

파일입출력

파일입출력이란?

- ▶ 휘발성 메모리(RAM)에 저장된 데이터 Data들을 비휘발성 메모리(NVM, NVRAM)인 HDD, SSD에 옮기거나 반대로 HDD, SSD에 있는 Data를 RAM에 옮겨 사용
- ▶ 헤더파일 : <fstream>
- ▶ 클래스 : ofstream, ifstream, fstream

ofstream

- ▶ Output File Stream : 파일 쓰기 전용

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ofstream file_output;
    file_output.open("output.txt");
    if (file_output.is_open())
    {
        //파일쓰기.
        char arr[13] = "Hello World!"; //Hello World!W0
        file_output.write(arr, 12);    //W0(null)문자를 제외하고 쓴다.
    }
    file_output.close();
}
```

ofstream

- ▶ 파일 열기
- ▶ **Open(...)**
 - void open(const char* fileName, ios_base::openmode mode = ios_base::out)
 - void open(const string& fileName, ios_base::openmode mode = ios_base::out)
- ▶ **fileName** : 파일의 이름
- ▶ **Mode** : 파일 모드 상수
 - 매개변수 default 값으로 ios_base::out 사용

```
...  
int main()  
{  
    ofstream file_output("output.txt");  
    ...  
}
```

ofstream

- ▶ 파일 닫기
- ▶ close()
 - void close()

※파일 닫기를 하지 않으면 **메모리 누수**가 발생!!

```
...  
int main()  
{  
    ...  
    file_output.close();  
}
```

ofstream

- ▶ 파일 Open 확인
- ▶ **is_open()**
 - bool is_open() const
- ▶ **fail()**
 - bool fail() const

```
...  
if (!file_output.fail()) //! : 논리 부정 연산자  
{  
    //파일쓰기.  
    ...  
}  
...
```

```
...  
if (file_output)  
{  
    //파일쓰기.  
    ...  
}  
...
```

ofstream

- ▶ 파일 쓰기
- ▶ write(...)
 - ostream& write(const char* str, streamsize count)

```
...  
{  
    char arr[13] = "Hello World!";  
    file_output.write(arr, 12);  
}  
...
```

```
#include <string>  
...  
{  
    string str = "Hello World!";  
    file_output.write(str.c_str(), str.size());  
}  
...
```

```
...  
{  
    file_output << "Hello World!";  
}  
...
```

ifstream

- ▶ Input File Stream : 파일 읽기 전용

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
    ifstream file_input;
    file_input.open("input.txt");
    if (file_input.is_open())
    {
        char c;
        while (file_input.get(c))
        {
            cout << c << endl;
        }
    }
    file_input.close();
}
```


ifstream

- ▶ 파일 열기
- ▶ **Open(...)**
 - void open(const char* fileName, ios_base::openmode mode = ios_base::in)
 - void open(const string& fileName, ios_base::openmode mode = ios_base::in)
- ▶ **fileName** : 파일의 이름
- ▶ **Mode** : 파일 모드 상수
 - 매개변수 default 값으로 ios_base::in 사용

...

```
int main()
{
    ifstream file_input("input.txt");
    ...
}
```

ifstream

- ▶ 파일 닫기
- ▶ close()
 - void close()

※파일 닫기를 하지 않으면 **메모리 누수**가 발생!!

```
...  
int main()  
{  
    ...  
    file_input.close();  
}
```

ifstream

- ▶ 파일 Open 확인
- ▶ **is_open()**
 - bool is_open() const
- ▶ **fail()**
 - bool fail() const

```
...  
    if (!file_input.fail())  
    {  
        ...  
    }  
...
```

```
...  
    if (file_input)  
    {  
        ...  
    }  
...
```

ifstream

- ▶ 파일 읽기

- ▶ **get(...)**

- ifstream& get(char& ch)

- ▶ **getline(...)**

- ifstream& getline ifstream& lstr, string str)

```
...
string str;
while (!file_input.eof())
{
    getline(file_input, str); //한 줄(Wn를 만나기 전)을 읽는다.
    cout << str << endl;
}
...
```

```
...
while (!file_input.eof())
{
    file_input >> str; //공백 문자(' ')가 나올 때까지 읽는다.
    cout << str << endl;
}
...
```

fstream

▶ File Stream : 파일 읽기 쓰기 대응

```
...
int main()
{
    fstream file_stream;
    file_stream.open("file.txt", ios_base::out);
    if (file_stream.is_open())
    {
        file_stream << "Hello World!!" << endl;
        file_stream << "File Stream";
    }
    file_stream.close();

    file_stream.open("file.txt", ios_base::in);
    if (file_stream.is_open())
    {
        string str;
        while (!file_stream.eof())
        {
            getline(file_stream, str);
            cout << str << endl;
        }
    }
    file_stream.close();
}
```

fstream

▶ 파일 모드 상수 모음

파일 모드 상수	설명
<code>ios_base::in</code>	파일을 읽는 것만 가능한 모드로 개방
<code>ios_base::out</code>	파일을 쓰는 것만 가능한 모드로 개방
<code>ios_base::ate</code>	파일을 개방할 때 파일의 끝으로 포인터를 이동
<code>ios_base::app</code>	파일의 맨 끝에서부터 데이터를 추가
<code>ios_base::trunc</code>	파일이 있으면 해당 파일의 모든 데이터를 지우고 개방
<code>ios_base::binary</code>	바이너리 모드로 개방

fstream

▶ 자주 사용되는 파일 모드 상수 조합

파일 모드 상수	설명
<code>ios_base::out ios_base::trunc</code>	파일이 없으면 새 파일을 만들고 , 파일이 있으면 해당 파일의 모든 데이터를 지우고 쓰는 것만 가능한 모드로 개방
<code>ios_base::out ios_base::app</code>	파일이 없으면 새 파일을 만들고 , 파일이 있으면 해당 파일의 맨 끝에서부터 데이터를 추가하는 쓰는 것만 가능한 모드로 개방
<code>ios_base::in ios_base::out</code>	파일을 읽고 쓰는 것이 가능한 모드로 개방 파일이 없을 경우 새로 만들지 않는다
<code>ios_base::in ios_base::out ios_base::trunc</code>	파일이 없으면 새 파일을 만들고 , 파일이 있으면 해당 파일의 모든 데이터를 지우고 읽고 쓰는 것이 가능한 모드로 개방