

8 주차 Homework

학번

32151671

이름

박민혁

[Python 복습] 주피터와 NumPy 를 이용해 다음 결과를 구하시오. 실행 결과를 캡처 또는 내용을 붙여넣는다.

```
import numpy as np
```

1. 1 부터 100 사이의 홀수를 만들어 변수 x 에 저장하고 출력하시오.

```
x = np.arange(1, 101, 2)
x # print(x)
```

```
array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
        35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
        69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99])
[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95
 97 99]
```

2. x 를 5 행 10 열의 배열로 변환하여 y 에 저장하고 출력하시오.

```
y = x.reshape(5, 10)
y # print(y)
```

```
array([[ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19],
       [21, 23, 25, 27, 29, 31, 33, 35, 37, 39],
       [41, 43, 45, 47, 49, 51, 53, 55, 57, 59],
       [61, 63, 65, 67, 69, 71, 73, 75, 77, 79],
       [81, 83, 85, 87, 89, 91, 93, 95, 97, 99]])
```

```
[[ 1  3  5  7  9 11 13 15 17 19]
 [21 23 25 27 29 31 33 35 37 39]
 [41 43 45 47 49 51 53 55 57 59]
 [61 63 65 67 69 71 73 75 77 79]
 [81 83 85 87 89 91 93 95 97 99]]
```

3. y의 차원을 확인하시오.

```
y.ndim # print(y.ndim)
```

2

4. y의 배열의 모양(shape)을 확인하시오.

```
y.shape # print(y.shape)
```

(5, 10)

5. y의 크기(총 몇 개의 값으로 구성되었는지)를 구하시오.

```
y.size # print(y.size)
```

50

6. x에서 마지막 10개의 값을 출력하시오

```
x[-10 : ] # print(x[-10 : ])
```

array([81, 83, 85, 87, 89, 91, 93, 95, 97, 99])

[81 83 85 87 89 91 93 95 97 99]

7. x의 값을 하나 걸러 하나씩 출력하시오

```
x[: : 2] # print(x[: : 2])
```

```
array([ 1,  5,  9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 65,
        69, 73, 77, 81, 85, 89, 93, 97])
[ 1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93
 97]
```

8. x의 값을 뒤에서부터 하나 걸러 하나씩 출력하시오

```
x[: : -2] # print(x[: : -2])
```

```
array([99, 95, 91, 87, 83, 79, 75, 71, 67, 63, 59, 55, 51, 47, 43, 39, 35,
        31, 27, 23, 19, 15, 11,  7,  3])
array([99, 95, 91, 87, 83, 79, 75, 71, 67, 63, 59, 55, 51, 47, 43, 39, 35,
        31, 27, 23, 19, 15, 11,  7,  3])
```

9. 7과 8의 결과를 곱하여 z에 저장하시오

```
z = x[: : 2] * x[: : -2]
z # print(z)
```

```
array([ 99, 475, 819, 1131, 1411, 1659, 1875, 2059, 2211, 2331, 2419,
        2475, 2499, 2491, 2451, 2379, 2275, 2139, 1971, 1771, 1539, 1275,
        979,  651,  291])
[ 99 475 819 1131 1411 1659 1875 2059 2211 2331 2419 2475 2499 2491
 2451 2379 2275 2139 1971 1771 1539 1275 979 651 291]
```

10. 2,4,6,8,10의 숫자 패턴을 5번 반복하는 배열을 만들어 w에 저장하고 출력하시오.

```
w = np.tile(np.arange(2, 11, 2), 5)
w # print(w)
```

```
array([ 2,  4,  6,  8, 10,  2,  4,  6,  8, 10,  2,  4,  6,  8, 10,  2,  4,
        6,  8, 10,  2,  4,  6,  8, 10])
[ 2  4  6  8 10  2  4  6  8 10  2  4  6  8 10  2  4  6  8 10  2  4  6  8
 10]
```

11. x 와 w 를 결합하여 새로운 배열 k 를 만드시오.

```
k = np.concatenate((x, w))
k # print(k)
```

```
array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
        35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
        69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99,  2,
         4,  6,  8, 10,  2,  4,  6,  8, 10,  2,  4,  6,  8, 10,  2,  4,  6,
         8, 10,  2,  4,  6,  8, 10])

[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47
 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95
 97 99  2  4  6  8 10  2  4  6  8 10  2  4  6  8 10  2  4  6  8 10  2  4
  6  8 10]
```

12. k 에 있는 모든 요소의 합을 구하고 결과를 출력하시오

```
np.sum(k) # print(np.sum(k))
```

2650

13. k 의 평균을 구하고 결과를 출력하시오

```
np.mean(k) # print(np.mean(k))
```

35.333333333333336

14. k 의 중앙값을 구하고 결과를 출력하시오

```
np.median(k) # print(np.median(k))
```

25.0

15. k의 표준편차를 구하고 결과를 출력하시오

```
np.std(k) # print(np.std(k))
```

31.436002007606216

16. k 안에 있는 값 중 최대값과 최소값을 구하고 출력하시오.

```
print('max : {0}, min : {1}'.format(max(k), min(k)))
```

max : 99, min : 1

17. k의 값 중 평균보다 큰 값의 개수를 구하시오

```
((k > np.mean(k)).sum()) # print(((k > np.mean(k)).sum()))
```

32

18. k의 값 중 평균보다 큰 값만 출력하시오

```
k[k > np.mean(k)] # print(k[k > np.mean(k)])
```

```
array([37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69,  
       71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99])
```

```
[37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83  
 85 87 89 91 93 95 97 99]
```

19. k의 값 중 평균 보다 크고 평균에 표준편차를 더한 값보다 작은 값만 출력하시오.

```
k[(k > np.mean(k)) & (k < (np.mean(k) + np.std(k)))]  
# print(k[(k > np.mean(k)) & (k < (np.mean(k) + np.std(k)))])
```

```
array([37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65])
```

```
[37 39 41 43 45 47 49 51 53 55 57 59 61 63 65]
```

20. k 를 오름차순으로 정렬하고 결과를 출력하시오.

```
np.sort(k) # print(np.sort(k))
```

```
array([ 1,  2,  2,  2,  2,  2,  3,  4,  4,  4,  4,  4,  5,  6,  6,  6,  6,
        6,  7,  8,  8,  8,  8,  8,  9, 10, 10, 10, 10, 10, 11, 13, 15, 17,
       19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51,
       53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85,
       87, 89, 91, 93, 95, 97, 99])

[ 1  2  2  2  2  2  3  4  4  4  4  4  5  6  6  6  6  6  7  8  8  8  8  8
  9 10 10 10 10 10 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45
 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93
 95 97 99]
```

21. k 를 내림차순으로 정렬하고 결과를 출력하시오.

```
np.sort(k)[: -1] # print(np.sort(k)[: -1])
```

```
array([99, 97, 95, 93, 91, 89, 87, 85, 83, 81, 79, 77, 75, 73, 71, 69, 67,
       65, 63, 61, 59, 57, 55, 53, 51, 49, 47, 45, 43, 41, 39, 37, 35, 33,
       31, 29, 27, 25, 23, 21, 19, 17, 15, 13, 11, 10, 10, 10, 10, 9,
        8,  8,  8,  8,  8,  7,  6,  6,  6,  6,  6,  5,  4,  4,  4,  4,  4,
        3,  2,  2,  2,  2,  2,  2,  1])

[99 97 95 93 91 89 87 85 83 81 79 77 75 73 71 69 67 65 63 61 59 57 55 53
 51 49 47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 10 10 10
 10 10 9  8  8  8  8  8  7  6  6  6  6  6  5  4  4  4  4  4  3  2  2  2
  2  2  1]
```

22. 0 으로 채워진 3 행 2 열의 배열을 만들고 출력하시오.

```
array = np.zeros((3, 2), dtype = np.int)
array # print(array)
```

```
array([[0, 0],
       [0, 0],
       [0, 0]])

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

23. 1로 채워진 2행 3열의 배열을 만들고 출력하시오.

```
array = np.ones((2, 3), dtype = np.int)
array # print(array)
```

```
array([[1, 1, 1],      [[1 1 1]
       [1, 1, 1]])      [1 1 1]]
```

24. 5로 채워진 길이가 10인 1차원 배열을 만들고 출력하시오.

```
array = np.full(10, 5)
array # print(array)
```

```
array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
[5 5 5 5 5 5 5 5 5 5]
```

25. 2,4,6,8,10을 각 숫자 요소별 5번씩 반복하는 배열을 만들고 출력하시오.

```
np.repeat(np.arange(2, 11, 2), 5) # print(np.repeat(np.arange(2, 11, 2), 5))
```

```
array([ 2,  2,  2,  2,  2,  4,  4,  4,  4,  4,  6,  6,  6,  6,  6,  8,  8,
        8,  8,  8, 10, 10, 10, 10, 10])
[ 2  2  2  2  2  4  4  4  4  4  6  6  6  6  6  8  8  8  8  8 10 10 10 10
 10]
```

26. 2,4,6,8,10의 숫자 패턴을 5번 반복하는 배열을 만들고 출력하시오.

```
np.tile(np.arange(2, 11, 2), 5) # print(np.tile(np.arange(2, 11, 2), 5))
```

```
array([ 2,  4,  6,  8, 10,  2,  4,  6,  8, 10,  2,  4,  6,  8, 10,  2,  4,
        6,  8, 10,  2,  4,  6,  8, 10])
[ 2  4  6  8 10  2  4  6  8 10  2  4  6  8 10  2  4  6  8 10  2  4  6  8
 10]
```