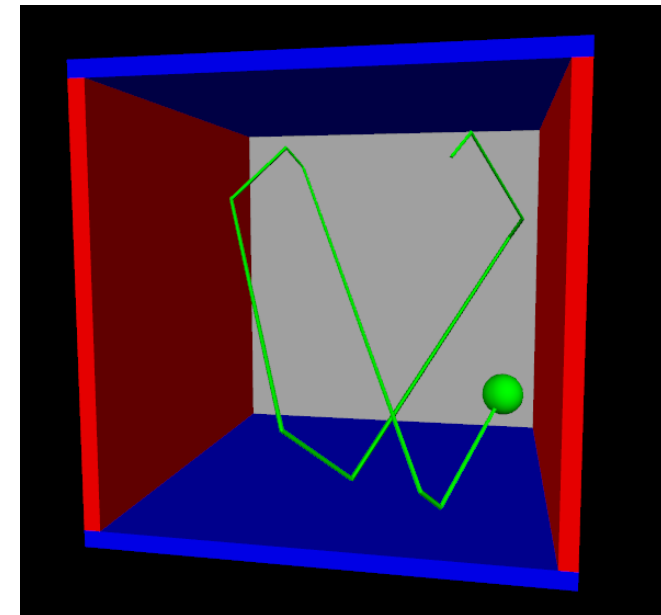
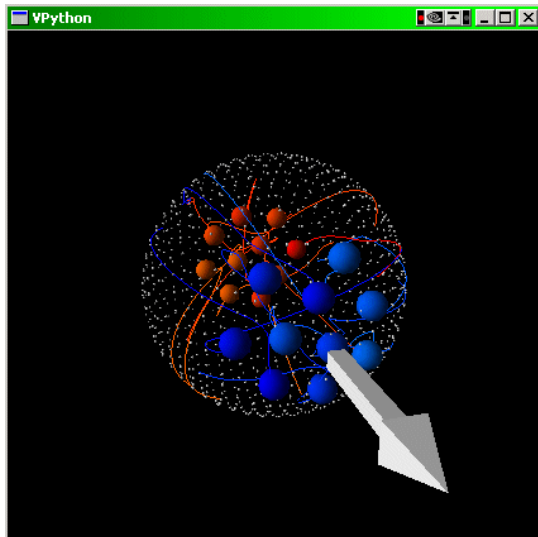


# vPython



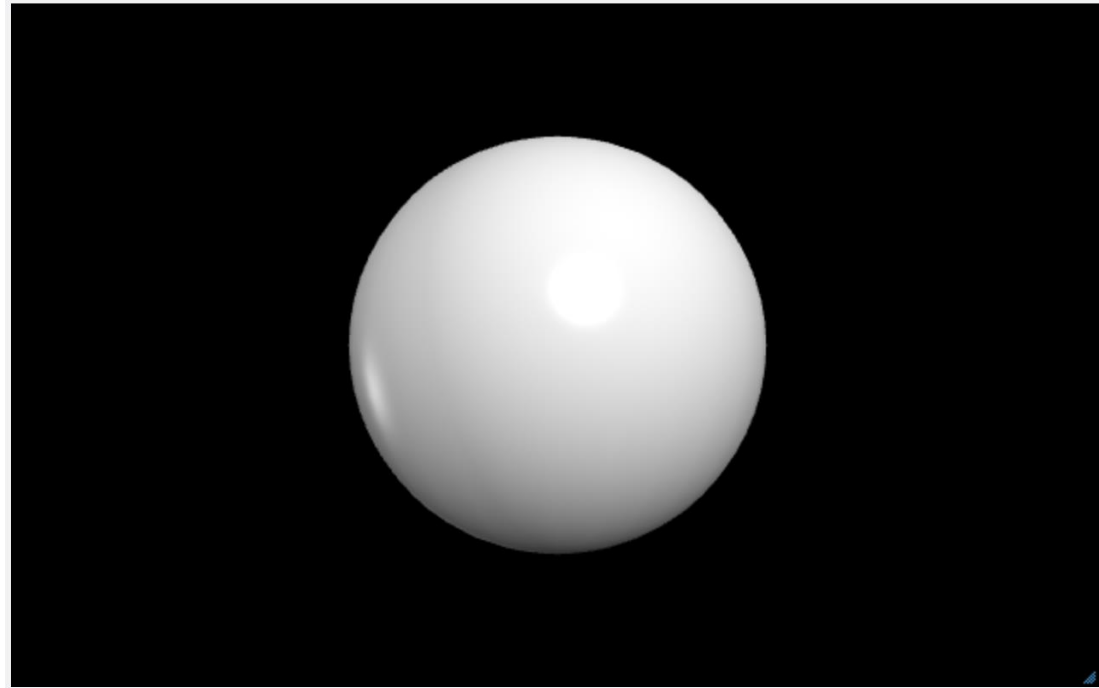
# 도움 자료

- 예제 : <https://www.glowscript.org/#/user/GlowScriptDemos/folder/Examples/>
- Tutorial : [https://www.glowscript.org/docs/VPythonDocs/VPython\\_Intro.pdf](https://www.glowscript.org/docs/VPythonDocs/VPython_Intro.pdf)
- 도움말 : <https://www.glowscript.org/docs/VPythonDocs/index.html>
- vpython & 물리 :
  - [https://www.youtube.com/watch?v=pzKJ\\_R-lpho&list=PLO7G1xLUawYDrpr7bSk\\_c\\_XABpy-Ohatc&index=55](https://www.youtube.com/watch?v=pzKJ_R-lpho&list=PLO7G1xLUawYDrpr7bSk_c_XABpy-Ohatc&index=55)
  - [https://www.youtube.com/watch?v=FyEWafwMJf8&list=PLO7G1xLUawYDrpr7bSk\\_c\\_XABpy-Ohatc&index=56](https://www.youtube.com/watch?v=FyEWafwMJf8&list=PLO7G1xLUawYDrpr7bSk_c_XABpy-Ohatc&index=56)
  - [https://www.youtube.com/watch?v=RdXXdkQ4foE&list=PLO7G1xLUawYDrpr7bSk\\_c\\_XABpy-Ohatc&index=57](https://www.youtube.com/watch?v=RdXXdkQ4foE&list=PLO7G1xLUawYDrpr7bSk_c_XABpy-Ohatc&index=57)

Web VPython 3.2

sphere()

회전 : 마우스 오른쪽 버튼 클릭 후 드래그  
카메라 거리(줌인/줌아웃) : 마우스 휠



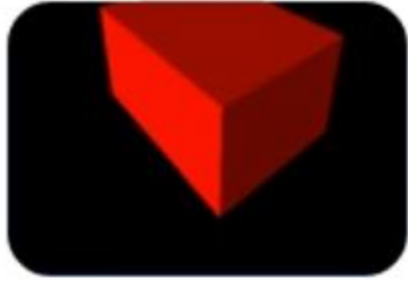
sphere 객체의 속성(변수)

pos : 위치	# 기본값 : vector(0, 0, 0)
color : 색상	# 기본값 : color.white or vector(1, 1, 1)
radius : 반지름	# 기본값 : 0.5

...

도움말 링크 참조

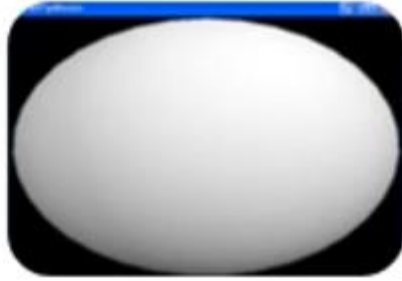
# vPython에서 사용할 수 있는 그래픽 객체들



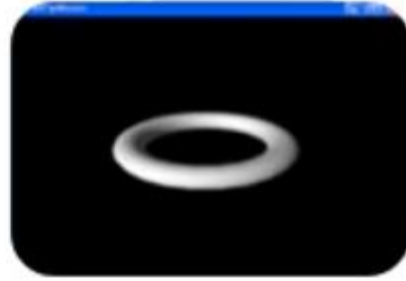
1. ~~상자(box)~~



2. 구(sphere)



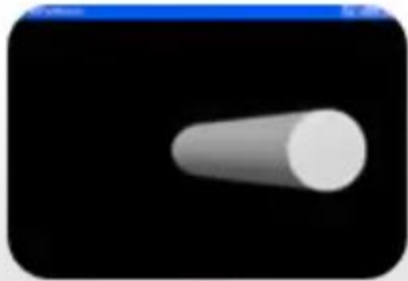
3. 타원체  
(ellipsoid)



4. 고리(ring)



5. 원뿔(cone)



6. 원통  
(cylinder)



7. 화살표  
(arrow)



8. 원형나선  
(helix)



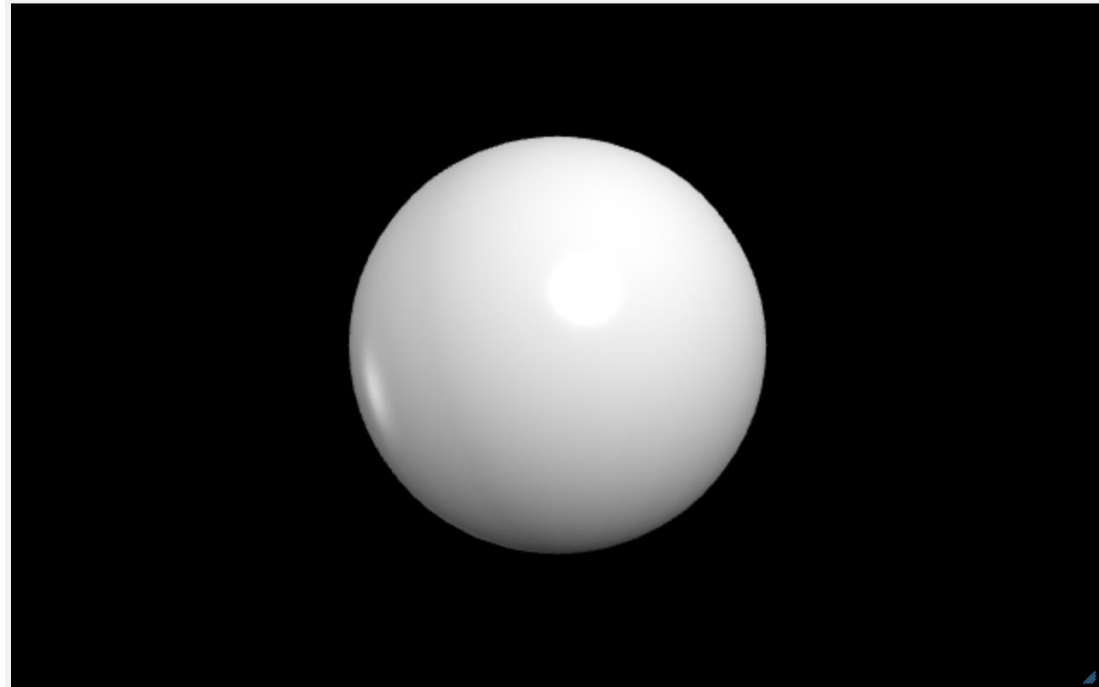
9. 피라미드  
(pyramid)



10. 곡선  
(curve)

Web VPython 3.2

```
sphere()  
sphere()
```



회전 : 마우스 오른쪽 버튼 클릭 후 드래그  
카메라 거리(줌인/줌아웃) : 마우스 휠

Web VPython 3.2

```
sphere(pos=vector(-5,0,0))    # print(1, 2, end=' ') # 디폴트 파라미터는 pos=vector(0, 0, 0)  
sphere(pos=vector(5,0,0))
```

Web VPython 3.2

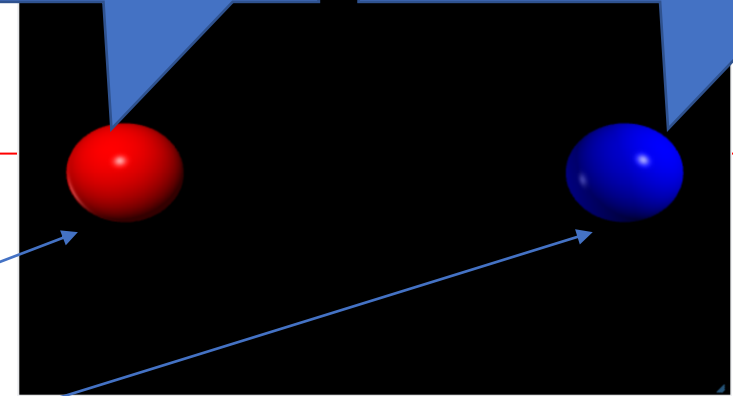
```
sphere(pos=vector(-5,0,0), color=vector(1,0,0))  
sphere(pos=vector(5,0,0), color=color.blue)
```

Web VPython 3.2

```
sphere(pos=vector(-5,0,0), color=vector(1,0,0))  
sphere(pos=vector(5,0,0), color=color.blue)
```

pos = vector(-5, 0, 0)  
color = vector(1, 0, 0)  
radius = 0.5(기본값)

pos = vector(5, 0, 0)  
color = vector(0, 0, 1)  
radius = 0.5(기본값)



실행 후 1/0.5초 후 아래처럼 공 색깔을 바꾸고 싶다.  
왼쪽공의 색깔을 **color.green**으로  
오른쪽공의 색깔을 **color.red**로 바꾸고 싶다...????

객체를 변수에 담아 변수의 이름으로... 속성변수를 변경

Web VPython 3.2

```
lball = sphere(pos=vector(-5,0,0), color=vector(1,0,0))  
rball = sphere(pos=vector(5,0,0), color=color.blue)
```

```
rate(0.5) # 1/0.5 초 delay
```

```
lball.color = color.green  
rball.color = color.red
```

sphere 객체의 속성(변수)

pos : 위치 # 기본값 : vector(0, 0, 0)  
color : 색상 # 기본값 : color.white or vector(1, 1, 1)  
radius : 반지름 # 기본값 : 0.5  
...  
도움말 링크 참조

## Web VPython 3.2

```
lball = sphere(pos=vector(-5,0,0), color=vector(1,0,0))  
rball = sphere(pos=vector(5,0,0), color=color.blue)
```

```
rate(0.5)
```

```
lball.color = color.green  
rball.color = color.red
```

#왼쪽 공의 반지름을 2로 변경하려면????(속성변수는 radius)

```
lball.radius = 2
```

#오른쪽 공을 위쪽으로 3만큼 움직이려면???

```
rball.pos = rball.pos + vector(0, -3, 0)
```

Web VPython 3.2

```
lball = sphere(pos=vector(-5,0,0), color=vector(1,0,0))  
rball = sphere(pos=vector(5,0,0), color=color.blue)
```

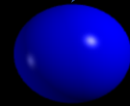
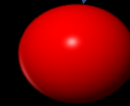
```
# lball 객체에 mass라는 변수를 추가하고 값을 10으로 하고 싶다면?  
lball.mass = 10
```

```
print(lball.mass)
```

pos = vector(-5, 0, 0)  
color = vector(1, 0, 0)  
radius = 0.5(기본값)

**mass = 1(변수추가)**

pos = vector(5, 0, 0)  
color = vector(0, 0, 1)  
radius = 0.5(기본값)







# 객체 생성

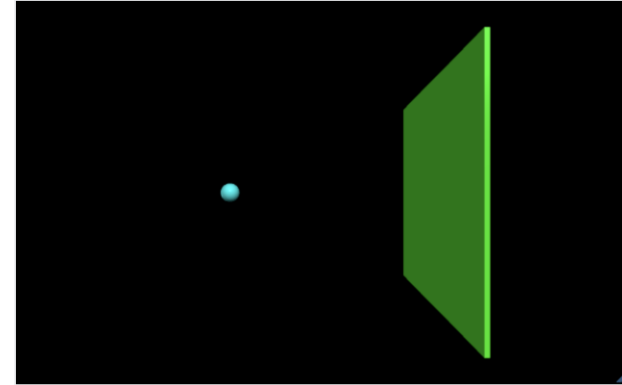
Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

```
ball.velocity = vector(25,0,0)    # ball에 사용자 속성 변수 추가
deltat = 0.005
t = 0
ball.pos = ball.pos + ball.velocity * deltat
```

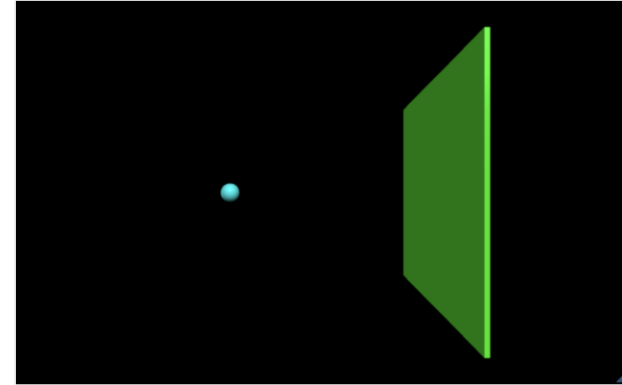


# 객체 속성 설정

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)  
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

```
ball.velocity = vector(25,0,0)    # ball에 사용자 속성 변수 추가  
deltat = 0.005  
t = 0  
ball.pos = ball.pos + ball.velocity * deltat
```



# ball의 이동 설정

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

```
ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0
```

```
while t < 3:
    ball.pos = ball.pos + ball.velocity*deltat
    t = t + deltat
```

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)

ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0

while t < 3:
    rate(100) # 1/100초 대기
    ball.pos = ball.pos + ball.velocity*deltat
    t = t + deltat
```

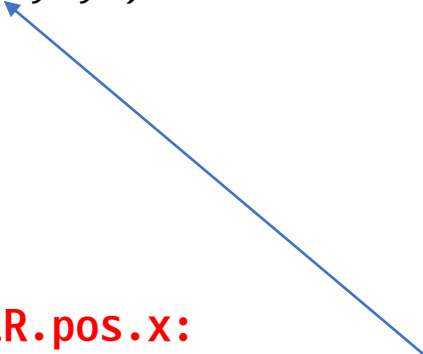
# ball이 wallR에 부딪혔을 때 튕어 나오기

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

```
ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0
```

```
while t < 3:
    rate(100)
    if ball.pos.x > wallR.pos.x:
        ball.velocity.x = -ball.velocity.x
    ball.pos = ball.pos + ball.velocity*deltat
    t = t + deltat
```



# 왼쪽 벽 추가 및 왼쪽 벽에 부딪치면 튕어 나오기

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
wallL = box(pos=vector(-6,0,0), size=vector(0.2,12,12), color=color.green)
```

```
ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0
```

```
while t < 3:
    rate(100)
    if ball.pos.x > wallR.pos.x:
        ball.velocity.x = -ball.velocity.x
    if ball.pos.x < wallL.pos.x:
        ball.velocity.x = -ball.velocity.x
    ball.pos = ball.pos + ball.velocity*deltat
    t = t + deltat
```

# 자취 남기기

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan, make_trail=True, retain=70)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
wallL = box(pos=vector(-6,0,0), size=vector(0.2,12,12), color=color.green)

ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0

while t < 3:
    rate(100)
    if ball.pos.x > wallR.pos.x:
        ball.velocity.x = -ball.velocity.x
    if ball.pos.x < wallL.pos.x:
        ball.velocity.x = -ball.velocity.x
    ball.pos = ball.pos + ball.velocity*deltat
    t = t + deltat
```



Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
wallL = box(pos=vector(-6,0,0), size=vector(0.2,12,12), color=color.green)
ball.velocity = vector(25,0,0)
#ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0

vscale = 0.1
varr = arrow(pos=ball.pos, axis=vscale*ball.velocity, color=color.yellow)
scene.autoscale = False
ball.trail = curve(color=ball.color)

ball.pos = ball.pos + ball.velocity*deltat

while t < 3:
    rate(100)
    if ball.pos.x > wallR.pos.x:
        ball.velocity.x = -ball.velocity.x
    if ball.pos.x < wallL.pos.x:
        ball.velocity.x = -ball.velocity.x
    ball.pos = ball.pos + ball.velocity*deltat

    varr.pos=ball.pos
    varr.axis=vscale*ball.velocity
    ball.trail.append(pos=ball.pos)

    t = t + deltat
```

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)

wallL = box(pos=vector(-6,0,0), size=vector(0.2,12,12), color=color.green)

wallT = box(pos=vector(0,6,0), size=vector(12,0.2,12), color=color.blue)
wallB = box(pos=vector(0,-6,0), size=vector(12,0.2,12), color=color.blue)
wallF = box(pos=vector(0,0,6), size=vector(12,12,0.2), color=color.red, opacity=0.2)
wallRE = box(pos=vector(0,0,-6), size=vector(12,12,0.2), color=color.red)
```

```
ball.velocity = vector(25,30,-35)
#ball.velocity = vector(25,0,0)
deltat = 0.005
t = 0
```

```
vscale = 0.1
varr = arrow(pos=ball.pos, axis=vscale*ball.velocity, color=color.yellow)
scene.autoscale = False
ball.trail = curve(color=ball.color)
```

```
ball.pos = ball.pos + ball.velocity*deltat
```

```
while True:
    rate(100)
    if ball.pos.x > wallR.pos.x:
        ball.velocity.x = -ball.velocity.x
    if ball.pos.x < wallL.pos.x:
        ball.velocity.x = -ball.velocity.x

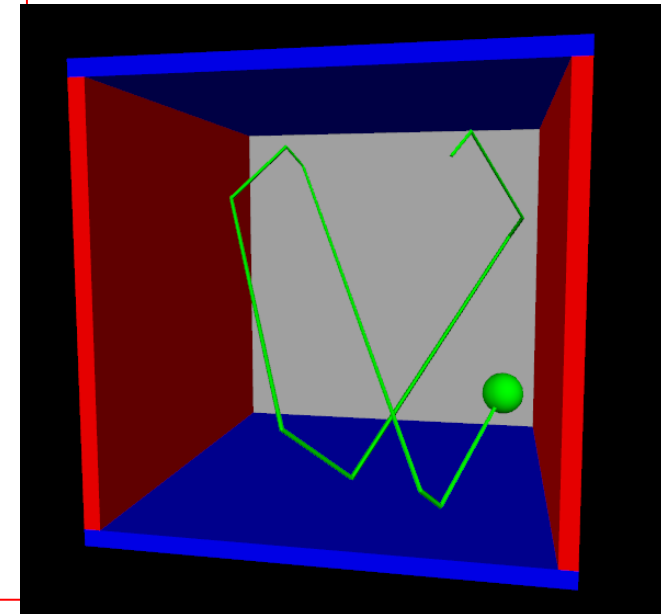
    if ball.pos.y > wallT.pos.y:
        ball.velocity.y = -ball.velocity.y
    if ball.pos.y < wallB.pos.y:
        ball.velocity.y = -ball.velocity.y

    if ball.pos.z > wallF.pos.z:
        ball.velocity.z = -ball.velocity.z
    if ball.pos.z < wallRE.pos.z:
        ball.velocity.z = -ball.velocity.z

    ball.pos = ball.pos + ball.velocity*deltat

    varr.pos=ball.pos
    varr.axis=vscale*ball.velocity
    ball.trail.append(pos=ball.pos)

    t = t + deltat
```



## Web VPython 3.2

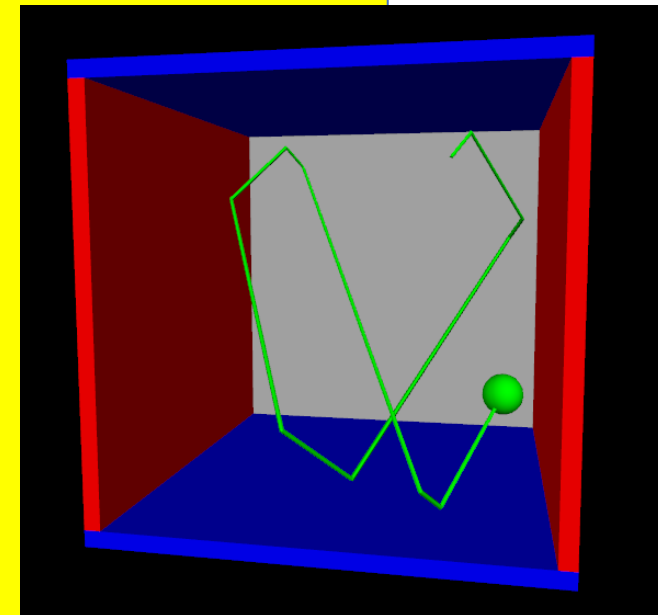
```
side = 4.0
thk = 0.3
s2 = 2*side - thk
s3 = 2*side + thk

wallR = box (pos=vector( side, 0, 0), size=vector(thk, s2, s3), color = color.red)
wallL = box (pos=vector(-side, 0, 0), size=vector(thk, s2, s3), color = color.red)
wallB = box (pos=vector(0, -side, 0), size=vector(s3, thk, s3), color = color.blue)
wallT = box (pos=vector(0, side, 0), size=vector(s3, thk, s3), color = color.blue)
wallBK = box(pos=vector(0, 0, -side), size=vector(s2, s2, thk), color = color.gray(0.7))

ball = sphere (color = color.green, radius = 0.4, make_trail=True, retain=200)
ball.mass = 1.0
ball.p = vector (-0.15, -0.23, +0.27)

side = side - thk*0.5 - ball.radius

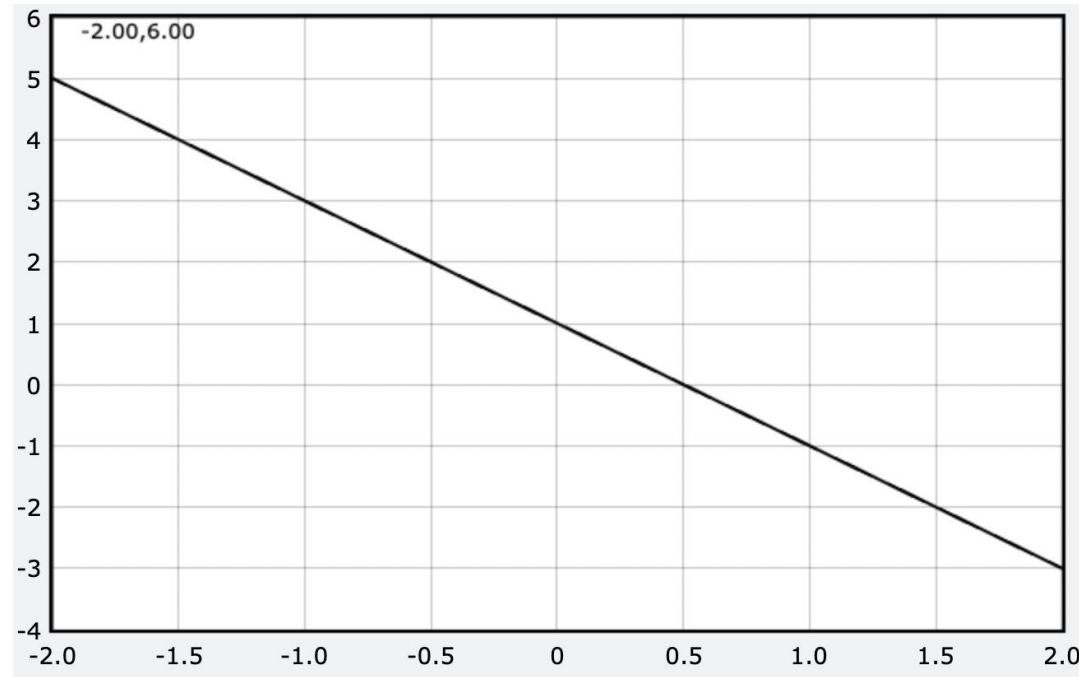
dt = 0.3
while True:
    rate(200)
    ball.pos = ball.pos + (ball.p/ball.mass)*dt
    if not (side > ball.pos.x > -side): ball.p.x = -ball.p.x
    if not (side > ball.pos.y > -side): ball.p.y = -ball.p.y
    if not (side > ball.pos.z > -side): ball.p.z = -ball.p.z
```



그래프 그리기...

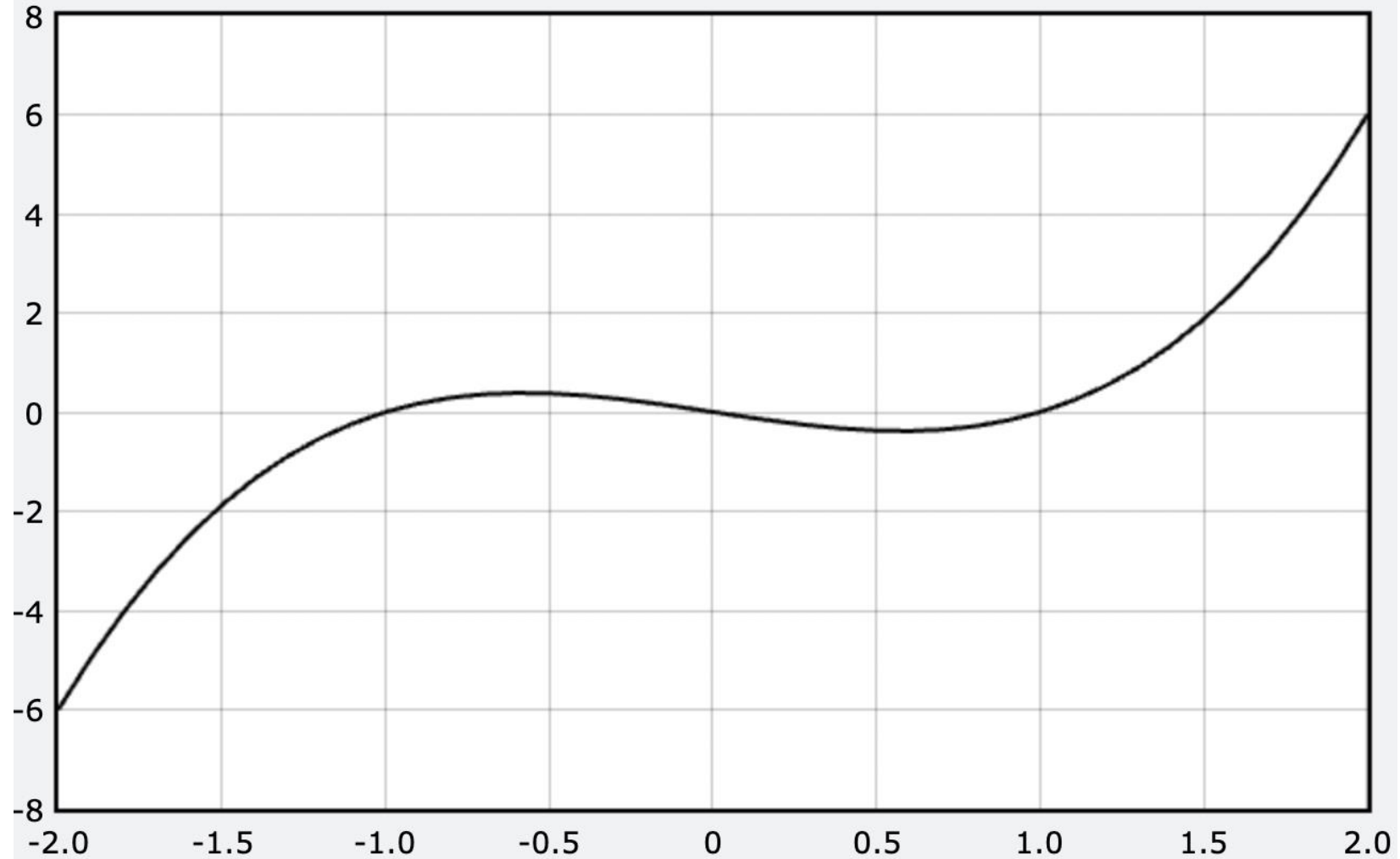
## Web VPython 3.2

```
g = gcurve()  
for x in arange(-2.0, 2.1, 0.1):  
    g.plot(pos=(x, -2*x+1))
```



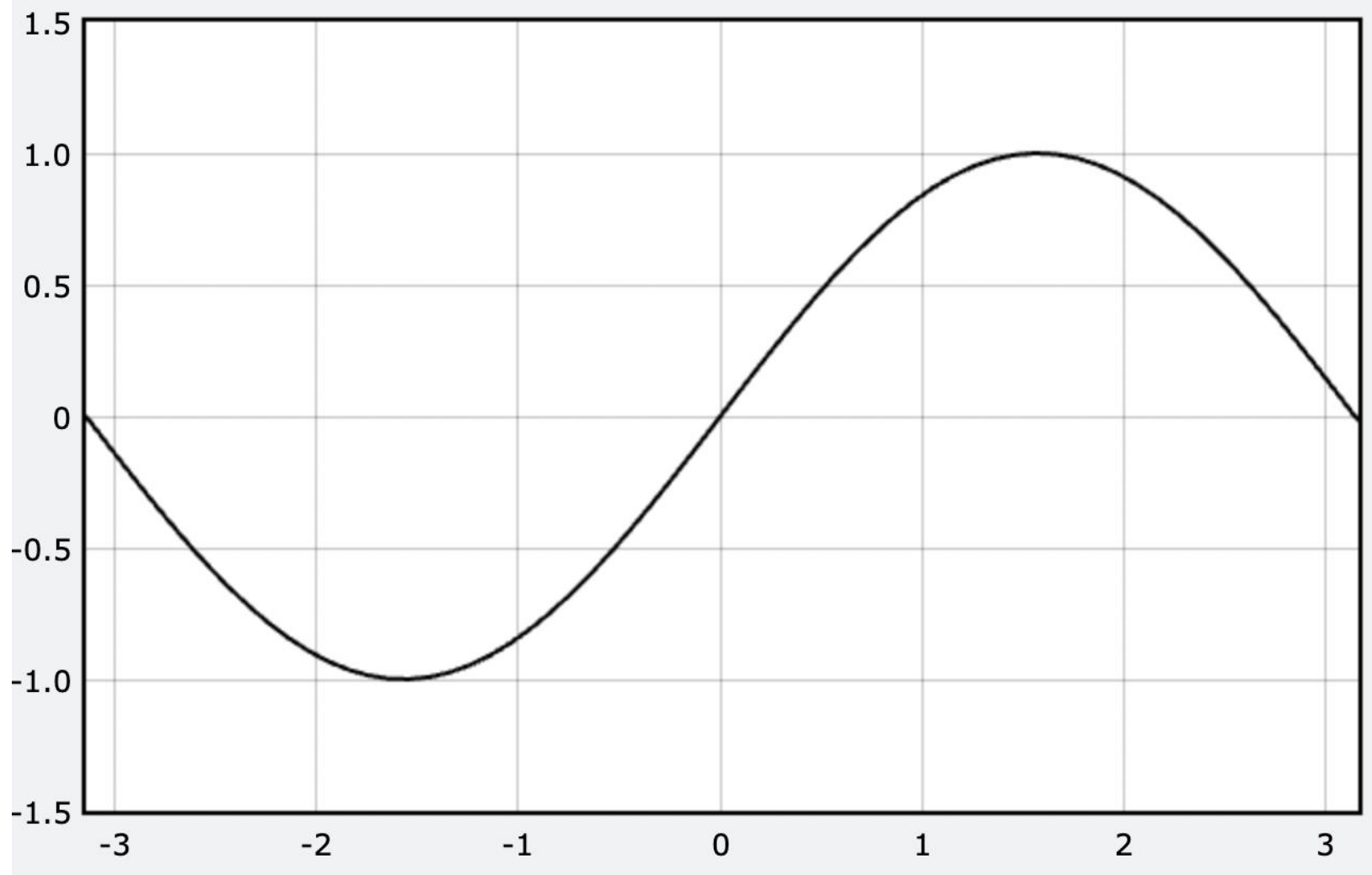
Web VPython 3.2

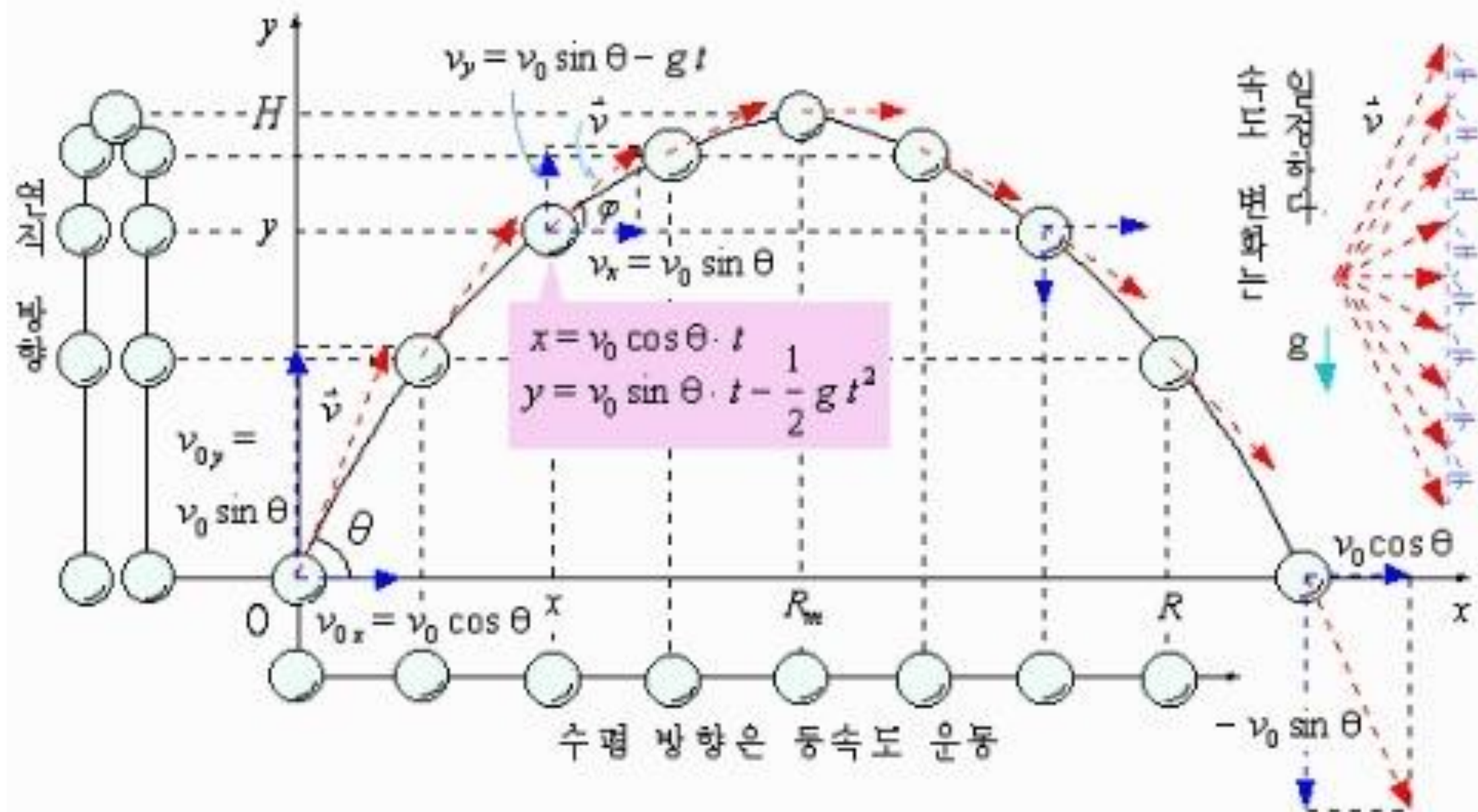
```
g = gcurve()  
for x in arange(-2.0, 2.1, 0.1):  
    g.plot(pos=(x, -2*x+1))
```



Web VPython 3.2

```
g = gcurve()  
for x in arange(-pi, pi+0.1, 0.1):  
    g.plot(pos=(x, sin(x)))
```





비스듬히 위로 던져 올린 물체의 운동



## Web VPython 3.2

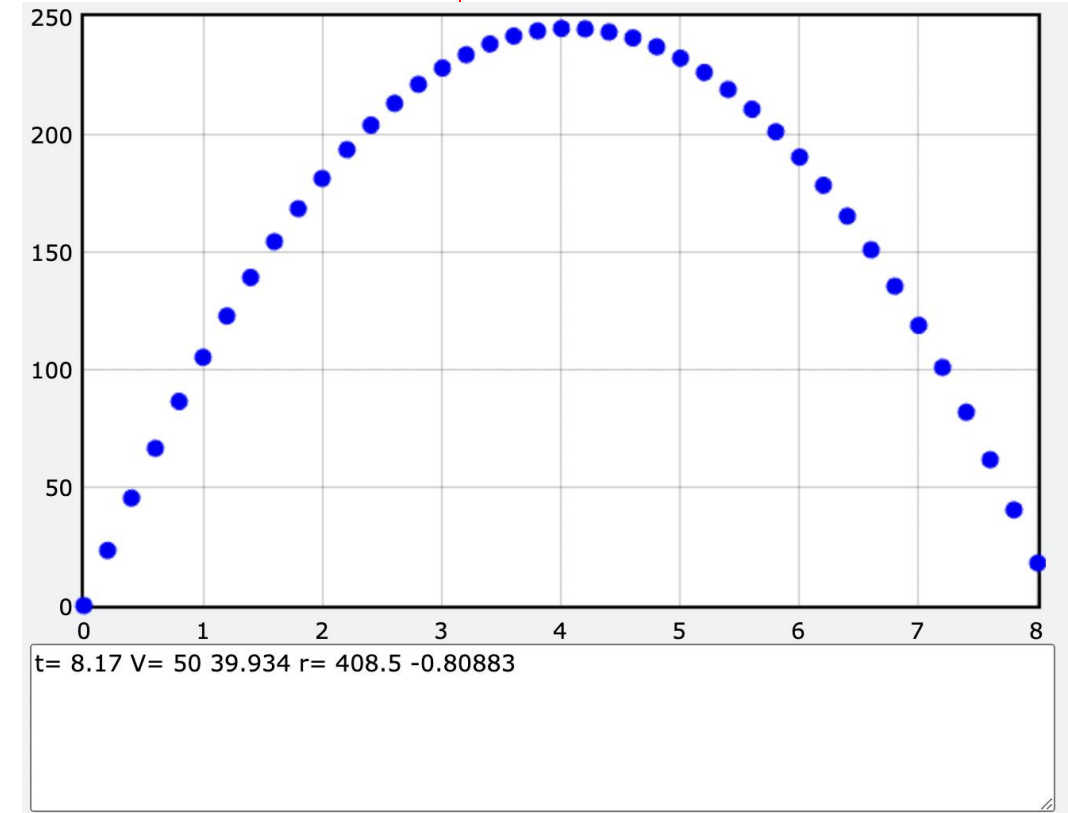
```
m=0.1                # Kg
g=vector(0,-9.8,0)   # m/s^2; 중력가속도
r0=vector(0,0,0)     # m
v0=vector(50,120,0)  # m/s
h=gdots(color=color.blue, size=10)
dt=0.01

for t in arange(0, 10+dt, dt):
    v=v0+g*t
    r=r0+v*t+(g/2)*t**2

    rate(1/dt)

    if r.y>=0:
        if(int(t/dt)%20==0):
            h.plot(pos=(t,r.y))

    if r.y<0:
        print("t=",t , "V=", v.x, v.y, "r=", r.x, r.y)
        break
```



## Web VPython 3.2

```
g=vector(0,-9.8,0) # m/s^2
vm=100              # m/s
r0=vector(0,0,0)
h=gdots(color=color.blue, size=5)

for angle in arange(0, 90+5, 5):
    theta=angle*pi/180. #radian
    v0=vector(vm*cos(theta), vm*sin(theta), 0)
    for t in arange(0, 16, 0.1):
        rate(50)
        v=v0+g*t
        r=r0+v*t+(g/2)*t**2
        if r.y>=0:
            h.plot(pos=(r.x, r.y))
        elif r.y<=0:
            print("각도=", angle, "체공시간=", t, "수평도달거리=", r.x)
            break
```

