

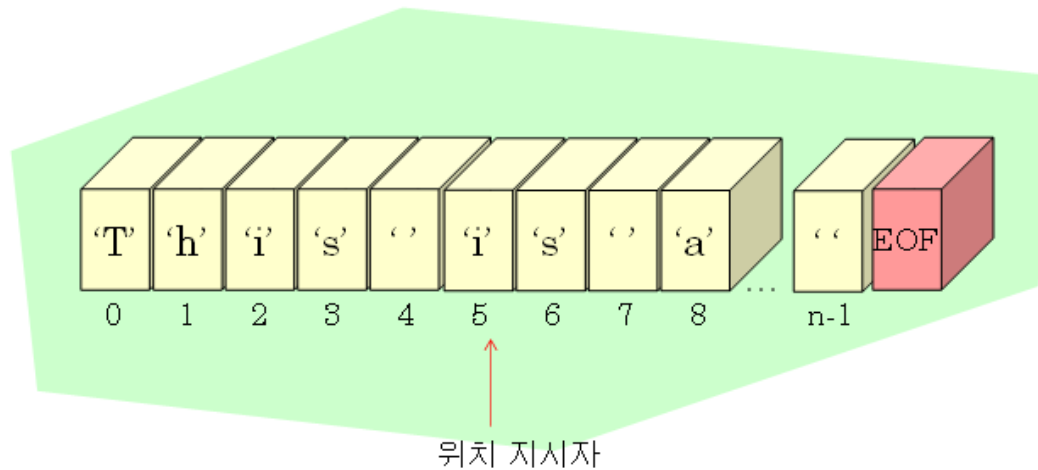
제7장 파일과 예외처리

파일의 필요성



논리적인 파일 구조

- 파일 안에는 바이트들이 순차적으로 저장되어 있고 맨 끝에는 EOF(end-of-file) 마커가 있다.
- 모든 파일은 입출력 동작이 발생하는 위치를 나타내는 위치 표시자(position indicator)를 가지고 있다.



파일의 논리적인 구성

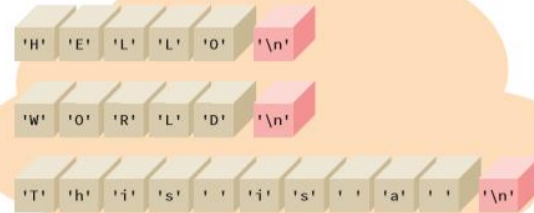
파일의 종류

텍스트 파일(text file)

이진 파일(binary file)



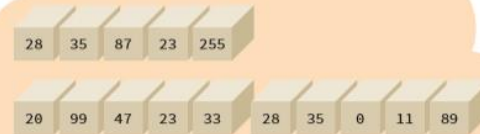
텍스트 파일: 문자로 구성된 파일



텍스트 파일



이진파일: 데이터로 구성된 파일



이진 파일

텍스트 파일 읽고 쓰기

전체적인 구조



파일 객체

```
infile = open( "input.txt" , "r" )
```

```
...
```

```
infile.close()
```

파일의 이름(name)

파일을 여는 모드(mode)

파일 모드

파일 모드	모드 이름	설명
"r"	읽기 모드(read mode)	파일의 처음부터 읽는다.
"w"	쓰기 모드(write mode)	파일의 처음부터 쓴다. 파일이 없으면 생성된다. 만약 파일이 존재하면 기존의 내용은 지워진다.
"a"	추가 모드(append mode)	파일의 끝에 쓴다. 파일이 없으면 생성된다.
"r+"	읽기와 쓰기 모드	파일에 읽고 쓸 수 있는 모드이다. 모드를 변경하려면 seek()가 호출되어야 한다.



"r"

파일을 처음부터 읽는다.



"w"

파일의 처음부터 쓴다.
만약 파일이 존재하면
기존의 내용이 지워진다.



"a"

파일의 끝에 쓴다.
파일이 없으면 생성된다.

예제

```
infile = open("phones.txt", "r")  
s = infile.read(10)  
print(s);  
infile.close()
```

홍길동 010-12

예제

```
infile = open("phones.txt", "r")
s = infile.readline()
print(s);
s = infile.readline()
print(s);
s = infile.readline()
print(s);
infile.close()
```

```
홍길동 010-1234-5678
김철수 010-1234-5679
김영희 010-1234-5680
```


예제

```
infile = open("phones.txt", "r")
line = infile.readline()
while line != "":
    print(line);
    line = infile.readline()
infile.close()
```

```
홍길동 010-1234-5678
김철수 010-1234-5679
김영희 010-1234-5680
```

예제

```
infile = open("phones.txt", "r")
for line in infile:
    line = line.rstrip()
    print(line)
infile.close()
```

```
홍길동 010-1234-5678
김철수 010-1234-5679
김영희 010-1234-5680
```

파일에 데이터 쓰기

```
import os.path

outfile = open("phones.txt", "w")

if os.path.isfile("phones.txt"):
    print("동일한 이름의 파일이 이미 존재합니다. ")
else :
    outfile.write("홍길동 010-1234-5678")
    outfile.write("김철수 010-1234-5679")
    outfile.write("김영희 010-1234-5680")

outfile.close()
```

LAB: 매출 파일 처리

입력 파일에는 상점의 하루 매출이 한 줄에 정수로 기록되어 있다. 예를 들면 다음과 같다.

sales.txt

```
1000000  
1000000  
1000000  
500000  
1500000
```

출력 파일은 다음과 같아야 한다.

summary.txt

```
총매출 = 5000000  
평균 일매출 = 1000000.0
```

SOLUTION

```
# 입력 파일 이름과 출력 파일 이름을 받는다.
infilename = input("입력 파일 이름: ");
outfilename = input("출력 파일 이름: ");

# 입력과 출력을 위한 파일을 연다.
infile = open(infilename, "r")
outfile = open(outfilename, "w")

# 합계와 횟수를 위한 변수를 정의한다.
sum = 0
count = 0

# 입력 파일에서 한 줄을 읽어서 합계를 계산한다.
for line in infile:
    dailySale = int(line)
    sum = sum + dailySale
    count = count + 1

# 총매출과 일평균 매출을 출력 파일에 기록한다.
outfile.write("총매출 = "+ str(sum)+"\n")
outfile.write("평균 일매출 = "+ str(sum/count) + "\n")

infile.close()
outfile.close()
```

텍스트 입출력 기법

데이터 추가하기

```
outfile = open("phones.txt", "a")  
  
outfile.write("최무선 010-1111-2222")  
outfile.write("정중부 010-2222-3333")  
  
outfile.close()
```

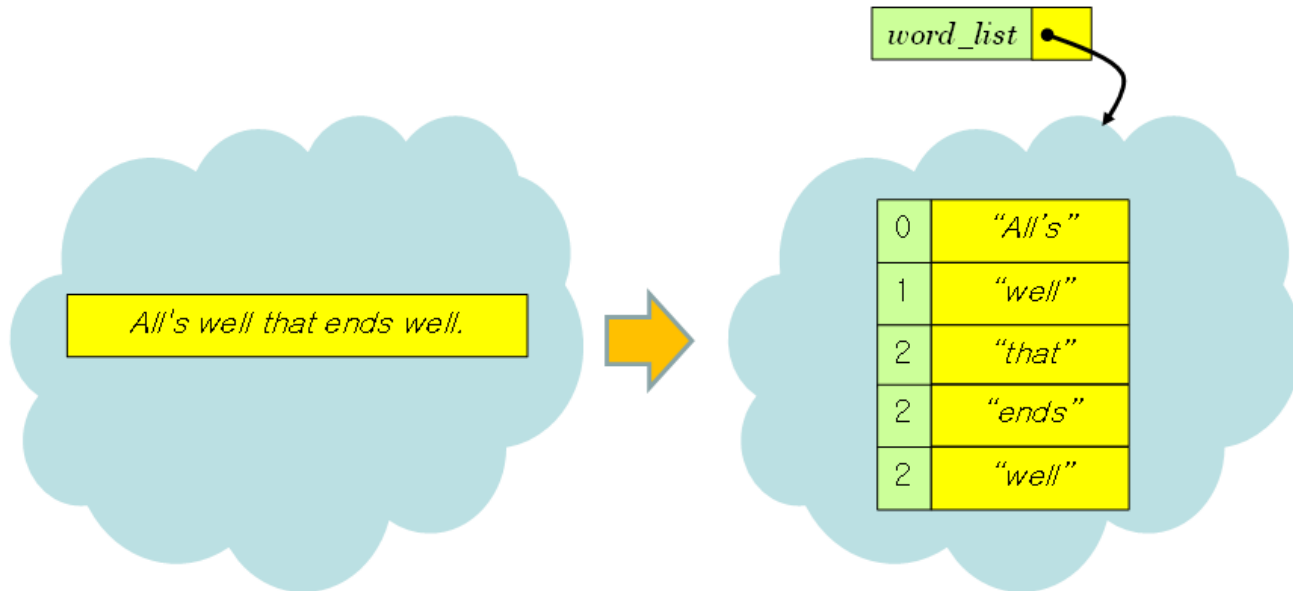
텍스트 입출력 기법

줄바꿈 기호 삭제하기

```
infile = open("proverbs.txt", "r")
for line in infile:
    line = line.rstrip()
    print(line);
infile.close()
```

텍스트 입출력 기법

파일에서 단어 읽기




```
infile = open("proverbs.txt", "r")
for line in infile:
    line = line.rstrip()
    word_list = line.split()
    for word in word_list:
        print(word);
infile.close()
```

All's
well
that
ends
well.
...
flock
together.

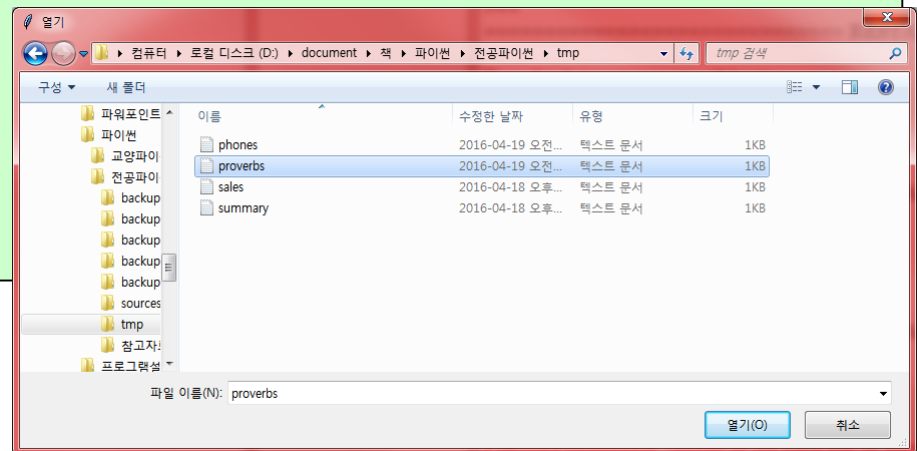
파일 대화 상자

```
from tkinter import *  
from tkinter.filedialog import askopenfilename  
from tkinter.filedialog import asksaveasfilename
```

```
readFile = askopenfilename()  
if( readFile != None):  
    infile = open(readFile, "r")
```

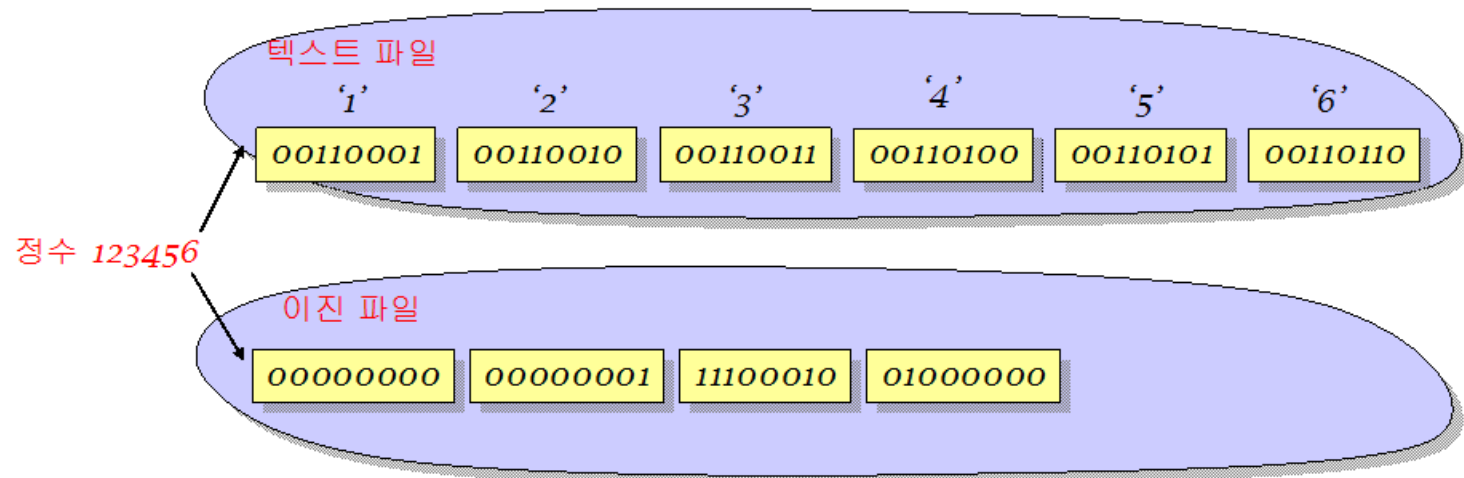
```
for line in infile.readlines():  
    line = line.strip()  
    print(line)
```

```
infile.close()
```



이진 파일

이진 파일(binary file)은 데이터가 직접 저장되어 있는 파일이다.



LAB: 이미지 파일 복사하기

- 예를 들어 평문 “come to me”은 “FRPH WR PH”으로 바뀐다. 시저 암호 방식을 이용하여서 파일을 암호화하고 복호화하는 프로그램을 작성하라.

원본 파일 이름을 입력하시오: 123.png
복사 파일 이름을 입력하시오: kkk.png
123.png를 kkk.png로 복사하였습니다.



SOLUTION

```
filename1 = input("원본 파일 이름을 입력하시오: ");
filename2 = input("복사 파일 이름을 입력하시오: ");

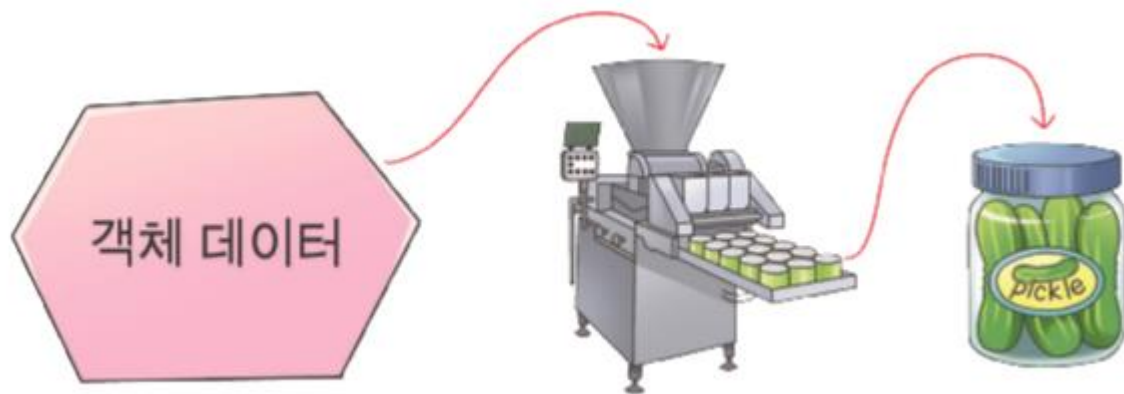
infile = open(filename1, "rb")
outfile = open(filename2, "wb")

# 입력 파일에서 1024 바이트씩 읽어서 출력 파일에 쓴다.
while True:
    copy_buffer = infile.read(1024)
    if not copy_buffer:
        break
    outfile.write(copy_buffer)

infile.close()
outfile.close()
print(filename1+"를 " +filename2+"로 복사하였습니다. ")
```

객체 입출력

pickle 모듈의 dump()와 load() 메소드를 사용하면 객체를 쓰고 읽을 수 있다.



에외 처리

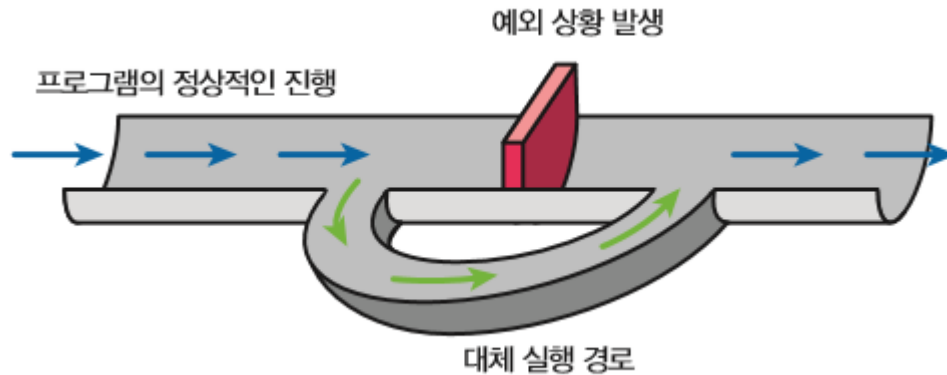
오류가 발생할 수 있다!



```
>>> (x, y)=(2, 0)
>>> z=x/y
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    z=x/y
ZeroDivisionError: division by zero
>>>
```

예외 처리의 개념

- 오류가 발생 했을 때 오류를 사용자에게 알려주고 모든 데이터를 저장하게 한 후에 사용자가 우아 하게(gracefully) 프로그램을 종료할 수 있도록 하는 것이 바람직하다.



파이썬 예외 처리

전체적인 구조



try :

예외가 발생할 수 있는 문장

except 오류내용 :

예외를 처리하는 문장

예제

```
(x,y) = (2,0)
try:
    z = x/y
except ZeroDivisionError:
    print ("0으로 나누는 예외")
```

0으로 나누는 예외

예제

```
while True:
    try:
        n = input("숫자를 입력하시오 : ")
        n = int(n)
        break
    except ValueError:
        print("정수가 아닙니다. 다시 입력하시오. ")
print("정수 입력이 성공하였습니다!")
```

```
숫자를 입력하시오 : 23.5
정수가 아닙니다. 다시 입력하시오.
숫자를 입력하시오 : 10
정수 입력이 성공하였습니다!
```

예제

```
try:
    fname = input("파일 이름을 입력하세요: ")
    infile = open(fname, "r")
except IOError:
    print("파일 " + fname + "을 발견할 수 없습니다.")
```

파일 이름을 입력하세요: kkk.py
파일 kkk.py을 발견할 수 없습니다.

다중 예외 처리 구조

전체적인 구조



try :

예외가 발생할 수 있는 문장

except ExceptionI :

ExceptionI이면 이 블록이 실행된다.

except ExceptionII :

ExceptionII이면 이 블록이 실행된다.

else :

예외가 없는 경우에 실행된다.

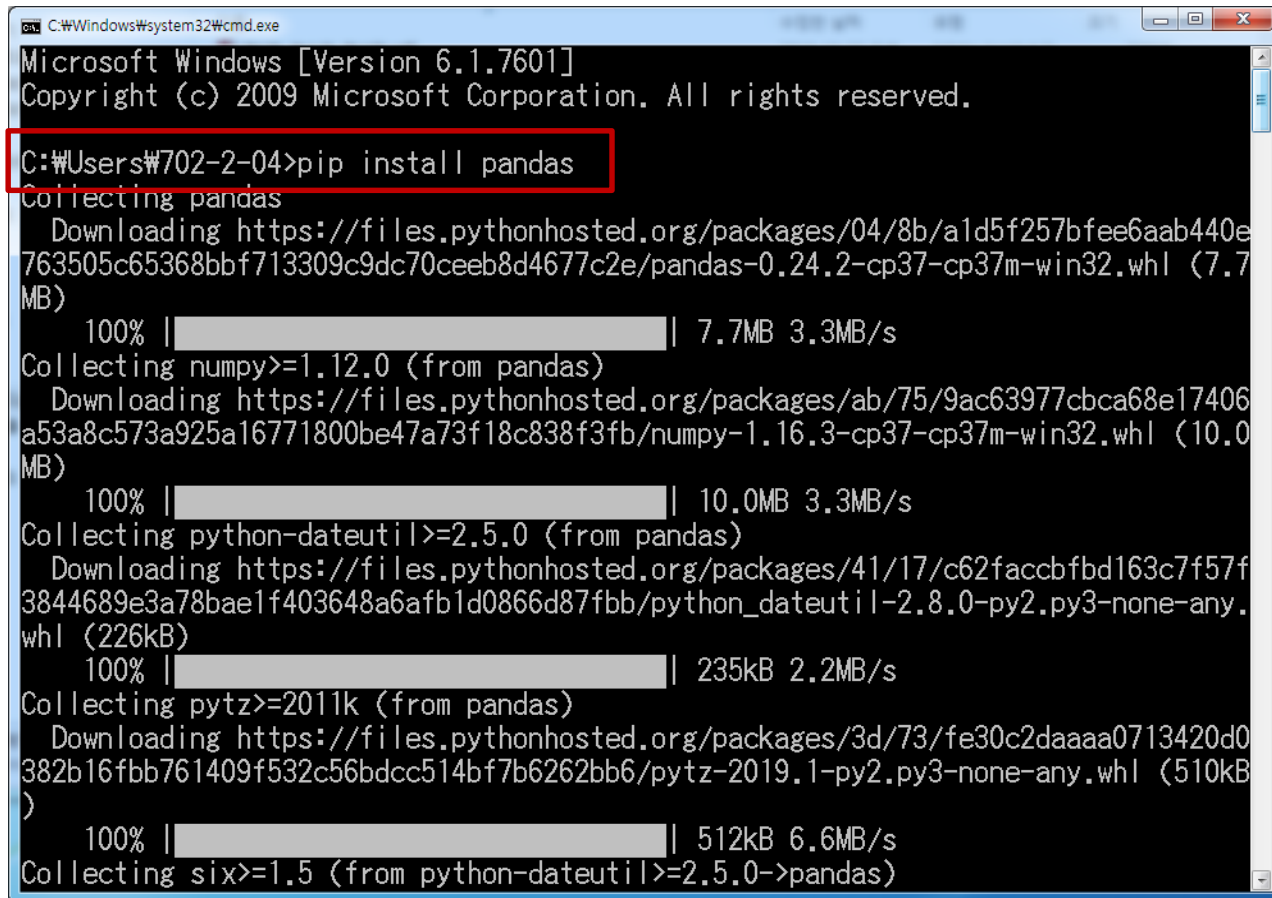
예제

```
try:
    fh = open("testfile", "w")
    fh.write("테스트 데이터를 파일에 씁니다!!")
except IOError:
    print("Error: 파일을 찾을 수 없거나 데이터를 쓸 수 없습니다. ")
else:
    print("파일에 성공적으로 기록하였습니다. ")
    fh.close()
```

파일에 성공적으로 기록하였습니다.

외부 패키지 설치 : 특수파일 입출력을 위한

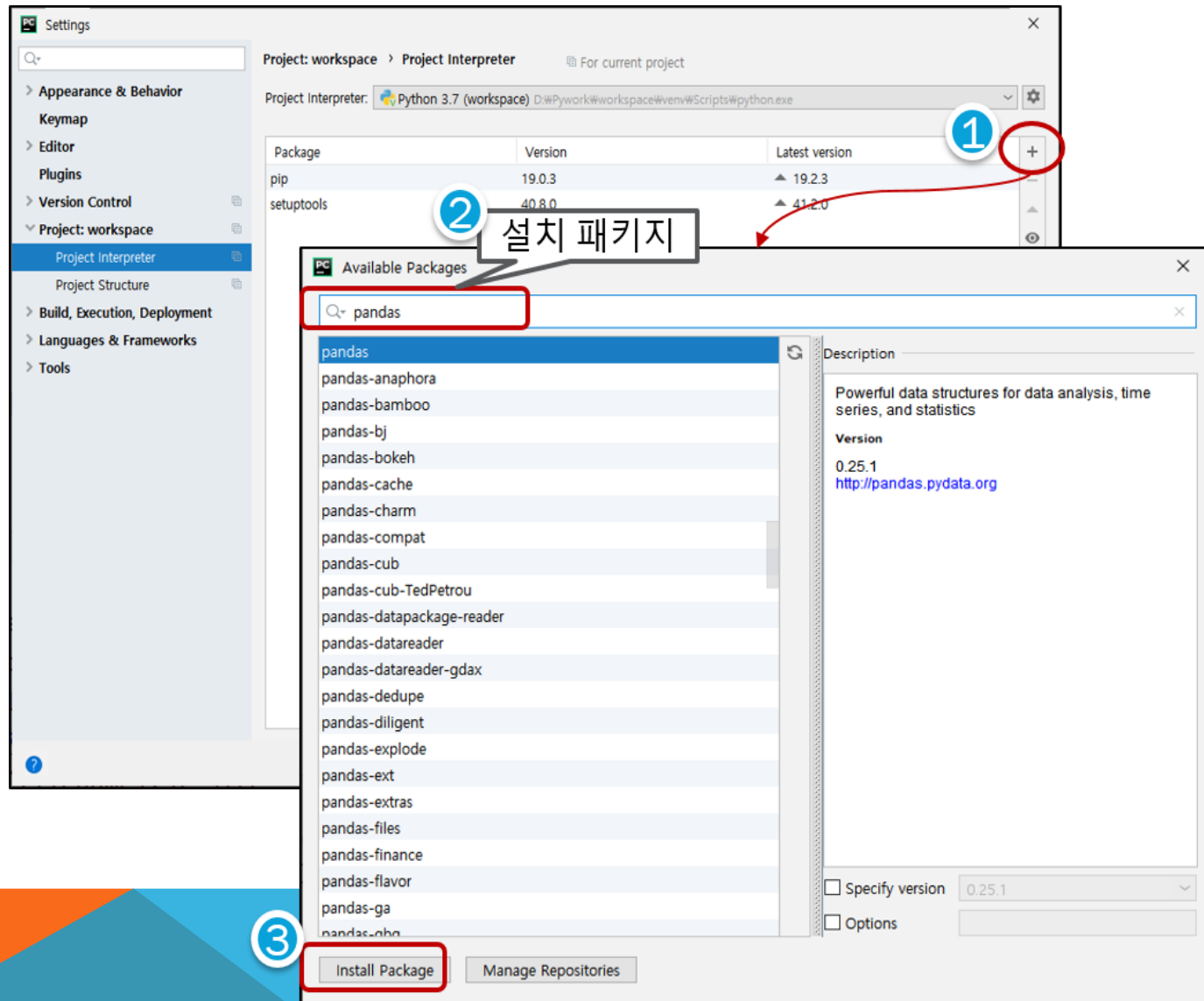
- 명령줄(command line)에서 pip 명령어 이용 패키지 설치



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\702-2-04>pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/04/8b/a1d5f257bfee6aab440e763505c65368bbf713309c9dc70ceeb8d4677c2e/pandas-0.24.2-cp37-cp37m-win32.whl (7.7MB)
    100% |#####| 7.7MB 3.3MB/s
Collecting numpy>=1.12.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/ab/75/9ac63977cbca68e17406a53a8c573a925a16771800be47a73f18c838f3fb/numpy-1.16.3-cp37-cp37m-win32.whl (10.0MB)
    100% |#####| 10.0MB 3.3MB/s
Collecting python-dateutil>=2.5.0 (from pandas)
  Downloading https://files.pythonhosted.org/packages/41/17/c62facbfbfd163c7f57f3844689e3a78bae1f403648a6afb1d0866d87fbb/python_dateutil-2.8.0-py2.py3-none-any.whl (226kB)
    100% |#####| 235kB 2.2MB/s
Collecting pytz>=2011k (from pandas)
  Downloading https://files.pythonhosted.org/packages/3d/73/fe30c2daaaa0713420d0382b16fbb761409f532c56bdcc514bf7b6262bb6/pytz-2019.1-py2.py3-none-any.whl (510kB)
    100% |#####| 512kB 6.6MB/s
Collecting six>=1.5 (from python-dateutil>=2.5.0->pandas)
```

- 파이참(Pycharm)에서 패키지 설치



- 엑셀(excel) 파일 읽기를 위한 패키지 설치

