

## 🔨 핵심 정리

- stream iterator

### 1. 스트림 반복자 ( stream iterator )

- 입/출력 스트림에서 요소를 읽거나 쓰기 위한 반복자.
- `#include <iterator>`
- 4가지 종류의 반복자 제공

스트림 반복자	출력 대상	출력 형태
<code>ostream_iterator</code>	<code>basic_ostream</code>	서식화된 출력
<code>ostreambuf_iterator</code>	<code>basic_ostreambuf</code>	CharT 출력
<code>istream_iterator</code>	<code>basic_istream</code>	서식화된 입력
<code>istreambuf_iterator</code>	<code>basic_istreambuf</code>	CharT 입력

### 2. `ostream_iterator`

- 출력 스트림(`ostream`, `ofstream`, `ostringstream`)을 사용해서 출력을 하는 반복자
- `copy` 등의 알고리즘 함수를 사용해서 스트림에 출력 할 때 사용.

## 핵심 정리

- stream iterator

### 1. ostream\_iterator 생성

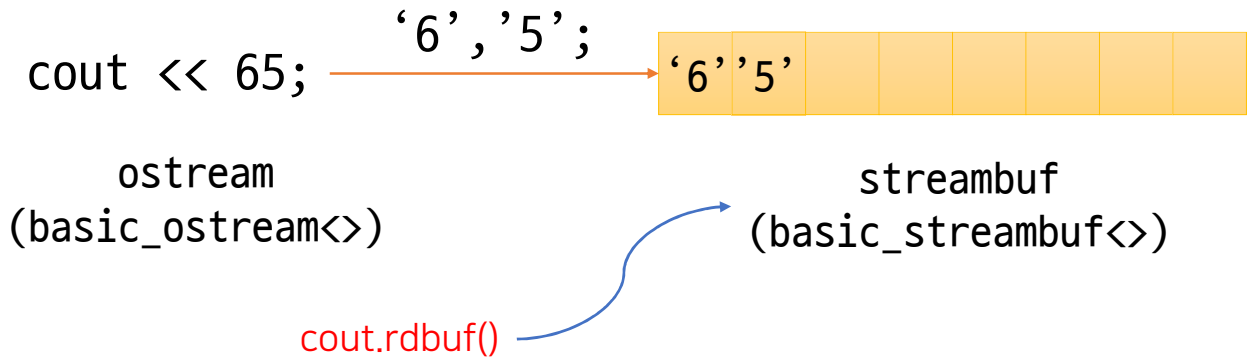
- `ostream_iterator<ostream_type& stream, const CharT* delim>`
- `ostream_iterator<ostream_type& stream>`

### 2. ostream\_iterator 연산

연산	연산의 결과
++	no-op
*	no-op
=	write to stream

## 🔨 핵심 정리

- stream iterator



- `rdbuf()`

stream 객체가 사용하는 streambuf 의 포인터를 반환 하는 stream의 멤버 함수

- `basic_streambuf<>::sputc()`

streambuf 에 한 문자를 출력하는 함수.

- `ostream_iterator`

stream(cout)을 사용해서 화면 출력 - 서식화된 출력

- `ostreambuf_iterator`

streambuf 를 사용해서 화면 출력 - 문자 타입만 출력

## 핵심 정리

- stream iterator

- ostreambuf\_iterator

streambuf 를 사용해서 화면 출력 - 문자 타입만 출력

```
ostreambuf_iterator(streambuf_type* buffer);
```

```
ostreambuf_iterator(ostream_type& stream);
```

## 🔨 핵심 정리

- stream iterator

### 1. 스트림 반복자 ( stream iterator )

- 입/출력 스트림에서 요소를 읽거나 쓰기 위한 반복자.
- `#include <iterator>`
- 4가지 종류의 반복자 제공

스트림 반복자	출력 대상	출력 형태
<code>ostream_iterator</code>	<code>basic_ostream</code>	서식화된 출력
<code>ostreambuf_iterator</code>	<code>basic_ostreambuf</code>	CharT 출력
<code>istream_iterator</code>	<code>basic_istream</code>	서식화된 입력
<code>istreambuf_iterator</code>	<code>basic_istreambuf</code>	CharT 입력

### 2. `istream_iterator` vs `istreambuf_iterator`

- `istream_iterator` 는 white space(space, tab등..)을 입력 받지 못하지만 `istreambuf_iterator` 를 사용하면 white space 입력 받을 수 있다.

## 핵심 정리

- stream iterator

### 1. 입력 받을 때

- cin : 버퍼에 입력된 white space( space, tab, enter 등)을 제외하고 입력 받는다.
- streambuf 에서 직접 꺼내 오면 white space 를 입력 받을 수 있다.

## 🔨 핵심 정리

- stream iterator

### 1. istream\_iterator

- 디폴트 생성자는 **end of stream** 을 나타낸다.

### 2. istream\_iterator vs istreambuf\_iterator

- **istream\_iterator** 는 **white space(space, tab등..)**을 입력 받지 못하지만 **istreambuf\_iterator** 를 사용하면 white space 입력 받을 수 있다.