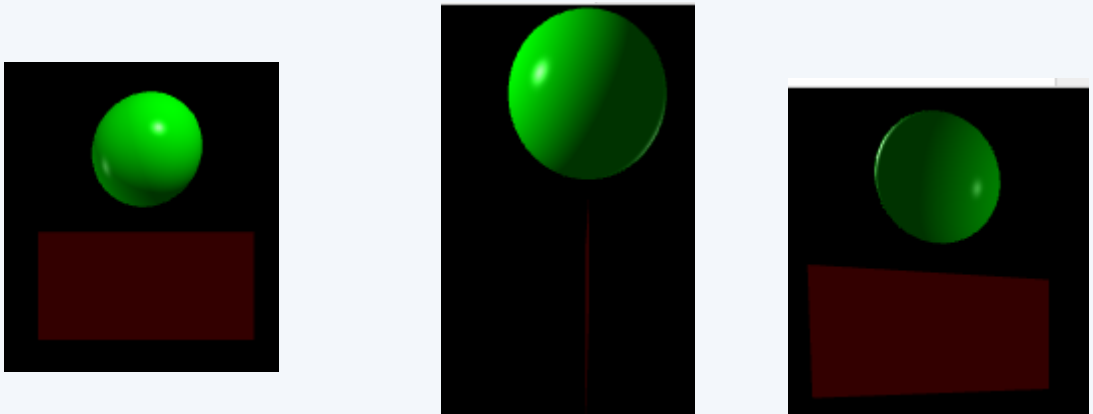
Z Axis



Right click and turn

If I set z to 0

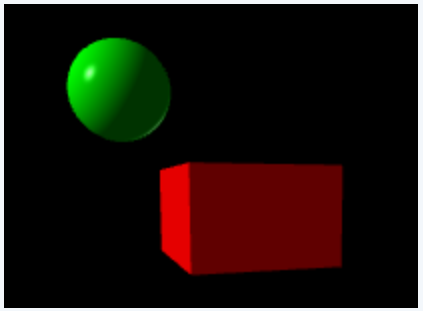
```
redbox=box(pos=vector(4,2,0), size=vector(8,4,0),color=color.red)
ball=sphere(pos=vector(4,7,0),radius=2,color=color.green)
```



Different Z values?

- Let's try different Z values
- redbox=box(pos=vector(4,2,3), size=vector(8,4,6),color=color.red)
  ball=sphere(pos=vector(4,7,8),radius=2,color=color.green)

- Move around the view



Print Vector

- ball = sphere(pos=vector(0,1,0),color=color.red, radius=0.1)
- print(ball.pos)

- What is returned from the print?

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

- Ball1.pos is the vector position of the ball.

- Try:
- print(ball.pos.y) – what do you think will print?
- 1

Some Physics

- What is a vector in physics?

Some definitions of a Vector:

  - *A quantity that has both magnitude and direction. A quantity that only has magnitude would be a scalar.*
  - *A quantity that contains more than one piece of information. Three dimensional vectors have three components. A vector with only one piece of information (a 1D vector), is a scalar.*

- Example: velocity, force, acceleration.

Vectors

- Vectors-displacement:

https://www.khanacademy.org/math/precalculus/vectors-precalc/vector-addition-subtraction/v/adding-vectors

- Vector Addition:

https://trinket.io/glowscript/8dc4452eb9

Vectors Addition

- Vector objects in VPython are similar to vectors in science and engineering.
- VPython allows you to create 3D vector quantities and perform vector operations on them.

```
v1 = vector(1,2,3)
v2 = vector(10,20,30)
print(v1+v2) # displays <11 22 33>
print(2*v1)  # displays <2 4 6>
```

- You can refer to individual components of a vector:
- v2.x is 10, v2.y is 20, v2.z is 30
- v2.mag or mag(v2)gives you the magnitude of the vector.

Another Example

- Vector Addition:

  a = vector(1,2,3)
  b = vector(4,5,6)
  c=a+b

  If you print *c* , you will find that it is a vector with components (5, 7, 9.).

- Scalar Multiplication
  - Multiply vector with scalar

  a = vector(1,2,3)

  d = 3*a

  d is a vector with components (3, 6, 9)

Vector addition and scalar multiplication

- Vector addition and scalar multiplication

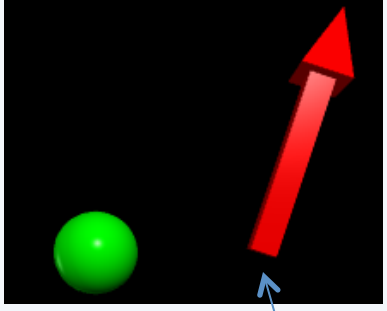
```
< 0, 1, 0 >   ball = sphere(pos=vector(0,1,0),color=color.red, radius=0.1)
              print(ball.pos)
< 0, 2, 0 >   m=2*ball.pos
              print(m)
              a=m + vec(2,-3,-1)
< 2, -1, -1 > print(a)
```

Display a Vector in VPython

- Vectors
  - Usually arrows
  - Vectors have magnitude and direction
  - Draw from one point to another
- Draw arrow from origin to center of ball
  - **arrow(pos=(0,0,0), axis=ball.pos)**

  Begins at pos Ends at axis
- Arbitrary arrow
  - **arrow(pos=(1,2,3), axis=(0,2,-1))**

Arrow



- Arrow does not have a radius, it is defined by have far it stretches and in which direction.

```
arrow(pos=vector(1,0,0), axis=vector(1, 3, 0), color=color.red)
```

Location of arrow's tail

Stretching 1 unit in X, 3 in Y and 0 in Z

What if we changed our mind and want the arrow in the same position as the circle whose x value was -1?

```
arrow(pos=vector(-1,0,0), axis=vector(+1, +3, 0),
color=color.red)
```

Variables

- What if you have multiple spheres and want to refer to attributes from each one?
- Let's store the objects in a variable!

```
Sue=sphere(pos=vector(-1,0,0),  radius=.25, color=color.red)
Bob = sphere(pos=vector(1,1,0),  radius=.15, color=color.orange)
arrow(pos=vector(0,0,0), axis=vector(0, 1, 0), color=color.yellow)
```

Then you can say:

Bob.color=color.red

Or use Sue.pos OR Sue.radius

Example – I want the arrow to start where the sphere, Sue, begins:

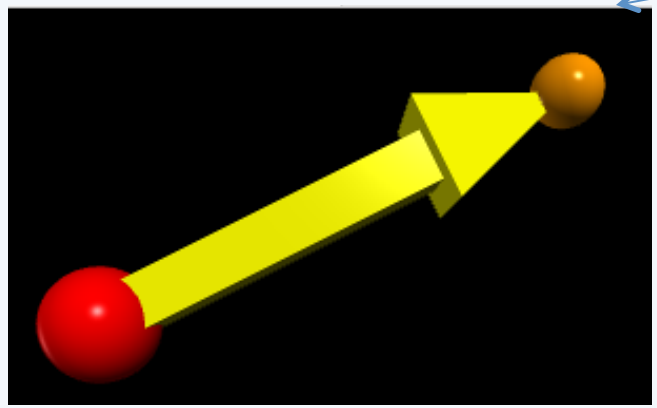
```
arrow(pos=Sue.pos, axis=vector(0, 1, 0), color=color.yellow)
```

Axis

- What if we want the arrow to reach from Sue's to Bob's position?

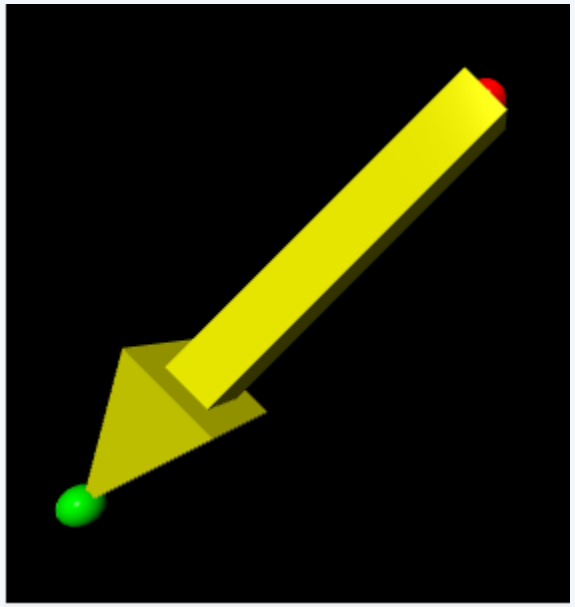
arrow(pos=Sue.pos, axis=Bob.pos-Sue.pos, color=color.yellow)

Bob's position-Sue's position would give you the distance you want it stretched.

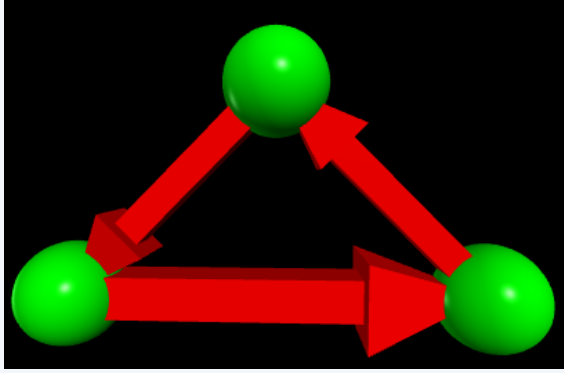


Another Example

```
ball1=sphere(pos=vector(0,1,0), radius=0.1, color=color.red)
ball2=sphere(pos=vec(-2,-1,0), radius=0.1, color=color.green)
dist=ball2.pos-ball1.pos
Arrow1=arrow(pos=ball1.pos, axis=dist, color=color.yellow)
```



In Class Exercise



- https://trinket.io/home
- Using variables for their names create
- 3 spheres. Attributes:
  - Set a position (vector – choose values for x and y, z can be 0)
  - radius (you can use .5)
  - color
- 3 arrows. Attributes:
  - pos – set position appropriately using the spheres' positions.
  - axis - set appropriately using what you know from the spheres' positions
  - Connect them as shown in the diagram

Extra Time:

- What other shapes can you draw? Check out:

http://www.glowscript.org/docs/VPythonDocs/index.html

Look at the drop down on the left to see some other shapes and try them out.