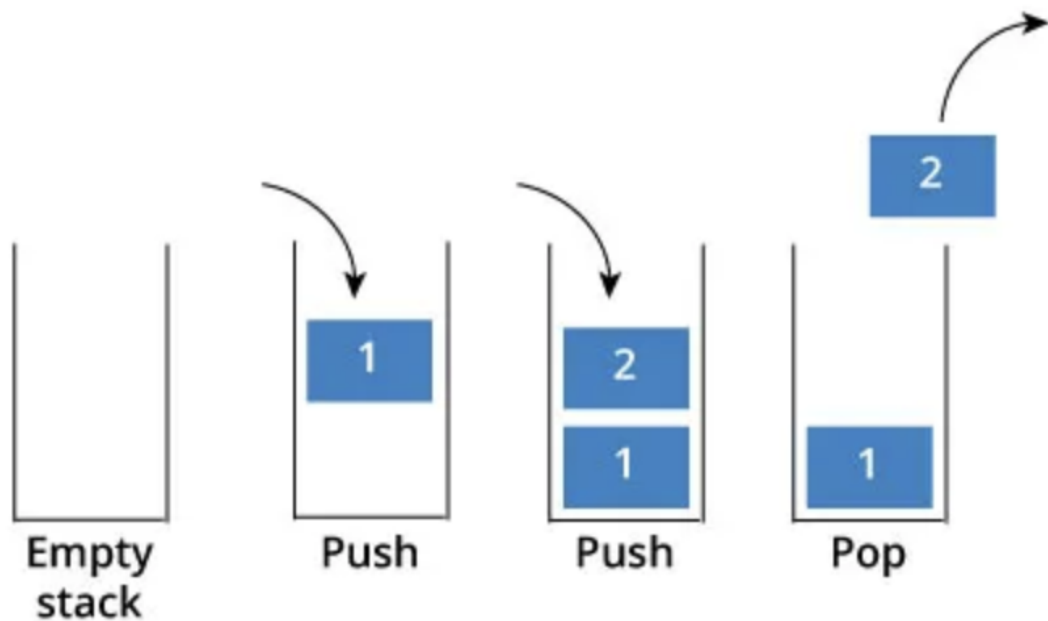


# 스택

🕒 작성일시	@June 9, 2023 8:31 PM
📎 자료	
# 주차	3

배열은 원하는 위치에 원소를 넣고 뺄 수 있다. 하지만 단지 원소를 추가/ 삭제하는 것 이상의 제어를 할 수 있는 형태의 자료구조가 필요한 경우가 있는데, 바로 배열과 유사한 자료구조인 스택과 큐가 있다.

## 스택이란?



LIFO(Last In First Out, 후입선출)원리에 따라 정렬된 컬렉션이다.

- 스택의 자료는 항상 동일한 종단점에서 추가/ 삭제
- 가장 최근 자료는 꼭대기(top)근처에, 가장 오래된 자료는 바닥(base) 근처에 위치한다.

## 스택 구현

스택 구현하는 데 필요한 메소드

- **push(원소(들))**: 스택 **꼭대기**에 새 원소(들)를 추가  
✓ 스택의 꼭대기에만 원소를 넣을 수 있다
- **pop()**: 스택의 꼭대기에 있는 원소를 반환하고 해당 원소는 스택에서 삭제
- **peek()**: 스택 꼭대기에 있는 원소를 반환하되 스택은 변경하지 ㅇ낳는다
- **isEmpty()**: 스택에 원소가 하나도 없으면 true, 스택의 크기가 0보다 크면 false 반환
- **clear()**: 스택의 모든 원소 삭제
- **size()**: 스택의 원소 개수를 반환

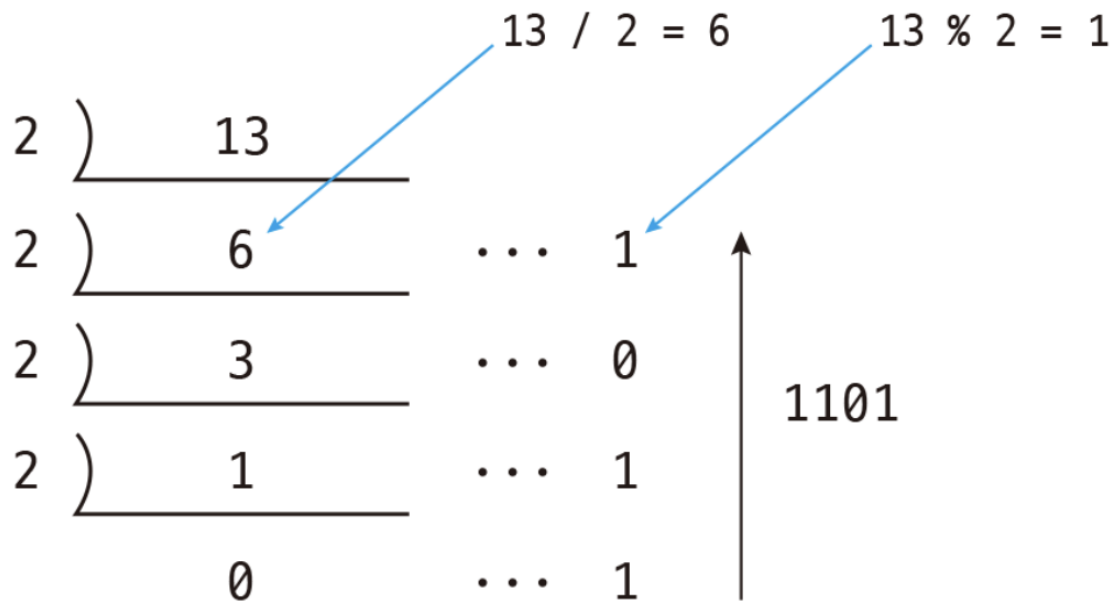
```
function Stack() {
  const items = [];

  this.push() = function(element) {
    items.push(element);
  }
  this.pop() = function() {
    items.pop();
  }
  this.peek = function() {
    return items[items.length-1];
  }
  this.isEmpty = function() {
    return items.length === 0
  }
  this.size = function() {
    return items.length;
  }
  this.clear = function() {
    items = [];
  }
}
```

## 10진수에서 2진수로 변환

컴퓨터에서 모든 것은 0과 1, 두 개의 이진수로만 표시되므로 컴퓨터 과학에선 2진법 표기가 중요하다.

10진수를 2진수로 바꾸려면 **나눈셈의 몫이 0이 될 때까지 2로 나누면 된다.**



```
function getBinaryString(decNumber) {
  const remainderStack = [];
  let remainder;
  let binaryString = '';
  // push
  while(decNumber > 0) {
    remainder = Math.floor(decNumber % 2);
    remainderStack.push(remainder);
    decNumber = Math.floor(decNumber / 2);
  }
  // pop
  while(remainderStack.length !== 0) {
    binaryString += remainderStack.pop().toString();
  }
  return binaryString;
}
```

2진법 이외의 진법에도 대응하도록 변경해보자.

10진수를 2진수로 변경할 때는 나머지가 0 아니면 1이었다. 만약 8진수로 바꾼다면 나머지는 0~7이 될 것이고, 16진수하면 숫자 0~9에 A~F(10~15에 해당)까지 필요하다. 그래서 이런 값들을 바꾸는 로직이 필요하다.

```
function baseConverter(decNumber, base) {  
  const remainderStack = [];  
  let remainder;  
  let baseString = '';  
  const digits = '0123456789ABCDEF';  
  
  while(decNumber > 0) {  
    remainder = Math.floor(decNumber % base);  
    remainderStack.push(remainder);  
    decNumber = Math.floor(decNumber / base);  
  }  
  
  while(remainderStack.length !== 0) {  
    baseString += digits[remainderStack.pop()];  
  }  
  
  return baseString;  
}
```