



9주차 2차시

컴퓨팅사고력과 코딩

스크래치 응용 예제 III



동양미래대학교



동의과학대학교



영진전문대학교
YEUNGJIN UNIVERSITY



전주비전대학교



충북보건과학대학교
CHUNGBUK HEALTH & SCIENCE UNIVERSITY

학습목표

- 정렬(소트, sort)의 개념 및 필요성
- 다양한 정렬 알고리즘
- 선택 정렬(Selection Sort) 알고리즘 상세
- 스크래치에서 선택 정렬 알고리즘을 활용하여 정렬 기능을 구현한다.

학습목차

1. 정렬의 개념 및 필요성

2. 다양한 정렬 알고리즘

3. 선택 정렬 알고리즘 상세

4. [예제] 선택 정렬 알고리즘 구현

정렬의 개념 및 필요성

[정렬(소트) Sort]

- **정렬** : 순서 없이 저장된 데이터를 일정한 순서대로 열거하는 행위
 - ▶ **오름차순** : 값이 작은 것부터 큰 순으로 정렬
 - ▶ **내림차순** : 값이 큰 것부터 작은 순으로 정렬



1. 정렬의 개념 및 필요성

[일상 생활에서의 정렬된 데이터 활용]



영어 사전

(출처 : <https://news.join.com/article/4412924>)

순위	변동	게임명	장르	점유율(%)	사용시간(시)
1		리그 오브 레전드	RTS	45.28%	1,454,960
2	▲1	서든어택	FPS	7.49%	240,508
3	▲1	배틀그라운드	FPS	7.01%	225,139
4	▼2	피파온라인4	스포츠	6.64%	213,438
5		오버워치	FPS	4.51%	145,052
6		메이플 스토리	RPG	4.41%	141,663
7		아이온	RPG	3.12%	100,349
8	▲1	던전앤파이터	RPG	2.27%	73,037
9	▼1	스타크래프트	RTS	2.20%	70,577
10		월드 오브 워크래프트	RPG	1.29%	41,333

게임 순위

(출처 : http://www.gametrics.com/news/News02_View.aspx?seqid=45080)

국가	확진자	사망자	완치	치명 (%)	완치 (%)	발생률*
1 미국US	28,262,656 (+301)	497,177 (+2)	18,225,135 (+439)	1.8	64.5	85,385
2 인도IN	10,919,616	155,789	10,624,838	1.4	97.3	7,913
3 브라질BR	9,834,513	239,294	8,745,424	2.4	88.9	46,267
4 러시아RU	4,086,090	80,520	3,607,036	2.0	88.3	27,999
5 영국GB	4,038,078	117,166	2,160,515	2.9	53.5	59,483
6 프랑스FR	3,465,163	81,814	238,753	2.4	6.9	53,087
7 스페인ES	3,056,035	64,747	N/A	2.1	N/A	65,363
8 이탈리아IT	2,721,879	93,577	2,225,519	3.4	81.8	45,018

코로나 현황

(출처 : <https://coronaboard.kr/>)

영화 랭킹			
조희순	평점순 (현재상영영화)	평점순 (모든영화)	2021.02.14
순위	영화명	평점	변동폭
1	그린 북	★★★★★ 9.59	평점주기 - 0
2	가버나움	★★★★★ 9.59	평점주기 - 0
3	디지털 머드벤처 라스트 에볼루션 : 인연	★★★★★ 9.53	평점주기 - 0
4	베일리 어게인	★★★★★ 9.53	평점주기 - 0
5	먼 훗날 우리	★★★★★ 9.51	평점주기 - 0
6	원더	★★★★★ 9.51	평점주기 - 0
7	극장판 바이올렛 에버가든	★★★★★ 9.50	평점주기 - 0
8	아일라	★★★★★ 9.48	평점주기 - 0
9	당갈	★★★★★ 9.48	평점주기 ↑ 1

영화 순위

(출처 : <https://movie.naver.com/movie/sdb/rank/rmovie.nhn?sel=pnt&date=20210214>)

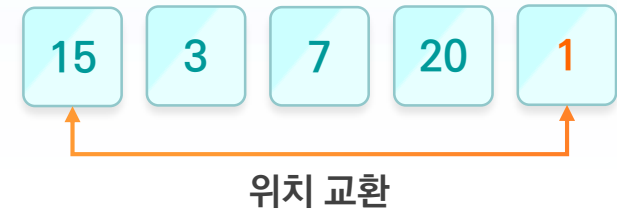
[정렬의 필요성]

- 원하는 정보를 **검색**하는데 드는 **시간 단축**
 - ▶ 만약 영어 사전에 단어들이 순서 없이 기록되어 있다면, 내가 원하는 단어를 찾으려면?
→ 순차 탐색
 - ▶ 영어 사전에는 단어들이 알파벳 순서로 기록되어 있기 때문에 내가 원하는 단어를 쉽게 찾을 수 있음
- **정보를 보다 가치 있게** 활용 가능
 - ▶ 사용자의 의사결정에 도움을 줄 수 있음

다양한 정렬 알고리즘

[다양한 정렬 알고리즘]

정렬 알고리즘명	정렬 방식
선택 정렬(Selection sort)	정렬되지 않은 데이터 중에서 가장 작은 값을 찾아 앞자리 숫자와 교환 하는 방법
버블 정렬(Bubble sort)	이웃한 데이터끼리 크고 작음을 따져 서로 위치를 교환하는 방법
버킷 정렬(Bucket sort)	첫째자리 수만큼 통(버킷)을 준비하여 그곳에 데이터를 분류하고 정렬하는 방법
삽입 정렬(Insertion sort)	새로운 데이터의 크기를 따져 이미 정렬된 데이터 사이에 삽입하는 방법
퀵 정렬(Quick sort)	임의의 기준 값을 선택하고 작은 데이터는 기준 값 왼쪽에, 큰 데이터는 기준 값 오른쪽에 가져다 놓는 방법



[다양한 정렬 알고리즘]

정렬 알고리즘명

정렬 방식

선택 정렬(Selection sort)

정렬되지 않은 데이터 중에서 가장 작은 값을 찾아
앞자리 숫자와 교환하는 방법

버블 정렬(Bubble sort)

이웃한 데이터끼리 크고 작음을 따져 **서로 위치를
교환**하는 방법

버킷 정렬(Bucket sort)

첫째자리 수만큼 통(버킷)을 준비하여 그곳에 데이터를
분류하고 정렬하는 방법

삽입 정렬(Insertion sort)

새로운 데이터의 크기를 따져 이미 정렬된 데이터 사이에
삽입하는 방법

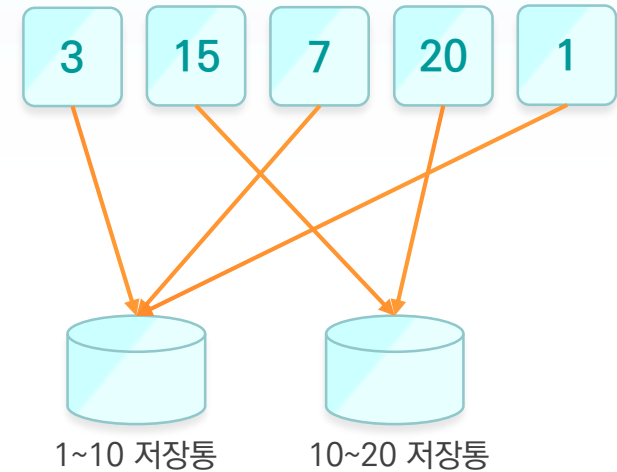
퀵 정렬 (Quick sort)

임의의 기준 값을 선택하고 작은 데이터는 기준 값 왼쪽에,
큰 데이터는 기준 값 오른쪽에 가져다 놓는 방법



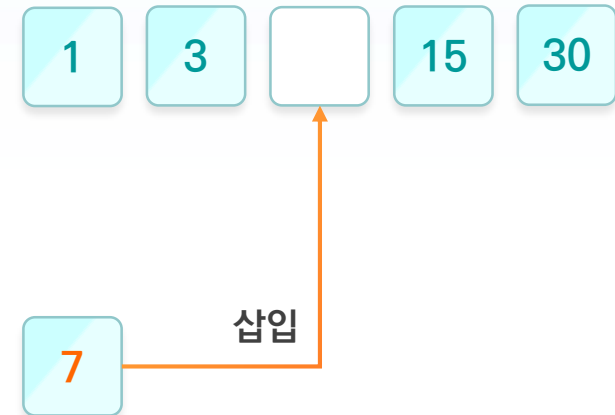
[다양한 정렬 알고리즘]

정렬 알고리즘명	정렬 방식
선택 정렬(Selection sort)	정렬되지 않은 데이터 중에서 가장 작은 값을 찾아 앞자리 숫자와 교환하는 방법
버블 정렬(Bubble sort)	이웃한 데이터끼리 크고 작음을 따져 서로 위치를 교환하는 방법
버킷 정렬(Bucket sort)	첫째자리 수만큼 통(버킷)을 준비하여 그곳에 데이터를 분류하고 정렬하는 방법
삽입 정렬(Insertion sort)	새로운 데이터의 크기를 따져 이미 정렬된 데이터 사이에 삽입하는 방법
퀵 정렬 (Quick sort)	임의의 기준 값을 선택하고 작은 데이터는 기준 값 왼쪽에, 큰 데이터는 기준 값 오른쪽에 가져다 놓는 방법



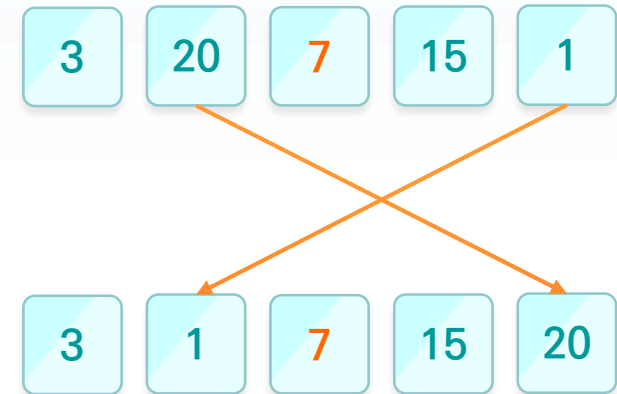
[다양한 정렬 알고리즘]

정렬 알고리즘명	정렬 방식
선택 정렬(Selection sort)	정렬되지 않은 데이터 중에서 가장 작은 값을 찾아 앞자리 숫자와 교환하는 방법
버블 정렬(Bubble sort)	이웃한 데이터끼리 크고 작음을 따져 서로 위치를 교환하는 방법
버킷 정렬(Bucket sort)	첫째자리 수만큼 통(버킷)을 준비하여 그곳에 데이터를 분류하고 정렬하는 방법
삽입 정렬(Insertion sort)	새로운 데이터의 크기를 따져 이미 정렬된 데이터 사이에 삽입 하는 방법
퀵 정렬 (Quick sort)	임의의 기준 값을 선택하고 작은 데이터는 기준 값 왼쪽에, 큰 데이터는 기준 값 오른쪽에 가져다 놓는 방법



[다양한 정렬 알고리즘]

정렬 알고리즘명	정렬 방식
선택 정렬(Selection sort)	정렬되지 않은 데이터 중에서 가장 작은 값을 찾아 앞자리 숫자와 교환하는 방법
버블 정렬(Bubble sort)	이웃한 데이터끼리 크고 작음을 따져 서로 위치를 교환하는 방법
버킷 정렬(Bucket sort)	첫째자리 수만큼 통(버킷)을 준비하여 그곳에 데이터를 분류하고 정렬하는 방법
삽입 정렬(Insertion sort)	새로운 데이터의 크기를 따져 이미 정렬된 데이터 사이에 삽입하는 방법
퀵 정렬 (Quick sort)	임의의 기준 값을 선택 하고 작은 데이터는 기준 값 왼쪽에, 큰 데이터는 기준 값 오른쪽에 가져다 놓는 방법



선택 정렬 알고리즘 상세

[선택 정렬(Selection Sort) 알고리즘 소개]

- ① 미정렬 리스트에서 가장 작은 수 찾기
- ② 미정렬 리스트의 맨 앞 항목과 가장 작은 수를 교환
- ③ 모두 정렬될 때까지 1, 2번 반복



코딩하기

[예제] 선택 정렬 알고리즘 구현하기



문제정의 - 선택 정렬 구현하기

- ‘점수로 정렬’ 버튼을 클릭하면, 점수 순으로 오름차순 정렬을 수행한다.
- ‘플레이어명으로 정렬’ 버튼을 클릭하면, 플레이어명 순으로 오름차순 정렬을 수행한다. (가나다 순)

▶ 결과물 (미리보기)

The image shows a Scratch project window with two main data lists and several variables.

점수		플레이어	
1	1220	1	정몽주
2	1270	2	이퇴계
3	1279	3	신사임당
4	1329	4	논개
5	1284	5	전봉준
6	1290	6	지석영
7	1290	7	유동주

Below the lists, there are four variable monitors:

- first: 30
- last: 30
- n: 31
- temp: 3000
- min: 29

Two purple text boxes are overlaid on the right side of the project window:

- 점수로 정렬 (Sort by score)
- 플레이어명으로 정렬 (Sort by player name)

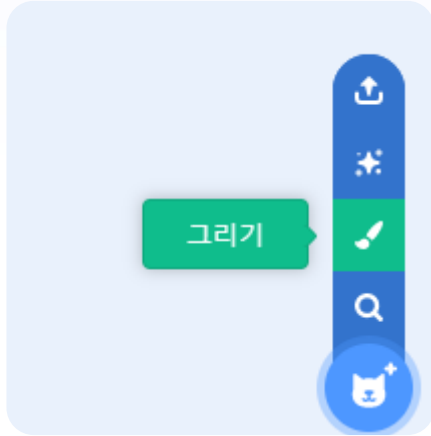
[사전 준비물]

- 점수와 플레이어 데이터를 리스트에 저장하기 (9-1주차. 이진 탐색 실습과 동일한 파일)
 - ▶ 점수.txt → 점수 리스트에 저장
 - ▶ 플레이어.txt → 플레이어 리스트에 저장
- first, last 변수 생성하기 : 미정렬 리스트의 처음과 끝을 가르키는 변수
- min : 미정렬 리스트에서 최소값을 갖는 항목 인덱스를 저장하기 위한 변수
- n : 최소값을 찾을 때 사용하는 반복횟수
- temp : 두 값의 위치를 바꿀 때 사용하는 임시 변수

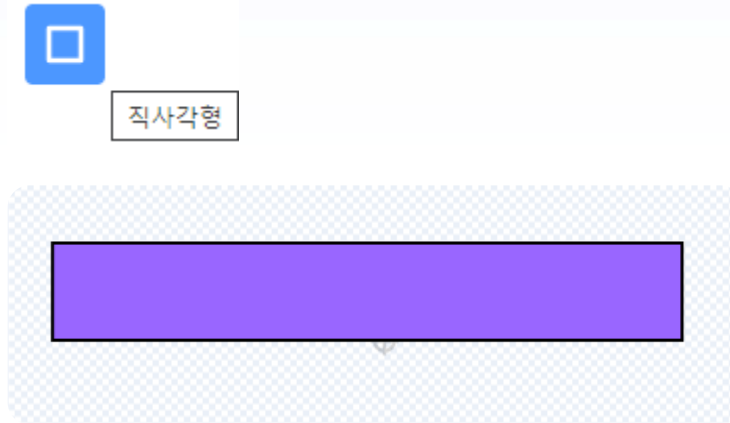
점수		플레이어	
1	3000	1	강감찬
2	2500	2	을지문덕
3	2300	3	이순신
4	2150	4	광개토...
5	2000	5	문무왕
6	1900	6	단군
7	1700	7	한글
+ 길이 30 =		+ 길이 30 =	
first	1	temp	0
last	30	min	0

[사전 준비물]

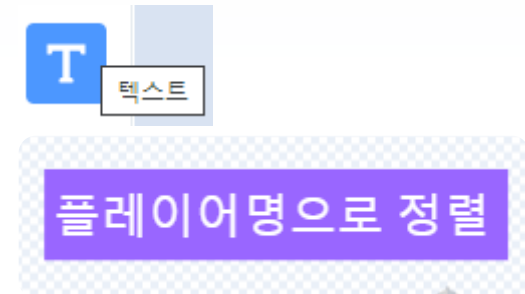
- ‘플레이어명으로 정렬’ 버튼 스프라이트 준비하기



① 스프라이트 > 그리기



② 직사각형 선택 후, 사각형 박스 그리기



③ 텍스트 선택 후, 입력



1. 스크립트 작성

- 실행 버튼(🚩)을 클릭했을 때
 - ▶ 변수초기화하기
 - ▶ 오름차순정렬하기

이 스프라이트를 클릭했을 때

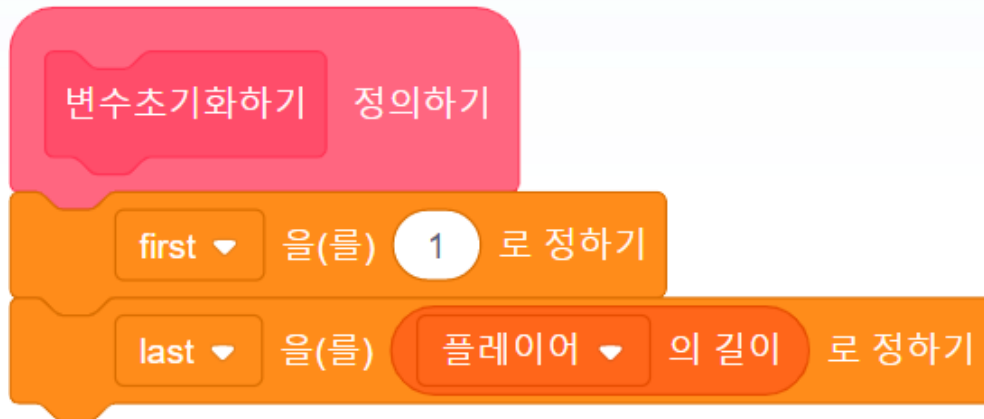
변수초기화하기

오름차순정렬하기



1. 스크립트 작성

- 변수초기화하기 함수 정의



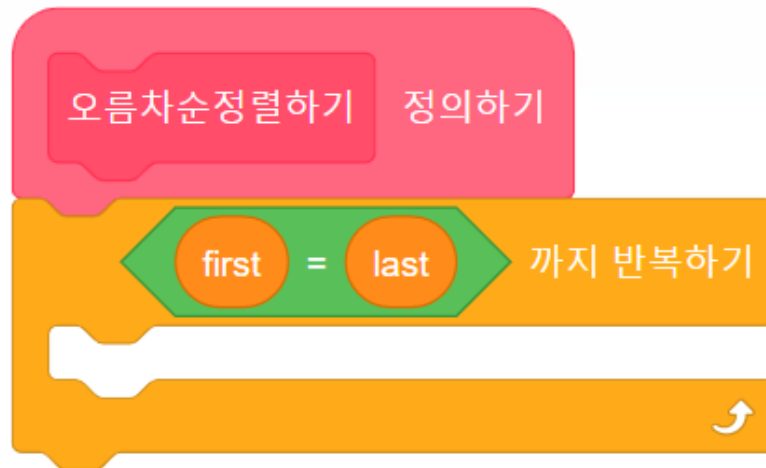
처음에는 리스트 전체가 모두 미정렬 상태이므로,
First는 리스트의 첫번째 항목을 가르키도록 1로,
Last는 리스트의 마지막을 가르키도록 리스트의 길이로 초기화



1. 스크립트 작성

- 오름차순정렬하기

① 모두 정렬할 때까지 반복하기 (first = last)





1. 스크립트 작성

• 오름차순정렬하기

- ② first부터 last까지의 숫자 중 최소값이 들어있는 위치를 찾기 위해 n과 min을 first값으로 초기화





1. 스크립트 작성

오름차순정렬하기

- 3 미정렬 리스트(first~last)에서
최소값 찾기



4. 예제 - 선택 정렬 알고리즘 구현하기 ❖ 코딩하기



1. 스크립트 작성

오름차순정렬하기

- 4 미정렬 리스트의 맨 앞에 있는 항목과 최소값을 교환하기



4. 예제 - 선택 정렬 알고리즘 구현하기 ❖ 코딩하기



1. 스크립트 작성

● 전체 스크립트

이 스프라이트를 클릭했을 때

변수초기화하기

오름차순정렬하기

변수초기화하기 정의하기

first ▾ 을(를) 1 로 정하기

last ▾ 을(를) 플레이어 ▾ 의 길이 로 정하기

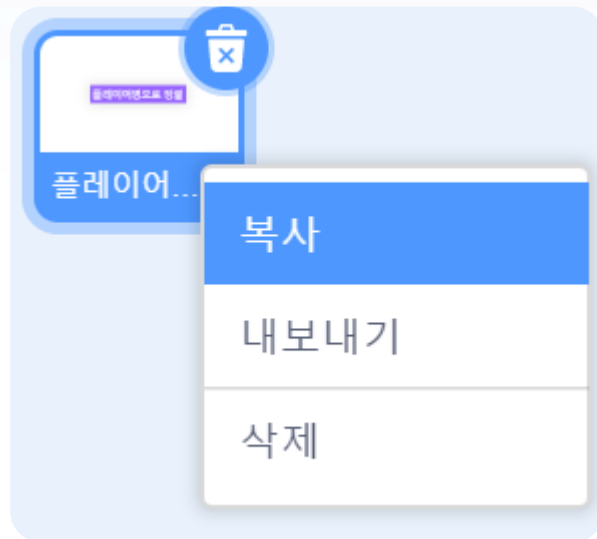


4. 예제 - 선택 정렬 알고리즘 구현하기 ❖ 코딩하기



1. 스크립트 작성

• 점수로 정렬하기



플레이어 정렬 스프라이트 복사하기

오름차순정렬하기 정의하기

최소값 항목 인덱스를 찾는 조건만 플레이어 리스트에서 점수 리스트로 변경

```

first = last 까지 반복하기
n 을(를) first 로 정하기
min 을(를) first 로 정하기
n > last 까지 반복하기
만약 플레이어 리스트의 n 번째 항목 < 플레이어 리스트의 min 번째 항목 (이)라면
    min 을(를) n 로 정하기
n 을(를) 1 만큼 바꾸기
만약 first = min 이(가) 아니다 (이)라면
    temp 을(를) 플레이어 리스트의 first 번째 항목 로 정하기
    플레이어 리스트의 first 번째 항목을 플레이어 리스트의 min 번째 항목 으로 바꾸기
    플레이어 리스트의 min 번째 항목을 temp 으로 바꾸기
    temp 을(를) 점수 리스트의 first 번째 항목 로 정하기
    점수 리스트의 first 번째 항목을 점수 리스트의 min 번째 항목 으로 바꾸기
    점수 리스트의 min 번째 항목을 temp 으로 바꾸기
first 을(를) 1 만큼 바꾸기
  
```



2. 스크립트 실행

- 스크립트 실행

점수

1	1220
2	1270
3	1279
4	1329
5	1284
6	1290
7	1299

플레이어

1	정몽주
2	이퇴계
3	신사임당
4	논개
5	전봉준
6	지석영
7	유동준

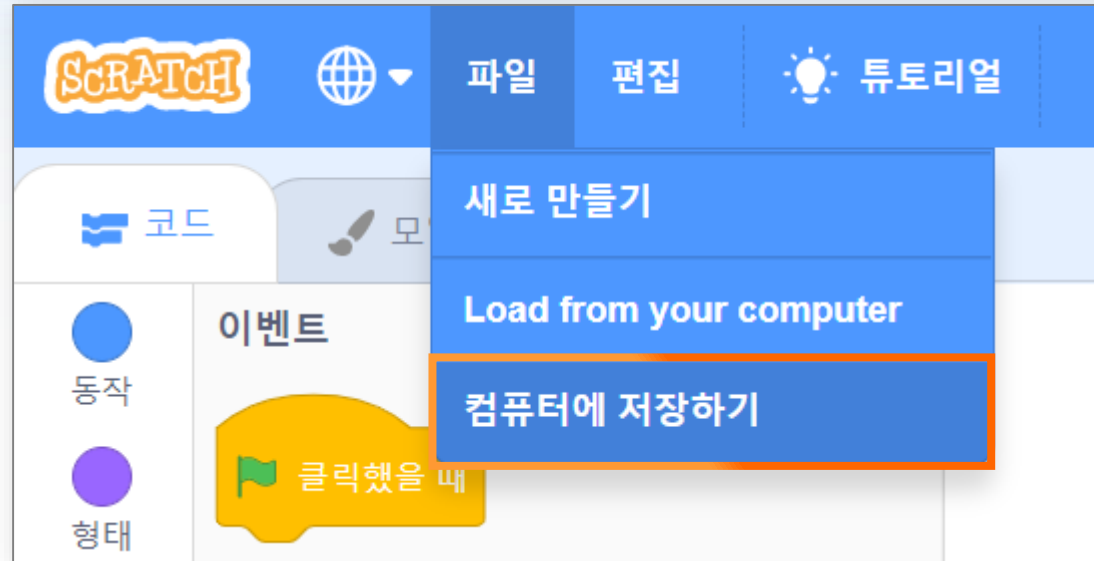
first 30 temp 3000
last 30 min 29
n 31

점수로 정렬
플레이어명으로 정렬



2. 스크립트 실행

- 프로젝트 저장



프로젝트명

9-2-선택정렬알고리즘구현.sb3

학습정리

- ☑ 정렬은 순서 없이 저장된 데이터를 일정한 순서대로 열거하는 행위이다.

 - ▶ 오름차순 : 값이 작은 것부터 큰 순으로 정렬
 - ▶ 내림차순 : 값이 큰 것부터 작은 순으로 정렬
- ☑ 정렬이 필요한 이유는 다음과 같다.

 - ▶ 원하는 정보를 검색하는 데 드는 시간 단축
 - ▶ 정보를 보다 가치 있게 활용 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.