

7. 파일 입출력 기초

프로그래밍 기초 교육



파일의 종류

텍스트(text) 파일

- 캐릭터(char)들로 데이터가 저장되는 파일

이진(binary) 파일

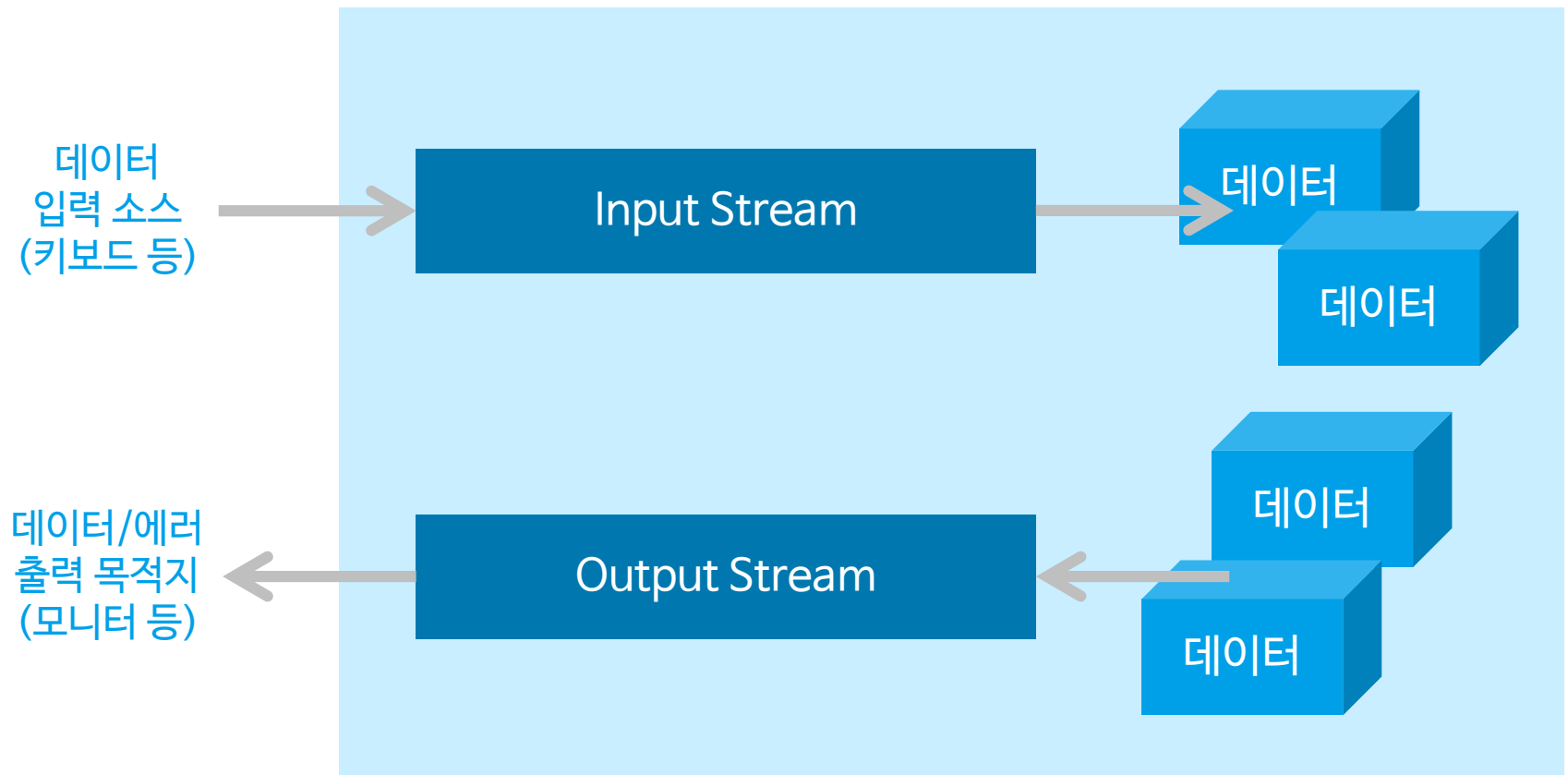
- 01010111.. 이런 이진수(기계어)로 변환(인코딩)되어 저장되는 파일
- 텍스트 파일 역시 이진수로 저장되겠지만, 이진 파일에는 텍스트 외에 다른 형식의 정보가 더 들어 있다.

```
0000000 0000 0001 0001 1010 0010 0001 0004 0128
0000010 0000 0016 0000 0028 0000 0010 0000 0020
0000020 0000 0001 0004 0000 0000 0000 0000 0000
0000030 0000 0000 0000 0010 0000 0000 0000 0204
0000040 0004 8384 0084 c7c8 00c8 4748 0048 e8e9
0000050 00e9 6a69 0069 a8a9 00a9 2828 0028 fdfe
0000060 00fc 1819 0019 9898 0098 d9d8 00d8 5857
0000070 0057 7b7a 007a bab9 00b9 3a3c 003c 8888
0000080 8888 8888 8888 8888 288e be88 8888 8888
0000090 3b83 5788 8888 8888 7667 778e 8828 8888
00000a0 d61f 7abd 8818 8888 467c 585f 8814 8188
00000b0 8b06 e8f7 88aa 8388 8b3b 88f3 88bd e988
00000c0 8a18 880c e841 c988 b328 6871 688e 958b
00000d0 a948 5862 5884 7e81 3788 1ab4 5a84 3eec
00000e0 3d86 dcb8 5cbb 8888 8888 8888 8888 8888
00000f0 8888 8888 8888 8888 8888 8888 8888 0000
0000100 0000 0000 0000 0000 0000 0000 0000 0000
*
0000130 0000 0000 0000 0000 0000 0000 0000
000013e
```

위키백과 아이콘은 오른쪽과 같이 저장된다.
이미지 파일은 이진 파일의 일종이다.

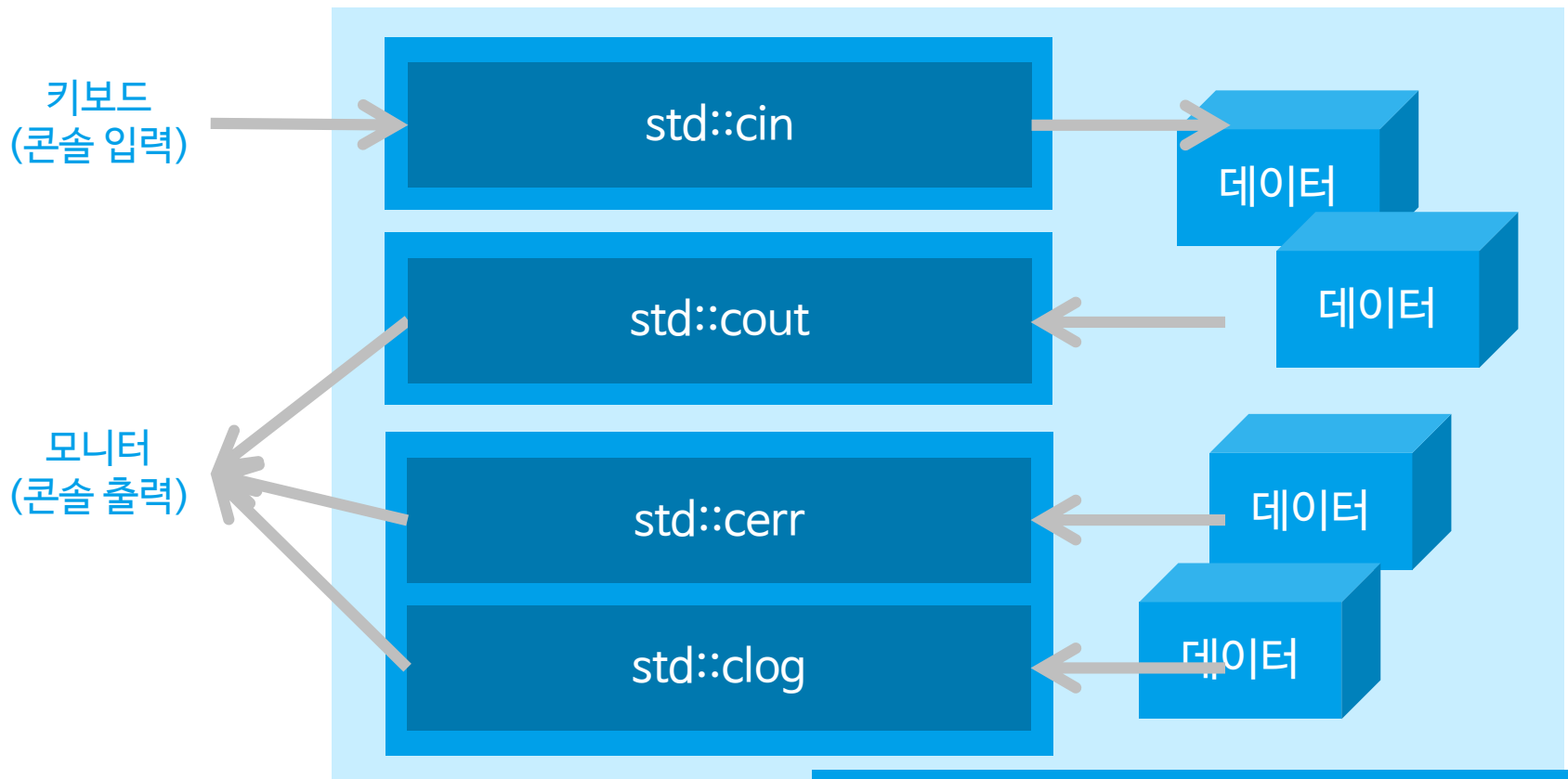
스트림

- 데이터가 오고 갈 때 사용되는 추상적인 통로이다.



표준 스트림

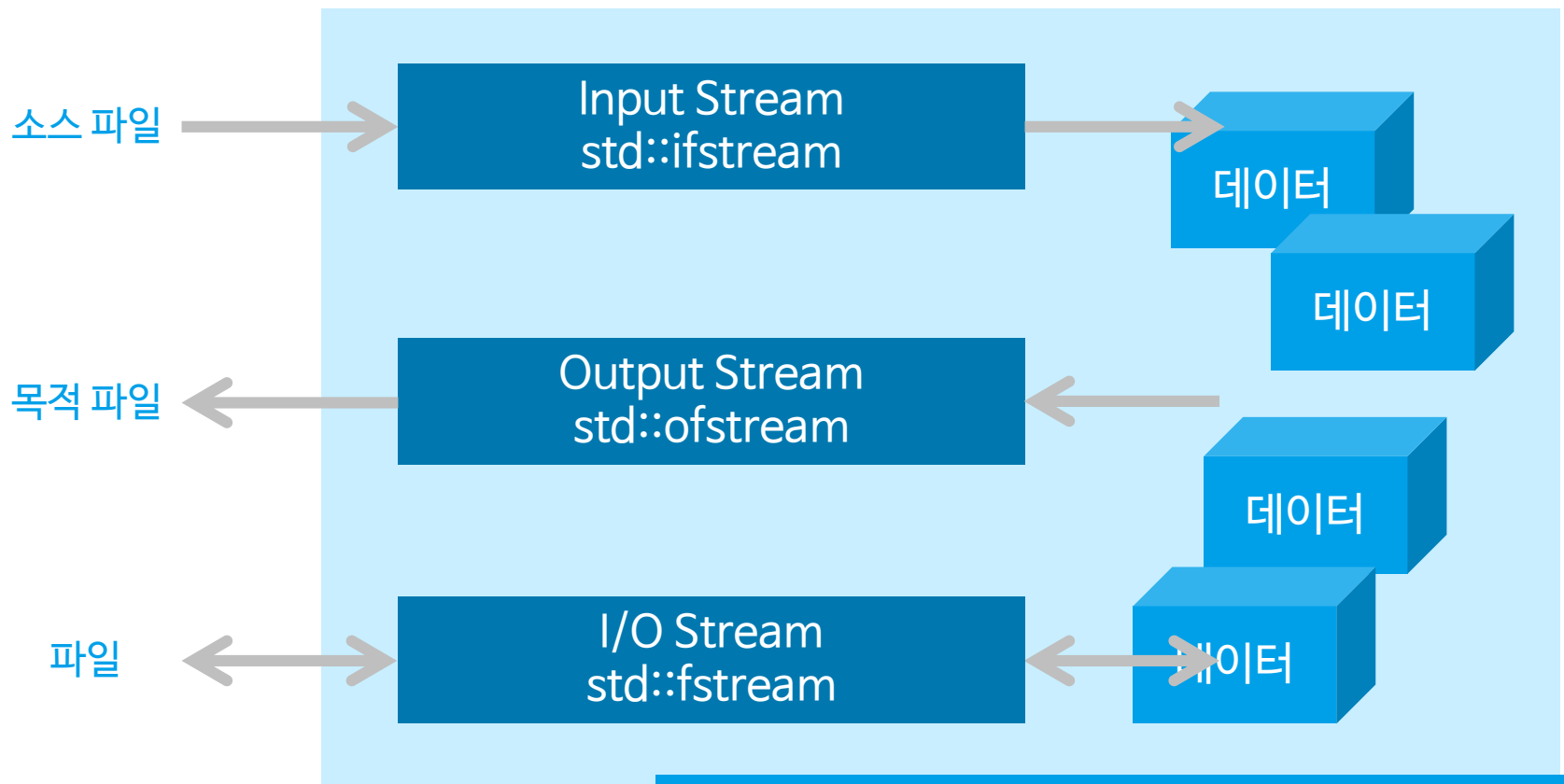
- 표준 스트림에는 표준 입력(stdin), 표준 출력(stdout), 표준 오류(stderr)가 있으며, C++에서는 아래 4가지 형식으로 지원한다.



표준 스트림은 자동으로 만들어지고 연결되고 해제된다.

파일 스트림

- C++의 파일 스트림에는 3종류가 있다.



파일 스트림은 코드 상에서 직접 만들고 연결하고 해제시켜야 한다.

파일 상태

- 파일 상태에는 읽기 상태, 쓰기 상태, 오류 상태가 있다.

읽기 상태
read state

파일에서 읽을 수 있다.

쓰기 상태
write state

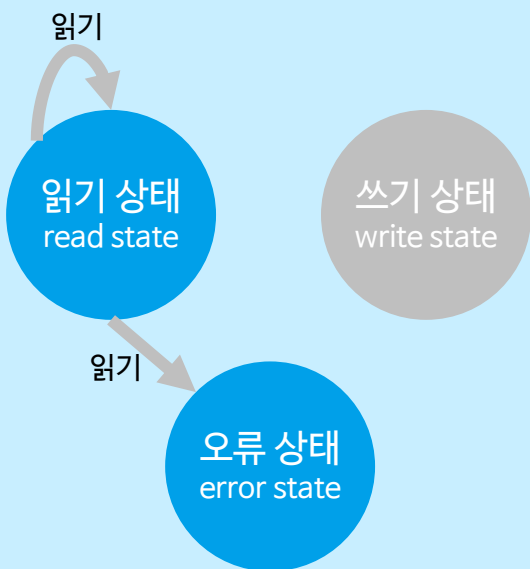
파일에서 쓸 수 있다.

오류 상태
error state

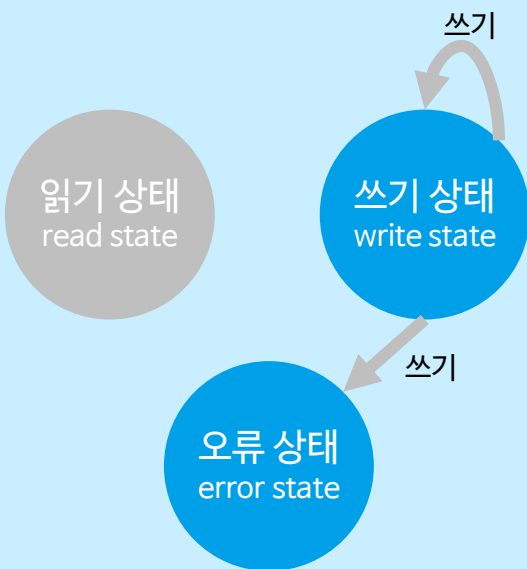
읽거나 쓰다가 오류가 발생해 오류 상태가 되면 읽거나 쓸 수 없다.
clear 해주어 이전의 정상적인 상태로 되돌아간다.

파일 상태

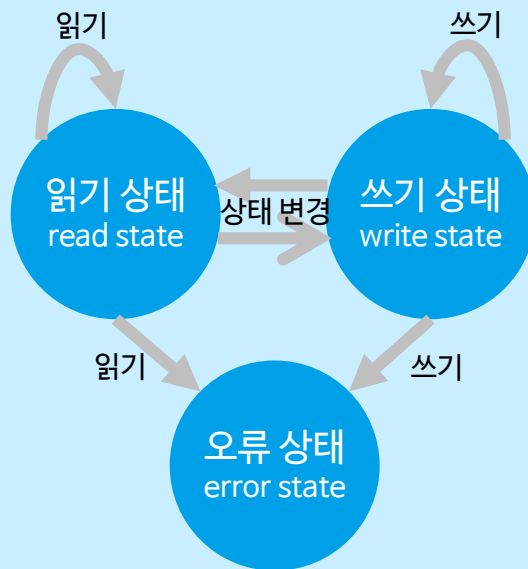
읽기 상태 read state



쓰기 상태 write state



갱신 상태 update (read+write) state



파일 출력

- "file.txt"라는 파일이 만들어지고, 열어보면 "파일 출력"이라고 적혀있다.

Visual Studio(Windows)에는

- 프로젝트 폴더의 소스 코드와 같은 위치에 있다.

Xcode(macOS)에는

1. Product 메뉴를 누른다.
2. Scheme option > Edit Scheme option을 선택한다.
3. Options Tab에서 Working Directory에 체크를 한다.
4. 작업 디렉터리를 선택하고 코드를 실행시키면 그 위치에 파일이 만들어진다.

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

int main() {
    ofstream fs; 파일 출력 스트림 만들기
    fs.open("file.txt"); 파일 출력 스트림 연결

    fs << 405 << endl
        << 406 << '\t'
        << 409;

    fs.close(); 파일 출력 스트림 연결 해제
    return 0;

}
```


파일 마커

- 프로그램이 파일을 사용할 때, 현재 작업 중인 위치를 마커(marker)로 표시한다.

파일 마커



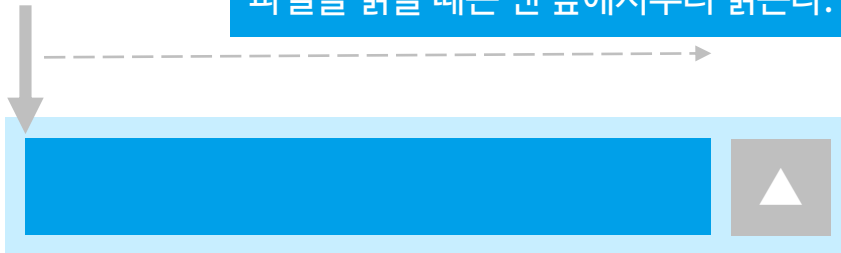
쓰기 모드로 새 파일이 만들어졌고,
내용과 관계 없이 파일 마커는 맨 앞에서부터 시작된다.

파일이 끝나는 부분에는
EOF(end-of-file)라는
문자가 있다.

파일 입력

- 아까 만들어진 "file.txt"을 그대로 두고 코드만 다시 적어보자.
- 공백 문자(띄어쓰기, 탭(\t), 줄바꿈 (\n))를 기준으로 분리된다.

파일을 읽을 때는 맨 앞에서부터 읽는다.



405
406
409

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

int main() {
    ifstream fs; 파일 입력 스트림 만들기
    fs.open("file.txt"); 파일 입력 스트림 연결
    int data1, data2, data3;
    fs >> data1 >> data2 >> data3;
    fs.close(); 파일 입력 스트림 연결 해제

    cout << data1 << endl;
    cout << data2 << endl;
    cout << data3 << endl;
    return 0;
}
```

파일 스트림 오류 처리

- 파일이 열리지 않으면 스트림 객체는 false를 반환한다.
- 오류를 처리할 때는 cout보다 cerr를 사용하는 것이 더 좋다. (둘 다 콘솔에는 출력된다.)

Wa는 알림(alert)이나 알람(alarm)을 나타낸다. 콘솔에는 보이지 않는다.

- exit(int)는 main()에서의 return int와 같은 역할을 한다.

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

int main() {
    ifstream fs;
    fs.open("notexist.txt");
    if (!fs) {
        cerr << "\aERROR 100: "
              << "cannot opening file\n";
        exit(100);
    }
    fs.close();
    if (fs.fail()) {
        cerr << "\aERROR 1002 "
              << "cannot closing file\n";
        exit(102);
    }
    return 0;
}
```

파일 스트림 사용해보기

한 줄에 임의의 수 10개 씩 10줄, 총 100개의 임의의 수를 파일로 출력하시오.

파일 출력 예시

```
209    20917  19754  31641  3158    19122  12243  2563    24623  27433
11950  20466   7454   1375   30014   9198   29793  10033   13624   7915
20716  11854  14822  29210  9719    27811  19145  20229   1720    28951
8578   5595   27506  18553  30451   15880  18350  19126   27553   11529
21960  18512  20316  28543  15278   11275  3107   2178    27877   25834
14402  19255  28109  11353  5053    29349  23589  31916   29937   27588
24785  19400  26745  1495   28379  24096  2127   31996   29625   17892
20019  4866   23758  25204  25937   10874  5683   29723   16158   6161
26294  6138   28547  7709   19747  16450  5181   18037   17091   31911
19332  3718   1827   10001  32234  6563   22806  18580   32647   23965
```

위에서 만든 파일을 사용하여, 각 줄의 평균을 콘솔에 출력하시오.

결과 예시

```
0번째 줄의 평균: 16166.3
1번째 줄의 평균: 14182.2
2번째 줄의 평균: 18417.7
3번째 줄의 평균: 18312.1
4번째 줄의 평균: 17488
5번째 줄의 평균: 22055.1
6번째 줄의 평균: 20654
7번째 줄의 평균: 16838.3
8번째 줄의 평균: 17710.5
9번째 줄의 평균: 17167.3
```

출력 스트림 +

stream.width(int)

- 바로 뒤의 출력물에 대해 칸을 만들어준다. 기본적으로 오른쪽 정렬이다.
- setw(int)와 같다.

stream.fill(char)

- 만들어진 칸이 비어있을 때 채울 문자를 정한다.
- setfill(char)과 같다.

stream.precision(int)

- 실수를 출력할 때 소수점 이하 몇 째 자리까지 출력할지 정한다.
- setprecision(int)와 같다.

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

```
int main() {
    fstream fs;
    fs.open("file.txt", fstream::out);
```

```
    fs.width(10);
    fs << "Hello" << endl;
    fs.fill('0'); fs.width(10);
    fs << 12345 << endl;
    fs.precision(8);
    fs << (1.0 / 7.0) << endl;
```

```
    return 0;
```

```
}
```

fstream은 입출력 모두 된다.
대신 ofstream처럼 없는
파일을 만들어주지는 않기
때문에 fstream::out을
붙여준다.

파일 스트림 뿐만 아니라 cout과
같은 스트림에서도 사용할 수 있다.

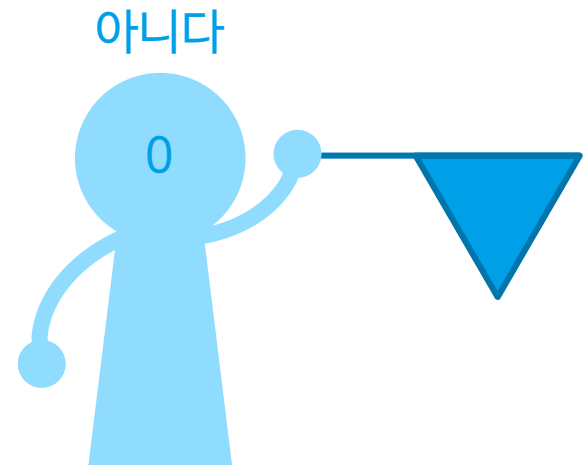
```
    Hello
    0000012345
    0.14285714
```

플래그

- 특정 상태를 나타내는 1비트 변수를 '플래그(flag)'라고 부른다.



```
stream.setf(std::ios::[플래그이름]);
```



```
stream.unsetf(std::ios::[플래그이름]);
```

플래그: skipws

- 띄어쓰기, 탭과 같은 공백 (whitespace) 문자를 무시한다.
- 기본적으로 set되어 있다.

(스페이스)의 아스키 값: 32
z의 아스키 값: 122
(줄바꿈)의 아스키 값: 10

[스페이스][문자]를 입력하세요: z
받은 문자: 122

cin의 플래그: skipws
받은 문자: 10

버퍼(데이터가 전달되어 사용 되기 전에 임시 저장되는 위치)에 남아 있던 줄바꿈 문자가 무시 되지 않고 들어왔다.

[스페이스][문자]를 입력하세요: z
받은 문자: 32
받은 문자: 122
받은 문자: 10

매번 파일 열어보기 번거로우므로
표준 출력 스트림으로 실습해보자.

```
char cInput;  
cout << "[스페이스][문자]를 입력하세요: ";  
cin >> cInput;  
cout << "받은 문자: " << (int)cInput << endl;  
cout << endl;
```

```
cin.unsetf(ios::skipws);  
cout << "cin의 플래그: skipws를 해제"  
<< endl;  
cin >> cInput;  
cout << "받은 문자: " << (int)cInput << endl;  
cout << endl;
```

```
cout << "[스페이스][문자]를 입력하세요: ";  
cin >> cInput;  
cout << "받은 문자: " << (int)cInput << endl;  
cin >> cInput;  
cout << "받은 문자: " << (int)cInput << endl;  
cin >> cInput;  
cout << "받은 문자: " << (int)cInput << endl;
```

플래그: adjustfield, left, right

- 너비가 지정된 칸에서의 정렬을 정한다.
- `ios::left`와 `ios::right`는 `ios::adjustfield`와 함께 사용한다.

```
cout << "기본 정렬 : |";  
cout.width(10);  
cout << "00000" << "|" << endl;
```

```
cout.setf(ios::left, ios::adjustfield);  
cout << "왼쪽 정렬 : |";  
cout.width(10);  
cout << "00000" << "|" << endl;
```

```
cout.setf(ios::right, ios::adjustfield);  
cout << "오른쪽 정렬: |";  
cout.width(10);  
cout << "00000" << "|" << endl;
```

```
기본 정렬 : |      00000|  
왼쪽 정렬 : |00000  |  
오른쪽 정렬: |      00000|
```


플래그: scientific, fixed

- `ios::scientific`은 숫자를 과학적 표기법(1자리 소수와 10의 멍수 (몇 제곱인지)로 표시)으로 나타낸다.
- `ios::fixed`는 소수점으로 나타낸다. `precision(int)`과 함께 쓰이며 기본 값은 6이다.

```
cout.setf(ios::scientific);  
cout << "과학적 표기법      : "  
      << 314.1592 << endl;  
cout.unsetf(ios::scientific);  
  
cout.setf(ios::fixed);  
cout << "고정소수점 표기법: "  
      << 314.1592 << endl;
```

```
과학적 표기법      : 3.141592e+02  
고정소수점 표기법: 314.159200
```

파일 입출력: 텍스트 파일 만들기

- 텍스트를 자유롭게 입력하고
ctrl+c를 눌러 끝내면 파일에 적은
텍스트가 그대로 저장되어 있다.

get(char)과 put(char)은 스트림(본인)을
반환하기 때문에 키보드에서 입력 받을 수
있는 한 계속해서 while문이 돈다.

샘플 텍스트

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariat. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.

```
ofstream fs;  
cout << "파일 만들기 시작" << endl;  
  
fs.open("text.txt");  
if (!fs) {  
    cerr << "\aERROR 100: "  
        << "cannot opening file\n";  
    exit(100);  
}
```

스트림의 get(char)은
공백 문자까지 다 받는다.

```
char cInput;  
while (cin.get(cInput))  
    fs.put(cInput);  
  
fs.close();  
cout << "파일 만들기 끝" << endl;
```

파일 입출력: 텍스트 파일 복사하기

- 입력 스트림에서 내용을 받아
그대로 출력 스트림으로 보낸다.

```
ifstream ifs;
ofstream ofs;
cout << "파일 복사 시작" << endl;

ifs.open("text.txt");
if (!ifs) {
    cerr << "\aERROR 100: "
         << "cannot opening file\n";
    exit(100);
}

ofs.open("copy.txt");
if (!ofs) {
    cerr << "\aERROR 102: "
         << "cannot opening file\n";
    exit(100);
}
```

```
char cInput;
```

받을 수 있는 게 끝나면 while문이 끝난다.

```
while (ifs.get(cInput))
    ofs.put(cInput);

ifs.close();
ofs.close();
cout << "파일 복사 끝" << endl;
```

파일 입출력: 문자와 줄 수 세기

```
ifstream fs;
fs.open("text.txt");
if (!fs) {
    cerr << "\Aerror 100: "
         << "cannot opening file\n";
    exit(100);
}
```

```
char current;
char pre;
int countChar = 0;
int countLine = 0;
```

```
while (fs.get(current)) {
    if (current != '\n')
        countChar++;
    else
        countLine++;

    pre = current;
}
```

현재 보고 있는 문자를 작전에 봤던 문자로 만든다.

```
if (pre != '\n')
    countLine++;
```

마지막 문자가 줄바꿈 문자면 마지막 줄에는 아무 내용도 없으므로 줄을 세지 않지만, 줄바꿈 문자가 아니라면 while문에서 마지막 줄을 세지 않았을 것이기 때문에 줄 수를 센다.

```
cout << "문자 수: " << countChar << endl;
cout << "줄 수 : " << countLine << endl;
```

```
fs.close();
```

줄바꿈 문자가 아니면 문자 수를 센다.

줄바꿈 문자면 줄이 바뀐 것이므로 줄 수를 센다.

파일 입출력 사용해보기

텍스트 파일에 포함된 단어의 개수를 세는 프로그램을 작성하시오.
단어는 공백문자(띄어쓰기, 탭, 줄바꿈)으로 나누어져 있다.

두 개의 텍스트 파일의 내용을 하나로 합친 파일을 저장하는 프로그램을 작성하시오.

샘플 텍스트

On its 60th anniversary in 2009, Kyung Hee University prepared for a second leap forward while seeking a new paradigm of future university development. It reflected on the fundamental mission of the University and established a new vision and goals for becoming a true university of the future.

샘플 텍스트

In this new vision for a university that transcends the traditional notion of the university as a mere physical location and the limitations of an exclusive academe, Kyung Hee University will rise to international prestige and develop into an international center of research, education, and praxis that brings together reality and imagination, the basics and applied sciences, and theory and practice.

파일 입출력 사용해보기

텍스트 파일의 내용을 두 칸 들여쓰기 하는 프로그램을 작성하시오.

샘플 텍스트

On its 60th anniversary in 2009, Kyung Hee University prepared for a second leap forward while seeking a new paradigm of future university development. It reflected on the fundamental mission of the University and established a new vision and goals for becoming a true university of the future.

결과 예시

On its 60th anniversary in 2009, Kyung Hee University prepared for a second leap forward while seeking a new paradigm of future university development. It reflected on the fundamental mission of the University and established a new vision and goals for becoming a true university of the future.

텍스트 파일의 내용의 너비가 고정되도록 정렬하는 프로그램을 작성하시오.

샘플 텍스트

In this new vision for a university that transcends the traditional notion of the university as a mere physical location and the limitations of an exclusive academe, Kyung Hee University will rise to international prestige and develop into an international center of research, education, and praxis that brings together reality and imagination, the basics and applied sciences, and theory and practice.

결과 예시

In this new vision for a university that transcends the traditional notion of the university as a mere physical location and the limitations of an exclusive academe, Kyung Hee University will rise to international prestige and develop into an international center of research, education, and praxis that brings together reality and imagination, the basics and applied sciences, and theory and practice.

파일 입출력 사용해보기

학생들의 성적표를 출력하는 프로그램을 작성하시오. 점수는 0~100점이다.
학생 마다 5개 과목의 점수가 있으며, 평균을 구한 다음 각 학생의 평균과 학점, 과목의 평균을 출력하면 된다.
점수는 텍스트 파일에 저장되어 있으며, 파일이 없는 경우 새로 입력 받는다.
파일에서 각 점수는 띄어쓰기 또는 탭으로, 학생은 줄바꿈으로 구분한다.
성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 50점 이상이면 D 그리고 그 미만이면 F다.

결과 예시

점수 데이터가 없습니다.

> 몇 명의 학생입니까? 5

학생마다 5개의 점수를 입력해주세요.

> 학생 1의 점수: 20 30 40 50 60

> 학생 2의 점수: 30 40 50 60 70

> 학생 3의 점수: 40 50 60 70 80

> 학생 4의 점수: 50 60 70 80 90

> 학생 5의 점수: 60 70 80 90 100

데이터 파일이 만들어졌습니다.

	과목 1	과목 2	과목 3	과목 4	과목 5	평 균
학생 1	20 F	30 F	40 F	50 D	60 D	40 F
학생 2	30 F	40 F	50 D	60 D	70 C	50 D
학생 3	40 F	50 D	60 D	70 C	80 B	60 D
학생 4	50 D	60 D	70 C	80 B	90 A	70 C
학생 5	60 D	70 C	80 B	90 A	100 A	80 B
평 균	40	50	60	70	80	60

파일 출력 예시

20 30 40 50 60
30 40 50 60 70
40 50 60 70 80
50 60 70 80 90
60 70 80 90 100

2017.05.02. 프로그래밍 기초 (2017-1)
with D.com

1010
01