Sure, here are some alternative examples of keyframe animation configurations with math formulas:

- Translation:

Code snippet

```
translation_x: "0:(0), 100:(10), 200:(-10), 300:(0)"
```

This animation would move the object 10 units to the right, then 10 units to the left, and then back to its original position.

Code snippet

```
translation_y: "0:(0), 100:(sin(2*pi*t/100)), 200:(cos(2*pi*t/100)), 
300:(0)"
```

This animation would move the object up and down in a sinusoidal motion.

- Rotation:

Code snippet

```
rotation_3d_x: "0:(0), 100:(pi/2), 200:(-pi/2), 300:(0)"
```

This animation would rotate the object 90 degrees around the x-axis, then 90 degrees in the opposite direction, and then back to its original position.

Code snippet

```
rotation_3d_y: "0:(0), 100:(pi), 200:(-pi), 300:(0)"
```

This animation would rotate the object 180 degrees around the y-axis, then 180 degrees in the opposite direction, and then back to its original position.

- Scaling:

Code snippet

```
scaling_x: "0:(1), 100:(2), 200:(1), 300:(1)"
```

This animation would scale the object to twice its original size, then back to its original size.

Code snippet

```
scaling_y: "0:(1), 100:(sin(2*pi*t/100)), 200:(cos(2*pi*t/100)),
300:(1)"
```

This animation would scale the object up and down in a sinusoidal motion.

These are just a few examples of how you can use math formulas to create keyframe animation configurations. The possibilities are endless!