

01. Tensorflow에서 변수 표현

In [3]:

```
1 import tensorflow as tf
2
3 # 상수 정의하기 --- (※1)
4 a1 = tf.constant(10, name="a")
5 b1 = tf.constant(20, name="b")
6 c1 = tf.constant(30, name="c")
```

In [4]:

```
1 # 변수 정의하기 --- (※2)
2 v = tf.Variable(0, name="v") # 변수 v 선언 및 초기값 0
3
4 # 데이터 플로우 그래프 정의하기 --- (※3)
5 calc_op = a1 + b1 + c1
6 assign_op = tf.assign(v, calc_op) # v에 calc_op 대입
7
8 # 세션 실행하기 --- (※4)
9 sess = tf.Session()
10 sess.run(assign_op)
11
12 # v의 내용 출력하기 --- (※5)
13 print( sess.run(v) )
```

60

02. Placeholder(플레이스 홀더)

- 템플릿처럼 값을 넣을 공간을 만들어두는 기능
- 플레이스홀더를 이용하면 데이터 플로우 그래프를 구축시에 값을 넣지 않고,
- 이후 세션을 실행할 때 그릇에 값을 넣을 수 있음.

```
In [5]: 1 import tensorflow as tf
2
3 # 플레이스홀더 정의하기 --- (※1)
4 a = tf.placeholder(tf.int32, [3]) # 정수 자료형 3개를 가진 배열
5
6 # 배열을 모든 값에 2배하는 연산 정의하기 --- (※2)
7 b = tf.constant(2)
8 x2_op = a * b
9
10 # 세션 시작하기 --- (※3)
11 sess = tf.Session()
12
13 # 플레이스홀더에 값을 넣고 실행하기 --- (※4)
14 r1 = sess.run(x2_op, feed_dict={ a:[1, 2, 3] })
15 print(r1)
16 r2 = sess.run(x2_op, feed_dict={ a:[10, 20, 10] })
17 print(r2)
```

```
[2 4 6]
[20 40 20]
```

03. 플레이스 홀더를 선언할 때, 요소의 개수를 고정시키지 않기(실습)

- 배열의 모든 값에 10배를 하는 연산을 정의
- 첫번째 리스트 값은 [1,2,3,4,5]
- 두번째 리스트 값은 [10,20]

실습pdf 참조

```
In [ ]:
```

```
1
```