# Chapter 2: TensorFlow

## Ex1: Constants, Sequences, và Random Values

### Constant Value Tensors

- Câu 1: Tạo một tensor có shape [2, 3] với tất cả các phần tử ban đầu = 0
- Câu 2: Cho X là một tensor như sau [[1,2,3], [4,5,6]]. Tạo 1 tensor có cùng shape và dtype như X với các phần tử ban đầu = 0
- Câu 3: Tạo một tensor có shape [2, 3] với tất cả các phần tử ban đầu là 1
- Câu 4: Cho X là một tensor như sau [[1,2,3], [4,5,6]]. Tạo 1 tensor có cùng shape và dtype như X với các phần tử ban đầu = 1
- Câu 5: Tạo một tensor có shape [3, 2] với tất cả các phần tử ban đầu là 5
- Câu 6: Tạo một constant tensor [[1, 3, 5], [4, 6, 8]], dtype=float32
- Câu 7: Tạo một constant tensor có shape [2, 3], với tất cả các phần tử ban đầu là 4

### Sequence

- Câu 1: Tạo 1-D tensor gồm 50 phần tử khoảng cách đều nhau từ 5 đến 10
- Câu 2: Tạo 1 tensor như sau: [10, 12, 14, 16, ..., 100]-

### Random Tensor

- Câu 1: Tạo một tensor ngẫu nhiên có shape [3, 2], với các phần tử từ phân bố chuẩn của giá trị trung bình = 0 (normal distribution of mean), độ lệch chuẩn = 2 (standard deviation)
- Câu 2: Tạo một random tensor có shape [3, 2], với các phần tử nằm trong một uniform distribution trong khoảng từ 0 đến 2.

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
from __future__ import print_function
import tensorflow as tf
import numpy as np
```

In [3]:
```
tf.__version__
```

Out[3]:
```
'2.5.0'
```

In [4]:
```
from datetime import date
date.today()
```

Out[4]:
```
datetime.date(2021, 10, 8)
```

# Constant Value Tensors

## Câu 1: Tạo một tensor có shape [2, 3] với tất cả các phần tử ban đầu = 0

In [5]:
```
out = tf.zeros([2, 3])
tf.print(out)
```

```
[[0 0 0]
 [0 0 0]]
```

## Câu 2: Cho X là một tensor như sau [[1,2,3], [4,5,6]]. Tạo 1 tensor có cùng shape và dtype như X với các phần tử ban đầu = 0

In [6]:
```
_X = np.array([[1,2,3], [4,5,6]])
X = tf.convert_to_tensor(_X)
out = tf.zeros_like(X)
tf.print(out)
```

```
[[0 0 0]
 [0 0 0]]
```

## Câu 3: Tạo một tensor có shape [2, 3] với tất cả các phần tử ban đầu là 1

In [7]:
```
out = tf.ones([2, 3])
tf.print(out)
```

```
[[1 1 1]
 [1 1 1]]
```

## Câu 4: Cho X là một tensor như sau [[1,2,3], [4,5,6]]. Tạo 1 tensor có cùng shape và dtype như X với các phần tử ban đầu = 1

In [8]:
```
_X = np.array([[1,2,3], [4,5,6]])
X = tf.convert_to_tensor(_X)
out = tf.ones_like(X)
tf.print(out)
```

```
[[1 1 1]
 [1 1 1]]
```

### Câu 5: Tạo một tensor có shape [3, 2] với tất cả các phần tử ban đầu là 5

In [9]:
```
out3 = tf.constant(5, shape=[3, 2])
tf.print(out3)
```

```
[[5 5]
 [5 5]
 [5 5]]
```

### Câu 6: Tạo một constant tensor [[1, 3, 5], [4, 6, 8]], dtype=float32

In [10]:
```
out = tf.constant([[1, 3, 5], [4, 6, 8]], dtype=tf.float32)
tf.print(out)
```

```
[[1 3 5]
 [4 6 8]]
```

### Câu 7: Tạo một constant tensor có shape [2, 3], với tất cả các phần tử ban đầu là 4

In [11]:
```
out3 = tf.constant(4, shape=[2, 3])
tf.print(out3)
```

```
[[4 4 4]
 [4 4 4]]
```

## Sequence

### Câu 1: Tạo 1-D tensor gồm 50 phần tử khoảng cách đều nhau từ 5 đến 10

In [12]:
```
out = tf.linspace(5., 10., 50)
tf.print(out)
```

```
[5 5.10204077 5.20408154 ... 9.79591846 9.89795876 10]
```

### Câu 2: Tạo 1 tensor như sau: [10, 12, 14, 16, ..., 100]

In [13]:
```
out = tf.range(10, 101, 2)
tf.print(out)
```

```
[10 12 14 ... 96 98 100]
```

# Random Tensor

## Câu 1: Tạo một tensor ngẫu nhiên có shape [3, 2], với các phần tử từ phân bố chuẩn của giá trị trung bình = 0 (normal distribution of mean), độ lệch chuẩn = 2 (standard deviation)

In [14]:
```
X = tf.random.normal([3, 2], 0, 2.)
tf.print(X)
```

```
[[0.272729248 0.567091]
 [-0.298118681 -2.30317354]
 [-3.96515965 1.41219771]]
```

## Câu 2: Tạo một random tensor có shape [3, 2], với các phần tử nằm trong một uniform distribution trong khoảng từ 0 đến 2.

In [15]:
```
out = tf.random.uniform([3, 2], 0, 2)
tf.print(out)
```

```
[[1.72838426 0.125823975]
 [1.49826 0.773273468]
 [1.43737864 1.49947023]]
```