

## rotate3DVectorTest

```
% create column vectors with simple direction for rotations along the axes
% without tilts in other achses.
x = [-1; 0; 0];
y = [0; -1; 0];
z = [0; 0; -1];

% set angle step width in degree to rotate at choosen axes (x, y, or z)
angle = 90;
```

### Test 1: output dimensions

```
rotated = rotate3DVector(x, 0, 0, angle);
assert(isequal(size(rotated), [3, 1]))
rotated = rotate3DVector([x x x x x], 0, 0, angle);
assert(isequal(size(rotated), [3, 6]))
```

### Test 2: rotate vectors in x-axes

```
rotated = rotate3DVector([x y z], 0, 0, 0); % 0 degree
assert(isequal(rotated, [-1 0 0; 0 -1 0; 0 0 -1]))

rotated = rotate3DVector([x y z], angle, 0, 0); % 90 degree
assert(isequal(rotated, [-1 0 0; 0 0 1; 0 -1 0]))

rotated = rotate3DVector([x y z], 2 * angle, 0, 0); % 180 degree
assert(isequal(rotated, [-1 0 0; 0 1 0; 0 0 1]))

rotated = rotate3DVector([x y z], 3 * angle, 0, 0); % 270 degree
assert(isequal(rotated, [-1 0 0; 0 0 -1; 0 1 0]))

rotated = rotate3DVector([x y z], 4 * angle, 0, 0); % 360 degree
assert(isequal(rotated, [-1 0 0; 0 -1 0; 0 0 -1]))
```

### Test 3: rotate vectors in y-axes

```
rotated = rotate3DVector([x y z], 0, 0, 0); % 0 degree
assert(isequal(rotated, [-1 0 0; 0 -1 0; 0 0 -1]))

rotated = rotate3DVector([x y z], 0, angle, 0); % 90 degree
assert(isequal(rotated, [0 0 -1; 0 -1 0; 1 0 0]))

rotated = rotate3DVector([x y z], 0, 2 * angle, 0); % 180 degree
assert(isequal(rotated, [1 0 0; 0 -1 0; 0 0 1]))

rotated = rotate3DVector([x y z], 0, 3 * angle, 0); % 270 degree
assert(isequal(rotated, [0 0 1; 0 -1 0; -1 0 0]))

rotated = rotate3DVector([x y z], 0, 4 * angle, 0); % 360 degree
assert(isequal(rotated, [-1 0 0; 0 -1 0; 0 0 -1]))
```

### Test 4: rotate vectors in z-axes

```
rotated = rotate3DVector([x y z], 0, 0, 0); % 0 degree
assert(isequal(rotated, [-1 0 0; 0 -1 0; 0 0 -1]))

rotated = rotate3DVector([x y z], 0, 0, angle); % 90 degree
assert(isequal(rotated, [0 1 0; -1 0 0; 0 0 -1]))

rotated = rotate3DVector([x y z], 0, 0, 2 * angle); % 180 degree
assert(isequal(rotated, [1 0 0; 0 1 0; 0 0 -1]))

rotated = rotate3DVector([x y z], 0, 0, 3 * angle); % 270 degree
assert(isequal(rotated, [0 -1 0; 1 0 0; 0 0 -1]))

rotated = rotate3DVector([x y z], 0, 0, 4 * angle); % 360 degree
assert(isequal(rotated, [-1 0 0; 0 -1 0; 0 0 -1]))
```