

스택 Stack

스택(Stack)의 이해

- 한쪽은 막히고 한쪽은 뚫려있는 통에 비유
- 한쪽만 뚫려있기 때문에 '먼저 들어간 것이 나중에 나온다'는 특성을 지님
- 'LIFO(Last-In, First-Out)구조의 자료구조'라고 함

예) 프링글스 통



스택의 ADT (Abstract Data Type)

- ADT는 추상 자료형으로 기능을 뜻함
- push : 아이টে를 넣는다.
- pop : 아이টে를 꺼낸다.
- peek : 제일 위에 뭐가 있는지 들여다 본다.
- empty : 통이 비었는지 확인한다.

cf) C++에서 peek는 top으로 가능

Stack 선언

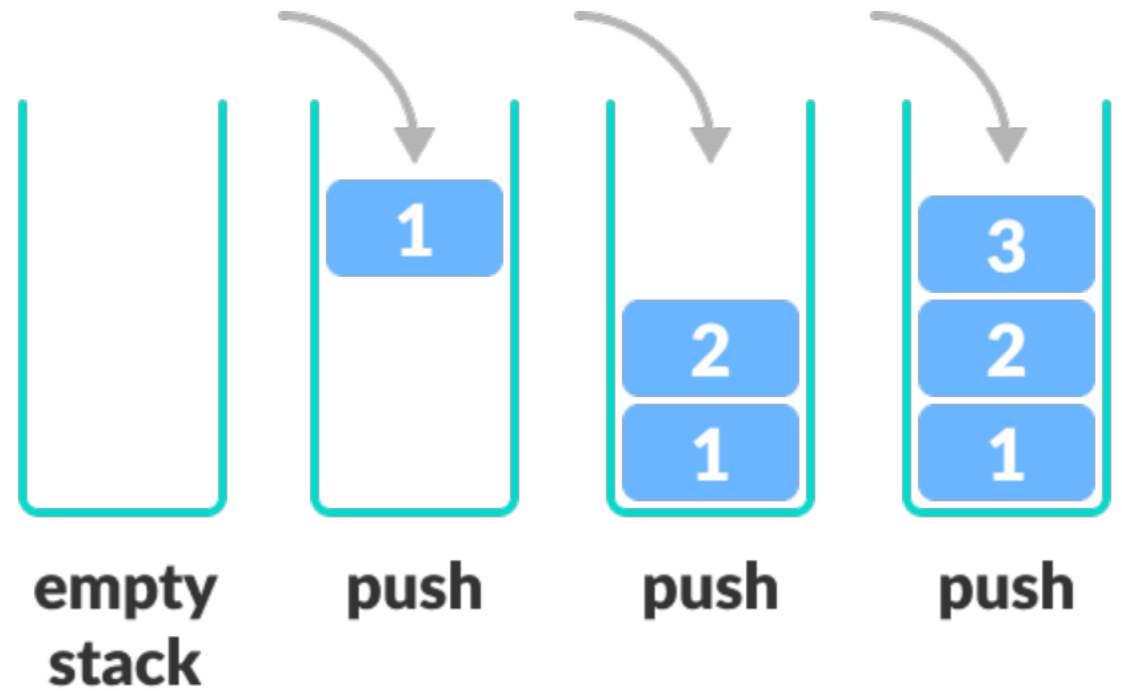
```
#include <iostream>
#include <stack> // stack 헤더파일
using namespace std;

int main() {
    stack<int> int_s;    // int형 스택 선언
    stack<char> char_s; // char형 스택 선언
    return 0;
}
```

Stack 값 추가

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

int main() {
    stack<int> s; // int형 스택 선언
    s.push(1);    // stack에 값 1 추가
    s.push(2);    // stack에 값 2 추가
    s.push(3);    // stack에 값 3 추가
    return 0;
}
```

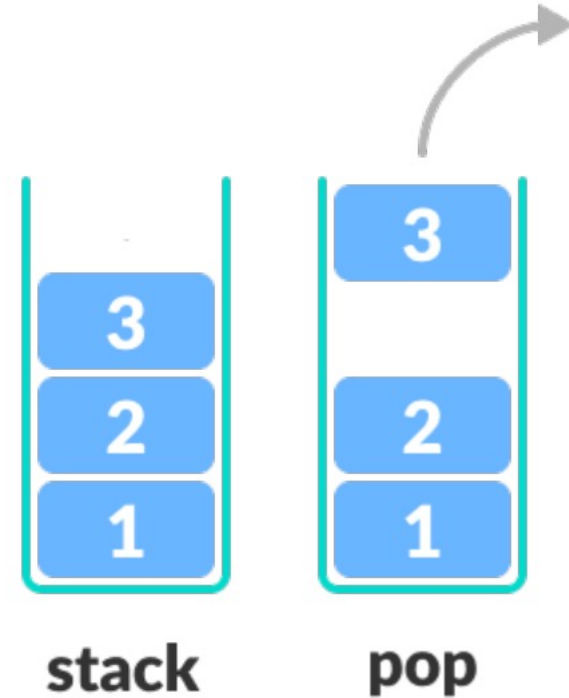


오른쪽 그림과 같이 추가됨

Stack 값 삭제

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

int main() {
    stack<int> s; // int형 스택 선언
    s.push(1);    // stack에 값 1 추가
    s.push(2);    // stack에 값 2 추가
    s.push(3);    // stack에 값 3 추가
    s.pop();      // stack에 값 제거
    return 0;
}
```

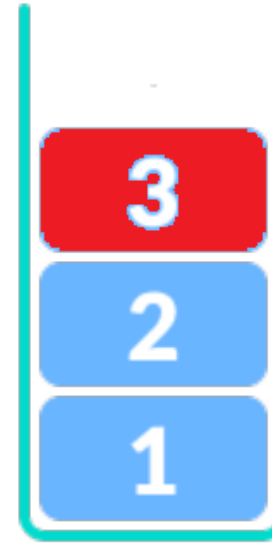


오른쪽 그림과 같이 빠져나감

Stack의 가장 상단의 값 출력

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

int main() {
    stack<int> s;           // int형 스택 선언
    s.push(1);             // stack에 값 1 추가
    s.push(2);             // stack에 값 2 추가
    s.push(3);             // stack에 값 3 추가
    s.top();               // stack의 가장 상단 값 반환
    cout << s.top() << endl; // 3
    return 0;
}
```



stack

3

오른쪽 그림과 같이 빠져나감

Stack 기타 기능

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

int main() {
    stack<int> s1;           // int형 스택 선언
    stack<int> s2;           // int형 스택 선언
    s1.push(1);              // stack s1에 값 1 추가
    s1.push(2);              // stack s1에 값 2 추가
    cout << s1.size() << endl; // stack s1의 크기를 반환
    // console>> 2
    cout << s1.empty() << endl; // stack s1이 비어있는지 bool 값을 반환
    // console>> 0
    s1.swap(s2);             // stack s1과 s2의 값 교환
    return 0;
}
```


Stack 여담

- 배열이나 벡터와 달리 index를 사용할 수 없음
 - └ 값에 접근할 수 없음(오직 가장 위에 있는 값만 가능)
- C++에서 벡터를 stack 대신 사용 가능
 - └ `stack(push, pop, top) == vector(push_back, pop_back, back, etc)`
 - └ 벡터가 상위호환 느낌

Stack 연습문제

예제1) 스택 [실4] – 스택 사용법 연습

<https://www.acmicpc.net/problem/10828>

예제2) 괄호 [실4] – 스택을 이용한 전형적인 문제

<https://www.acmicpc.net/problem/9012>

예제3) 후위 표기식2 [실3] – 스택을 이용한 전형적인 문제 5번과 비슷

<https://www.acmicpc.net/problem/1935>

예제4) 도키도키 간식드리미 [실3] – 조금 생각해보면 가능

<https://www.acmicpc.net/problem/12789>

예제5) 천재 수학자 성필 [실3] – 3번과 비슷

<https://www.acmicpc.net/problem/15815>

예제6) 오큰수 [골4] – 고난이도

<https://www.acmicpc.net/problem/17298>