

스칼라, 벡터, 매트릭스, 텐서를 numpy를 이용하여 만들기

```
In [58]: #넘파이 import 하기  
import numpy as np
```

```
In [59]: a = np.array(3.0)#스칼라 만들기
```

```
In [60]: b = np.arange(4)# 벡터  
print(b)
```

```
[0 1 2 3]
```

```
In [61]: c = np.arange(20).reshape(5,4)#매트릭스  
print(c)  
print(c.shape)
```

```
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]  
 [12 13 14 15]  
 [16 17 18 19]]  
(5, 4)
```

```
In [62]: d = np.arange(24).reshape(2,3,4)# 텐서  
print(d)
```

```
[[[ 0  1  2  3]  
   [ 4  5  6  7]  
   [ 8  9 10 11]]  
  
 [[12 13 14 15]  
   [16 17 18 19]  
   [20 21 22 23]]]
```

numpy의 poly1d를 이용하여 미분하기

```
In [63]: var = np.poly1d([2,2,1]) #np.poly1d를 이용하여 임의의 함수를 정의하기  
#print(var)
```

```
In [64]: print(var) #함수의 형태 보기
```

```
      2  
2 x + 2 x + 1
```

```
In [65]: derivative = np.polyder(var,m=1)# 미분구하기  
#print(derivative)
```

```
In [66]: print(derivative) # 미분 형태 보기
```

```
4 x + 2
```

```
In [67]: x = -1  
#equation = np.polyval(derivative,x)  
#print(equation) # 미분된 함수에 x에 -1 값 넣기  
print(derivative(x))
```

```
-2
```

```
In [68]: f = np.poly1d([2,0,12]) # 함수를 2x^2 + 12 로 만들기
          print(f)

          2
          2 x + 12
```

```
In [69]: x = 3.0 #임의의 값 설정
```

```
In [70]: der = np.polyder(f,m=1) #미분된 함수 구하기
          print(der)
```

```
4 x
```

```
In [71]: lr = 0.1           # learning rate
          x = 3.0           # 새로운 x 구하기
          #print(x, np.polyval(f,x), np.polyval(der,x))      # x값 프린트
          print(x,f(x),der(x))

          3.0 30.0 12.0
```

```
In [72]: for i in range(20): # 반복하기
          x = x-lr*der(x)
          print(f'x: {x:.5f} f(x): {f(x):.5f}')
```

```
x: 1.80000 f(x): 18.48000
x: 1.08000 f(x): 14.33280
x: 0.64800 f(x): 12.83981
x: 0.38880 f(x): 12.30233
x: 0.23328 f(x): 12.10884
x: 0.13997 f(x): 12.03918
x: 0.08398 f(x): 12.01411
x: 0.05039 f(x): 12.00508
x: 0.03023 f(x): 12.00183
x: 0.01814 f(x): 12.00066
x: 0.01088 f(x): 12.00024
x: 0.00653 f(x): 12.00009
x: 0.00392 f(x): 12.00003
x: 0.00235 f(x): 12.00001
x: 0.00141 f(x): 12.00000
x: 0.00085 f(x): 12.00000
x: 0.00051 f(x): 12.00000
x: 0.00030 f(x): 12.00000
x: 0.00018 f(x): 12.00000
x: 0.00011 f(x): 12.00000
```

```
In [ ]:
```

```
In [ ]:
```