x, y, z

```python
import pyquaternion


# Create a quaternion representing a rotation of +90 degrees about positive y axis.
my_quaternion = pyquaternion.Quaternion(axis=[0, 1, 0], degrees=90)

my_vector = [0, 0, 4]
my_rotated_vector = my_quaternion.rotate(my_vector)

print('\nBasic Rotation')
print('-------------')
print('My Vector: {}'.format(my_vector))
print('Performing rotation of {angle} deg about {axis}'.format(angle=my_quaternion.degrees, axis=my_quaternion.axis))
print('My Rotated Vector: {}'.format(my_rotated_vector))


# Create another quaternion representing no rotation at all
null_quaternion = pyquaternion.Quaternion(axis=[0, 1, 0], angle=0)

print('\nInterpolated Rotation')
print('---------------------')

# The following will create a sequence of 9 intermediate quaternion rotation objects
for q in pyquaternion.Quaternion.intermediates(null_quaternion, my_quaternion, 9, include_endpoints=True):
    my_interpolated_point = q.rotate(my_vector)
    print('My Interpolated Point: {point}\t(after rotation of {angle} deg about {axis})'.format(
        point=my_interpolated_point, angle=round(q.degrees, 4), axis=q.axis
    ))

print('Done!')
```

y转90°

0,0,4 [illegible]

format: y转90°

String Rotation

(q [illegible],

q.degrees

q.axis.