

 CODINGO  X  posco

K-Digital Training 스마트 팩토리 3기

파일입출력

파일 입출력

파일 객체 = open(파일이름, 파일 열기 모드)

파일객체.close()

[파일 열기 모드]

r : 파일을 읽기 위해서 연다.(기본값)

w : 파일에 데이터를 쓰기 위해서 연다.

x : 새로운 파일을 생성해야 한다.

이때, 같은 이름의 파일이 존재하면 error 가 발생한다.

a : 파일의 뒷부분에 데이터를 추가하기 위해 파일을 연다.

b : 파일을 바이너리 데이터로 본다.

t : 파일을 텍스트 데이터로 본다.(줄바꿈, 인코딩 등을 자동으로 처리)

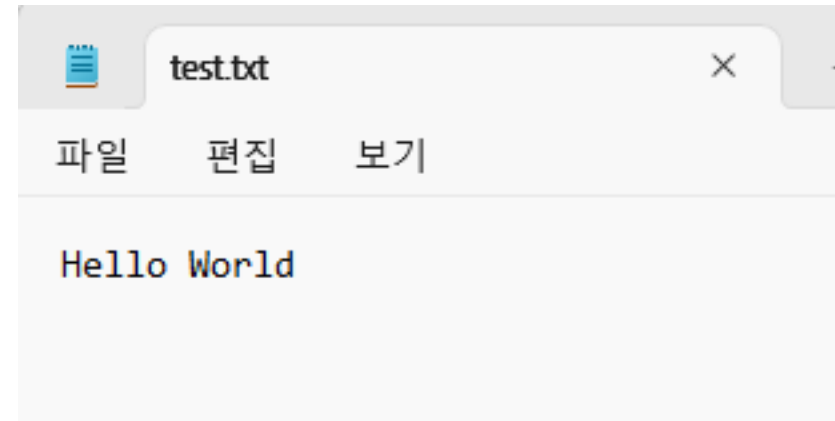
+ : 기존 파일 업데이트를 위해서 파일을 연다.

파일 입출력

w : 파일에 데이터를 쓰기 위해서 연다.

단, 기존의 내용이 모두 사라지고 새로 작성함.

```
f = open( "./test.txt", "w" )  
f.write( "Hello World \n" )  
f.close()
```



파일 입출력

r : 파일을 읽기 위해서 연다.(기본값)

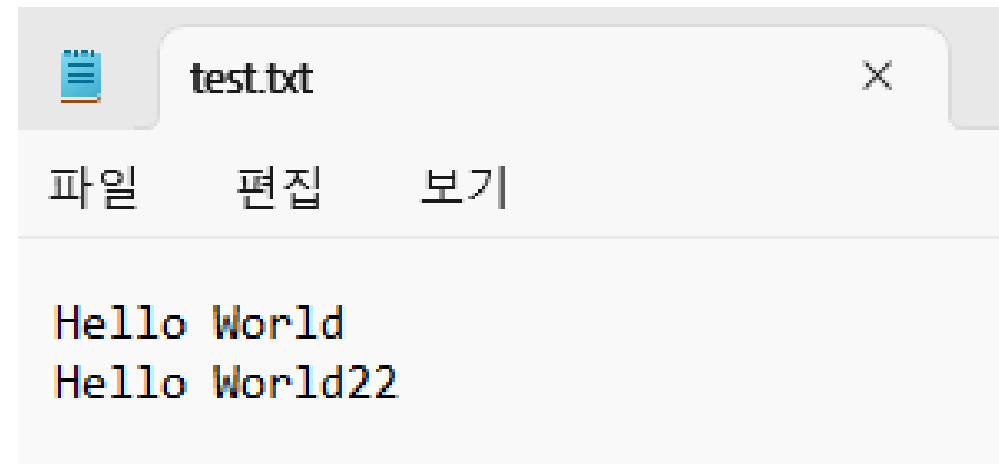
```
f2 = open( "./test.txt", "r" )  
print(f2.read())  
f2.close()
```

```
===== RESTART: C:\#U  
Hello World
```

파일 입출력

a : 파일의 뒷부분에 데이터를 추가하기 위해 파일을 연다.

```
f3 = open( "./test.txt", "a" )  
f3.write( "Hello World22 \n" )  
f3.close()
```



파일 입출력

주의! 파일을 다 사용했다면, 파일 닫아주기!

파일을 닫아주지 않으면 우리가 예기치 못한 이슈가 발생할 수 있다!

ex) write으로 작성한 데이터가 저장되지 않음.

```
f = open( "./test.txt", "w" )  
f.write( "Hello World \n" )  
  
f2 = open( "./test.txt", "r" )  
print(f2.read())  
..
```

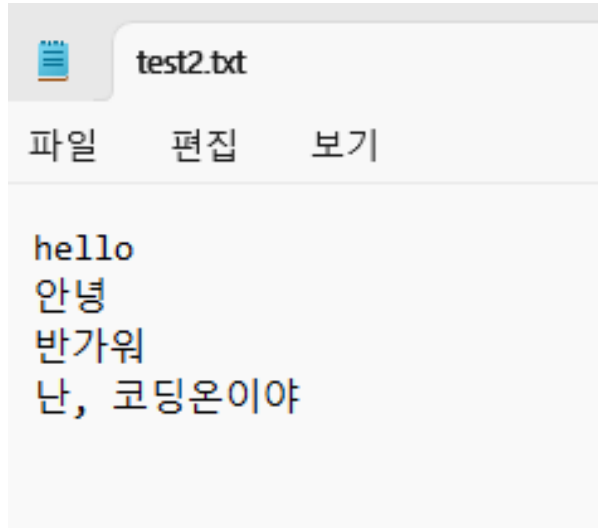
```
>>> |==== RESTART: C  
~~~~~  
아무런 내용도 출력되지 않음
```

파일 입출력

파일 읽기 모드 메소드

- read() : 파일의 내용 전체를 문자열로 반환
- readlines() : 파일의 모든 줄을 읽고, 각각의 줄을 요소로 갖는 리스트를 반환.
- seek(위치) : 해당하는 위치로 커서 이동. (* 한글은 2의 배수)
보통은 seek(0)을 이용하여 줄 앞으로 이동할 때 많이 사용
- tell() : 커서의 현재 위치 반환

파일 입출력



```
f4 = open("./test2.txt", "r")  
print(f4.read())
```

```
f4.seek(0) #커서를 쥔 앞으로 이동  
print(f4.tell())
```

```
context = f4.readlines()  
print(context)
```

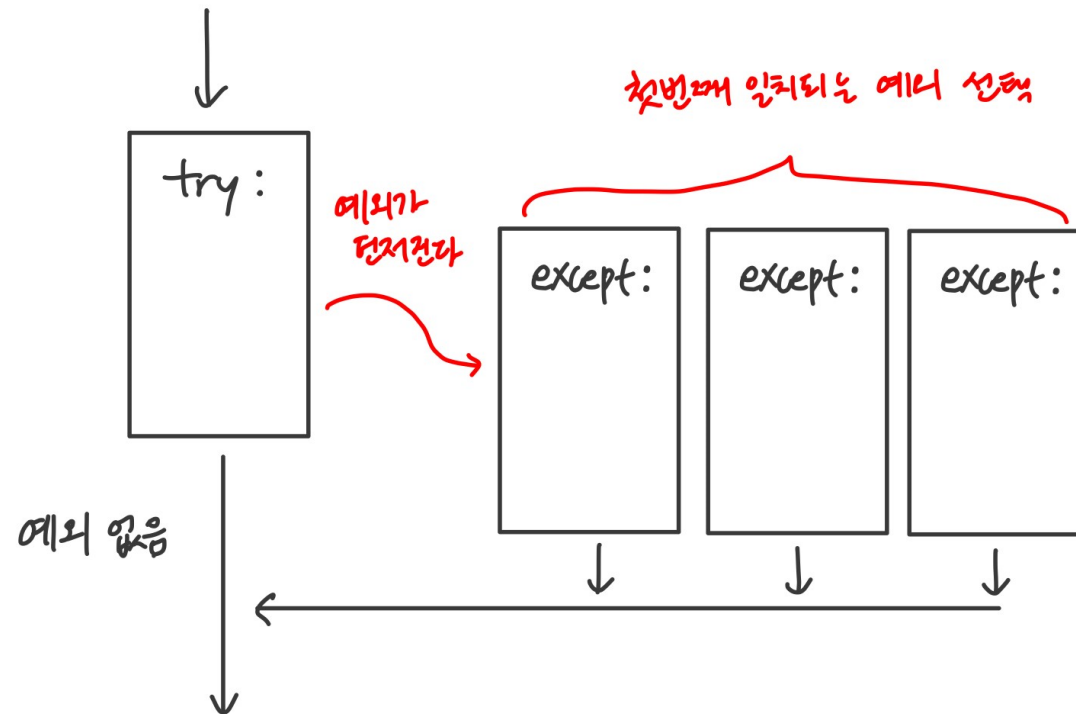
```
f4.close()
```

```
hello  
안녕  
반가워  
난, 코딩온이야  
0  
['hello\n', '안녕\n', '반가워\n', '난, 코딩온이야']
```

예외처리

예외 처리란?

에러가 발생할만한 부분을 예측하여, 미리 예외 상황에 대한 처리를 하는 것
Try 블록에서 발생한 예외를 except 블록에서 처리한다.



예외 처리

Try 블록에서 발생한 예외를 except 블록에서 처리한다.

try:

예외가 발생할 수 있는 문장

except 오류내용:

예외를 처리하는 문장

try:

예외가 발생할 수 있는 문장

except 오류내용1:

예외를 처리하는 문장

except 오류내용2:

예외를 처리하는 문장

else:

예외가 없는 경우에 실행

발생 가능한 예외 종류

- IndexError: 리스트 인덱스 범위 오류

```
shop = ["반팔", "청바지", "이어폰", "키보드"]
```

```
print(shop[2]) # 이어폰
```

```
print(shop[10])
```

Traceback (most recent call last):

File "C:/Users/Lily/github/lecture-dataAnalysis/python/exception_lec.py", line 4, in <module>

print(shop[10]) # IndexError : 리스트 길이를 넘는 인덱스로 요소에 접근하려할 때

IndexError: list index out of range

- ValueError: 부적절한 값을 가진 인자를 받았을 때 발생하는 에러

```
number = int("hello")
```

```
print(shop.index("없는값"))
```

Traceback (most recent call last):

File "C:/Users/Lily/github/lecture-dataAnalysis/python/exception_lec.py", line 8, in <module>

number = int("hello")

ValueError: invalid literal for int() with base 10: 'hello'

발생 가능한 예외 종류

- ZeroDivisionError: 0으로 나눌 때 발생하는 오류 `print(5/0)`

- NameError: 존재하지 않는 변수를 호출할 때 `a = 1`
`print(b)`

- FileNotFoundError: 존재하지 않는 파일을 호출할 때

```
file = open("./test3.txt", "r")
```

Traceback (most recent call last):

File "C:/Users/Lily/github/lecture-dataAnalysis/python/exception_lec.py", line 22, in <module>

file = open("./test3.txt", "r")

FileNotFoundError: [Errno 2] No such file or directory: './test3.txt'

예외처리 예시

```
try:
    num = int(input('숫자 입력: ')) # 예외가 발생할 수 있는 문장
except ValueError: # 오류 내용
    print('정수가 아님! 정수를 입력해주세요!') # 예외를 처리하는 안내 메시지
```

숫자 입력: 10 숫자 입력: 12.3
정수가 아님! 정수를 입력해주세요!

```
try:
    num = int(input('숫자 입력: '))
except ValueError as msg: # 오류 내용 (as 별칭)
    print(msg) #시스템이 보내는 예외 메시지 출력 가능
```

숫자 입력: 12.5
invalid literal for int() with base 10: '12.5'

예외처리 예시

```
# 다중 예외처리
try:
    num = [1, 2, 3, 4]
    print(who) # 예외가 발생할 수 있는 문장 (NameError)
    print(num[3]/0) # 예외가 발생할 수 있는 문장 (ZeroDivisionError)
    print(num[100]) # 예외가 발생할 수 있는 문장 (IndexError)
except NameError:
    print("존재하지 않는 변수 호출")
except IndexError:
    print("인덱스 에러 발생")
except ZeroDivisionError:
    print("0으로 나눌 수 없음!")
```

결과 : 존재하지 않는 변수 호출

=> NameError가 먼저 나오므로 IndexError, ZeroDivisionError 오류는 발생하지 않음

예외처리 예시

```
# 다중 예외처리
try:
    num = [1, 2, 3, 4]
    # print(who) # 예외가 발생할 수 있는 문장 (NameError)
    # print(num[3]/0) # 예외가 발생할 수 있는 문장 (ZeroDivisionError)
    # print(num[100]) # 예외가 발생할 수 있는 문장 (IndexError)
except NameError:
    print("존재하지 않는 변수 호출")
except IndexError:
    print("인덱스 에러 발생")
except ZeroDivisionError:
    print("0으로 나눌 수 없음!")
else:
    print("예외없음! 성공적!!")
```

예외없음! 성공적!!

예외 처리

예외 타입 지정하지 않는 방법!

```
try:  
    예외가 발생할 수 있는 문장  
except:  
    예외 발생시 실행할 문장
```

모든 종류 예외 보려면

```
try:  
    print(a[10])  
except Exception as err:  
    print(err)
```

람다 함수:

```
def add(a, b)  
    return a + b
```

=>

```
print(((lambda a, b : a + b)(1, 2)))
```

```
try:  
    num = [1, 2, 3, 4]  
    print(who)  
except:  
    print("알 수 없는 예외가 발생했습니다!")
```

알 수 없는 예외가 발생했습니다!