입력과 출력

표준 입출력(Standard Input/Output)

표준 입력 함수

- `input()` 함수: 콘솔 창을 통해서 사용자 입력을 받는 표준 입력 함수
- `input()` 함수 안에 입력 받기 위한 질문을 텍스트로 넣을 수 있음

```
name = input("이름: ")
print(name)
```

```
이름: 이수안
이수안
```

[Lab] 섭씨 온도를 화씨 온도로 변환

- 섭씨 온도를 화씨 온도로 변환하는 공식: $F = (C * 1.8) + 32$

```
celsius = float(input("섭씨 온도: "))
fahrenheit = (celsius * 1.8) + 32
print("화씨 온도:", fahrenheit)
```

```
섭씨 온도: 28
화씨 온도: 82.4
```

[Lab] 구구단 중의 하나의 단을 입력받아 계산

- 출력할 단을 입력
- 해당 단의 계산 결과 출력

```
i = int(input("출력할 단: "))
for j in range(1, 10):
  print("{0} x {1} = {2}".format(i, j, i * j))
```

```
출력할 단: 9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
```

표준 출력 함수

- `print()` 함수: 콘솔 창을 통해서 결과를 출력하는 표준 출력 함수
- sep='separator': 개체가 분리하는 방법 지정, 기본값은 ' '
- end='end': 끝에 출력할 항목 지정, 기본값은 '\n'
- file: 쓰기 방법이 있는 객체, 기본값은 sys.stdout
- flush: True일 경우 flush하고, False일 경우 버퍼, 기본값은 False

```
print("Hello Python")
print("안녕 파이썬")
print(7)
print(3.14)
print([1, 2, 3])

print("Hello", "Python", sep = "---")
print(1, 2, 3)
print(1, end=' ')
print(2, end=' ')
print(3, end=' ')
```

```
Hello Python
안녕 파이썬
7
3.14
[1, 2, 3]
Hello---Python
1 2 3
1 2 3
```

[Lab] 구구단 출력

- `print()` 함수를 사용하여 각 단마다 횡으로 출력

```
for i in range(2, 10):
  for j in range(1, 10):
    print("{0}x{1}={2}".format(i, j, i * j), end='₩t')
  print()
```

```
2x1=2	2x2=4	2x3=6	2x4=8	2x5=10	2x6=12	2x7=14	2x8=16	2x9=18
3x1=3	3x2=6	3x3=9	3x4=12	3x5=15	3x6=18	3x7=21	3x8=24	3x9=27
4x1=4	4x2=8	4x3=12	4x4=16	4x5=20	4x6=24	4x7=28	4x8=32	4x9=36
```

```
5x1=5   5x2=10  5x3=15  5x4=20  5x5=25  5x6=30  5x7=35  5x8=40  5x9=45
6x1=6   6x2=12  6x3=18  6x4=24  6x5=30  6x6=36  6x7=42  6x8=48  6x9=54
7x1=7   7x2=14  7x3=21  7x4=28  7x5=35  7x6=42  7x7=49  7x8=56  7x9=63
8x1=8   8x2=16  8x3=24  8x4=32  8x5=40  8x6=48  8x7=56  8x8=64  8x9=72
9x1=9   9x2=18  9x3=27  9x4=36  9x5=45  9x6=54  9x7=63  9x8=72  9x9=81
```

파일 입출력(File Input/Output)

파일 입출력 과정

- 파일의 입출력 과정은 단계를 가짐
- 파일을 열고, 파일을 읽거나 쓰고, 파일을 닫는 순서

파일 열기/닫기

- file.txt 파일을 생성하고, 열고 닫기

```
f = open("file.txt", 'w')
f = open("file.txt", 'r')
f.close()
```

```
!ls file.txt
```

```
file.txt
```

파일 모드(File Mode)

파일 모드	설명
(생략)	r과 동일한 모드
r	읽기 모드(기본값)
w	쓰기 모드, 기존에 파일이 있으면 덮어쓰기

파일 모드	설명
a	쓰기 모드, 기존에 파일이 있으면 이어서 쓰기(append)
+	읽기/쓰기 모드
t	텍스트 모드(기본값), 텍스트 파일을 처리
b	이진 모드, 이진 파일을 처리

텍스트 파일 쓰기

write()

- `write()`를 이용하여 파일에 쓰기

```
f = open("file.txt", 'w')
f.write("Hello Python")
f.close()
```

```
!cat file.txt
```

```
Hello Python
```

writelines()

- `writelines()`를 이용하여 list를 파일에 쓰기

```
list = ["One₩n", "Two₩n", "Three₩n"]
f = open("file.txt", 'w')
f.writelines(list)
f.close()
```

```
!cat file.txt
```

```
One
Two
Three
```

표준 입력 → 파일 쓰기

- 표준 입력을 `input()` 함수를 통해 입력 받고, `write()` 함수를 통해 파일에 쓰기

```
text = input("입력: ")
```

```
f = open("file.txt", 'w')
f.write(text)
f.close()
```

```
입력: 이수안
```

```
!cat file.txt
```

```
이수안
```

```
text = [str(text + '₩n') for text in input("여러 값 입력: ").split()]
f = open("file.txt", 'w')
f.writelines(text)
f.close()
```

```
여러 값 입력: 1 2 3 4 5
```

```
!cat file.txt
```

```
1
2
3
4
5
1
2
3
4
5
q
```

```
x = ""
text = ""
while x != 'q':
  x = input("반복 입력: ")
  text += x + '₩n'

f = open("file.txt", 'w')
f.writelines(text)
f.close()
```

```
반복 입력: 1
반복 입력: 2
반복 입력: 3
반복 입력: 4
반복 입력: 5
반복 입력: q
```

```
!cat file.txt
```

```
1
2
3
4
```

```
5
q
```

[Lab] 구구단 결과를 파일에 쓰기

- 구구단 결과를 파일에 쓰기

```
f = open("dan.txt", 'w')
for i in range(2, 10):
  for j in range(1, 10):
    f.write("{0} x {1} = {2}₩n".format(i, j, i * j))
  else:
    f.write("₩n")

f.close()
```

```
!cat dan.txt
```

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
```

```
5 x 8 = 40
5 x 9 = 45

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
```

텍스트 파일 출력

- 텍스트 파일 예제

```
!echo "동해물과 백두산이 마르고 닳도록" > anthem.txt
!echo "하느님이 보우하사 우리나라 만세" >> anthem.txt
```

readline()

- `readline()`를 이용하여 텍스트 파일을 라인 단위로 읽고 화면에 출력

```
f = open('anthem.txt', 'r')
while True:
  line = f.readline()
  if not line: break
  print(line)

f.close()
```

```
동해물과 백두산이 마르고 닳도록

하느님이 보우하사 우리나라 만세
```

readlines()

- `readlines()`를 이용하여 텍스트 파일의 여러 라인을 읽고 화면에 출력

```
f = open("anthem.txt", 'r')
lines = f.readlines()
print(lines)
f.close()
```

```
['동해물과 백두산이 마르고 닳도록₩n', '하느님이 보우하사 우리나라 만세₩n']
```

read()

- `read()`를 이용하여 텍스트 파일을 읽고 화면에 출력

```
f = open("anthem.txt", 'r')
data = f.read()
print(data)
f.close()
```

```
동해물과 백두산이 마르고 닳도록
하느님이 보우하사 우리나라 만세
```

tell()

- 파일 포인터 위치 변경

```
f = open("anthem.txt", 'r')
while True:
  print(f.tell())
  line = f.readline()
  if not line: break
  print(line)

f.close()
```

```
0
동해물과 백두산이 마르고 닳도록

46
하느님이 보우하사 우리나라 만세

92
```

seek()

- 파일 포인터 위치 이동

```
f = open("anthem.txt", 'r')
```

```
f.seek(46)
line = f.readline()
print(line)
f.seek(0)
line = f.readline()
print(line)
f.close()
```

```
 하느님이 보우하사 우리나라 만세

 동해물과 백두산이 마르고 닳도록
```

with 문

- 항상 파일을 `open()` 함수로 열고, `close()` 함수로 닫아야 하는 일을 자동으로 처리
- with문을 이용하면 with 블록 내에서 파일을 열고 벗어나면 파일을 닫음

```
f = open("file.txt", 'w')
f.write("Hello Python")
f.close()
```

```
with open("file.txt", 'w') as f:
  f.write("Hello Python")
```

[Lab] 표준 입력으로 받은 내용을 파일로 쓰기

- 사용자로부터 표준 입력 받음
- 받은 내용을 계속 파일 쓰기

```
f = open("file.txt", 'w')
line = ""
while True:
  line = input()
  if not line: break
  f.writelines(line + '₩n')

f.close()
```

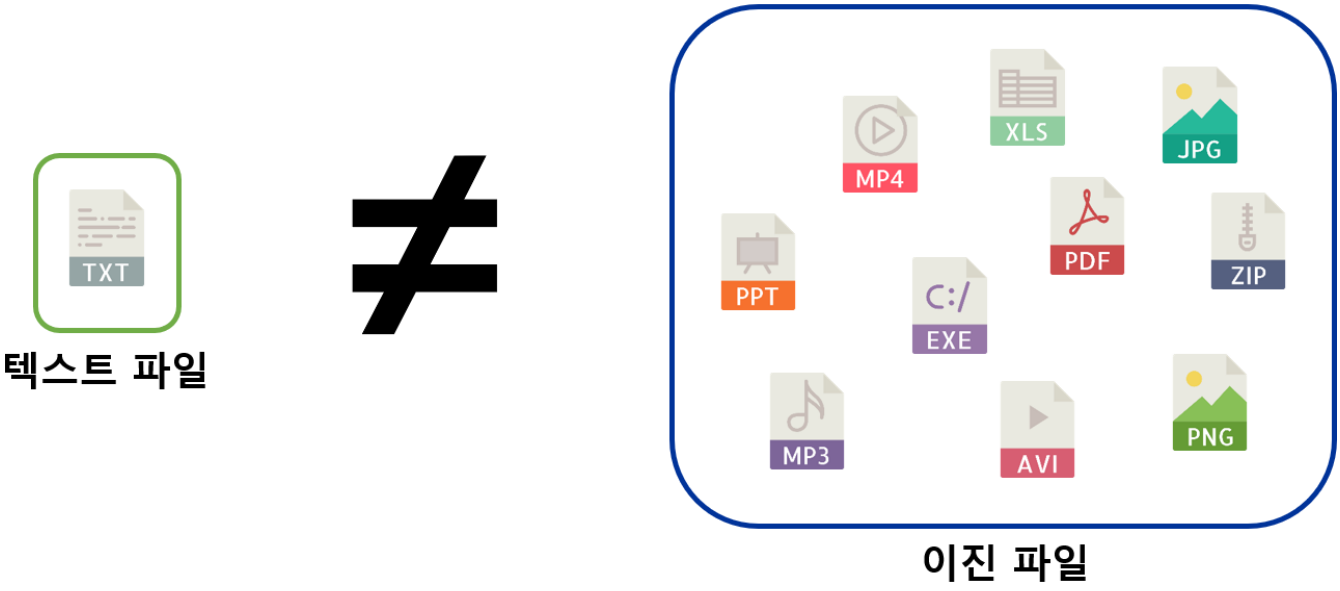
```
 동해물과 백두산이 마르고 닳도록
 하느님이 보우하사 우리나라 만세
```

```
!cat file.txt
```

```
 동해물과 백두산이 마르고 닳도록
 하느님이 보우하사 우리나라 만세
```

텍스트 파일 vs. 이진 파일

- 텍스트 파일(text file): 사람이 읽을 수 있는 텍스트로 구성된 파일
- 이진 파일(binary file): 텍스트가 아닌 비트 단위의 파일
- 이진 파일의 종류: 그림, 음악, 영상, 프로그램 파일 등



- 이진 파일(이미지) 예제

```
from google.colab import files
files.upload()
```


파일 선택 선택된 파일 없음 Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving unnamed.jpg to unnamed.jpg
{'unnamed.jpg': b'₩xff₩xd8₩xff₩xe0₩x00₩x10JFIF₩x00₩x01₩x01₩x00₩x00₩x01₩x00₩x01₩x00₩x00₩xff₩x


```
!ls
```

```
anthem.txt  dan.txt  file.txt  sample_data  unnamed.jpg
```

이진 파일 복사

- 하나의 이진 파일을 읽어서 다른 이진 파일로 쓰기 (복사)

```
input = open("unnamed.jpg", "rb")
output = open("unnamed_copy.jpg", "wb")
while True:
  fp = input.read(1)
  if not fp: break
  output.write(fp)

input.close()
```

```
output.close()
```

```
!ls
```

```
anthem.txt  dan.txt  file.txt  sample_data  unnamed_copy.jpg  unnamed.jpg
```

디렉토리 및 파일 처리

디렉토리 생성

- 디렉토리와 파일을 다루는 다양한 함수를 제공하는 shutil와 os 라이브러리
- `mkdir()`: 디렉토리 생성

```
import os
import shutil
os.mkdir('test')
os.mkdir('test/1')
os.mkdir('test/2')
os.mkdir('test/3')
```

```
!ls
```

```
anthem.txt  dan.txt  file.txt  sample_data  test  unnamed_copy.jpg  unnamed.jpg
```

```
!ls test
```

```
1  2  3
```

디렉토리 및 파일 복사

- `shutil.copy()`: 파일 복사
- `shutil.copytree()`: 디렉토리 전체 복사

```
import shutil
shutil.copy('unnamed.jpg', 'unnamed_copy2.jpg')
```

```
'unnamed_copy2.jpg'
```

```
!ls
```

```
anthem.txt  file.txt     test               unnamed_copy.jpg
dan.txt     sample_data  unnamed_copy2.jpg  unnamed.jpg
```

```
import shutil
shutil.copytree('test', 'test2')
```

```
'test2'
```

```
!ls
```

```
anthem.txt  file.txt     test   unnamed_copy2.jpg  unnamed.jpg
dan.txt     sample_data  test2  unnamed_copy.jpg
```

```
!ls test2
```

```
1  2  3
```

디렉토리 및 파일 확인

- `isdir()`: 디렉토리 존재 확인
- `isfile()`: 파일 존재 확인
- `exists()`: 디렉토리/파일 존재 확인

```
import os.path
print(os.path.isdir('test'))
print(os.path.isdir('test/1'))
print(os.path.isdir('test/4'))
print(os.path.isfile('unnamed.jpg'))
print(os.path.isfile('unnamed_copy.jpg'))
print(os.path.isfile('unnamed_copy2.jpg'))
print(os.path.exists('test2'))
print(os.path.exists('test2/1'))
print(os.path.exists('test2/4'))
```

```
True
True
False
True
True
True
True
True
False
```

디렉토리 목록 보기

- `os.walk()`: 디렉토리 목록 보기

```
import os
for dir_name, dir_list, file_names in os.walk('./'):
  for file_name in file_names:
    print(os.path.join(dir_name, file_name))
```

```
./anthem.txt
./unnamed.jpg
./unnamed_copy.jpg
./dan.txt
```

```
./unnamed_copy2.jpg
./file.txt
./.config/config_sentinel
./.config/.last_update_check.json
./.config/.last_survey_prompt.yaml
./.config/gce
./.config/.last_opt_in_prompt.yaml
./.config/active_config
./.config/.metricsUUID
./.config/logs/2020.06.17/16.18.24.878976.log
./.config/logs/2020.06.17/16.18.44.263522.log
./.config/logs/2020.06.17/16.18.30.389925.log
./.config/logs/2020.06.17/16.18.11.544396.log
./.config/logs/2020.06.17/16.17.52.116219.log
./.config/logs/2020.06.17/16.18.44.921040.log
./.config/configurations/config_default
./sample_data/README.md
./sample_data/anscombe.json
./sample_data/mnist_test.csv
./sample_data/california_housing_test.csv
./sample_data/mnist_train_small.csv
./sample_data/california_housing_train.csv
```

디렉토리 및 파일 삭제

- `os.remove()`: 파일 삭제
- `shutil.rmtree()`: 디렉토리 안에 모든 파일/디렉토리 삭제

```
import os
os.remove('unnamed_copy2.jpg')
os.remove('unnamed_copy.jpg')
```

```
!ls
```

```
anthem.txt  dan.txt  file.txt  sample_data  test  test2  unnamed.jpg
```

```
import shutil
shutil.rmtree('test')
shutil.rmtree('test2')
```

```
!ls
```

```
anthem.txt  dan.txt  file.txt  sample_data  unnamed.jpg
```

파일 크기

- `os.path.getsize()`: 파일의 크기를 바이트 단위로 출력

```
import os.path
print(os.path.getsize('unnamed.jpg'))
```

```
40229
```

파일 압축

- zipfile: 압축 관련 라이브러리
- `ZipFile()`: 압축 파일 열기
- `write()`: 압축 파일에 쓰기
- `extractall()`: 전체 압축 해제
- `close()`: 압축 파일 닫기

```
import zipfile
comp = zipfile.ZipFile('new.zip', 'w')
comp.write('unnamed.jpg')
comp.close()
```

```
!ls
```

```
anthem.txt  dan.txt  file.txt  new.zip  sample_data  unnamed.jpg
```

```
decomp = zipfile.ZipFile('new.zip', 'r')
decomp.extractall('new')
decomp.close()
```

```
!ls new
```

```
unnamed.jpg
```

```
```