In [ ]:

```
from IPython.display import display, Math
from sympy import Eq, solve, latex
from sympy.physics.units.systems import SI
from sympy.physics.units import Quantity, length, mass, time
```

In [ ]:

```
# デフォルトで用意されている変数 t, g と
# 物体 A に紐付く変数 x_A, y_A, v_{Ax}, v_{Ay} を生成
t = Quantity('t', latex_repr='t')
SI.set_quantity_dimension(t, dimension=time)
g = Quantity('g', latex_repr='g')
SI.set_quantity_dimension(g, dimension=length/time**2)
xA = Quantity('xA', latex_repr='x_A')
SI.set_quantity_dimension(xA, dimension=length)
yA = Quantity('yA', latex_repr='y_A')
SI.set_quantity_dimension(yA, dimension=length)
vAx = Quantity('vAx', latex_repr='v_{Ax}')
SI.set_quantity_dimension(vAx, dimension=length/time)
vAy = Quantity('vAy', latex_repr='v_{Ay}')
SI.set_quantity_dimension(vAy, dimension=length/time)
```

In [ ]:

```
# v_{0x} と v_{0y} を生成
v0x = Quantity('v0x', latex_repr='v_{0x}')
SI.set_quantity_dimension(v0x, dimension=length/time)
v0y = Quantity('v0y', latex_repr='v_{0y}')
SI.set_quantity_dimension(v0y, dimension=length/time)
```

In [ ]:

```
# 各変数が満たすべき方程式を作成
eqs = [Eq(vAx, v0x), Eq(vAy, v0y - g * t), Eq(xA, v0x * t), Eq(yA, v0y * t - g * t / 2)]
for eq in eqs: display(eq)
```

$$v_{Ax} = v_{0x}$$

$$v_{Ay} = -gt + v_{0y}$$

$$x_A = tv_{0x}$$

$$y_A = -\frac{gt}{2} + tv_{0y}$$

In [ ]:

```
# 初期値を代入
param = {t: 2, g: 2, v0x: 3, v0y: 4}
for eq in eqs: display(Math(f'{latex(eq.lhs)} = {latex(eq.rhs.subs(param))}'))
```

$$v_{Ax} = 3$$

$$v_{Ay} = 0$$

$$x_A = 6$$

$$y_A = 6$$