# Exercise

`Create a Python function that will take a vector as input and output a unit vector in the same direction. What happens when you input the zeros vector?`

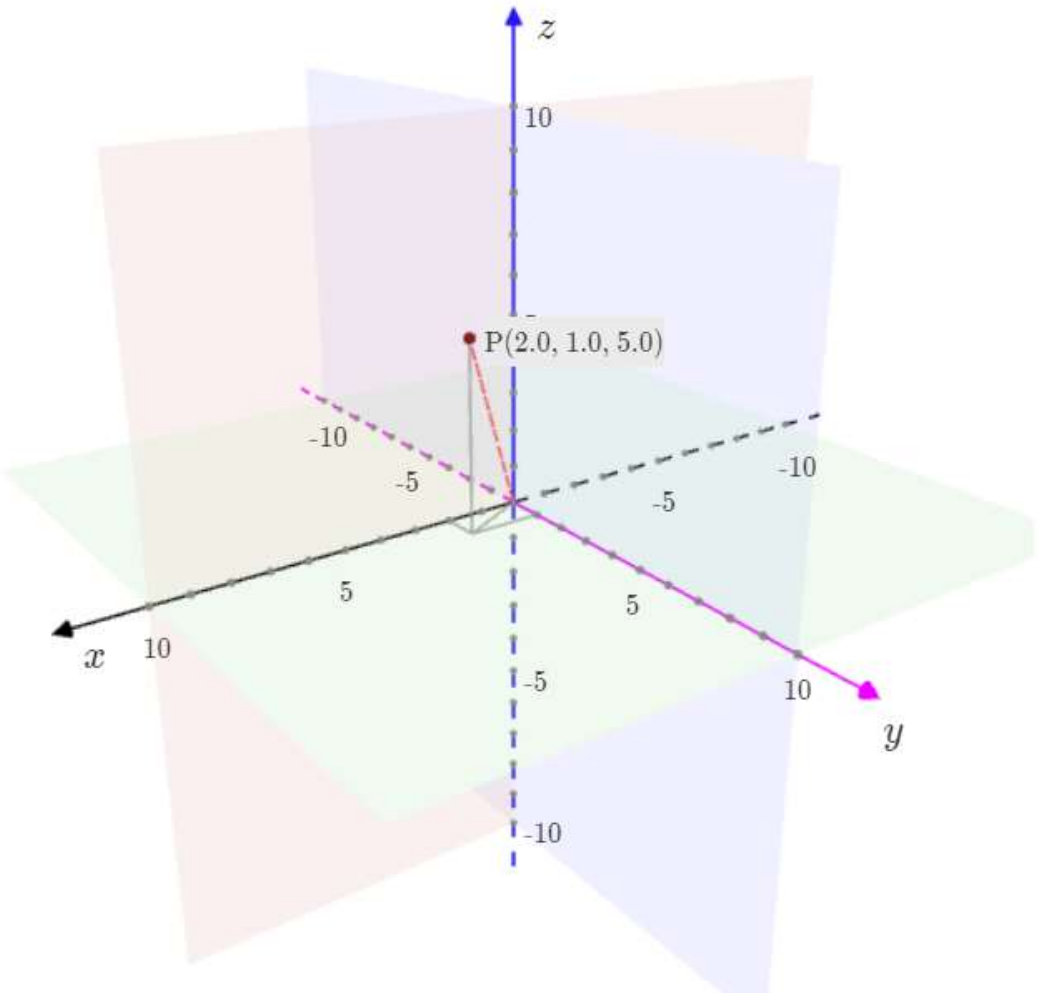
To do this... We'll have to recap a few stuff we've been on over the past weeks. Let's begin.

## Vectors

A **vector** is a quantity that has both **magnitude** (length) and **direction**. For example, a vector in two dimensions can be represented as `[3, 4]`, where:

- `3` is the component in the x-direction
- `4` is the component in the y-direction

We can also represent vectors in higher dimensions, such as `[1, 2, 3]` in three dimensions (x, y, z).



Example of a 3 dimensional Vector.

# Unit Vectors

A **unit vector** is a vector that has a **magnitude of 1** but points in the same direction as the original vector.

For example, if we start with a vector `[3, 4]`, the unit vector will point in the same direction but will have a length of 1.

The process of converting a vector to a unit vector is called **normalization**.

# Step 1: Calculate the Magnitude of a Vector

The **magnitude** (or length) of a vector `[a, b]` in two dimensions is calculated using the formula:

$$\text{Magnitude} = \sqrt{a^2 + b^2}$$

For a three-dimensional vector `[a, b, c]`, the formula is:

$$\text{Magnitude} = \sqrt{a^2 + b^2 + c^2}$$

Let's calculate the magnitude of a vector in Python.

In [ ]:
```python
import numpy as np

# Example vector
vector = np.array([3, 4])

# Calculate the magnitude
magnitude = np.linalg.norm(vector)
print("Magnitude of the vector:", magnitude)
```

```
Magnitude of the vector: 5.0
```

Remember our studies on Numpy. Numpy is used here for two main purposes: creating arrays and performing numerical operations.

`np.array([3, 4])` creates a NumPy array from the list [3, 4]

`np.linalg.norm(vector)` calculates the magnitude (or length) of the vector

# Step 2: Normalizing the Vector

To create a **unit vector**, we divide each component of the vector by its magnitude.

For a vector `[a, b]`, the unit vector `[u1, u2]` is given by:

$$u1 = \frac{a}{\text{Magnitude}}$$

$$u2 = \frac{b}{\text{Magnitude}}$$

Let's implement step in Python.

In [ ]:
```python
# Check if the magnitude is zero before normalizing
if magnitude == 0:
    raise ValueError("Cannot compute the unit vector of a zero vector.")

# Normalize the vector
unit_vector = vector / magnitude
print("Unit vector:", unit_vector)
```

```
Unit vector: [0.6 0.8]
```

## Unique Case: Handling the Zero Vector

A **zero vector** is a vector where all components are zero, such as `[0, 0]` in two dimensions or `[0, 0, 0]` in three dimensions.

The magnitude of a zero vector is zero, and since dividing by zero is undefined, we **cannot normalize a zero vector**. In such cases, the program should give a clear message or error indicating that normalization is not possible.

## Final Step: Creating a Function to Convert a Vector to a Unit Vector

Let's wrap everything into a Python function that:

1. Calculates the magnitude of the input vector.
2. Checks if the magnitude is zero (and handles the zero vector case).
3. Normalizes the vector if the magnitude is non-zero.

In [ ]:
```python
def to_unit_vector(vector):
    # Calculate the magnitude of the vector
    magnitude = np.linalg.norm(vector)

    # Check if the magnitude is zero (zero vector)
    if magnitude == 0:
        raise ValueError("Cannot compute the unit vector of a zero vector.")

    # Normalize the vector
    unit_vector = vector / magnitude
    return unit_vector

# Test the function
test_vector = np.array([3, 4])
unit_vector = to_unit_vector(test_vector)
print("Unit vector:", unit_vector)
```

```
Unit vector: [0.6 0.8]
```

## Testing the Function with a Zero Vector

Now, let's see what happens when we try to normalize a zero vector, such as `[0, 0]`.

In [ ]:
```python
# Test with a zero vector
zero_vector = np.array([0, 0])

try:
    unit_vector_zero = to_unit_vector(zero_vector)
    print("Unit vector:", unit_vector_zero)
except ValueError as e:
    print("Error:", e)
```

```
Error: Cannot compute the unit vector of a zero vector.
```

# Conclusion

- A unit vector has a magnitude of 1 and points in the same direction as the original vector.
- The process of converting a vector to a unit vector is called **normalization**.
- The zero vector cannot be normalized because its magnitude is zero, and division by zero is undefined.