



# 파이썬기초

8주차 - 2교시  
파이썬의 입출력 처리





## 학습내용

- 파이썬의 입출력 처리



## 학습목표

- 파이썬에서 파일 입출력 처리에 대해 이해하고 파일 입출력 코딩을 할 수 있다.



## 생각해 봅시다

프로그램을 기준으로  
파일을 “입력한다”, “출력한다”에 대해  
생각해봅시다.



01



# 파이썬의 입출력 처리



## 1 | 파이썬의 입출력 처리

### 01 파일 열기

#### 파일 열기

```
> 변수 = open("파일명", "파일모드")
```

# 01 파이썬의 입출력 처리

## 1 | 파이썬의 입출력 처리

### 01 파일 열기



#### 변수

프로그램 내부에서 사용할 파일(파일명)의 식별자



#### 파일명

운영체제에서 사용하는 파일명

→ 파일의 경로 포함하는 이름



#### 파일모드

파일 사용 방법

## 1 | 파이썬의 입출력 처리

### 02 파일모드

- 읽기 모드(r, r+, rb)

➤ 입력파일들은 프로그램의 **실행 전**에 미리 준비해야 함

- 쓰기모드('w', 'a' 모드)

➤ 출력파일들은 없으면 **새로 생성**해서 출력함

## 1 | 파이썬의 입출력 처리

### 02 파일모드

- 'w' 모드

➤ 기존 파일의 파일 내용을 **무시하고 새로 작성함**  
→ 즉, 기존 파일내용 삭제

- 'a' 모드

➤ 기존 파일(있는 경우) **끝에 추가**해서 출력함



## 1 | 파이썬의 입출력 처리

### 02 파일모드

파일모드	모드 의미	지정 파일이 없는 경우	지정 파일이 있는 경우
'r'	텍스트 파일 읽기	NULL 반환, 기본값	-
'w'	텍스트 파일 쓰기	신규 파일 생성	기존 파일 무시함
'a'	텍스트 파일 추가	신규 파일 생성	기존 파일 뒤에 추가함
'r+'	파일 갱신(읽고 쓰기)	NULL 반환	-
'rb'	이진 파일 읽기	NULL 반환, 기본값	-
'wb'	이진 파일 쓰기	신규 파일 생성	기존 파일 무시함

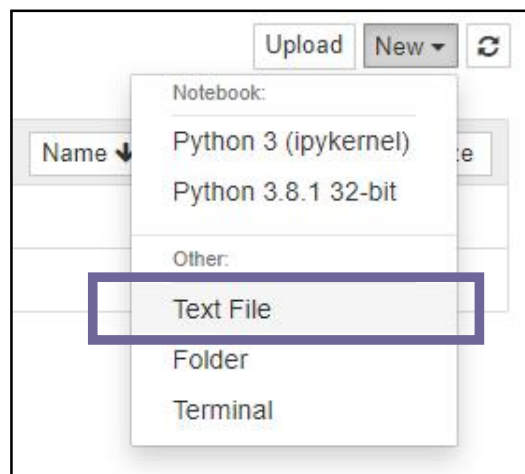
# 01 파이썬의 입출력 처리

## 1 | 파이썬의 입출력 처리

### 03 파일 처리 예제

01 Jupiter Notebook 내부에 새로운 폴더를 생성함

02 새로운 폴더 내부에서 새로운 텍스트파일 data.txt를 생성함

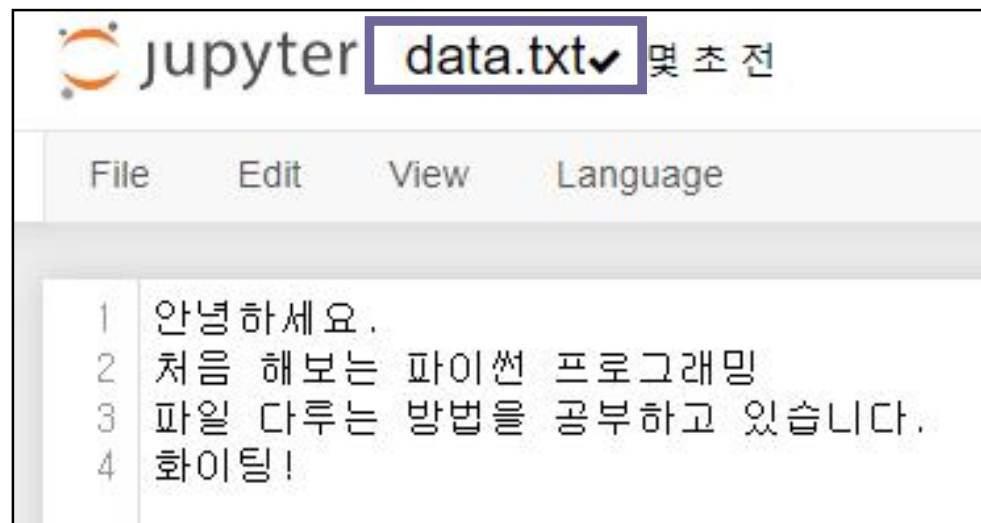


# 01 파이썬의 입출력 처리

## 1 | 파이썬의 입출력 처리

### 03 파일 처리 예제

#### 03 파일 생성 후 이름을 변경하고 내용을 추가함



The image shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "data.txt" in a blue box, with a checkmark and the Korean text "몇 초 전" (a few seconds ago) to its right. Below this is a menu bar with "File", "Edit", "View", and "Language". The main area contains a code cell with four lines of Korean text, numbered 1 to 4 on the left:

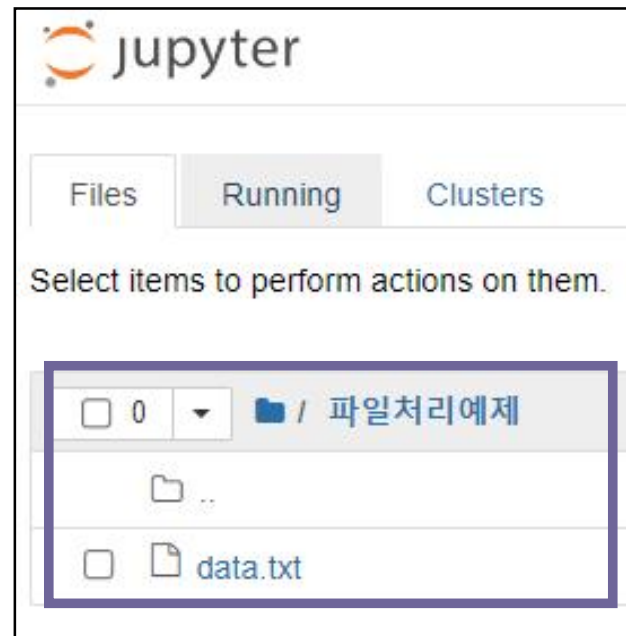
```
1 안녕하세요.  
2 처음 해보는 파이썬 프로그래밍  
3 파일 다루는 방법을 공부하고 있습니다.  
4 화이팅!
```

# 01 파이썬의 입출력 처리

## 1 | 파이썬의 입출력 처리

### 03 파일 처리 예제

#### 04 해당 폴더에 파일이 생성된 것을 확인함

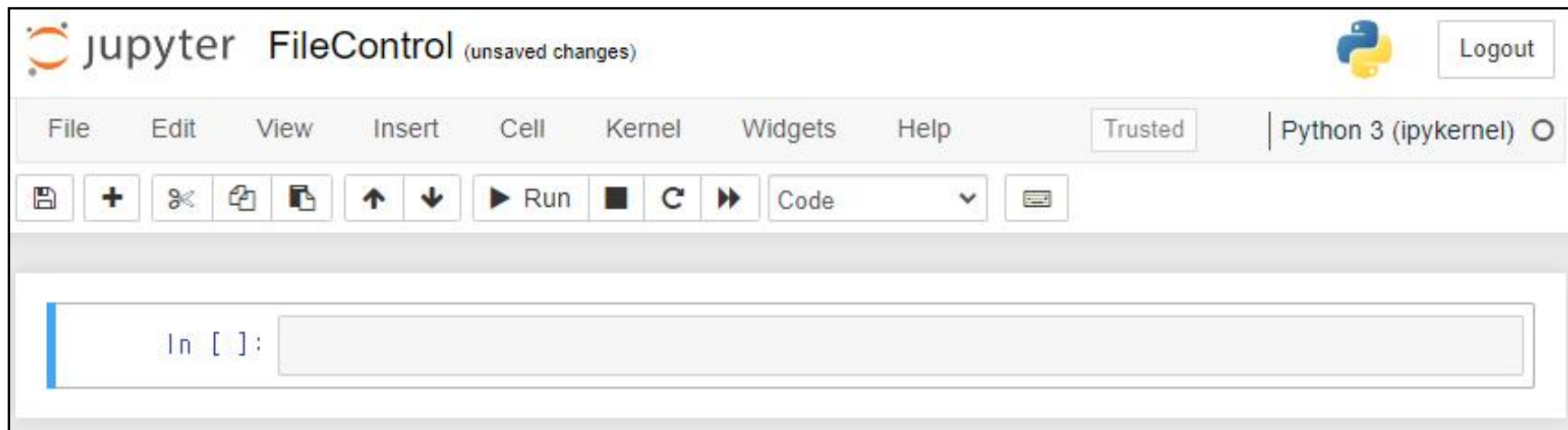


# 01 파이썬의 입출력 처리

## 1 | 파이썬의 입출력 처리

### 03 파일 처리 예제

#### 05 새로운 폴더 내부에서 파이썬 파일을 생성함



## 1 | 파이썬의 입출력 처리

### 04 파일 읽기 예제

```
In [1]: # 파일 열기
file1 = open("data.txt", "r", encoding='utf-8')

print("data.txt 파일 읽고 화면에 출력하기")
print("="*40)

tempStr = file1.read()
# read()는 파일의 내용 전체를 문자열로 반환
```

## 1 | 파이썬의 입출력 처리

### 04 파일 읽기 예제

```
In [1]: print(tempStr)
        print("="*40)

file1.close() # 파일을 다 사용한 후에는 닫아주어야 함
```

## 1 | 파이썬의 입출력 처리

### 04 파일 읽기 예제

#### 결과

**data.txt 파일 읽고 화면에 출력하기**

=====

**안녕하세요.**

**처음 해보는 파이썬 프로그래밍**

**파일 다루는 방법을 공부하고 있습니다.**

**화이팅!**

=====



## 1 | 파이썬의 입출력 처리

### 05 파일 쓰기 예제

```
In [6]: # 파일 쓰기
file2 = open("./write1.txt", "w", encoding="utf-8")

msg = input("메시지를 입력하세요.\n")

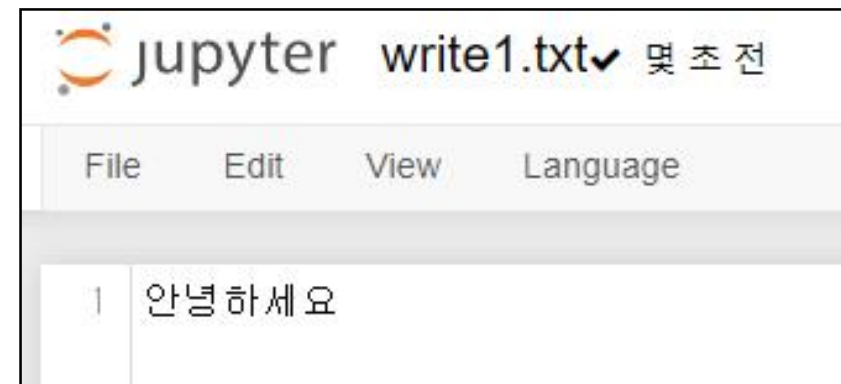
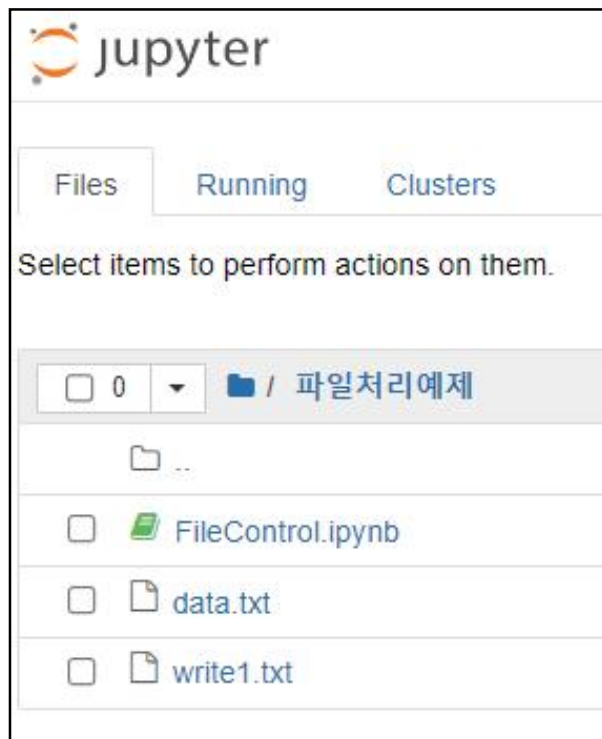
file2.write(msg)
file2.close()
```

**결과**    메시지를 입력하세요.  
안녕하세요

# 01 파이썬의 입출력 처리

## 2 | 결과

### 01 파일 생성 확인



## 2 | 결과

### 02 파일에 텍스트 추가하기 예제

```
In [10]: # 파일에 추가 쓰기
file3 = open("./write1.txt", "a", encoding="utf-8")

msg = input("메시지를 입력하세요.\n")

file3.write("\n")
file3.write(msg)

file3.close()
```

## 2 | 결과

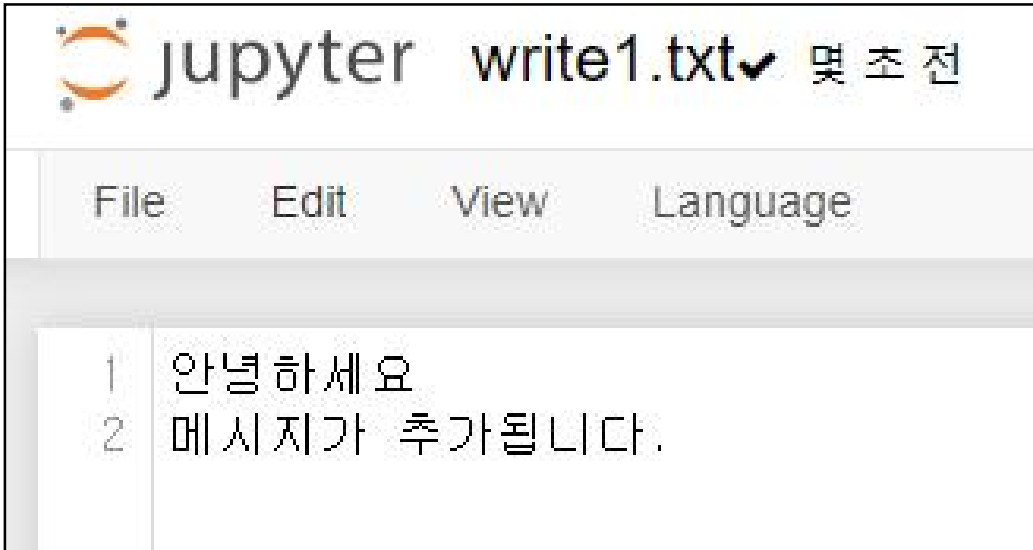
### 02 파일에 텍스트 추가하기 예제

#### 결과

메시지를 입력하세요.  
메시지가 추가됩니다.

## 2 | 결과

### 03 결과 확인



The image shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by the text "jupyter write1.txt" and a checkmark icon. Below this, there is a menu bar with the options "File", "Edit", "View", and "Language". The main area of the notebook displays two lines of text, numbered 1 and 2. Line 1 says "안녕하세요" and line 2 says "메시지가 추가됩니다."

```
jupyter write1.txt ✓ 몇 초 전
```

	File	Edit	View	Language
1	안녕하세요			
2	메시지가 추가됩니다.			

## 2 | 결과

### 04 with 블록을 이용한 파일 처리

- 파일을 사용하고 나면 반드시 **close** 해주어야 함
- 출력파일의 close 생략

➤ 출력 내용이 실제 파일에 반영되지 않음  
→ 반드시 **폐쇄**해야 함

#### with 블록 이용

블록이 끝날 때 자동으로 close() 실행

### 2 | 결과

#### 05 with 블록을 이용한 파일 처리 예제

```
In [15]: with open("./data.txt", "r", encoding="utf-8") as file4 :  
    print("data.txt 파일 읽고 화면에 출력하기")  
    print("="*40)  
  
    tempStr = file4.read()  
  
    # read()는 파일의 내용 전체를 문자열로 반환  
    print(tempStr)  
    print("="*40)
```

## 2 | 결과

### 05 with 블록을 이용한 파일 처리 예제

#### 결과

**data.txt 파일 읽고 화면에 출력하기**

=====

**안녕하세요.**

**처음 해보는 파이썬 프로그래밍**

**파일 다루는 방법을 공부하고 있습니다.**

**화이팅!**

=====





# 파이썬기초

NEXT

파일 입출력 함수

