

```
name = 'Bob'
print('Hello, {}'.format(name))
```

He

드라이브에서 찾기	
실습 모드에서 열기	
새 Python 3 노트	
새 Python 2 노트	
노트 열기...	Ctrl+O
노트 업로드...	
이름 바꾸기...	
휴지통으로 이동	
드라이브에 사본 저장...	
GitHub Gist로 사본 저장...	
GitHub에 사본 저장...	
저장	Ctrl+S
버전 저장 및 고정	Ctrl+M S
업데이트 기록	
.ipynb 다운로드	
.py 다운로드	
드라이브 미리보기 업데이트	
인쇄	Ctrl+P

```
for i in range(2,10):
    print('')
    print('{}단'.format(i))
    for j in range(1,10):
        print('{} * {} = {}'.format(i, j, i*j))
```




```
# for i in range(10):  
#     print(i, A[i])
```

```
B=1  
B=B+1  
print(B)
```



더블클릭 또는 Enter 키를 눌러 수정

```
print((1, 2, 3) + (4, 5, 6))  
print([1, 2, 3] + [4, 5, 6])  
print("Hello" + " " + "World")
```



```
# creating a tuple  
months = ('January', 'February', 'March', 'April', 'May', 'June', 'W  
'July', 'August', 'September', 'October', 'November', 'December')  
print(months[0])  
print("index of 7 ==> " , months[7])
```



```
for i in months:  
    print(i)
```



```
for i in range(12):  
    print(months[i])
```



```
t = ('john', 32, (2,3,4,5), 'hello', 213)  
print(t)  
print(t[2])  
print(t[2][1])  
print(t[:2]) #remember 2번째 전까지(미포함)  
print(t[2:])  
print(t[-1])  
print(t[-3])
```



```
t[2] = 1 #튜플 오류_ 고칠수가 없음_소괄호 안의 개체들
```



```
li = ['hallym', 1, 3.141572, 'hello']
print(li)
li[1] = 45
print(li)
li.append('September')#append_부록_맨마지막에 추가
print(li)
# 대괄호 안의 개체들은 고칠수 있음
```

```
v = []
for i in range(0,3):
    v.append(i)
print(v)
```

```
def Sum_(a,b):
    return a+b #결과를 돌려주다
```

```
c= Sum_(2,3)
print(c)
```

```
class Man:
    def __init__(self, name):
        self.name = name
        print("Initialized!")
    def hello(self):
        print("Hello " + self.name + " !")
    def goodbye(self):
        print("Good-bye " + self.name + " !")
```

```
m=Man("JQ")
```

```
m.hello()
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
import numpy as np #import란? 수입 as~로 np 즉, 무언갈 가져옴
import matplotlib.pyplot as plt#plt 그림그리기
from tensorflow.examples.tutorials.mnist import input_data# as없는 구문
```

```
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
```

```
print(mnist.train.images.shape)
print(mnist.test.images.shape)
```

```
idx=321
```

```

img1 = mnist.test.images[idx]
img1 = np.array(img1, dtype='float')

pixels = img1.reshape((28, 28))
plt.imshow(pixels, cmap='gray')
plt.title('mnist.test.images[{}].format(idx))
plt.show()

'mnist.test.images[{}].format(idx)

label1 = mnist.test.labels[idx]
print(label1)

label2 = np.argmax(label1)
print(label2)#one-hot encoding_하나의 뜨거운_강조요소

np.argmax([0, 1, 6, 3, 2, -2, 5, 7, 10]) #arg/_max/최대

np.argmax([0, 1, 0, 0, 0, 0, 0, 0, 0])

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

x_train = [1, 2, 3, 4, 5]
#y_train = [2, 4, 6, 8, 10]
y_train = [2+0.4+4, 4-0.43+4, 6+0.71+4, 8+0.007+4, 10-0.8+4] # Add some noise

W = tf.Variable(tf.random_normal([1]), name='weight')
b = tf.Variable(tf.random_normal([1]), name='bias')
W = tf.Variable(w0*tf.ones([1]), name='weight') #tf는 tensorflow
b = tf.Variable(b0*tf.ones([1]), name='bias')

hypothesis = x_train * W + b

cost = tf.reduce_mean(tf.square(hypothesis - y_train)) #스퀘어=제곱 //평균내기

optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.01)
train = optimizer.minimize(cost) #옵티마이저 수맞추기//미니멈

sess = tf.Session()

sess.run(tf.global_variables_initializer())

vw=[]
vb=[]
for step in range(3001):
    sess.run(train)
    w1 = sess.run(W)[0] # slope
    b1 = sess.run(b)[0] # bias
    vw.append(w1)

```

```
vb.append(b1)
if step % 100 == 0:
    print(step, sess.run(cost), w1, b1)

plt.plot (vw)

plt.plot (vb)

w1 = sess.run(W)[0] # slope
b1 = sess.run(b)[0] # bias
str1 = 'y = ' + str(w1) + 'x + ' + str(b1)
print(w1, b1)
print(str1)

plt.figure(1)
plt.plot(x_train, y_train, 'o')
x1 = np.linspace(np.min(x_train)-1, np.max(x_train)+1)
y1 = w1*x1 + b1
plt.plot(x1, y1)
plt.grid()
plt.title(str1)
```