



CHAP.5

파일입출력
예외처리
함수형
기타

파일입출력

Photo by Maksym Kaharlytskyi on Unsplash

리다이렉션 출력 콘솔 명령

```
a=[1,2,3,4,5,6]
for i in range(len(a)-1):
    print(a[i],",",end="")
print(a[len(a)-1])
```

```
C:\>python a.py
1 ,2 ,3 ,4 ,5 ,6
```

```
C:\>python a.py > a.txt
C:\>type a.txt
1 ,2 ,3 ,4 ,5 ,6
C:\>python a.py >> a.txt
1 ,2 ,3 ,4 ,5 ,6
1 ,2 ,3 ,4 ,5 ,6
```

리다이렉션 입력 콘솔 명령

```
a=[]  
for i in range(3):  
    b = input()  
    a.append(b)  
print(a)
```

```
C:\>python a.py  
1  
2   키보드 입력  
3  
['1', '2', '3']
```

```
C:\>echo 1 > a.txt  
C:\>echo 2 >> a.txt  
C:\>echo 3 >> a.txt  
C:\>type a.txt  
1  
2  
3
```

```
C:\>python a.py < a.txt   키보드 입력 대체  
['1', '2', '3']
```

파일 쓰기(w)-문자열을 쓴다, 숫자는 NO

<pre>a='Hello' f=open('a.txt', 'w') f.write(a) 기본적으로 줄을 f.write('\n') 띄우지 않는다. f.write(a) f.close()</pre>	<pre>a=1 #a=str(a) f=open('a.txt', 'w') f.write(a) f.write('\n') f.write(a) f.close()</pre>
<pre>C:\>python a.py C:\>type a.txt Hello Hello C:\></pre>	<pre>C:\>python a.py TypeError: write() argument must be str, not int C:\></pre> <p>파일은 기본적으로 문자열만 다룬다.</p>

파일 쓰기-추가하기(a)

```
f=open('a.txt', 'w')  
f.write('Hello') 줄을 띄우지 않는다.  
f.close()
```

```
f=open('a.txt', 'a')  
f.write('Hello')  
f.close()
```

```
C:\>python a.py  
C:\>type a.txt  
HelloHello  
C:\>
```

파일 읽기-문자열로 읽어온다.

```
C:\>echo 1 > a.txt
```

```
C:\>type a.txt
```

```
1
```

```
>>>f=open('a.txt', 'r')
```

```
>>>line = f.read()
```

```
'1\n'
```

```
>>>f.close()
```

다 문자로 읽어 들인다.

input() 함수와 비슷하다.
키보드에서 1을 누르면 '1'을 남긴다.

유닉스라는 운영체제에서는
사실 키보드도 파일로 다룬다.
'표준입력' 파일
stdin 이라는 이름도 있다.

파일 읽기

```
>>>f=open('a.txt', 'r')
>>>data = f.read()
'Hello\nHello'
>>>f.close()
>>>print(data)
Hello
Hello
```

파일에 있는 전체를 읽는다.

```
>>>f=open('a.txt', 'r')
>>>line = f.readline()
'Hello\n'
>>>line = f.readline()
'Hello'
>>>f.close()
```

파일에 있는 것을 한 줄씩 읽는다.

파일 한 줄씩 전체 읽기

```
>>>f=open('a.txt', 'r')
>>>line = f.readline()
>>>while(line != ''):      공백이 아닌 동안 계속 한 줄씩 읽는다.
...     print(line)
...     line = f.readline()
>>>f.close()
```

```
C:\>type a.txt
HelloWorld
```

```
f=open('a.txt', 'r')
dir(f)
```

```
['_CHUNK_SIZE', '__class__', '__del__', '__delattr__', '__dict__', '__dir__', '__doc__', '__enter__', '__eq__',
'__exit__', '__format__', '__ge__', '__getattr__', '__getstate__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__lt__', '__ne__', '__new__', '__next__', '__reduce__',
'__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '_checkClosed',
'_checkReadable', '_checkSeekable', '_checkWritable', '_finalizing', 'buffer', 'close', 'closed',
'detach', 'encoding', 'errors', 'fileno', 'flush', 'isatty', 'line_buffering', 'mode', 'name', 'newlines',
'read', 'readable', 'readline', 'readlines', 'seek', 'seekable', 'tell', 'truncate', 'writable', 'write',
'writelines']
```

여러 줄 읽기 - 리스트로 읽는다.

```
>>>f=open('a.txt', 'r')
>>>lines = f.readlines()
['Hello\n', 'Hello']
>>>f.close()
```

readlines는 리스트를 남긴다.

```
>>>f=open('a.txt', 'r')
>>>lines = f.readlines()
>>>f.close()
>>>for line in lines:
...     print(line)
Hello
    \n 때문에 한 줄 띄워진다
Hello
```

여러 줄 읽기

```
>>>f=open('a.txt', 'r')
>>>lines = f.readlines()
>>>f.close()
>>>for line in lines:
...     line=line[:-1]   맨 끝에 \n 없었다.
...     print(line)
Hello
Hello
```

문자열 리스트를 쓰기

```
f=open('a.txt', 'w')
n=f.write('Hello'); print(n)
f.close()
f=open('a.txt', 'a')
n=f.write('World'); print(n)
f.close()
```

```
C:\>5
C:\>5
C:\>type a.txt
HelloWorld
```

```
f=open('a.txt', 'w')
n=f.writelines(['Hello', 'World'])
print(n)
f.close()
```

리스트를 출력할 수 있다.
하나의 문자열로 이어서.

```
C:\>None
C:\>type a.txt
HelloWorld
```

파일이 없을 때의 대처 방법

```
try:
    f=open('k.txt', 'r')
    contents = f.read()
    f.close()
    print(contents)
except:
    print('File Not Found')
```

try 블록에서
예외사항이 발생하면

except 블록
을 수행한다.

```
C:\>python a.py
File Not Found
C:\>
```


파일 존재 여부 판정

```
def isFile(file):  
    try:  
        f=open(file, 'r')  
        f.close()  
        return True  
    except:  
        return False  
isFile('a.txt')
```

```
>>>import os
```

```
>>>os.path.isfile('./a.txt')
```

```
True
```

현재 폴더
/ 슬래쉬
방향에 주의

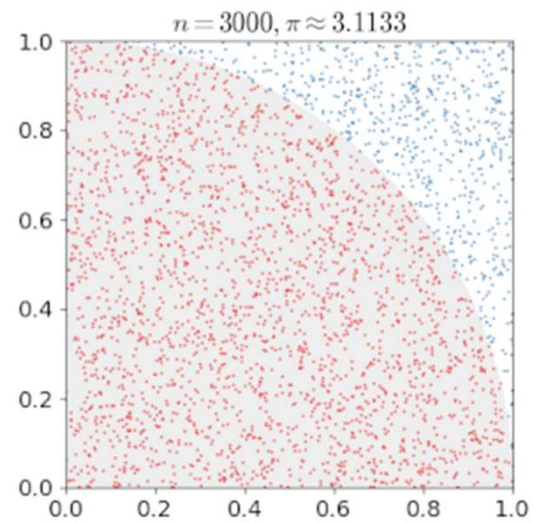
윈도우에서는 보통
역슬래쉬 2번 쓴다.
\\

파이썬에서는
유닉스처럼
/ 슬래쉬 한 개로 폴더 구분



Photo by Josh Hild on Unsplash

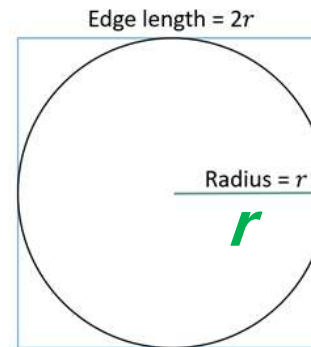
파이 추정, 원 그리기



https://upload.wikimedia.org/wikipedia/commons/thumb/8/84/Pi_30K.gif/330px-Pi_30K.gif

파이를 추정해 보자

```
>>>import math  
>>>math.pi  
3.141592653589793
```



$$\text{Area of Circle } A_{\text{circle}} = \pi r^2$$

$$\text{Area of Square } A_{\text{square}} = 4r^2$$

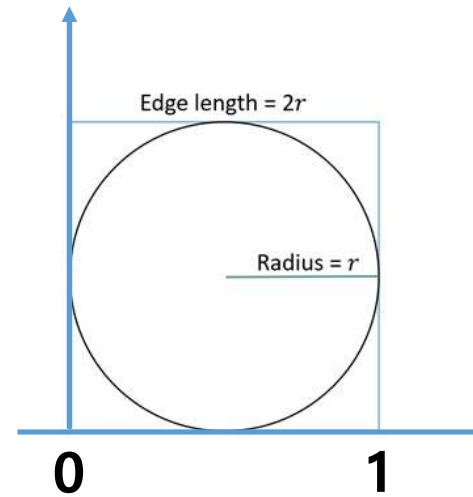
$$r^2 = \frac{A_{\text{circle}}}{\pi} \quad \rightarrow \quad \frac{A_{\text{circle}}}{\pi} = \frac{A_{\text{square}}}{4}$$
$$r^2 = \frac{A_{\text{square}}}{4} \quad \rightarrow$$

$$\pi = 4 \cdot \frac{A_{\text{circle}}}{A_{\text{square}}}$$

넓이는 공간을 채우고
있는 점의 수로 근사

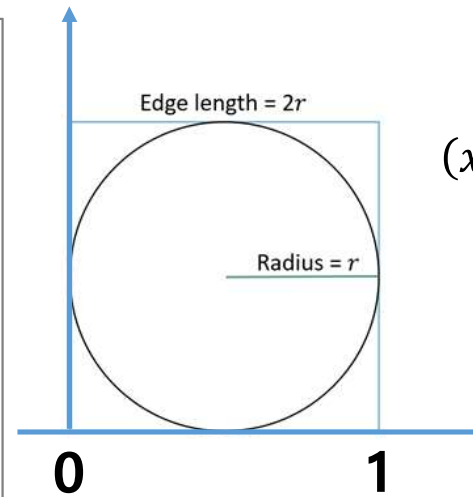
0~1사이의 임의의 값 꺼내 오기

```
>>>import random  
>>>a=random.random()  
0.7440043797747996
```



원 안에 속한 점 골라내기

```
>>>import random
>>>x=random.random()
>>>y=random.random()
>>>r=(x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)
>>>bInCircle = True if r<=0.25 else False
>>>print(x, y, bInCircle)
0.035 0.870 False
0.373 0.134 True
```



$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

$$x_c = 0.5$$

$$y_c = 0.5$$

$$(x - 0.5)^2 + (y - 0.5)^2 = r^2 \leq 0.25$$

이면 원 안의 점

파이 근사

Monte Carlo Method ~ approximation

```
N=1000
areaSquare = N
areaCircle = 0
for i in range(N):
    x = random.random()
    y = random.random()
    r=(x-0.5)*(x-0.5) + (y-0.5)*(y-0.5)
    bInCircle = True if r<=0.25 else False
    if bInCircle:
        areaCircle += 1
pi = 4 * areaCircle / areaSquare
print(pi)
```

3.12

N을 점점 크게 하면서 추이를 보자

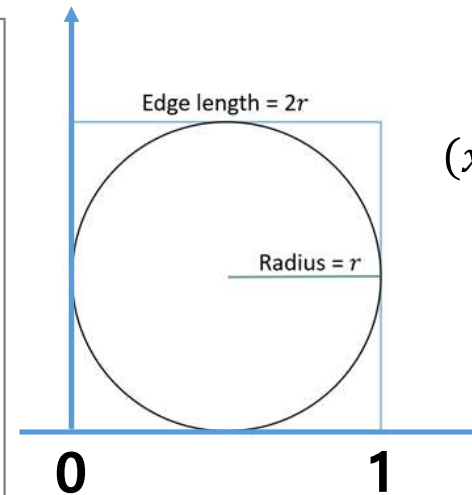
```
import random
def approximatePi(n):
    areaSquare = n
    areaCircle = 0
    for i in range(n):
        x = random.random()
        y = random.random()
        r=(x-0.5)*(x-0.5) + (y-0.5)*(y-0.5)
        bInCircle = True if r<=0.25 else False
        if bInCircle:
            areaCircle += 1
    pi = 4 * areaCircle / areaSquare
    return pi
```

```
for n in range(10000, 1000000, 100000):
    pi = approximatePi(n)
    print(n, pi)
```

```
10000 3.146
110000 3.14
210000 3.1415809523809526
310000 3.1477290322580647
410000 3.143551219512195
510000 3.142250980392157
610000 3.139193442622951
710000 3.140557746478873
810000 3.1415506172839507
910000 3.141560439560438
```

원 안에 속한 점 골라내서 파일로 저장

```
import random
N=10000 #1000
f = open('a.csv', 'w')   엑셀로 읽힌다.
for i in range(N):
    x = random.random()
    y = random.random()
    r=(x-0.5)*(x-0.5) + (y-0.5)*(y-0.5)
    bInCircle = True if r<=0.25 else False
    pos = str(x)+',' +str(y)+'\n'
    if bInCircle:   문자열로 바꾸어야 한다.
        f.write(pos)
f.close()
```



$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

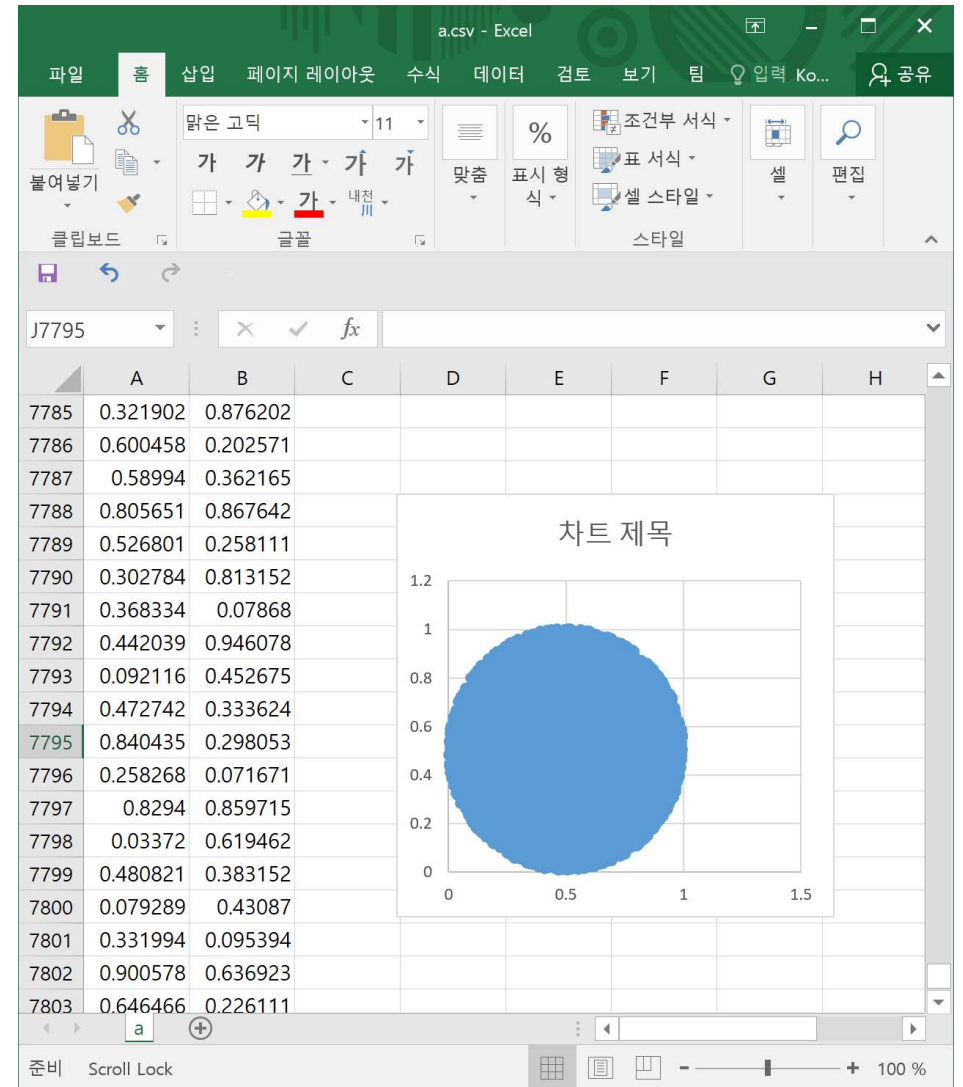
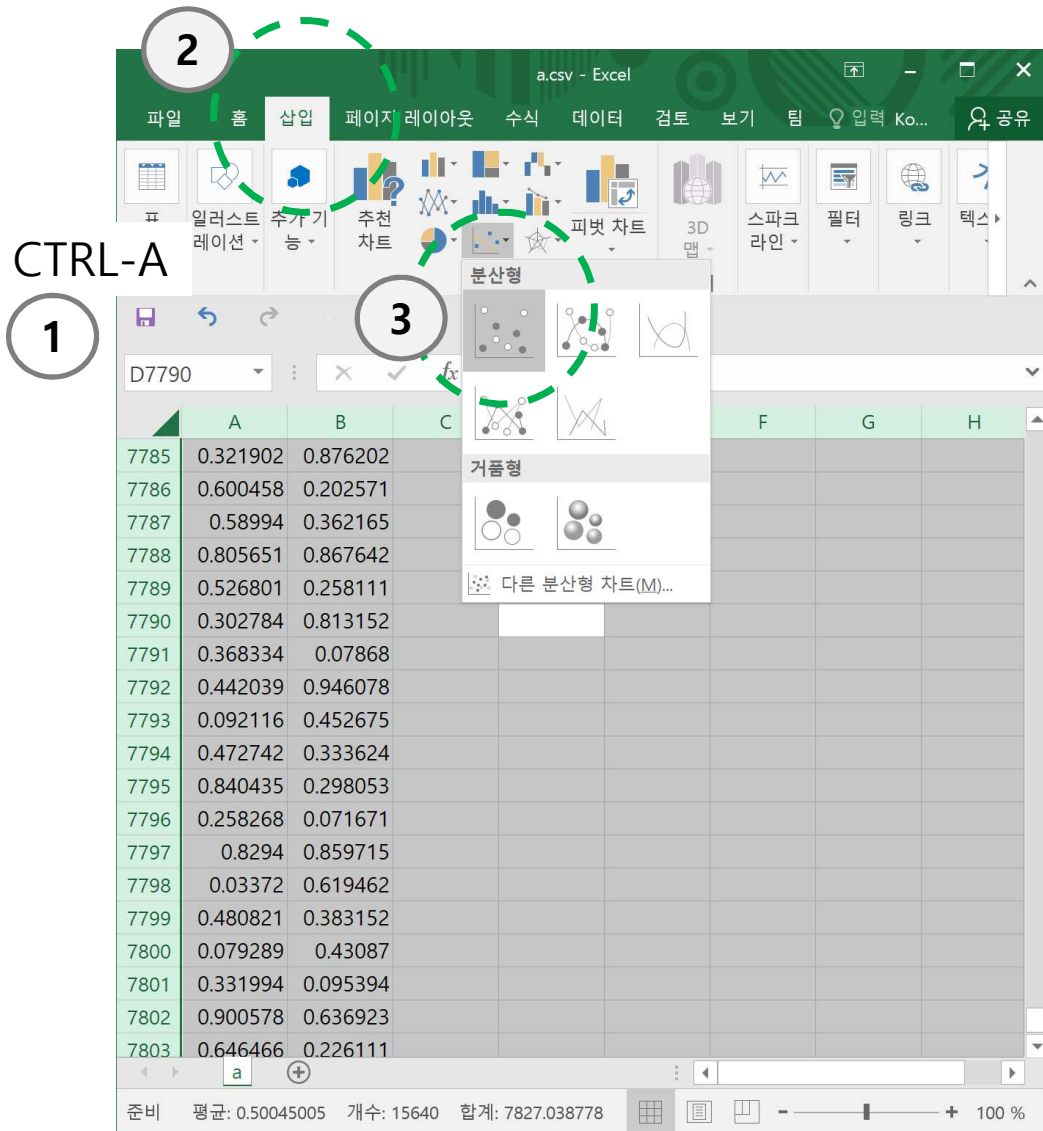
$$x_c = 0.5$$

$$y_c = 0.5$$

$$(x - 0.5)^2 + (y - 0.5)^2 = r^2 \leq 0.25$$

이면 원 안의 점

`pos = '{} , {} \n'.format(x,y)`



관련 패키지 활용

```
C:\>pip install matplotlib
```

```
import random
import matplotlib.pyplot as plt
N=10000
ptX = []; ptY = []
for i in range(N):
    x = random.random()
    y = random.random()
    r=(x-0.5)*(x-0.5) + (y-0.5)*(y-0.5)
    bInCircle = True if r<=0.25 else False
    if bInCircle:
        ptX.append(x); ptY.append(y)
```

```
plt.scatter(ptX, ptY)
plt.savefig('a.png')
plt.show()
```

