

SWP2 (2020 2학기)

python3 sum.py 명령어를 이용한 파이썬 파일 실행

```
sckhome@sckhome-ThinkPad-T440:~$ gedit sum.py
sckhome@sckhome-ThinkPad-T440:~$ ls
PycharmProjects  catkin_ws          git-example_05
Untitled.ipynb    class-01-SCKIMOSU  git-examples
anaconda3        examples.desktop   hello
sckhome@sckhome-ThinkPad-T440:~$ python3 sum.py
sum=55
sckhome@sckhome-ThinkPad-T440:~$ gedit sum.py
```



```
i=1; sum=0
while True:
    sum+=i
    if i==10:
        break
    i=i+1

print("sum=" +str(sum))
```

코드의 개선

```
sckhome@sckhome-ThinkPad-T440:~$ gedit sum.py
sckhome@sckhome-ThinkPad-T440:~$ gedit sum1.py
sckhome@sckhome-ThinkPad-T440:~$ python3 sum1.py
sum=55
sckhome@sckhome-ThinkPad-T440:~$ gedit sum1.py
```

```
i=1
sum=0

while i <= 10:
    sum+=i
    i+=1

print("sum=%d" %sum)
```

코드의 개선

```
sckhome@sckhome-ThinkPad-T440:~$ python3 sum2.py
sum = 55
sckhome@sckhome-ThinkPad-T440:~$ gedit sum2.py
```

```
sum=0
for i in range(1, 11):
    sum += i
print("sum = %d" %sum)
```

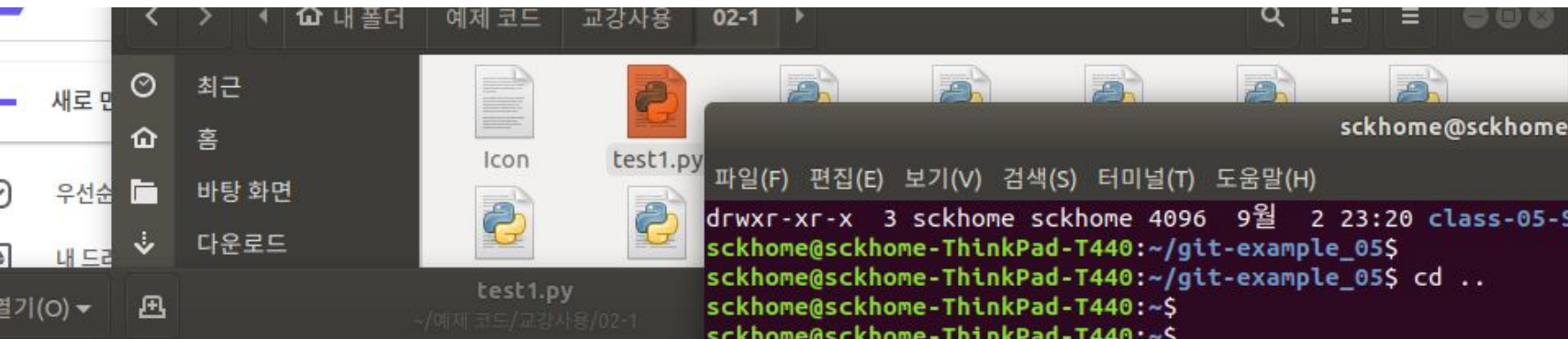
```
sckhome@sckhome-ThinkPad-T440:~/git-example_05/class-05-SCKIMOSU$ git diff
sckhome@sckhome-ThinkPad-T440:~/git-example_05/class-05-SCKIMOSU$ git diff --cached
diff --git a/readme.txt b/readme.txt
new file mode 100644
index 0000000..bdc6ff6
--- /dev/null
+++ b/readme.txt
@@ -0,0 +1 @@
+This is a sample file for git testing
sckhome@sckhome-ThinkPad-T440:~/git-example_05/class-05-SCKIMOSU$
```

git 사용법

https://backlog.com/git-tutorial/kr/intro/intro1_2.html

<https://github.com/kmu-cs-swp2-2018/>

2주차 1차시 내용



```
age = input('enter age: ')
age = int(age)
print(nage + 1)
print(age + 1) # test
```


Python ▼ 탭 너비: 8 ▼

```
sckhome@sckhome-ThinkPad-T440:
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
drwxr-xr-x 3 sckhome sckhome 4096 9월  2 23:20 class-05-SCKIMOSU
sckhome@sckhome-ThinkPad-T440:~/git-example_05$
sckhome@sckhome-ThinkPad-T440:~/git-example_05$ cd ..
sckhome@sckhome-ThinkPad-T440:~$
sckhome@sckhome-ThinkPad-T440:~$
sckhome@sckhome-ThinkPad-T440:~$
sckhome@sckhome-ThinkPad-T440:~$ python3
Python 3.7.1 (default, Dec 14 2018, 19:28:38)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> age=input('enter age: ')
enter age: 35
>>> nage=int(age)
>>> nage
35
>>> age
'35'
>>> print(nage+1)
36
>>> print(age+1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
>>>
```

```
phi = 3.14
x = 3
y = 4
print("Phi is %d" % phi)
print("Phi is %f" % phi)
print("Phi is %e" % phi)
print("x is %d, y is %d" % (x, y))
print("Hello %s" % input("Enter your name: "))
```



```
>>> print("phi is %d" #phi)
...
...
[4]+ 정지됨 python3
sckhome@sckhome-ThinkPad-T440:~$ python3
Python 3.7.1 (default, Dec 14 2018, 19:28:38)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> phi=3.14
>>> x=3
>>> y=4
>>> print("phi is %d" %ph)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ph' is not defined
>>> print("phi is %d" %phi)
phi is 3
>>> print("phi is %f" %phi)
phi is 3.140000
>>> print("phi is %e" %phi)
phi is 3.140000e+00
>>> print("x is %d, y is %d" %(x,y))
x is 3, y is 4
>>>
```

열기(O)  test5.py
~/SWP2_예제 코드/교강사용/02-1

test3.py x

```
x = int(input("Enter x: "))
y = int(input("Enter y: "))
if x < y:
    print(x, "is less than", y)
elif x == y:
    print(x, "is equal to", y)
else:
    print(x, "is greater than", y)
```

Python ▼ 탭 너비: 8 ▼

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
Enter x: 4
>>> x
4
>>> x1=input("Enter x: ")
Enter x: 5
>>> x=int(input("Enter x: "))
Enter x: 4
>>> x
4
>>> x1=input("Enter x1: ")
Enter x1: 6
>>> x1
'6'
>>> y=int(input("Enter y: "))
Enter y: 7
>>>
>>> if x < y:
...     print(x, " is less than", y)
... elif x==y:
...     print(x, " is equal to", y)
... else:
...     print(x, " is greatr than", y)
...
4 is less than 7
>>>
```

자 노트를 추가하려면 클릭하세요.

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io.py

file_io ▶ ⚙️ 🔍

Project ▾

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - input.txt
- External Libraries
- Scratches and Consoles

input.txt × file_io.py ×

```

1 # open(): 파일을 특정한 모드로 여는 함수. 읽을 때는 r, 쓸 때는 w
2 # read(): 파일 객체로부터 모든 내용을 읽는 함수
3
4 f=open("input.txt", "r", encoding= "UTF-8")
5 # f: file object (파일객체)
6
7 data=f.read()
8 print(data)
9 f.close() #해당 파일에 대한 리소스 사용이 끝남 |

```

 1: Project
 2: Favorites
 3: Structure

Run: file_io × ⚙️ -

```

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io.py
안녕하세요?
홍길동입니다.

```

Process finished with exit code 0

Python Console Terminal ▶ 4: Run ⚙️ 6: TODO

EventLog

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io.py

file_io

Project

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - input.txt
- External Libraries
- Scratches and Consoles

input.txt file_io.py

```

1  # open(): 파일을 특정한 모드로 여는 함수. 읽을 때는 r, 쓸 때는 w
2  # read(): 파일 객체로부터 모든 내용을 읽는 함수
3
4  f=open("input.txt", "r", encoding="UTF-8")
5  # f: file object (파일객체)
6  f.seek(9)
7  # seek(): 파일에서 읽을 위치를 선정해주는 함수
8  # 9 바이트의 위치에서부터 파일을 읽음
9  # 한글이 유니코드로 사용될 때, 파이썬에서는 한 글자당 3바이트를 할당함 (시스템마다 다름)
10 # 세요?
11 # 홍길동입니다.
12 # 세 번째 줄에는 무슨 말을 쓸까요? |
13
14 data=f.read()
15 print(data)
16 f.close() #해당 파일에 대한 리소스 사용이 끝남

```

Run: file_io

2: Favorites

Structure

```

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io.py
세요?
홍길동입니다.
세 번째 줄에는 무슨 말을 쓸까요?

```

Process finished with exit code 0

Python Console Terminal 4: Run 6: TODO

Event Log

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io.py

file_io ▶

Project

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - input.txt
- External Libraries
- Scratches and Consoles

```

17
18 print(data)
19 f.close() #해당 파일에 대한 리소스 사용이 끝남
20
21 # 두 번째 예제
22 f=open("input.txt", "r", encoding="UTF-8")
23
24 count=0
25 while count < 3:
26     data=f.readline()
27     count = count +1
28     print("%d번째 줄: %s" %(count, data))
29     # %s : 문자열이 출력될 형식지정자
30 f.close()

```

while count < 3

Run: file_io

```

/home/serkhome/PycharmProjects/swp2/venv/bin/python /home/serkhome/PycharmProjects/swp2/file_io.py
세요?
홍길동입니다.
세 번째 줄에는 무슨 말을 쓸까요?
1번째 줄: 안녕하세요?

2번째 줄: 홍길동입니다.

3번째 줄: 세 번째 줄에는 무슨 말을 쓸까요?

Process finished with exit code 0

```

Python Console Terminal ▶ 4: Run ⚙️ 6: TODO

Event Log

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io.py

file_io ▶ ⚙️ 🔍

Project ▾

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - input.txt
- External Libraries
- Scratches and Consoles

```

17
18 print(data)
19 f.close() #해당 파일에 대한 리소스 사용이 끝남
20
21 # 두 번째 예제
22 f=open("input.txt", "r", encoding="UTF-8")
23
24 count=0
25 while count < 3:
26     data=f.readline()
27     count = count +1
28     print("%d번째 줄: %s" %(count, data), end='|')
29     # %s : 문자열이 출력될 형식지정자
30 f.close()

```

while count < 3

Run: file_io ×

⚙️ -

```

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io.py
세요?
홍길동입니다.
세 번째 줄에는 무슨 말을 쓸까요?
1번째 줄: 안녕하세요?
2번째 줄: 홍길동입니다.
3번째 줄: 세 번째 줄에는 무슨 말을 쓸까요?
Process finished with exit code 0

```

Python Console Terminal ▶ 4: Run ⚙️ 6: TODO

Event Log

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io1.py

file_io1

Project
▼ swp2 ~/PycharmProjects/swp2

venv

file_io.py

file_io1.py

input.txt

External Libraries

Scratches and Consoles

```
1 # open(): 파일을 특정한 모드로 여는 함수. 읽을 때는 r, 쓸 때는 w
2 # read(): 파일 객체로부터 모든 내용을 읽는 함수
3 # readline(): 파일 객체로부터 한 줄씩 문자열을 읽는 함수
4 # readlines(): 전체 내용을 한 번에 리스트에 담는 함수
5
6 f=open("input.txt", "r", encoding="UTF-8")
7 list = f.readlines()
8 print(list)
```

Run: file_io1

```
/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io1.py
['안녕하세요?\n', '홍길동입니다.\n', '세 번째 줄에는 무슨 말을 쓸까요?']
```

Process finished with exit code 0

Z: Structure ★ 2: Favorites

Python Console Terminal 4: Run 6: TODO

Event Log

file_io2.py

```
list = f.readlines()
```

모든 줄을 읽은 내용을 list에 담았다가 list 원소 처리기법 활용할 수 있다.

```
for i, data in enumerate(list): #enumerate()는 각 원소에 접근 수 있도록 함
```

```
    print("%d번째 줄: %s" %(i+1, data), end="")
```

```
f.close() #file 을 open 하면 close 한다
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io2.py

file_io2

Project

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - file_io1.py
 - file_io2.py
 - input.txt
- External Libraries
- Scratches and Consoles

```

2      # read(): 파일 객체로부터 모든 내용을 읽는 함수
3      # readline(): 파일 객체로부터 한 줄씩 문자열을 읽는 함수
4      # readlines(): 전체 내용을 한 번에 리스트에 담는 함수
5
6      f=open("input.txt", "r", encoding= "UTF-8")
7      list = f.readlines()
8      for i, data in enumerate(list): #enumerate()는 각 원소에 접근 수 있도록 함
9          print("%d번째 줄: %s" %(i+1, data), end='')
10     f.close()
11     |

```

Run: file_io2

```

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io2.py
0번째 줄: 안녕하세요?
1번째 줄: 홍길동입니다.
2번째 줄: 세 번째 줄에는 무슨 말을 쓸까요?
Process finished with exit code 0

```

Structure
Favorites

Python Console Terminal 4: Run 6: TODO

Event Log

file_io3.py

특정 파일을 읽어서 그 객체를 **f** 라고 하겠다 고 설정한다.

with 구문이 나오는 순간 자동으로 파일 객체의 메모리가 자동 해제 되기 때문에

굳이 파일의 **close** 함수를 불러오지 않아도 자동으로 파일 객체에 사용되는 리소스가 자동으로

해제된다는 특징이 있다.

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io3.py

file_io3

Project
swp2 ~/PycharmProjects/swp2

venv

file_io.py

file_io1.py

file_io2.py

file_io3.py

input.txt

External Libraries

Scratches and Consoles

```

13 # 굳이 파일의 close 함수를 불러오지 않아도 자동으로 파일 객체에 사용되는 리소스가 자동으로
14 # 해제된다는 특징이 있다.
15
16 with open("input.txt", "r", encoding="UTF-8") as f:
17     list = f.readlines()
18     # 모드 줄을 읽 내용을 list에 담았다가 list 원소 처리기법 활용할 수 있다.
19     for i, data in enumerate(list): #enumerate()는 각 원소에 접근 수 있도록
20         # 리스트의 인덱스와 원소를 접근한다.
21         print("%d번째 줄: %s" %(i+1, data), end='')
22     # open하는 부분과 close 하는 부분이 생략된다.
23
24
25
26

```

with open("input.txt", "r", enc... > for i, data in enumerate(list)

Run: file_io3

⚙️

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io3.py

1번째 줄: 안녕하세요?

2번째 줄: 홍길동입니다.

3번째 줄: 세 번째 줄에는 무슨 말을 쓸까요?

Process finished with exit code 0

1: Structure 2: Favorites

Python Console Terminal 4: Run 6: TODO

Event Log

File Edit View Navigate Code Refactor Run Tools VCS Window Help

swp2 file_io4.py

file_io4 ▶ ⚙️ 🔍

Project ▾

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - file_io1.py
 - file_io2.py
 - file_io3.py
 - file_io4.py
 - input.txt
 - input1.txt
 - External Libraries
 - Scratches and Consoles

```

9      # dict는 key와 value, 키와 값으로 구성되어 있기 때문에
10     # 키:알파벳, 값: 빈도수
11     dict={}
12     data=f.read()
13     for i in data: # 각 문자열을 방문하면서 하나씩 체크한다
14         if i in dict: # 자료형에 키로 문자 알파벳이 포함되어 있다면
15             dict[i] +=1 # 빈도수를 1씩 증가, 특정한 알파벳이 여러 번 나오는 경우, 1
16         else:
17             dict[i]=1 # 그렇지 않으면 1로 설정, 특정한 알파벳을 처음 발견했을 경우.
18
19
20     return dict # 해당 자료형을 리턴한다.
21
22

```

process() > with open(filename,"r") as f > for i in data

Run: file_io4 × ⚙️

```

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io4.py
{'H': 2, 'e': 7, 'l': 2, 'o': 6, ' ': 10, 'J': 1, 'i': 3, 'n': 2, 'a': 5, '!': 1, '\n': 3, 'I': 2, '"': 1,

Process finished with exit code 0

```

IDE and Plugin Updates
PyCharm is ready to [update](#).

Python Console Terminal ▶ 4: Run ⚙️ TODO

Event Log

file_io3.py

```
with open("input.txt", "r", encoding= "UTF-8") as f:
```

```
    list = f.readlines()
```

```
    # 모드 줄을 읽 내용을 list에 담았다가 list 원소 처리기법 활용할 수 있다.
```

```
    for i, data in enumerate(list): #enumerate()는 각 원소에 접근 수 있도록
```

```
        # 리스트의 인덱스와 원소를 접근한다.
```

```
        print("%d번째 줄: %s" %(i+1, data), end="")
```

```
# open하는 부분과 close 하는 부분이 생략된다.
```

```
# 코드가 짧아지고 간결해 지는 것은 우리가 원하는 목표이다.
```

file_io4.py

a={'H': 2} a[0]='H', a[1]=2 가 들어가 있는 상황이다.

dict.items()는 키와 값(빈도수)을 가지는 한 개의 쌍이다.

이 때 정렬 기준은 dict.items()에서 얻은 **a**라는 키와 값의 쌍에서 **a[0]**는 키를 의미하고

a[1]은 값(빈도수)을 의미한다.

reverse=True는 값(빈도수)을 내림차순 정렬

print(dict)

dict는 가장 많은 빈도수를 가진 문자를 먼저 출력해준다.

file_io5

dict는 가장 많은 빈도수를 가진 문자를 먼저 출력해준다.

dict는 리스트 형태를 가진다.

[(' ', 10), ('e', 7), ('o', 6), ('a', 5), ('t', 4), ('i', 3), ('\n', 3), ('m', 3), ('s', 3), ('u', 3), ('d', 3), ('c', 3), ('H', 2), ('l', 2), ('n', 2), ('l', 2), ('h', 2), ('.', 2), ('r', 2), ('y', 2), ('J', 1), ('!', 1), ('"', 1), ('w', 1), ('?', 1), ('v', 1), ('p', 1)]

각 원소에 접근해 보면, (' ', 10) 튜플형태를 가지고 있는 것을 확인할 수 있다..

튜플의 첫 번째 원소를 **data**로 받고, 튜플의 두 번째 원소를 **count**로 받는다

첫 번째 원소는 ',', 'e'등으로 문자, **data** 가 되며, 두 번째 원소는 빈도수 **count**가 된다.

문자, 데이터 중에서 '\n' : 줄바꿈기호, 혹은 ':공백문자' 일때는 출력하지 않고 무시하고 넘어간다.

continue는 반복문을 무시하고 넘어간다

즉, 알파벳만 출력한다.

각 문자별 출현횟수를 출력한다

swp2 file_io5.py

file_io5 ▶ ⚙️ 🔍

Project ▾

- swp2 ~/PycharmProjects/swp2
 - venv
 - file_io.py
 - file_io1.py
 - file_io2.py
 - file_io3.py
 - file_io4.py
 - file_io5.py
 - input.txt
 - input1.txt
- External Libraries
- Scratches and Consoles

```

33
34 # dict는 리스트 형태를 가진다.
35 # [(' ', 10), ('e', 7), ('o', 6), ('a', 5), ('t', 4), ('i', 3), ('\n', 3), ('r', 3)]
36 # 각 원소에 접근해 보면, (' ', 10) 튜플형태를 가지고 있는 것을 확인할 수 있다.
37 # 튜플의 첫 번째 원소를 data로 받고, 튜플의 두 번째 원소를 count로 받는다
38 # 첫 번째 원소는 ' ', 'e'등으로 문자, data 가 되며, 두 번째 원소는 빈도수 count가 된다.
39 # 문자, 데이터 중에서 '\n' : 줄바꿈기호, 혹은 ' ':공백문자 일때는 출력하지 않고 무시하고 넘어간다
40 # continue는 반복문을 무시하고 넘어간다
41 # 즉, 알파벳만 출력한다.
42 # 각 문자별 출현횟수를 출력한다.
43
44 for data, count in dict:
45     if data == '\n' or data == ' ':
46         continue
47     print("%d 번 출현: [%c]" %(count, data))
48

```

Run: file_io5

⚙️ -

/home/sckhome/PycharmProjects/swp2/venv/bin/python /home/sckhome/PycharmProjects/swp2/file_io5.py

```

7 번 출현: [e]
6 번 출현: [o]
5 번 출현: [a]
4 번 출현: [t]
3 번 출현: [i]
3 번 출현: [m]
3 번 출현: [s]
3 번 출현: [u]
3 번 출현: [d]

```

IDE and Plugin Updates
PyCharm is ready to [update](#).