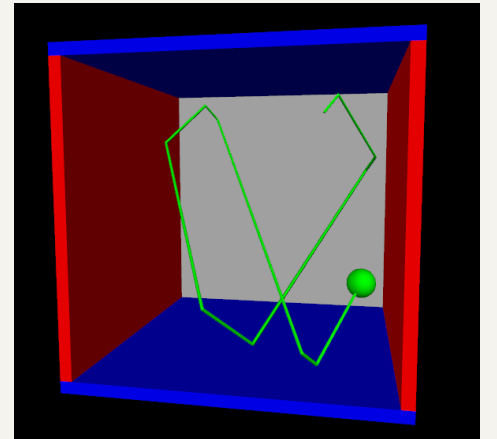
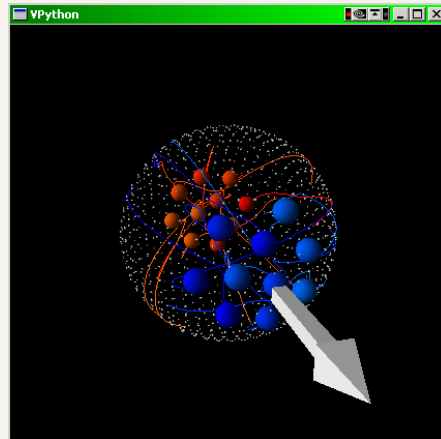


vPython을 활용한 visual simulation

<https://glowscript.org/>



핵심 키워드



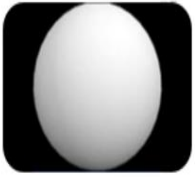
도움자료

- 예제 : <https://www.glowscript.org/#/user/GlowScriptDemos/folder/Examples/>
- Tutorial : 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-Ipho&list=PL07G1xLUawYDrpr7bSk_c_XABpy-0hatc&index=55
 - https://www.youtube.com/watch?v=FyEWafwMJf8&list=PL07G1xLUawYDrpr7bSk_c_XABpy-0hatc&index=56
 - https://www.youtube.com/watch?v=RdXXdkQ4foE&list=PL07G1xLUawYDrpr7bSk_c_XABpy-0hatc&index=57

vPython의 3차원 객체



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)

• 상자를 그리고 싶다면?

- `box()` # 기본형
- `box(pos=위치, color=색상, radius=크기)`

• 구를 그리고 싶다면?

- `sphere()` # 기본형
- `sphere(pos=위치, color=색상, radius=크기)`

• 스피링을 그리고 싶다면?

- `helix()` # 기본형
- `helix(위치, 색상, 크기)`

객체

객체 = 변수(값, 속성) + 함수(메서드)

```
sphere(pos=vector(0, 0, 0), color=vector(1, 1, 1), radius=2)  
혹은  
sphere()
```



속성(=변수값)
pos : vector(0, 0, 0)
color : vector(1, 1, 1)
radius : 2

위치변수 pos
pos = 벡터값 # 기본값 : vector(0, 0, 0)
원점

색상변수 color
color = 벡터값 # 기본값 : vector(1, 1, 1)
흰색

크기변수 radius
radius = 스칼라 # 기본값 : 2

... 나머지 변수와 기본값은 도움말 링크 참조

<https://www.glowscript.org/docs/VPythonDocs/sphere.html>

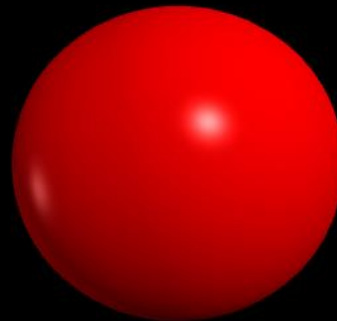
```
sphere(pos=vector(1, 0, 0))
```

속성(=변수값)
pos : vector(1, 0, 0)
color : vector(1, 1, 1)
radius : 2



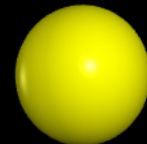
```
sphere(pos=vector(1, 0, 0), color=(1, 0, 0))
```

속성(=변수값)
pos : vector(1, 0, 0)
color : vector(1, 0, 0)
radius : 2



```
sphere(pos=vector(0, 1, 0), color=(1, 1, 0), radius=0.5)
```

속성(=변수값)
pos : vector(0, 1, 0)
color : vector(1, 1, 0)
radius : 0.5



객체를 변수에 할당해 보자...

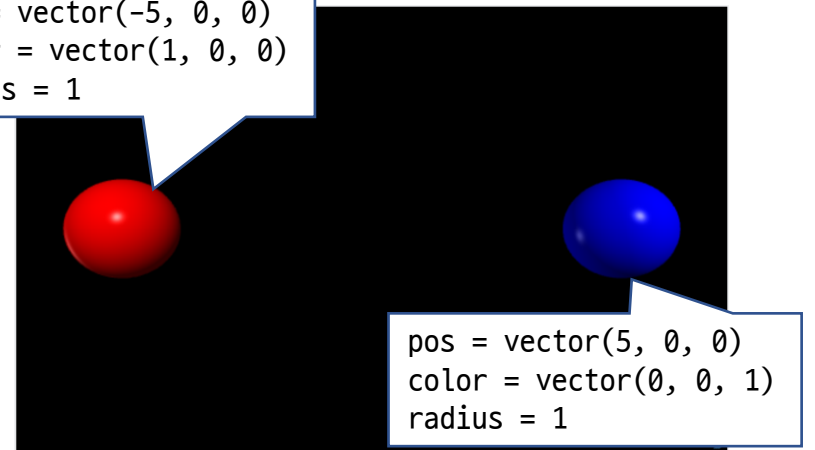
두 개의 구 객체를 생성하기

Web VPython 3.2

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

코드1

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

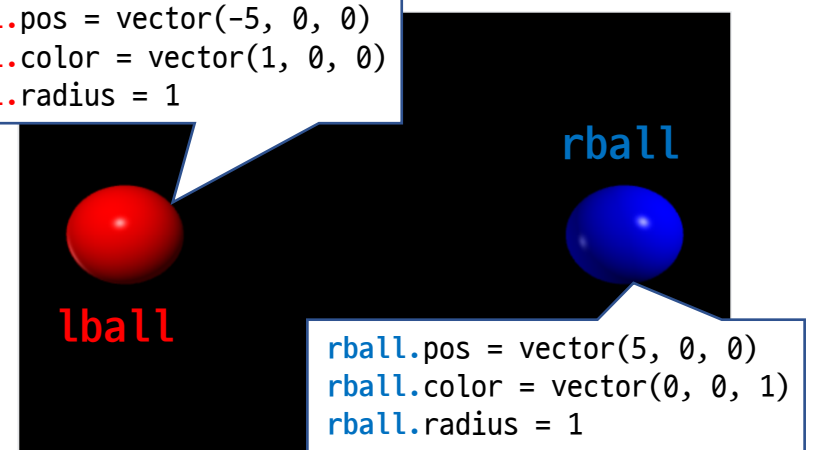


구 객체를 lball, rball 변수에 할당

Web VPython 3.2

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

```
lball.pos = vector(-5, 0, 0)
lball.color = vector(1, 0, 0)
lball.radius = 1
```



기존 구 객체의 색상이나 위치를 변경해 보기

구 객체를 lball, rball 변수에 할당

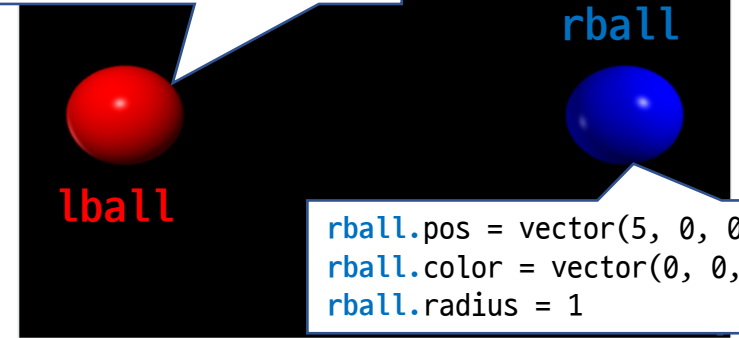
Web VPython 3.2

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

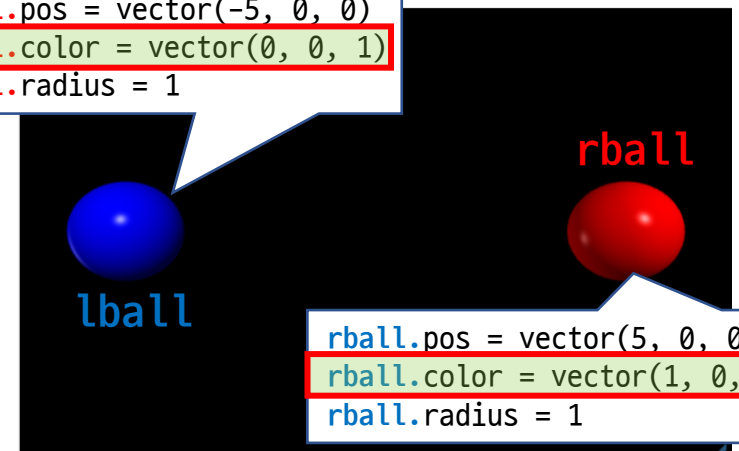
rate(0.5) # 0.5초 지연이 아님, (1/0.5)초 지연

```
lball.color = vector(0, 0, 1)
rball.color = vector(1, 0, 0)
```

```
lball.pos = vector(-5, 0, 0)
lball.color = vector(1, 0, 0)
lball.radius = 1
```



```
lball.pos = vector(-5, 0, 0)
lball.color = vector(0, 0, 1)
lball.radius = 1
```



기존 구 객체의 색상을 변경해 보기

Web VPython 3.2

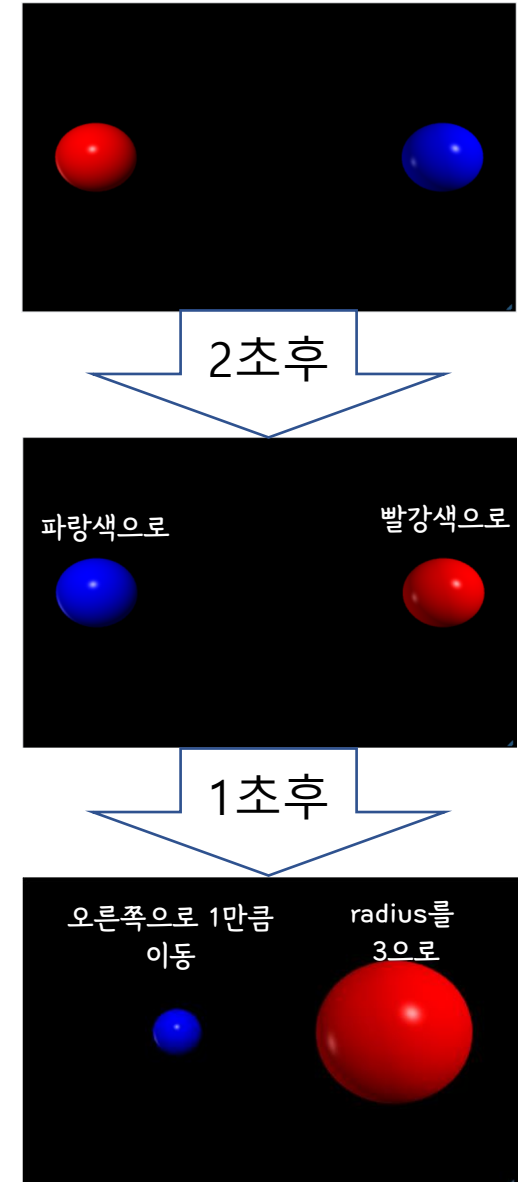
```
lball = sphere(pos=vector(-5,0,0), color=vector(1,0,0), radius=1)
rball = sphere(pos=vector(5,0,0), color=vector(0,0,1), radius=1)
```

rate(0.5) # 0.5초 지연이 아님, (1/0.5)초 지연

```
lball.color = vector(0, 0, 1)
rball.color = vector(1, 0, 0)
```

1초 지연 ??

```
# lball으로 오른쪽으로 1만큼 ??
# rball의 radius를 3으로
```



객체에 새로운 변수 추가

코드2

Web VPython 3.2

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

객체에 새로운 변수 추가

```
lball.cnsh = "충남과학고등학교"
```

cnsh라는 변수 추가

```
rball.mass = 1.0
```

질량 변수 추가

```
rball.velocity = vector(1, 0, 0)
```

속도 변수 추가

```
print(lball.cnsh)
print(rball.velocity)
```

```
lball.pos = vector(-5, 0, 0)
lball.color = vector(1, 0, 0)
lball.radius = 1
```

cnsh라는 변수가 추가

```
lball.cnsh = "충남과학고등학교"
```

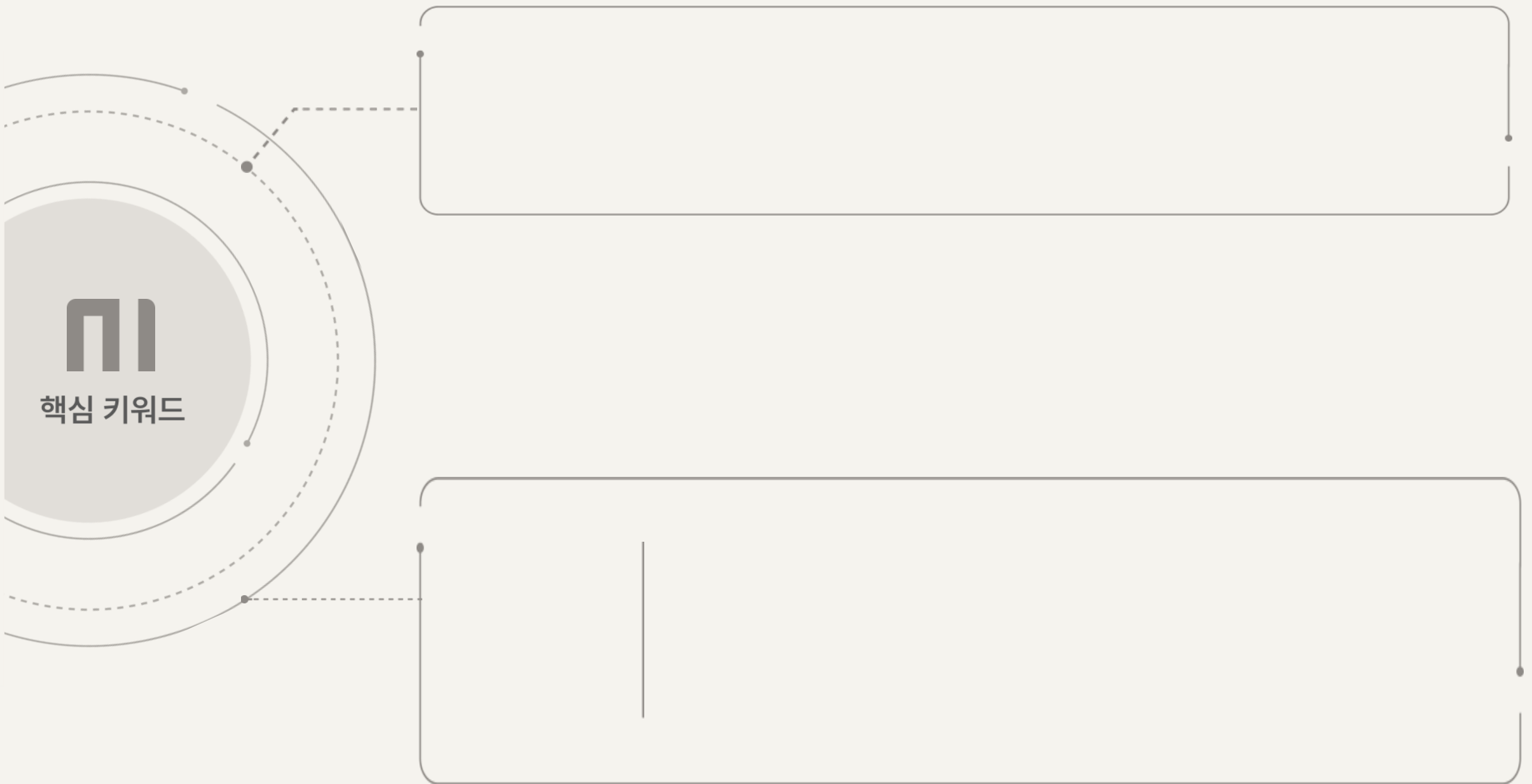


```
rball.pos = vector(5, 0, 0)
rball.color = vector(0, 0, 1)
rball.radius = 1
```

새로운 변수 추가

```
rball.mass = 1.0
```

```
rball.velocity = vector(1, 0, 0)
```

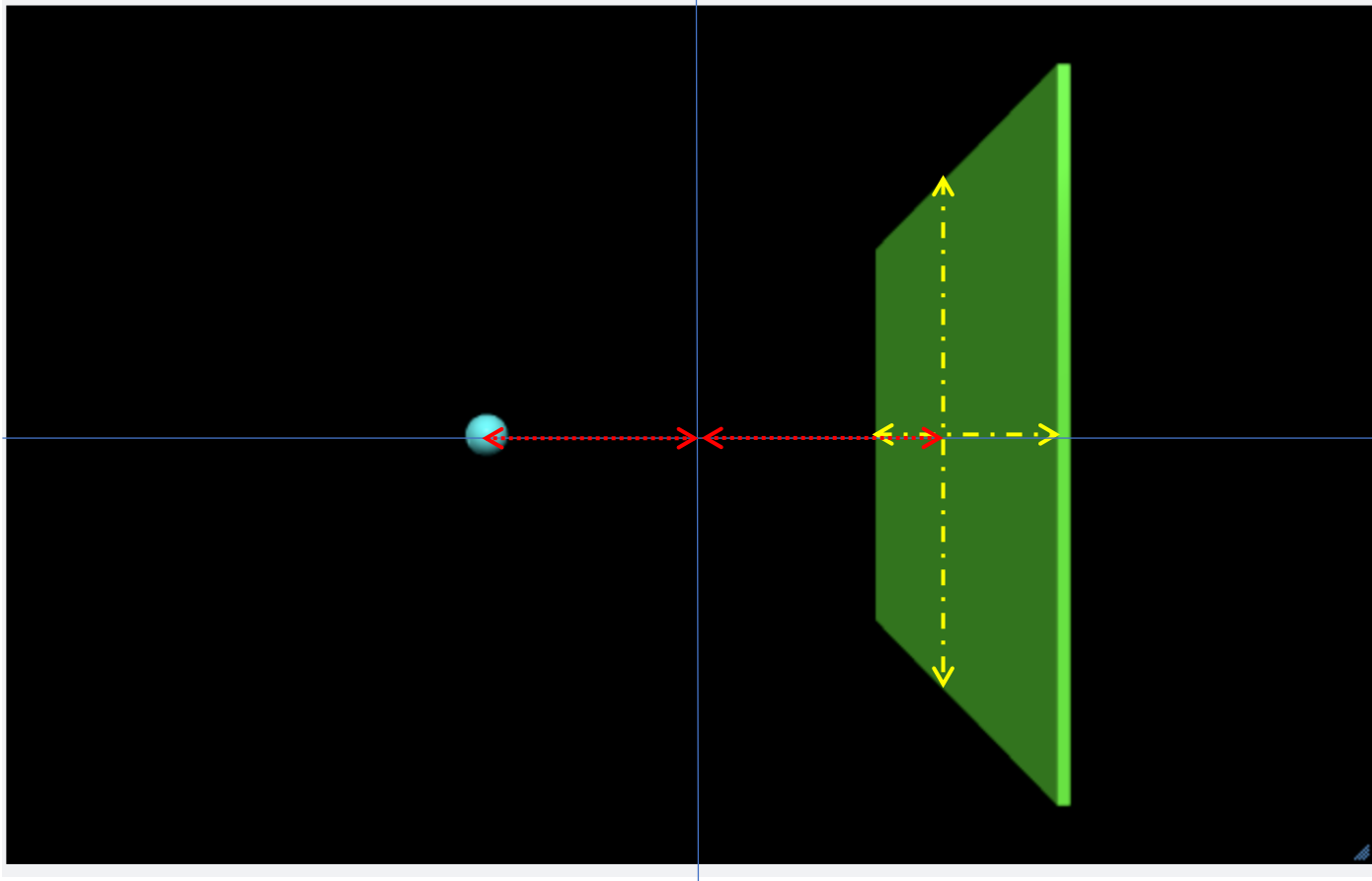



공간 만들기

코드3

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)
```



참고 : 사전 정의된 색상이 있음

Here are some examples of RGB colors, with names you can use in VPython:

vec(1,0,0) color.red

vec(1,1,0) color.yellow

vec(0,1,0) color.green

vec(1,0.6,0) color.orange

vec(0,0,1) color.blue

vec(0,1,1) color.cyan

vec(0.4,0.2,0.6) color.purple

vec(1,0,1) color.magenta

<https://www.glowscript.org/docs/VPythonDocs/color.html>

ball에 속도 변수 추가

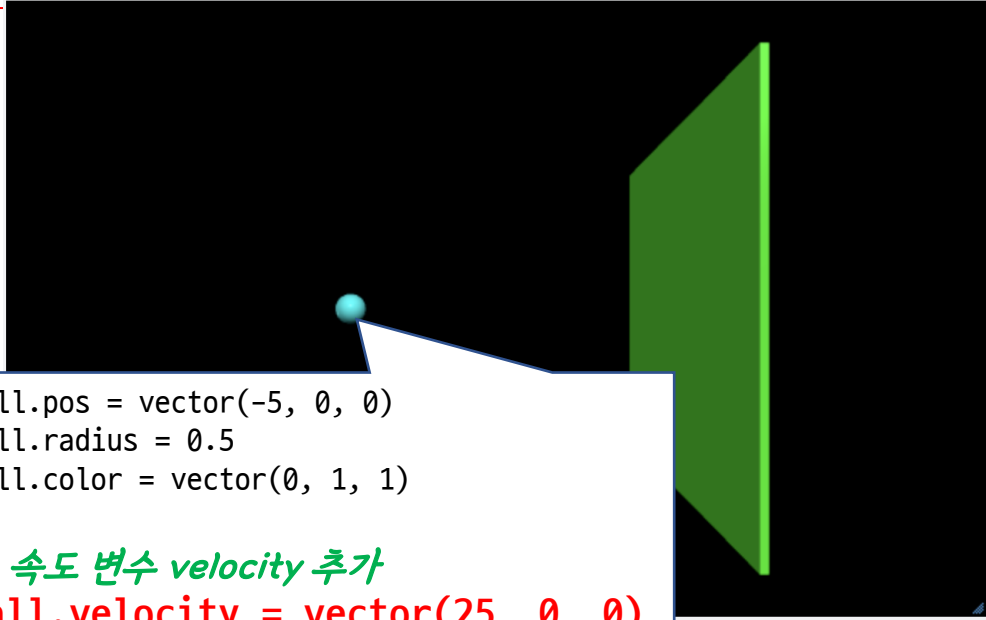
코드4

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)

t = 0
delta_t = 0.005
ball.velocity = vector(25,0,0)    # ball에 속도를 저장할 velocity 변수 추가
ball.pos = ball.pos + ball.velocity * delta_t    # ball의 처음 위치 + (delta_t초 동안 움직인 거리)
           # <-5, 0, 0> +    <25, 0, 0>    * 0.005
```

$$\begin{aligned} &\langle -5, 0, 0 \rangle + \langle 25, 0, 0 \rangle * 0.005 \\ &= \langle -5, 0, 0 \rangle + \langle 25*0.005, 0*0.005, 0*0.005 \rangle \\ &= \langle -5, 0, 0 \rangle + \langle 0.125, 0, 0 \rangle \\ &= \langle -4.875, 0, 0 \rangle \end{aligned}$$



```
ball.pos = vector(-5, 0, 0)
ball.radius = 0.5
ball.color = vector(0, 1, 1)
```

```
# 속도 변수 velocity 추가
ball.velocity = vector(25, 0, 0)
```

벡터 객체와 스칼라 값의 연산 예시

- $\text{vector}(1, 2, 3) + \text{vector}(2, 0, 0)$
= $\text{vector}(3, 2, 3)$
- $\text{vector}(1, 2, 3) - \text{vector}(2, 2, 2)$
= $\text{vector}(-1, 0, 1)$
- $\text{vector}(1, 2, 3) * 2$
= $\text{vector}(2, 4, 6)$
- $\text{vector}(1, 2, 3) / 2$
= $\text{vector}(0.5, 1, 1.5)$

ball을 연속적으로 움직이기

코드5

Web VPython 3.2

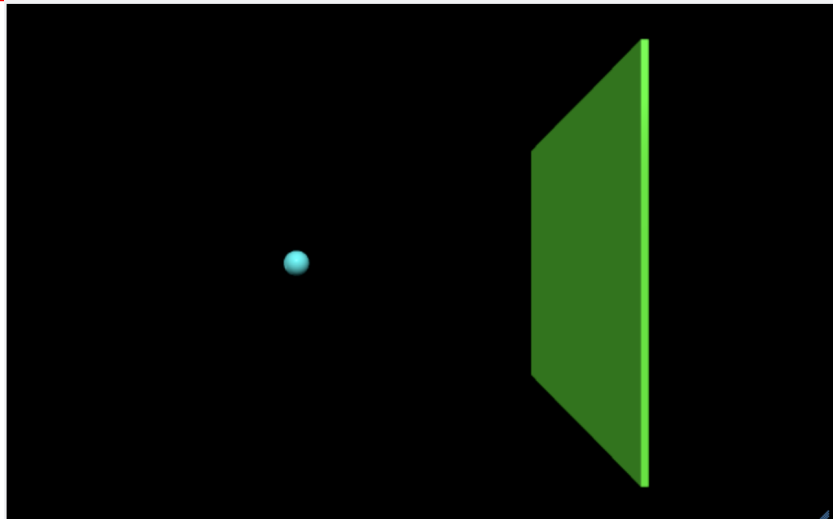
```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

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

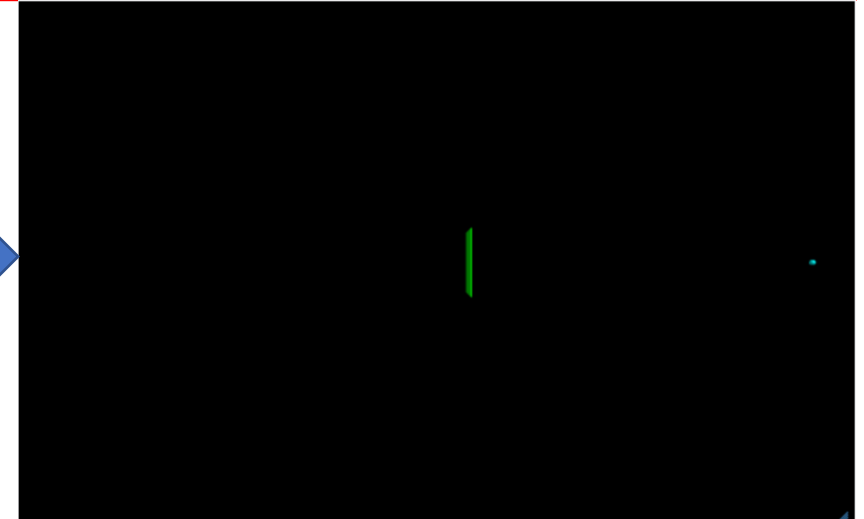
```
while t <= 3:
    ball.pos = ball.pos + ball.velocity * delta_t
    t = t + delta_t
```

최초

$(-5, 0, 0) + (25, 0, 0) * 0.005$
 $= (-5, 0, 0) + (25 * 0.005, 0 * 0.005, 0 * 0.005)$
 $= (-5, 0, 0) + (0.125, 0, 0)$
 $= (-4.875, 0, 0)$



한방에...(이게 아닌데???)



ball을 연속적으로 움직이기

Web VPython 3.2

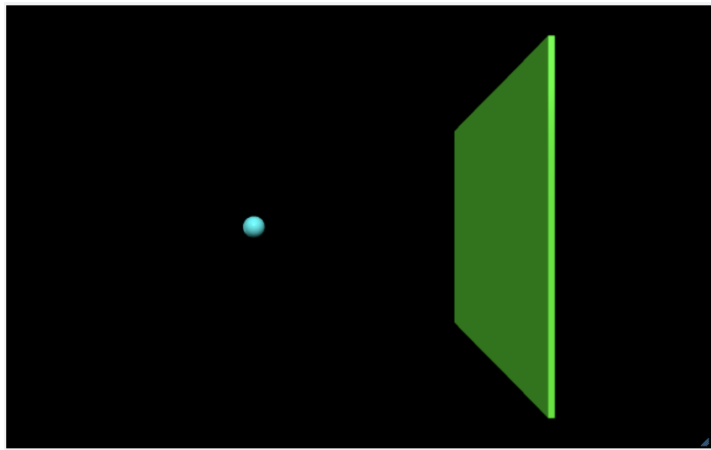
```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

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

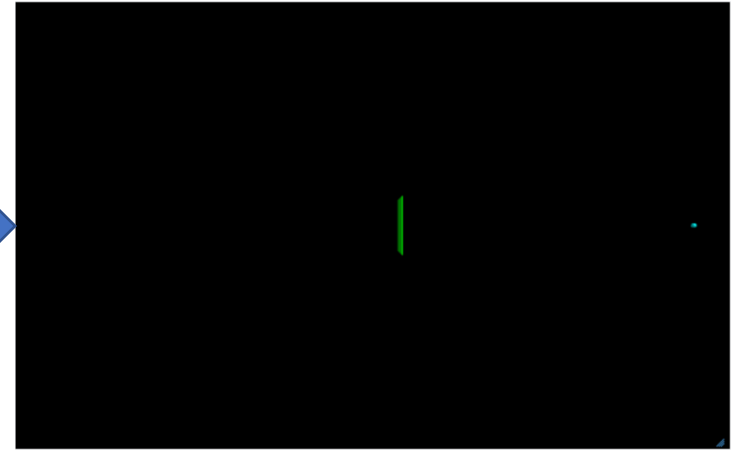
```
while t <= 3:
    rate(100)      # 1/100 sec초 지연
    ball.pos = ball.pos + ball.velocity * delta_t
    t = t + delta_t
```

t(시간)	ball.pos(공의 위치)	

0.000	<-5, 0, 0>	최초
0.005	<-4.875, 0, 0>	
0.010	<-4.75, 0, 0>	
0.015	<-4.625, 0, 0>	
... ..		
2.985	<69.625, 0, 0>	
2.990	<69.75, 0, 0>	
2.995	<69.875, 0, 0>	
3.000	<70, 0, 0>	마지막



1/100초에 한번씩 움직이도록...



ball이 WallR에 부딪히면 튜어나오기

코드6

Web Vpython 3.2

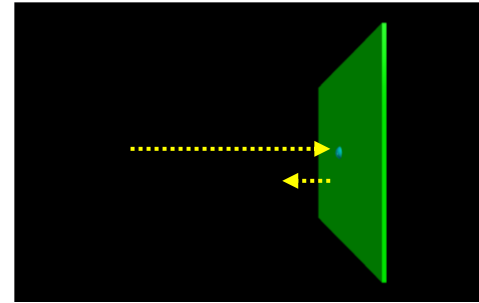
```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
```

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

```
while t <= 3:
    rate(100)
```

```
    if ball.pos.x > wallR.pos.x :
        ball.velocity.x = -ball.velocity.x
```

```
    ball.pos = ball.pos + ball.velocity * delta_t
    t = t + delta_t
```



t(시간)	ball.pos(공의 위치)	
0.000	$\langle -5, 0, 0 \rangle$	최초
0.005	$\langle -4.875, 0, 0 \rangle$	
0.010	$\langle -4.75, 0, 0 \rangle$	
0.015	$\langle -4.625, 0, 0 \rangle$	
...		
0.430	$\langle 5.750, 0, 0 \rangle$	
0.435	$\langle 5.875, 0, 0 \rangle$	
0.440	$\langle 6.000, 0, 0 \rangle$	
0.445	$\langle 6.125, 0, 0 \rangle$	

만약 ball.pos.X > wallR.pos.X 라면

$\langle ?, 0, 0 \rangle$ $\langle 6, 0, 0 \rangle$

$\text{ball.velocity} = \langle 25, 0, 0 \rangle * -1$
 $= \langle -25, 0, 0 \rangle$

왼쪽 벽 추가 및 왼쪽 벽에 부딪히면 튀어나오기

코드7

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
```


왼쪽 벽 추가 및 왼쪽 벽에 부딪히면 튕어나오기

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
```

```
if ball.pos.x > wallR.pos.x or ball.pos.x < wallL.pos.x:
    ball.velocity.x = -ball.velocity.x
```

공의 궤적 자취 남기기

코드8

Web VPython 3.2

```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan, make_trail=True, retain=30)
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 or ball.pos.x < wallL.pos.x:
        ball.velocity.x = -ball.velocity.x
    ball.pos = ball.pos + ball.velocity*deltat
    t = t + deltat
```

모든 벽 추가하기

코드9

Web VPython 3.2

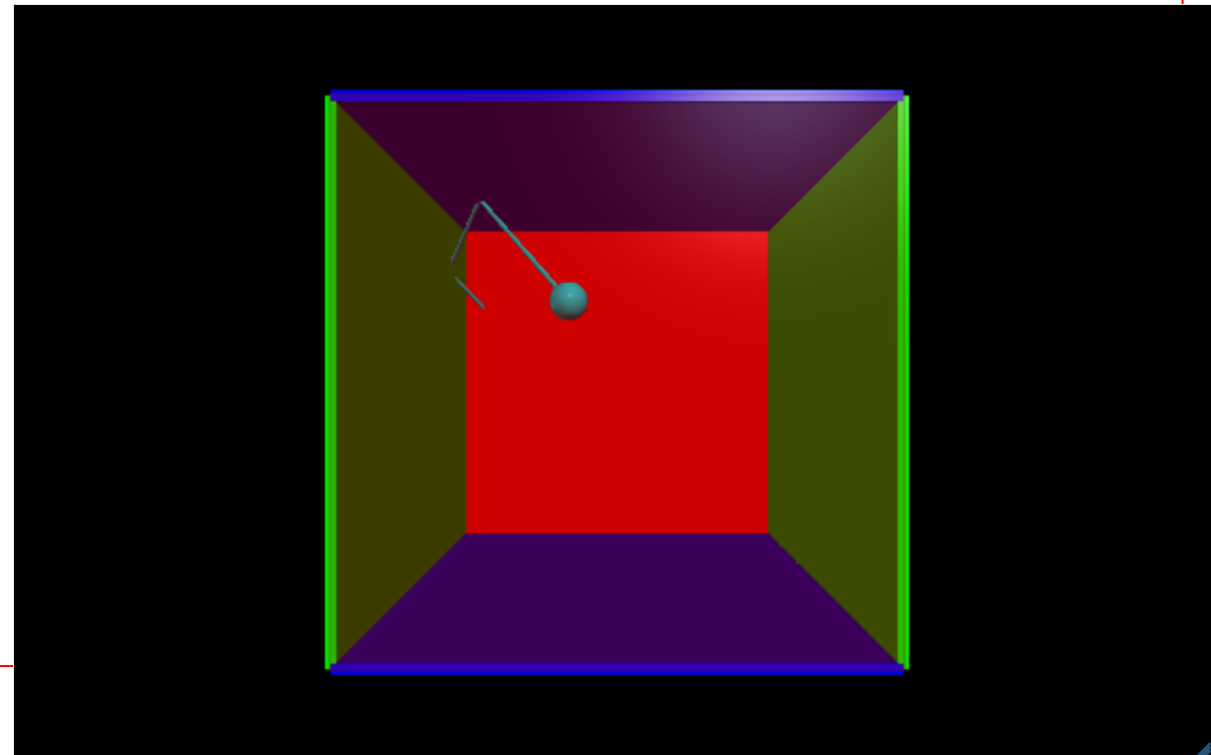
```
ball = sphere(pos=vector(-5,0,0), radius=0.5, color=color.cyan, make_trail=True, retain=30)
wallR = box(pos=vector(6,0,0), size=vector(0.2,12,12), color=color.green)
wallL = box(pos=vector(-6,0,0), size=vector(0.2,12,12), color=color.green)
wallT = box(pos=vector(0,6,0), size=vector(12,0.2,12), color=color.blue)
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, 0, 0)
ball.velocity = vector(25, 30, -35)
delta_t = 0.005
t = 0
```

```
while True:
    rate(100)

    if ball.pos.x > wallR.pos.x or ball.pos.x < wallL.pos.x:
        ball.velocity.x = -ball.velocity.x
    if ball.pos.y > wallT.pos.y or ball.pos.y < wallB.pos.y:
        ball.velocity.y = -ball.velocity.y
    if ball.pos.z > wallF.pos.z or ball.pos.z < wallRE.pos.z:
        ball.velocity.z = -ball.velocity.z

    ball.pos = ball.pos + ball.velocity*delta_t
    t = t + delta_t
```



탑재된 코드

코드10

Web VPython 3.2

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

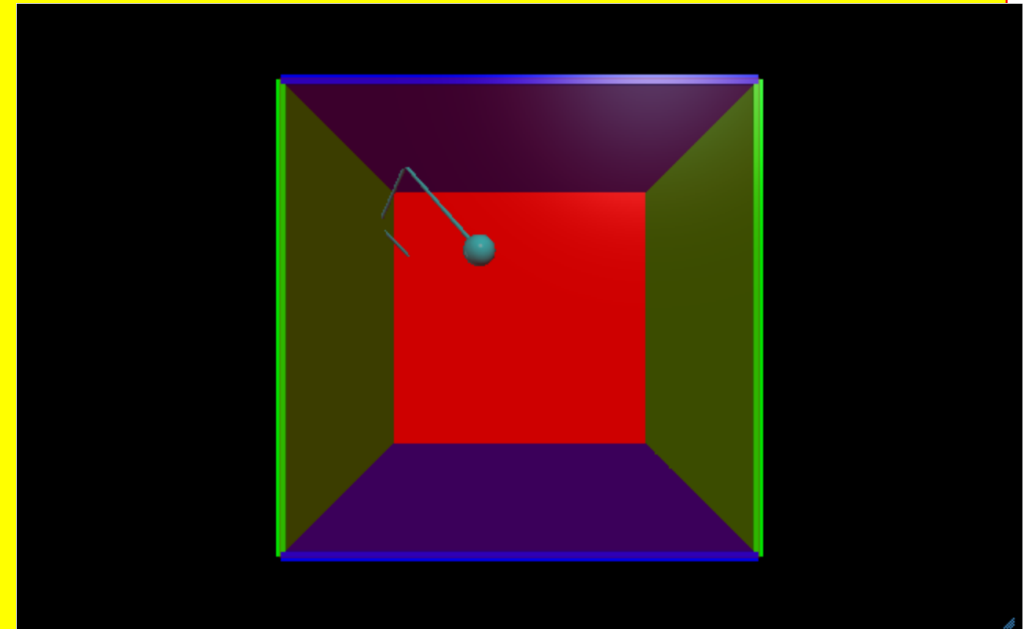
wallR = box(pos=vector( side, 0, 0), size=vector(thk, s2, s3), color = color.red)
wallL = box(pos=vector(-side, 0, 0), size=vector(thk, s2, s3), color = color.red)
wallB = box(pos=vector(0, -side, 0), size=vector(s3, thk, s3), color = color.blue)
wallT = box(pos=vector(0, side, 0), size=vector(s3, thk, 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
def move():
    rate(200, move)
    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

move()
```





그래프 그리기

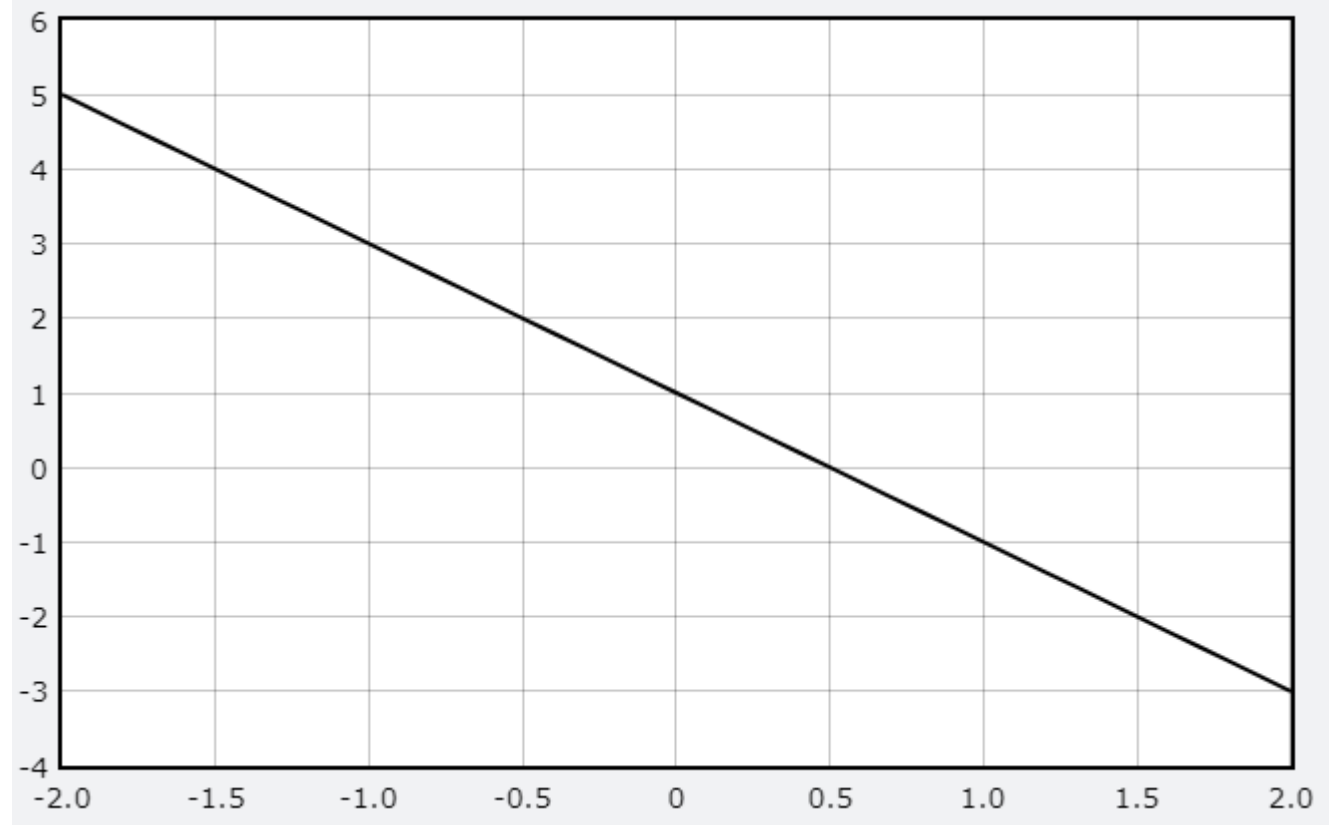
|

그래프 그리기

코드11

Web VPython 3.2

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

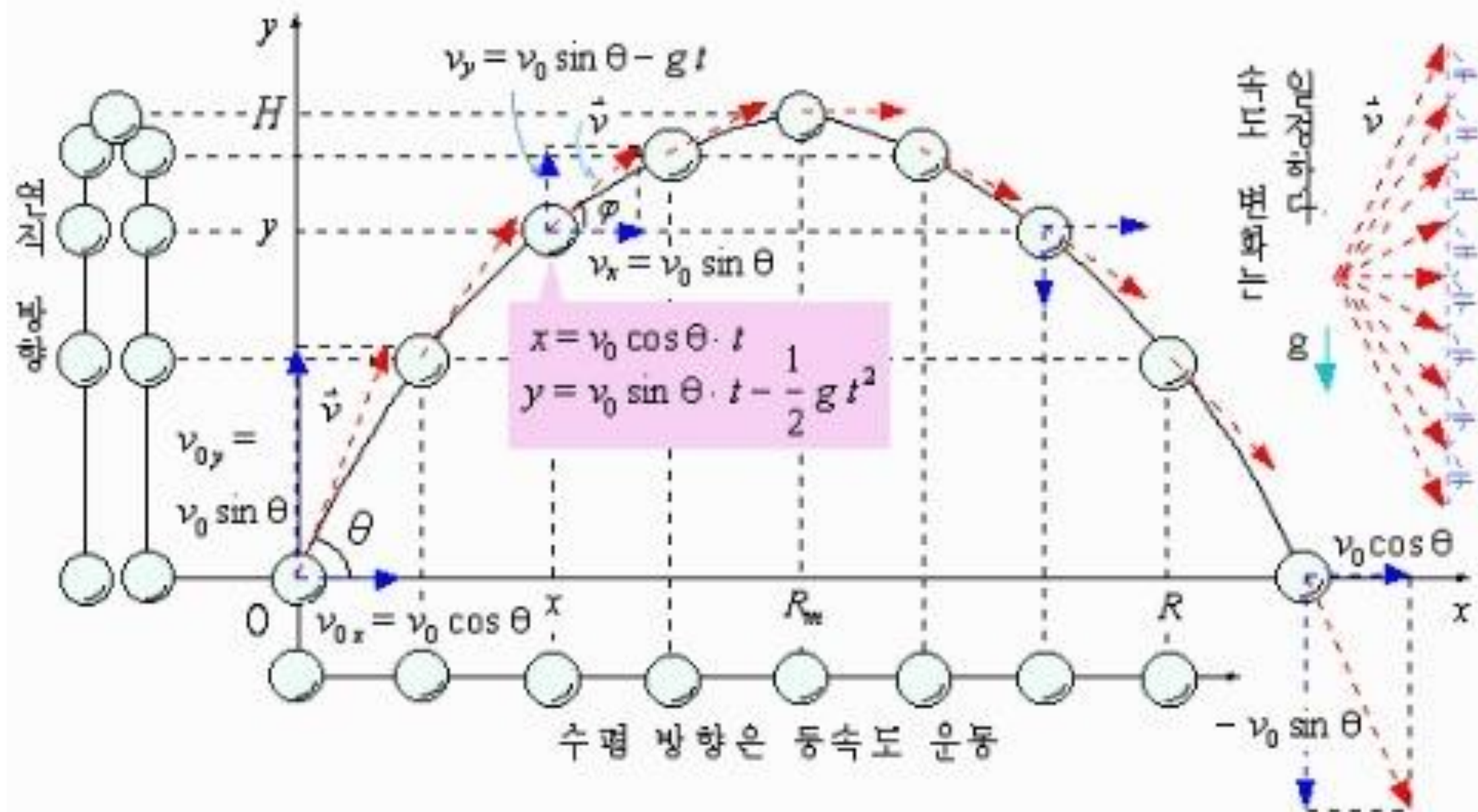


코드12

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

코드13

Web VPython 3.2

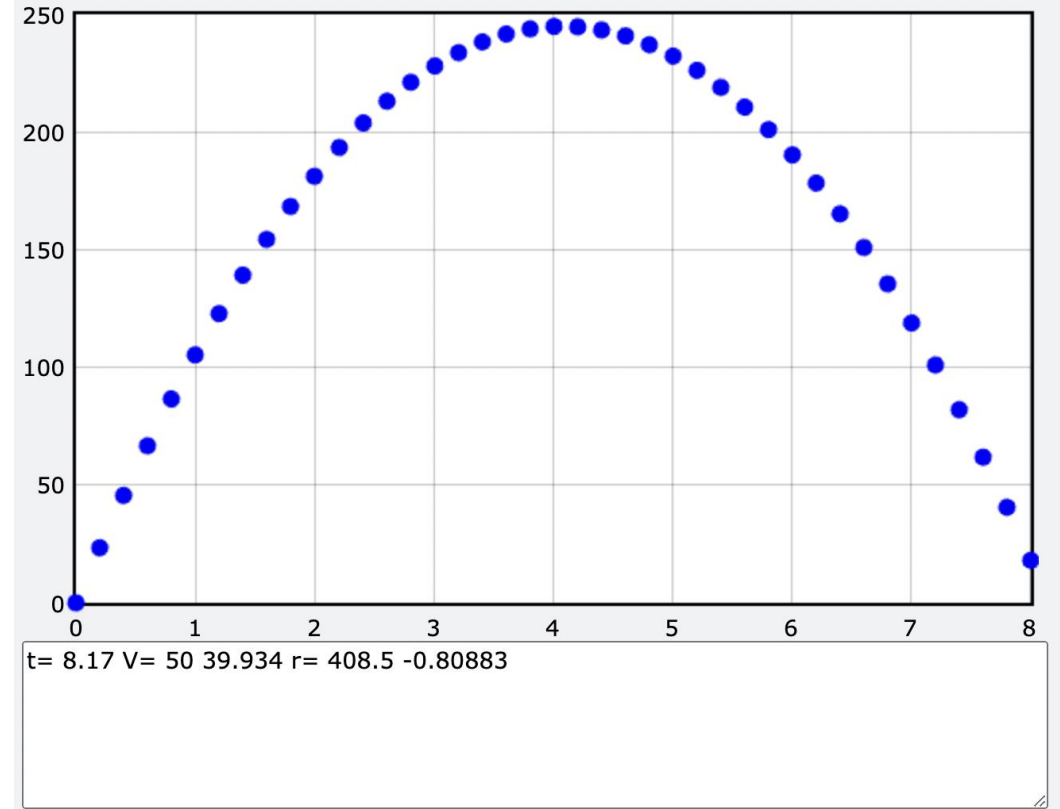
```
m = 0.1                # Kg
g = vector(0,-9.8,0)    # m/s^2; 중력가속도
r0 = vector(0,0,0)      # m
v0 = vector(50,120,0)   # m/s
h = gdots(color=color.blue, size=10)
dt = 0.01

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
```

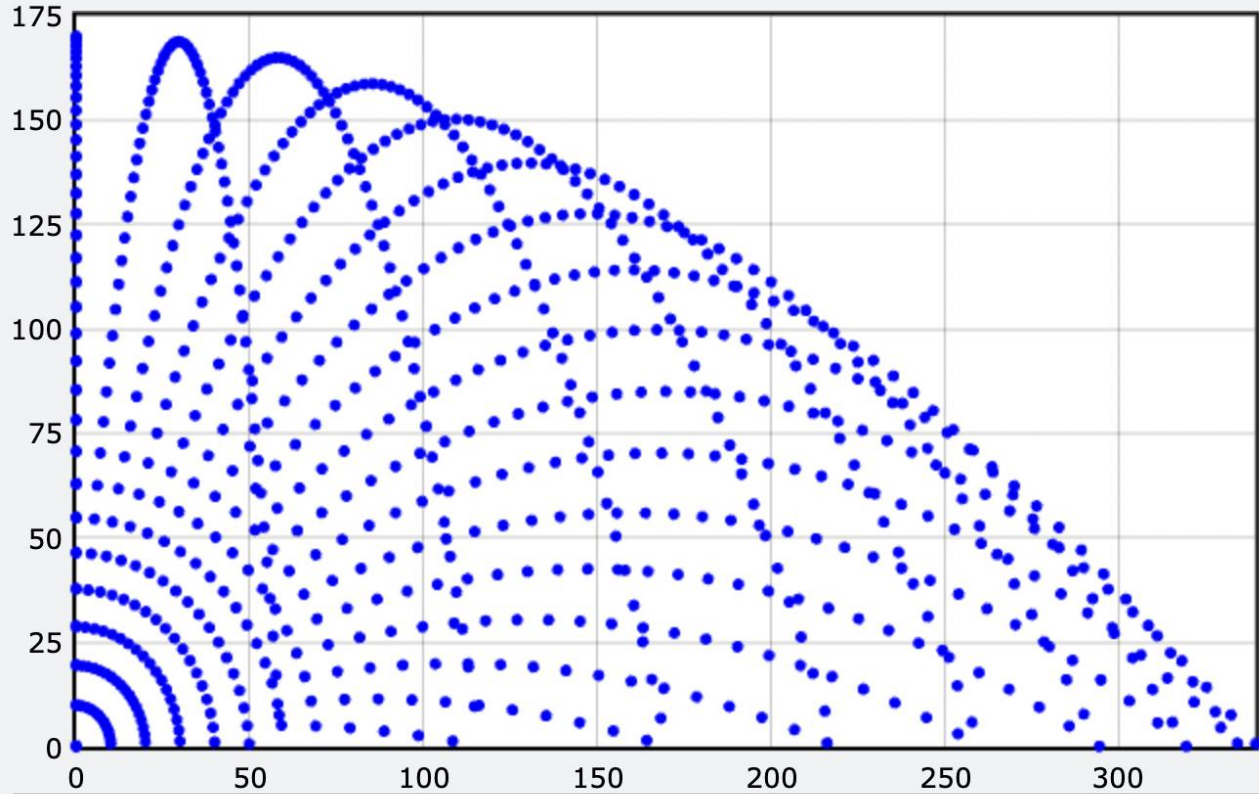


코드13

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
```



```
각도= 75 체공시간= 6.6 수평도달거리= 170.821
각도= 80 체공시간= 6.7 수평도달거리= 116.344
각도= 85 체공시간= 6.8 수평도달거리= 59.2659
각도= 90 체공시간= 6.9 수평도달거리= 4.22503e-14
```