

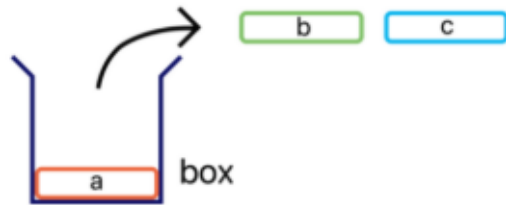
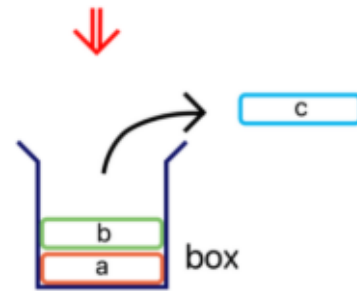
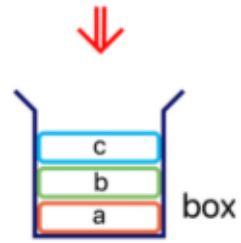
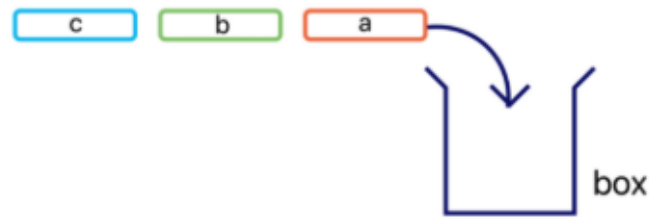
스택 (Stack)

25.06.21

김현지

스택이란?

- 스택(Stack)이란 한쪽 끝에서만 데이터를 넣고 뺄 수 있는 제한적으로 접근할 수 있는 후입선출(LIFO, Last In First Out) 형태의 선형 자료구조
ex) 브라우저의 뒤로 가기, 실행 취소, 재귀 함수 등
- top(peek) : 가장 최근에 저장한 데이터이자 먼저 삭제될 데이터
- push : 데이터 삽입
- pop : 데이터 삭제



스택의 연산

- pop() : 스택에서 가장 위에 있는 항목을 제거
- push(element) : element를 스택의 가장 맨 위에 추가
- peek() : 스택의 가장 위에 있는 항목을 반환
- isEmpty() : 스택이 비어있을 때 true를 반환
- size() : 현재 스택에 들어있는 요소의 개수 반환

스택 구현

```
class Stack {
  constructor() { //생성자 함수
    this.element = [];
  }

  push(element) {
    this.element.push(element);
    console.log('push : ' + element);
  }

  pop() {
    if (this.isEmpty()) {
      console.log('Empty!');
      return;
    }
    console.log('pop : ' + this.element.pop());
  }

  isEmpty() {
    return this.element.length === 0;
  }

  peek() {
    console.log('top element : ' + this.element[this.element.length - 1]);
  }
}
```

배열로 스택 구현

1. constructor() 함수로 element를 받을 배열 생성
2. push(element) 연산으로 element 추가
3. pop() 연산으로 element 삭제
4. isEmpty() 연산으로 스택이 비어있는지 확인
5. peek() 연산으로 스택의 가장 위에 있는 항목 반환

스택이 비어있어서 삭제할 요소가 없으면 Empty 출력
배열의 인덱스는 1이 아닌 0부터 시작이라 length에서 -1
this.element.length는 값이 아닌 인덱스라 this.element[]로 묶음

스택 구현

```
clear() {  
  this.element = [];  
}  
  
print() {  
  console.log('stack : ' + this.element.join(', '));  
}  
}  
  
const stack = new Stack();  
stack.push(1);  
stack.push(2);  
stack.push(3);  
stack.print();  
stack.pop();  
stack.peek();  
stack.pop();  
stack.peek();  
stack.pop();  
stack.peek();  
stack.pop();  
stack.print();
```

- 6. clear() 연산으로 배열 기화
- 7. print() 연산으로 console에 실행 결과 출력

실행결과

```
push : 1  
push : 2  
push : 3  
stack : 1, 2, 3  
pop : 3  
top element :2  
pop : 2  
top element :1  
pop : 1  
top element :undefined  
Empty!  
stack :
```

Big O Notation

- 요소 삽입, 삭제 시 $O(1)$
 - 자료구조의 크기와 상관없이 항상 같은 속도로 연산
- 특정 요소 탐색 시 $O(n)$
 - 특정 요소 탐색할 때 모든 요소를 한 번씩 찾게 됨 (반목문 사용)
 - 스택의 크기가 커질수록 동작이 많아짐

느낀 점

- 자바스크립트로 자료구조를 구현하는 건 처음이라 어색했지만 여러 자료를 찾아보며 스택에 대해 더 자세히 알 수 있었다.
- 다른 자료구조들도 스스로 구현해야겠다고 생각하였다.