



9주차 1차시

컴퓨팅사고력과 코딩

스크래치 응용 예제 II



동양미래대학교



동의과학대학교



영진전문대학교
YEUNGJIN UNIVERSITY



전주비전대학교



충북보건과학대학교
CHUNGBUK HEALTH & SCIENCE UNIVERSITY

학습목표

- 탐색(검색) 개념을 이해한다.
- 순서 없이 저장된 데이터에서 원하는 정보를 탐색할 수 있다.
- 순서가 있는 데이터에서 원하는 정보를 검색할 수 있다.

학습목차

1. 탐색(검색)이란?

2. 순차 탐색(Sequential Search)

3. 이진 탐색(Binary Search)

4. [예제] 이진 탐색 알고리즘 구현

탐색(검색)이란?

[탐색(검색) Search]

- **탐색** : 주어진 데이터에서 조건에 맞는 혹은 원하는 정보를 찾아내는 작업

순위 ▼	등급/평점	이름
1	2857	
2	2847	
2	2847	
4	2826	
5	2823	
6	2819	
7	2805	
8	2804	
9	2726	
9	2726	
11	2722	
12	2715	
13	2710	
13	2710	
15	2707	

🔍 캐릭터이름

출처: <https://worldofwarcraft.com/ko-kr/game/pvp/leaderboards/3v3>

순차 탐색(Sequential Search)

[순차 탐색 Sequential Search]

- **순서 없이 저장된 데이터**에서 원하는 정보를 탐색하는 방법
- 데이터를 **처음부터 끝까지 모두 탐색**해서 원하는 정보를 찾아냄
- 모든 데이터를 다 탐색해야 하므로, **전체 탐색(Full Search)**라고도 함

[순차 탐색 Sequential Search]

- 아래 데이터에서 '이순신'의 점수를 찾고자 한다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구



[순차 탐색 Sequential Search]

- 아래 데이터에서 '이순신'의 점수를 찾고자 한다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구

← 이순신 ✕

[순차 탐색 Sequential Search]

- 아래 데이터에서 '이순신'의 점수를 찾고자 한다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구



이순신	3	2300	이순신
-----	---	------	-----

동일한 플레이어명이 여러 개 있다면,
무조건 끝까지 모두 탐색해야 함

[순차 탐색 Sequential Search]

(전제조건 : 동일한 플레이어가 없는 경우)

- ‘강감찬’의 점수를 찾는다면? → 1번의 탐색으로 찾기 가능
- ‘김구’의 점수를 찾는다면? → 10번의 탐색으로 찾기 가능

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구

최악의 경우 모든 데이터를 다 탐색해야 함
(즉, 데이터의 개수만큼 비교 연산이 필요함)

이진 탐색(Binary Search)

[이진 탐색 Binary Search]

- **순서대로 저장된 데이터**에서 원하는 정보를 효율적으로 찾는 탐색 기법
- 찾고자하는 값과 데이터의 중간에 있는 값과 비교하여 크고 작음을 비교하는 방식
- 숫자 맞추기 게임 (상대방이 마음 속으로 정한 숫자를 맞추는 게임) 방식과 비슷

[이진 탐색 Binary Search]

- 아래 데이터에서 점수가 1900인 플레이어를 찾는다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구

1900

전체 데이터의 중간에
있는 값과 비교

[이진 탐색 Binary Search]

- 아래 데이터에서 점수가 1900인 플레이어를 찾는다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구

1900

전체 데이터의 중간에
있는 값과 비교

Low = 1

$$\text{Mid} = \frac{1+10}{2} = 5$$

(소수점 이하는 버림)

High = 10

[이진 탐색 Binary Search]

- 아래 데이터에서 점수가 1900인 플레이어를 찾는다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구

← 1900

Low = 6

$$\text{Mid} = \frac{6+10}{2} = 8$$

(소수점 이하는 버림)

High = 10

[이진 탐색 Binary Search]

- 아래 데이터에서 점수가 1900인 플레이어를 찾는다면?

순위	점수	플레이어명
1	3000	강감찬
2	2500	을지문덕
3	2300	이순신
4	2150	광개토대왕
5	2000	문무왕
6	1900	단군
7	1700	김유신
8	1650	우륵
9	1600	유관순
10	1540	김구



1900

6	1900	단군
---	------	----

Low = 6

$$\text{Mid} = \frac{6+7}{2} = 6$$

(소수점 이하는 버림)

High = 7

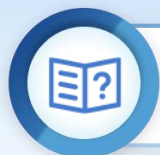
[이진 탐색(Binary Search)의 효율성]

- 순차 탐색(Sequential Search)의 경우, 최소 1번에서 최대 N^1 번의 비교를 통해서 원하는 정보를 탐색할 수 있음 $O(N)$
- 이진 탐색(Binary Search)의 경우, 최소 1번에서 최대 $\log_2 N$ 번의 비교를 통해 원하는 정보를 탐색할 수 있음 $O(\log_2 N)$
($N = 100$ 일 때, 최대 7번, $N = 1,000$ 일 때 최대 10번의 비교로 탐색 가능)
- 이와 같이 최악의 경우에 얼마나 오래 걸리는지(비교 횟수)에 따라 알고리즘의 성능을 평가하여, 이를 시간 복잡도(Time Complexity)라고 함

1) N 은 탐색하는 리스트의 길이(항목의 개수)

코딩하기

[예제] 이진 탐색(Binary Search) 구현하기

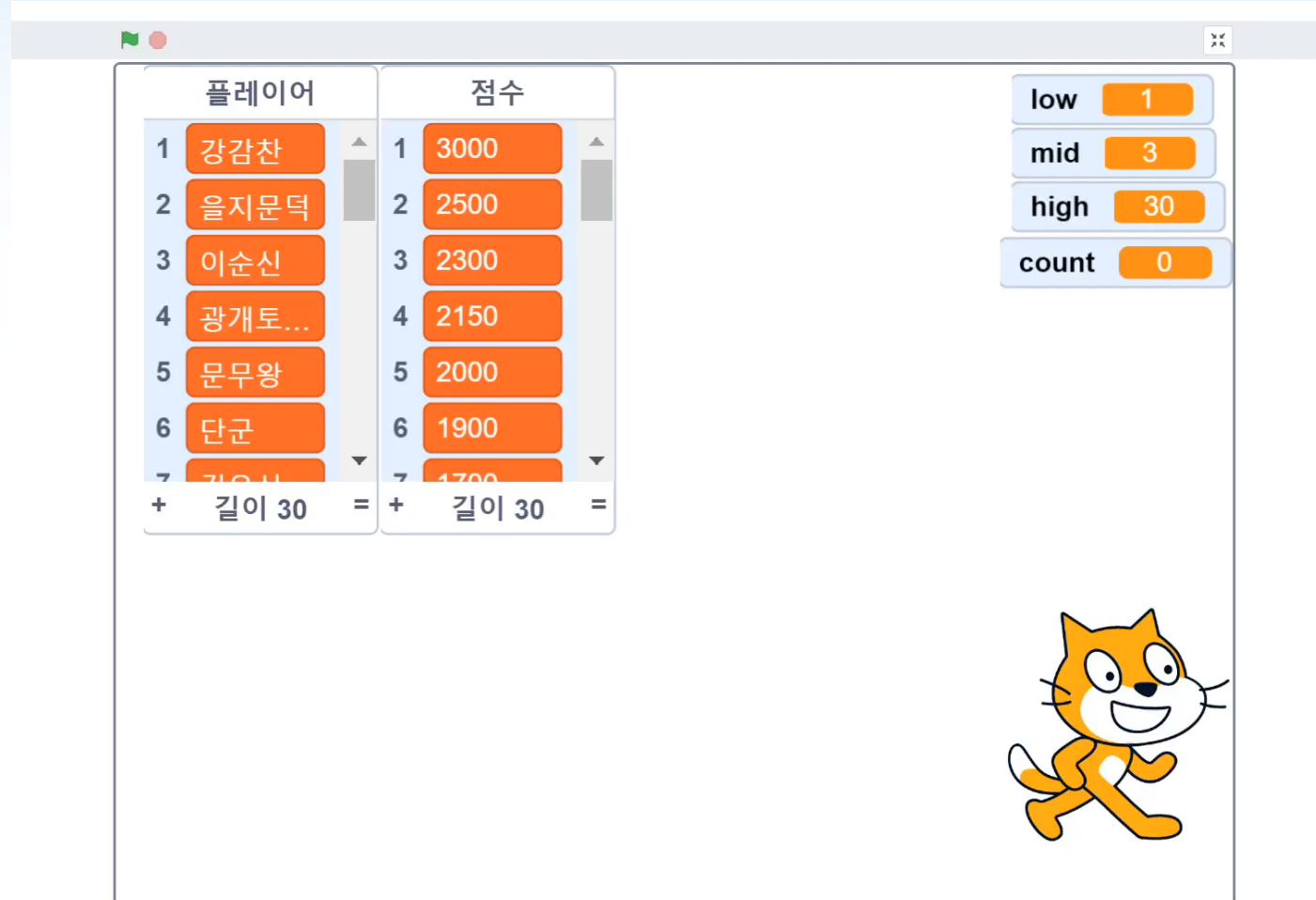


문제정의 - 입력 점수에 해당하는 플레이어 찾기

- ‘찾고 싶은 점수를 입력하세요’라고 묻고 기다리기
- 찾고 싶은 점수를 입력하면, 이진 탐색(Binary Search)를 통해 입력 점수에 해당하는 플레이어의 이름을 찾기
- 찾고 싶은 점수를 찾으면 ‘XX점인 플레이어는 YYYY입니다.’를 2초 동안 외치고, 이어서 ‘ZZ번 만에 찾았습니다.’를 2초 동안 외치기
- 만약 해당 점수를 가진 플레이어가 없다면, ‘해당 점수를 가진 플레이어를 찾을 수 없습니다’를 2초 동안 말하기

4. 예제 - 이진 탐색(Binary Search) 구현하기 ❖ 코딩하기

▶ 결과물 (미리보기)



The image shows a Scratch project window with a table of player names and scores, a search bar, and a cat character.

플레이어	점수
1 강감찬	1 3000
2 을지문덕	2 2500
3 이순신	3 2300
4 광개토...	4 2150
5 문무왕	5 2000
6 단군	6 1900
7 김유신	7 1700

Below the table, there are two search bars, each with a plus sign and the text "길이 30".

On the right side, there are four input fields for variables:

- low: 1
- mid: 3
- high: 30
- count: 0

A cat character is visible in the bottom right corner of the project window.

[사전 준비물]

- 점수와 플레이어 데이터를 리스트에 저장하기 (30명의 플레이어 이름과 점수 저장)
 - ▶ 점수.txt → 점수 리스트에 저장
 - ▶ 플레이어.txt → 플레이어 리스트에 저장
- low, high, mid 변수 생성하기 : 이진 탐색을 위한 인덱스 변수
- count : 이진 탐색에서 수행한 비교 횟수를 저장하는 변수

플레이어		점수	
1	강감찬	1	3000
2	을지문덕	2	2500
3	이순신	3	2300
4	광개토...	4	2150
5	문무왕	5	2000
6	단군	6	1900
7	간지...	7	1700
+ 길이 30 =		+ 길이 30 =	

low	0
mid	0
high	0
count	0

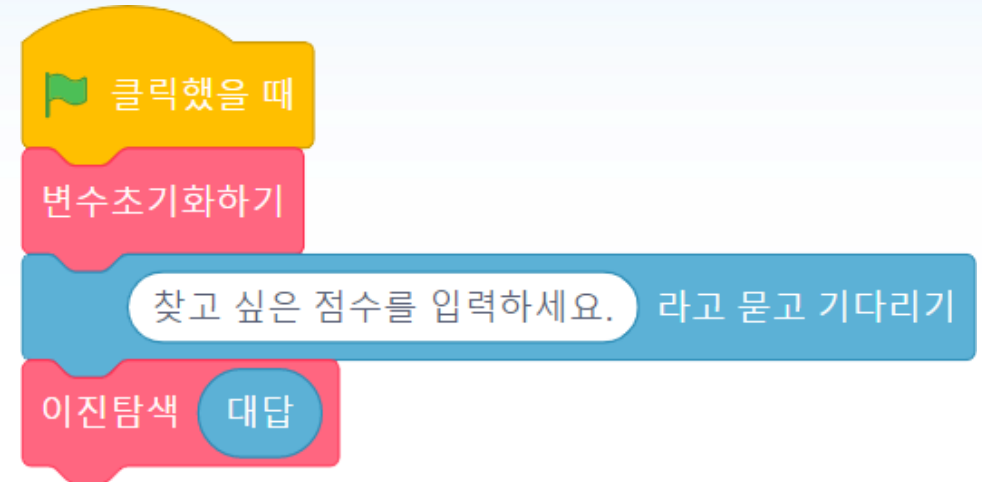


4. 예제 - 이진 탐색(Binary Search) 구현하기 ❖ 코딩하기



1. 스크립트 작성

- 실행 버튼(🚩)을 클릭했을 때
 - ▶ 변수초기화하기
 - ▶ ‘찾고 싶은 점수를 입력하세요.’라고 묻고 기다리기
 - ▶ 이진탐색하기





1. 스크립트 작성

- 변수초기화하기 함수 정의



Low가 리스트의 첫 항목을 가르키도록 1로 정하기

High가 리스트의 마지막 항목을 가르키도록
리스트의 길이로 정하기

비교횟수를 저장하는 count는 0으로 정하기

4. 예제 - 이진 탐색(Binary Search) 구현하기 ❖ 코딩하기



1. 스크립트 작성

- 이진탐색(목표점수)
함수 정의

```

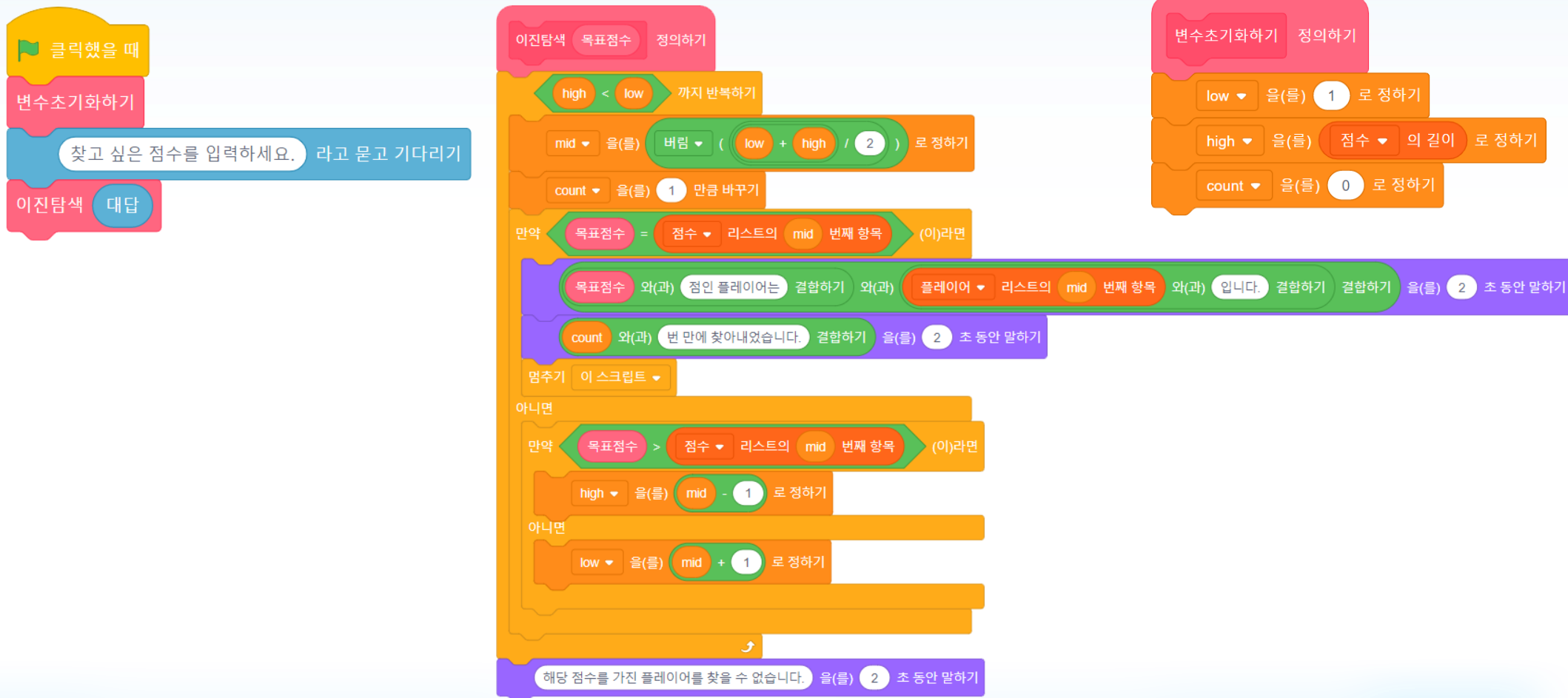
이진탐색 목표점수 정의하기
  high < low 까지 반복하기
    mid 을(를) ( low + high / 2 ) 로 정하기
    count 을(를) 1 만큼 바꾸기
    만약 목표점수 = 점수 리스트의 mid 번째 항목 (아)라면
      목표점수 와(과) 점인 플레이어는 결합하기 와(과) 플레이어 리스트의 mid 번째 항목 와(과) 입입니다. 결합하기 결합하기 을(를) 2 초 동안 말하기
      count 와(과) 번 만에 찾아내었습니다. 결합하기 을(를) 2 초 동안 말하기
    멈추기 이 스크립트
  아니면
    만약 목표점수 > 점수 리스트의 mid 번째 항목 (아)라면
      high 을(를) mid - 1 로 정하기
    아니면
      low 을(를) mid + 1 로 정하기
  해당 점수를 가진 플레이어를 찾을 수 없습니다. 을(를) 2 초 동안 말하기
  
```

4. 예제 - 이진 탐색(Binary Search) 구현하기 ❖ 코딩하기



1. 스크립트 작성

● 전체 스크립트





2. 스크립트 실행

- 스크립트 실행

플레이어		점수	
1	강감찬	1	3000
2	을지문덕	2	2500
3	이순신	3	2300
4	광개토...	4	2150
5	문무왕	5	2000
6	단군	6	1900
7	가야...	7	1700

+ 길이 30 = + 길이 30 =

low 1

mid 3

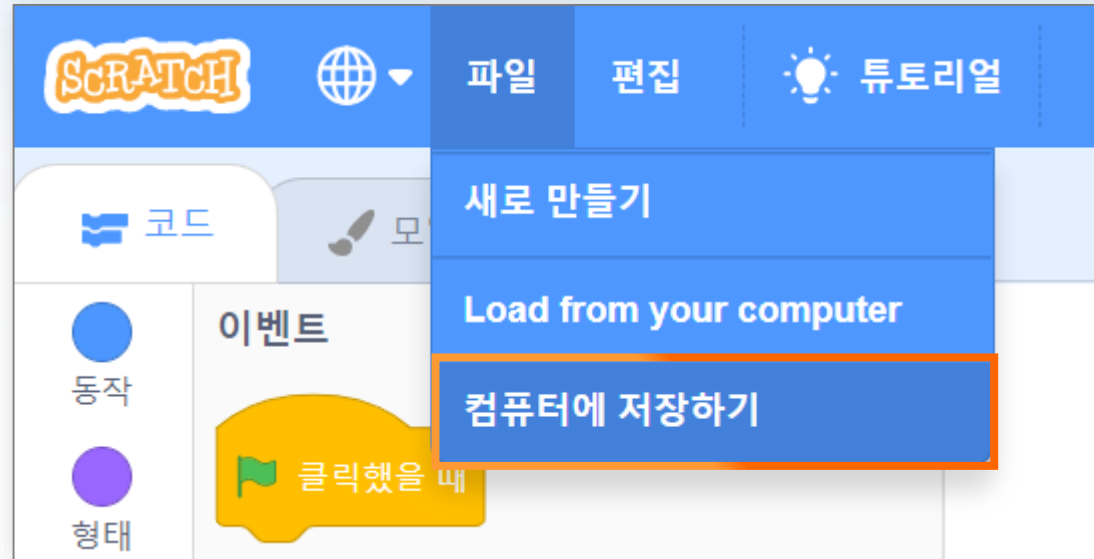
high 30

count 0



2. 스크립트 실행

- 프로젝트 저장



프로젝트명

9-1-이진탐색구현.sb3

학습정리

- ☑ **탐색(Search)**이란 주어진 데이터에서 조건에 맞는 혹은 원하는 정보를 찾아내는 작업이다.
- ☑ **순차 탐색(Sequential Search)**이란 순서 없이 저장된 데이터에서 원하는 정보를 탐색하는 방법이다.
 - ▶ 최대 N 번의 비교를 통해 원하는 정보 탐색 가능
- ☑ **이진 탐색(Binary Search)**이란 순서대로 저장된 데이터에서 원하는 정보를 효율적으로 찾는 탐색 기법이다.
 - ▶ 최대 $\log_2 N$ 번의 비교를 통해 원하는 정보 탐색 가능

본 수업자료는 저작권법 제 25조 2항에 따라
학교 수업을 목적으로 이용되었으므로,
본 수업자료를 외부에 공개, 게시하는 것을 금지하며,
이를 위반하는 경우 저작권 침해로서 관련법에 따라 처벌될 수 있습니다.