Aim

- Understand the basics of tensors
- Learn how to create tensors

In [3]:
```
import torch
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from IPython.display import Image
from IPython.core.display import HTML
```

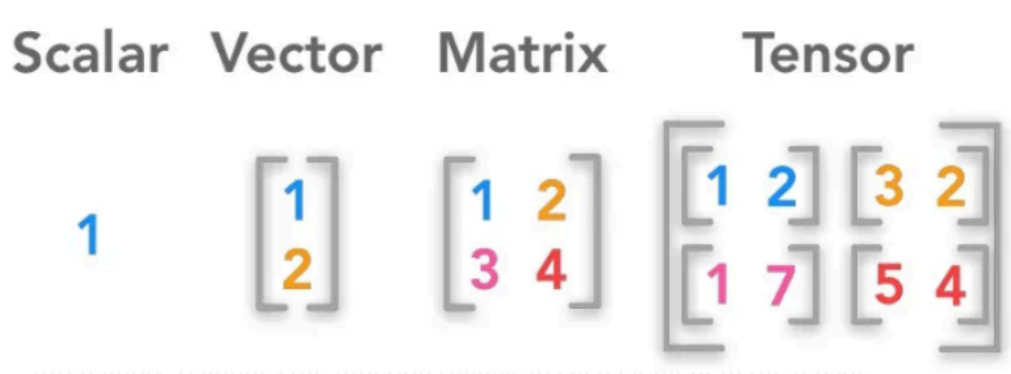
Introduction to Tensors

- A tensor is a datatype that represents multidimensional numerical data.
- torch.tensor creates an instance of a tensor.

Tensors can represent scalars (0-dimension), vectors (1-dimension), matrices (2-dimension), and higher-dimensional tensors (>2 dimensions).

In [4]:
```
TENSOR_DIM_VISUAL = "/Users/sunil/Yashvi/PyTorch/img/Scalar_Vector_Mat
Image(filename = TENSOR_DIM_VISUAL, width=500, height=500)
```

Out[4]:



Scalar

In [5]:
```
scalar = torch.tensor(7)
# Type of the scalar
print(f'Type of the scalar: {type(scalar)}')
# Get the dimension
print(f'Dimension: {scalar.ndim}')
# Get Python datatype from a scalar (int)
print(f'Get the value of a scalar: {scalar.item()}')
```

```
Type of the scalar: <class 'torch.Tensor'>
Dimension: 0
Get the value of a scalar: 7
```

Vector

In [6]:
```
vector = torch.tensor([1, 2, 3, 4])
# Get the dimension
print(f'Dimension: {vector.ndim}')
# Shape of a vector (size)
print(f'Shape of a vector: {vector.shape}')
# Access a scalar from vector
print(f'Access a scalar from vector: {vector[3].item()}')
```

```
Dimension: 1
Shape of a vector: torch.Size([4])
Access a scalar from vector: 4
```

Matrix

In [7]:
```
MATRIX = torch.tensor([[1, 2, 12], [3, 4, 34]])
# Get the dimension
print(f'Dimension: {MATRIX.ndim}')
# Shape of a matrix (size)
print(f'Shape of a matrix: {MATRIX.shape}')
```

```
Dimension: 2
Shape of a matrix: torch.Size([2, 3])
```

Tensor

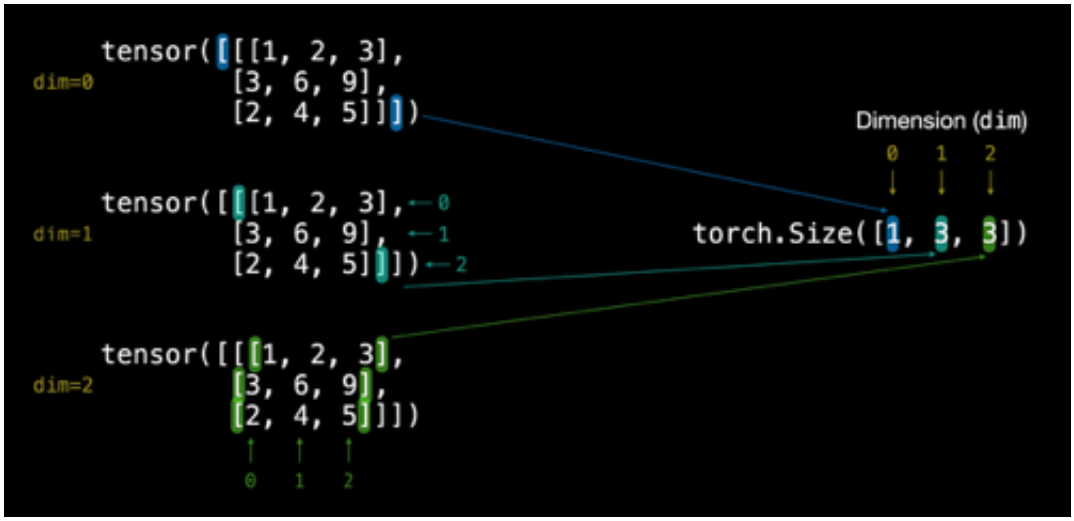
In [8]:
```
TENSOR = torch.tensor([[[1, 2, 3],
                        [3, 6, 9],
                        [2, 4, 5]]])
print(f'Dimention: {TENSOR.ndim}')
print(f'Shape of a vector: {TENSOR.shape}')
```

```
Dimention: 3
Shape of a vector: torch.Size([1, 3, 3])
```

Below diagram shows how to visualize the dimentions and shape of a tensor

In [9]:
```
TENSOR_DIM_VISUAL = "/Users/sunil/Yashvi/PyTorch/img/tensor_dimention_
Image(filename = TENSOR_DIM_VISUAL, width=500, height=500)
```

Out[9]:



- The shape of a vector is the size of the tensor across a specific dimension.
- The number of open square brackets indicates the dimension of a tensor.
- The number of elements within each square bracket forms the size of the tensor.
- For scalars and vectors, a lowercase variable is used. For matrices and tensors, an uppercase variable name is used.

Tensor Constructors

Random Tensors

In [10]:
```
def print_dim_size(tensor):
    print(f'Tensor: {tensor}')
    print(f'Dimension: {tensor.ndim}')
    print(f'Shape: {tensor.shape}')
    print('-------------------------#######################----------')

# Random vector
random_vector = torch.rand(3)
print_dim_size(random_vector)
# Random matrix
RANDOM_MATRIX = torch.rand(3, 4)
print_dim_size(RANDOM_MATRIX)
# Random tensor
RANDOM_TENSOR = torch.rand(3, 4, 5)
print_dim_size(RANDOM_TENSOR)
```

```
Tensor: tensor([0.2518, 0.6193, 0.6279])
Dimension: 1
Shape: torch.Size([3])
-------------------------#######################--------------------
-----
Tensor: tensor([[0.1161, 0.1004, 0.3043, 0.4153],
        [0.2817, 0.9007, 0.6152, 0.2507],
        [0.0953, 0.0513, 0.2774, 0.7158]])
Dimension: 2
Shape: torch.Size([3, 4])
-------------------------#######################--------------------
-----
Tensor: tensor([[[0.5854, 0.7855, 0.5888, 0.5847, 0.4712],
         [0.2976, 0.4946, 0.5490, 0.9866, 0.5227],
         [0.5141, 0.8636, 0.9911, 0.4598, 0.5055],
         [0.4693, 0.9248, 0.5605, 0.2646, 0.3851]],

        [[0.6428, 0.1994, 0.0452, 0.6236, 0.8777],
         [0.8255, 0.8431, 0.9469, 0.4122, 0.0056],
         [0.1923, 0.9496, 0.8077, 0.6793, 0.6325],
         [0.3084, 0.2466, 0.1293, 0.4753, 0.4295]],

        [[0.3932, 0.3431, 0.0610, 0.3036, 0.4806],
         [0.5940, 0.1803, 0.2193, 0.6335, 0.5051],
         [0.4184, 0.2860, 0.1053, 0.9838, 0.8827],
         [0.7358, 0.3315, 0.0125, 0.2228, 0.9178]]])
Dimension: 3
Shape: torch.Size([3, 4, 5])
-------------------------#######################--------------------
-----
```

Zero Tensors

In [11]:
```
ZERO_TENSOR = torch.zeros(3, 4)
print_dim_size(ZERO_TENSOR)
```

```
Tensor: tensor([[0., 0., 0., 0.],
        [0., 0., 0., 0.],
        [0., 0., 0., 0.]])
Dimension: 2
Shape: torch.Size([3, 4])
-------------------------#######################--------------------
-----
```

One Tensors

In [12]:
```
ONE_TENSOR = torch.ones(3, 4)
print_dim_size(ONE_TENSOR)
```

```
Tensor: tensor([[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]])
Dimension: 2
Shape: torch.Size([3, 4])
-------------------------#######################--------------------
-----
```

Identity Tensors

In [13]:
```
INDENTY_TENSOR = torch.eye(5)
print_dim_size(INDENTY_TENSOR)
```

```
Tensor: tensor([[1., 0., 0., 0., 0.],
        [0., 1., 0., 0., 0.],
        [0., 0., 1., 0., 0.],
        [0., 0., 0., 1., 0.],
        [0., 0., 0., 0., 1.]])
Dimension: 2
Shape: torch.Size([5, 5])
-------------------------#######################--------------------
-----
```

Other Constructors

In [14]:
```
torch.arange(0, 10)
```

Out[14]:
```
tensor([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

References

- https://youtu.be/Z_ikDlimN6A?si=5QW2TcZSJdMu6IIW
- https://dbourke.link/pt-github