

# Javascript 프로그래밍

---

컴퓨터의 개념 및 실습

2015.11.24 강의

IDB LAB 임유빈

yblim@idb.snu.ac.kr



# 목차

- 덧셈기
- 정렬 알고리즘 (Sorting Algorithm)
  - 버블 정렬 (Bubble sort)
  - 삽입 정렬 (Insertion sort)
  - 퀵 정렬 (Quick Sort)
- 검색 알고리즘 (Searching Algorithm)
  - 이진 검색 (Binary search)
  - 숫자 예측 프로그램

# 덧셈기

- 1부터 100까지 숫자의 덧셈

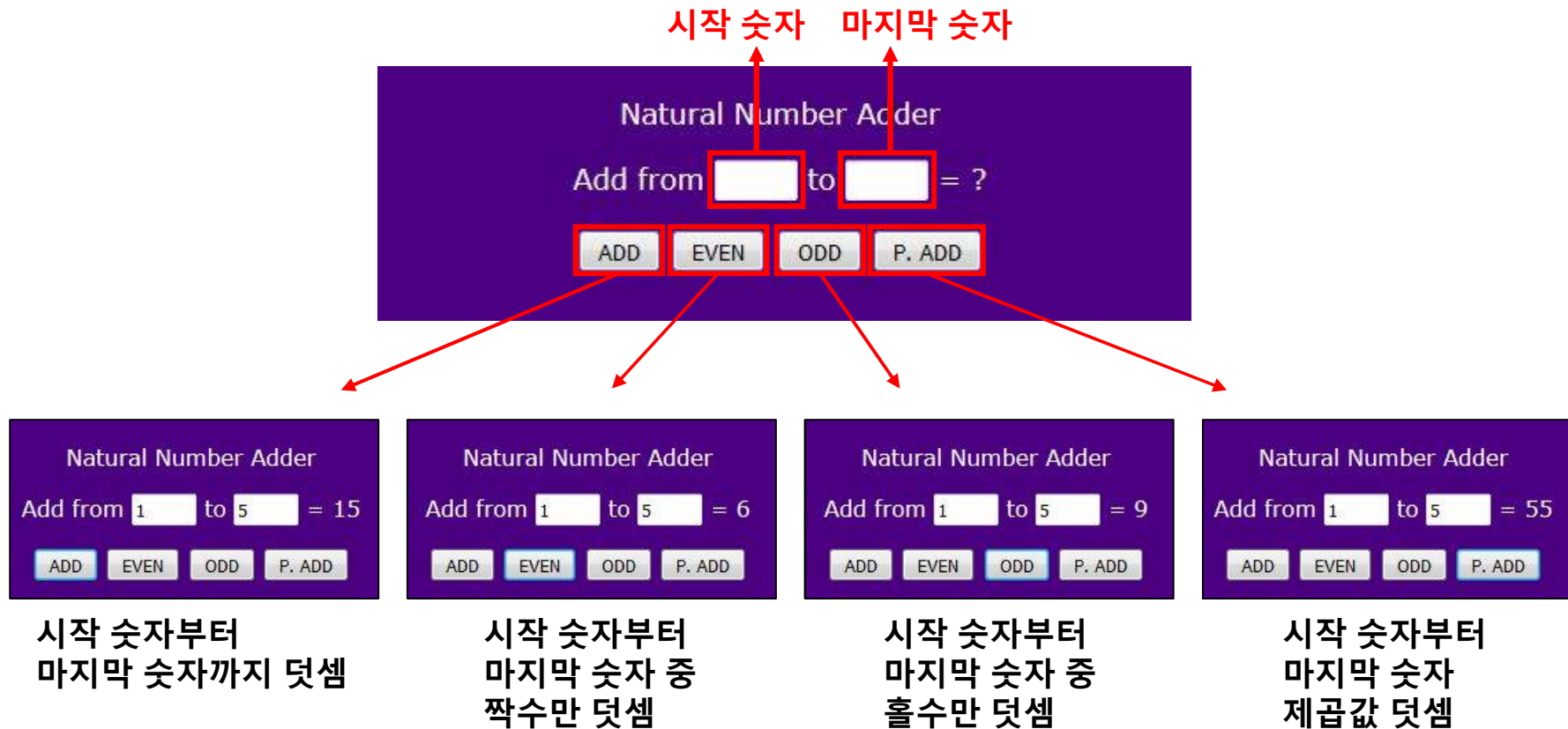
$$\sum_{n=1}^{100} n = \frac{n(n+1)}{2}$$

- 프로그래밍 언어로 작성

```
var sum=0;
for(var i=1; i<=100; i++){
    sum += i;
}
```

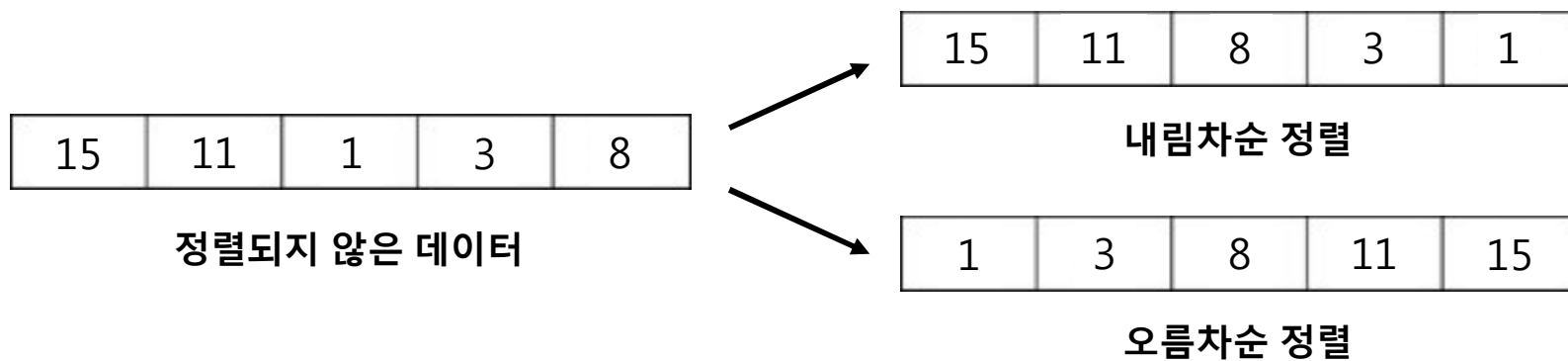
# 덧셈기 예제

- 다양한 덧셈기 만들기



# 정렬 알고리즘

- 데이터를 일정한 규칙에 따라 재배열
- 정보 정렬의 필요성
  - 이름 순 정렬
  - 생일 순 정렬
  - 학번 순 정렬



# 버블 정렬

- 버블 정렬 이름의 유래
  - 공기 방울이 물 속에서 물 위로 올라오듯이

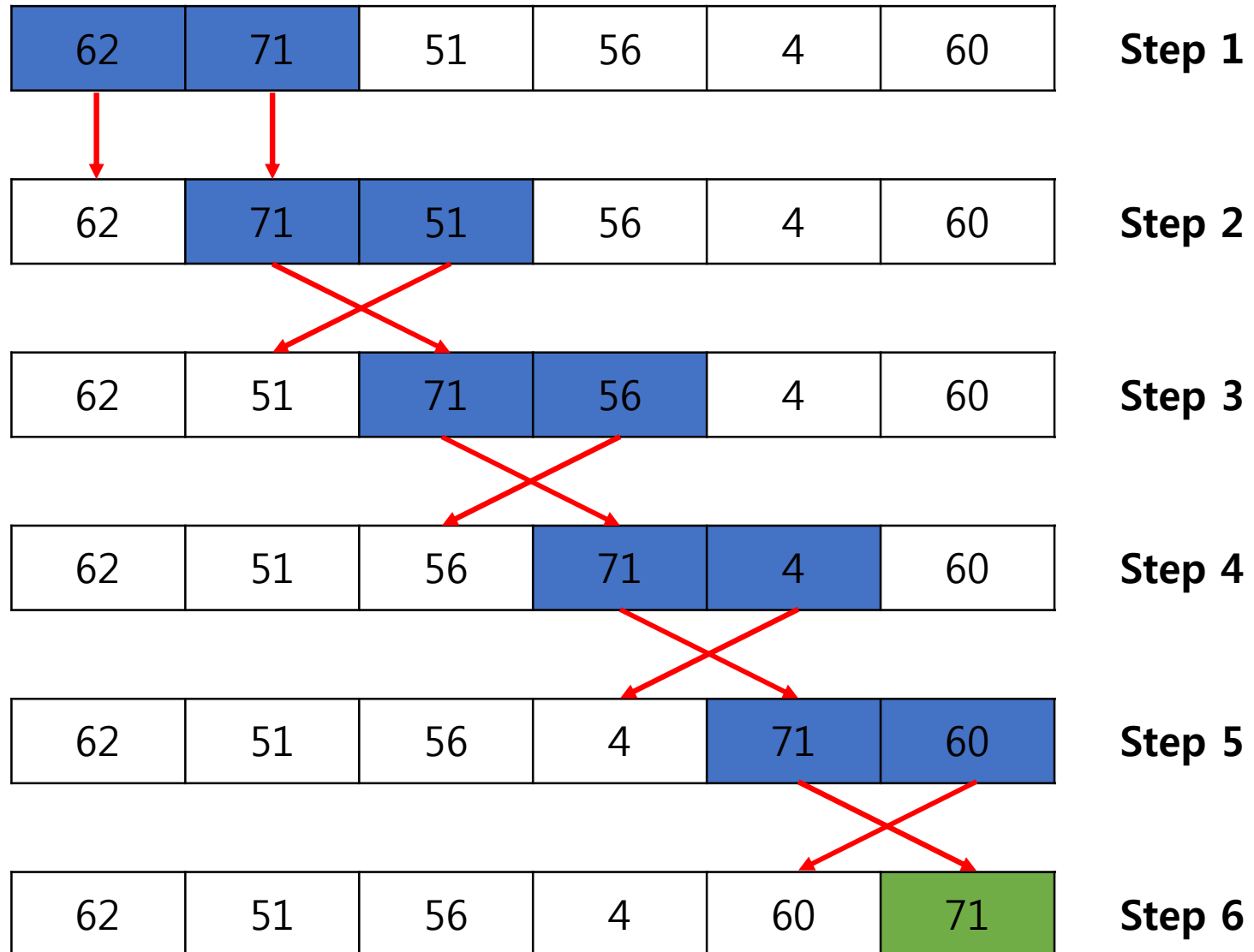




# 버블 정렬

- 가장 단순하고 이해하기 쉬운 정렬 알고리즘
- 알고리즘
  1. 인접한 데이터의 값 비교
  2. 대소관계에 따라 위치 변경
  3. 배열의 처음부터 마지막까지 1,2 반복
  4. 1~3과정 배열의 데이터 수 만큼 반복

# 버블 정렬





# 버블 정렬

## ■ 과정 4

62	51	56	4	60	71	Step 1
----	----	----	---	----	----	--------

51	56	4	60	62	71	Step 2
----	----	---	----	----	----	--------

51	4	56	60	62	71	Step 3
----	---	----	----	----	----	--------

4	51	56	60	62	71	Step 4
---	----	----	----	----	----	--------

4	51	56	60	62	71	Step 5
---	----	----	----	----	----	--------

4	51	56	60	62	71	Step 6
---	----	----	----	----	----	--------

# 삽입 정렬

- 정렬되지 않은 공간과 정렬된 공간 함께 운용
- 비정렬 데이터를 정렬된 부분의 적절한 위치에 삽입

비정렬 공간

62	71	51	56	4	60
71	51	56	4	60	
51	56	4	60		
56	4	60			
4	60				
60					

정렬 공간

62					
62	71				
51	62	71			
51	56	62	71		
4	51	56	62	71	
4	51	56	60	62	71

Step 1

Step 2

Step 3

Step 4

Step 5

Step 6

# 삽입 정렬

62	71	51	56	4	60
----	----	----	----	---	----

62	71	51	56	4	60
----	----	----	----	---	----

Step 1

62	71	51	56	4	60
----	----	----	----	---	----

Step 2

51	62	71	56	4	60
----	----	----	----	---	----

Step 3

51	56	62	71	4	60
----	----	----	----	---	----

Step 4

4	51	56	62	71	60
---	----	----	----	----	----

Step 5

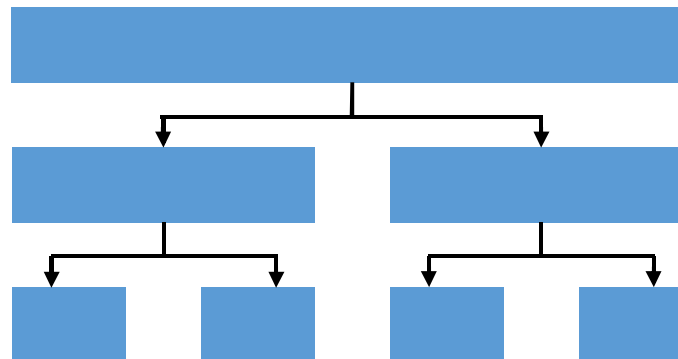
4	51	56	60	62	71
---	----	----	----	----	----

Step 6

■: 정렬 공간 ■: 비정렬 공간 ■: 분류할 데이터

# 퀵 정렬

- 일반적으로 정렬 알고리즘 중에서 가장 빠름
- 알고리즘
  1. 배열에서 기준 값 하나 선정
  2. 기준 값보다 작은 값은 앞으로, 큰 값은 뒤로
  3. 나뉘어진 구역에서 1,2를 반복  
구역의 데이터가 하나일 때까지
- Divide-and-conquer



# 퀵 정렬



■ : 기준 값 ■ : 기준 값보다 큰 값 ■ : 기준 값보다 작은 값

# 검색 알고리즘

- 데이터의 집합에서 원하는 값을 검색
- 순차 검색 (Sequential search)

1	2	3	4	6	7	9
[1]	[2]	[3]	[4]	[5]	[6]	[7]

검색 값

6
---

- 이진 검색 (Binary search)
  - 정렬된 데이터 집합을 이분화하며 검색

# 이진 검색

- 알고리즘

1. **min\_idx** = 배열의 처음 위치
2. **max\_idx** = 배열의 마지막 위치
3. **median\_idx** = (최소 위치 값 + 최대 위치 값) / 2
4. **if ( min\_idx != max\_idx )**
  - a. **if (search > median) min\_idx = median\_idx+1**
  - b. **if (search < median) max\_idx = median\_idx-1**
  - c. **if (search == median) return median**
5. **min\_idx**가 **max\_idx**보다 커질 때까지 3,4 반복  
위 경우에 해당하지 않으면 검색 값은 배열에 없음

# 이진 검색

## ■ 데이터가 있는 경우

[1]	[2]	[3]	[4]	[5]	[6]	[7]	검색 값
1	2	3	4	6	7	9	6

1	2	3	4	6	7	9	$4 < 6$
---	---	---	---	---	---	---	---------

최소 위치 값 = 1 중간 위치 값 =  $(1+7)/2 = 4$  최대 위치 값 = 7 중간 값 = 4 최소 위치 값 =  $4+1=5$

1	2	3	4	6	7	9	$7 > 6$
---	---	---	---	---	---	---	---------

최소 위치 값 = 5 중간 위치 값 =  $(5+7)/2 = 6$  최대 위치 값 = 7 중간 값 = 7 최대 위치 값 =  $6-1=5$

1	2	3	4	6	7	9	$6 = 6$
---	---	---	---	---	---	---	---------

최소 위치 값 = 5 중간 위치 값 =  $(5+5)/2 = 5$  최대 위치 값 = 5

중간 값 = 6

■ : 제외 공간



# 이진 검색

## ■ 데이터가 없는 경우

[1]	[2]	[3]	[4]	[5]	[6]	[7]	검색 값
1	2	3	4	6	7	9	5

1	2	3	4	6	7	9	$4 < 5$
---	---	---	---	---	---	---	---------

최소 위치 값 = 1 중간 위치 값 =  $(1+7)/2 = 4$  최대 위치 값 = 7 중간 값 = 4 최소 위치 값 =  $4+1=5$

1	2	3	4	6	7	9	$7 > 5$
---	---	---	---	---	---	---	---------

최소 위치 값 = 5 중간 위치 값 =  $(5+7)/2 = 6$  최대 위치 값 = 7 중간 값 = 7 최대 위치 값 =  $6-1=5$

1	2	3	4	6	7	9	$6 \neq 5$
---	---	---	---	---	---	---	------------

최소 위치 값 = 5 중간 위치 값 =  $(5+5)/2 = 5$  최대 위치 값 = 5

중간 값 = 6

■ : 제외 공간

# 숫자 예측 프로그램

- 1~100 사이의 숫자 중 임의로 선택된 숫자 예측

Expect number among 1 ~ 100

Expect number:

Range: (66 ~ 74)

Result: Correct! Number is 70

# 숫자 예측 프로그램

- 기능

- EXPECT 버튼

1. 범위 밖 숫자 입력으로 들어올 때 처리

Expect number among 1 ~ 100

Expect number:

Range: (1 ~ 100)

Result: Out of Range

Expect number among 1 ~ 100

Expect number:

Range: (1 ~ 100)

Result: Out of Range

Expect number among 1 ~ 100

Expect number:

Range: (66 ~ 74)

Result: Out of Range

2. 예측 값이 실제 값보다 큰 지 작은지 체크

Expect number among 1 ~ 100

Expect number:

Range: (6 ~ 100)

Result: Higher than 5

Expect number among 1 ~ 100

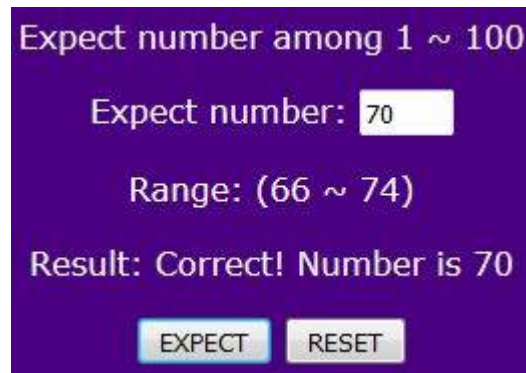
Expect number:

Range: (6 ~ 94)

Result: Lower than 95

# 숫자 예측 프로그램

- 기능
  - 3. 예측 값과 실제 값이 일치 시



- RESET 버튼  
프로그램을 초기 상태로 되돌림