

교육용프로그래밍언어기초(스크래치)

6주차 - 2교시

난수와 재귀 호출





 난수의 개념

 난수 생성 기능 연습

 재귀호출의 개념

 재귀호출을 활용한 알고리즘

 스크래치의 재귀 프로시저를 활용한 프로그래밍 연습

학습목표

- ① 난수의 개념을 이해할 수 있다.
- ② 스크래치의 난수 생성 기능을 사용할 수 있다.
- ③ 재귀호출의 개념을 이해할 수 있다.
- ④ 재귀호출을 활용한 알고리즘을 이해할 수 있다.
- ⑤ 스크래치의 재귀 프로시저를 활용해서 프로그래밍 할 수 있다.



1 난수의 개념





1

난수의 개념

1 난수의 이해

1 난수

난수

정해진 범위 내에서 무작위로 추출된 수

1

난수의 개념

1 난수의 이해

2 난수가 사용되는 예

- 복권 추첨에서 당첨번호 생성
- 게임에서 무작위로 등장하는 적군이나 아이템들
- 예측할 수 없게 유입되는 손님이나 작업들의 시뮬레이션

1

난수의 개념

2 컴퓨터의 난수 생성

1 컴퓨터의 난수 생성 방법

- 이미 정해진 계산 방법을 사용하기 때문에 매번 생성하는
난수의 배열이 동일함

예 새로 실행할 때마다 같은 패턴으로 적군이 나타나는 게임

- 이런 문제를 해결하기 위해, 계속 변하는 컴퓨터의 현재 시각을 씨앗 값으로 매번 다른 난수의 배열 생성

Point

컴퓨터의 시간을 조정하면 결국 다른 난수 배열 생성이 가능함

1

난수의 개념

2 컴퓨터의 난수 생성

2 의사난수

- 같은 조건 하에서는 동일한 배열로 생성되는 난수
- 컴퓨터에 의해 생성되는 모든 난수는 의사난수

1

난수의 개념

2 컴퓨터의 난수 생성

3 컴퓨터의 난수 생성

- 컴퓨터의 난수 생성에 **씨앗 값을 사용하지 않는 경우**

첫 번째 난수: 23, 45, 2, 14....

두 번째 난수: 23, 45, 2, 14....

매번 같은 난수의 배열이 생성됨

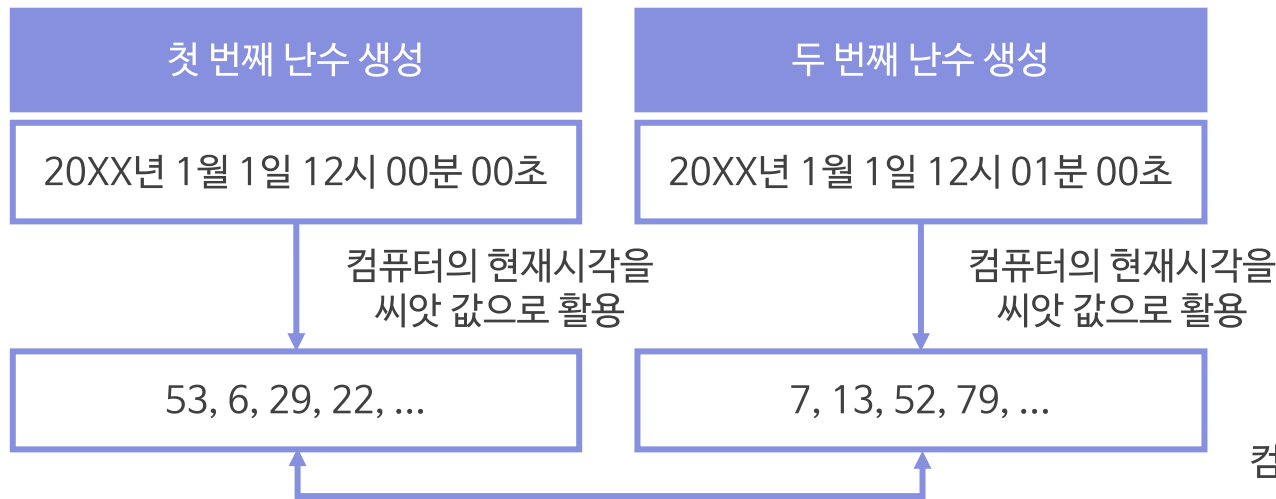
1

난수의 개념

2 컴퓨터의 난수 생성

3 컴퓨터의 난수 생성

- 컴퓨터의 난수 생성에 **씨앗 값을 사용하는 경우**



매번 다른 난수의 배열이 생성됨
컴퓨터의 현재시각을 조정하면 같은
난수의 배열 생성도 가능



2 난수 생성 기능 연습

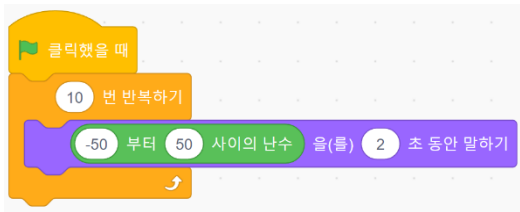


2 난수 생성 기능 연습

1 스크래치의 난수 생성 기능

1 정수 또는 실수를 무작위로 생성하는 자료 블록 제공

- [코드] 탭의 [연산] 메뉴에 존재



- 〈정수로만 범위를 나타내면 정수로 난수를 생성〉

2 난수 생성 기능 연습

1 스크래치의 난수 생성 기능

1 정수 또는 실수를 무작위로 생성하는 자료 블록 제공

- [코드] 탭의 [연산] 메뉴에 존재

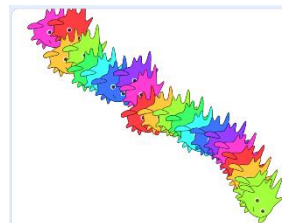


- 〈한쪽이라도 실수로 범위를 나타내면 실수로 난수를 생성〉

2 고보가 임의의 방향으로 움직이게 하기

1 예제 프로그램의 개요

- ① 프로그램을 실행하면 고보 스프라이트는 실행 화면 중앙에 정지해 있습니다.
- ② 고보 스프라이트를 클릭하면 현재 바라보는 방향 각도보다 10만콤작은 수부터 10만콤 큰 수 사이의 무작위 각도를 바라보며 일정 거리만큼 이동합니다(일정 거리 이동 후 다시 같은 방식으로 바라보는 방향을 계산하여 이동하는 행동을 무한히 반복).
- ③ 고보 스프라이트는 벽에 닿으면 반대 방향으로 튕겨져 이동합니다.
- ④ 고보 스프라이트는 이동하면서 색이 계속 변하면서 무한히 복제를 만들어 자신의 뒤에 꼬리를 다는 행동을 수행합니다.
- ⑤ 고보의 복제는 복제 당시 고보의 색을 그대로 반영하며 고보 스프라이트의 뒤로 일정 거리만큼 이동하다 사라집니다.

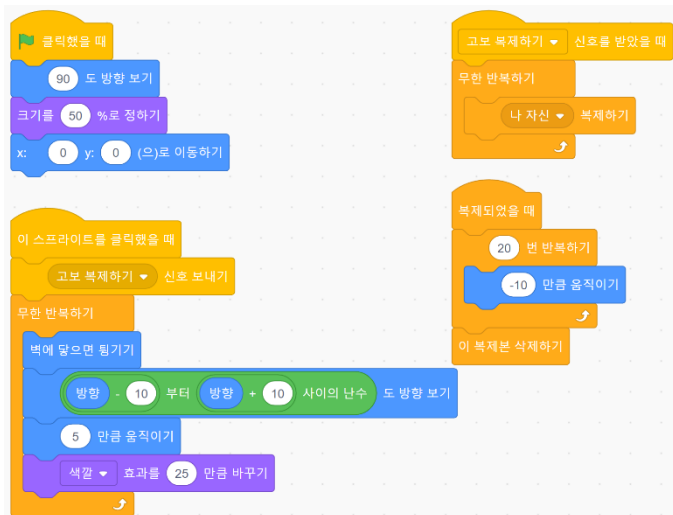


2

난수 생성 기능 연습

2 고보가 임의의 방향으로 움직이게 하기

2 고보 스프라이트의 행동



2

난수 생성 기능 연습

③ 난수를 활용한 화살로 풍선 맞추기

1 예제 프로그램의 개요

- ① 풍선종류1: 원래 크기 대비 40% 크기의 보라색 풍선으로 0.1초마다 거리 15만큼씩 올라갑니다. 화살로 맞히면 30만큼의 점수가 추가됩니다.
- ② 풍선 종류 2: 원래 크기 대비 50% 크기의 파란색 풍선으로 0.1초마다 거리 10만큼씩 올라갑니다. 화살로 맞히면 20만큼의 점수가 추가됩니다.
- ③ 풍선 종류 3: 원래 크기 대비 60%크기의 노란색 풍선으로 0.1초마다 거리 5만큼씩 올라갑니다. 화살로 맞히면 10만큼의 점수가 추가됩니다.

2

난수 생성 기능 연습

③ 난수를 활용한 화살로 풍선 맞추기

1 예제 프로그램의 개요

- 〈풍선이 생성되는 시간 간격〉

풍선1: 매번 5초에서 7초 사이의 간격으로 생성

풍선2: 매번 3초에서 5초 사이의 간격으로 생성

풍선3: 매번 2초에서 4초 사이의 간격으로 생성

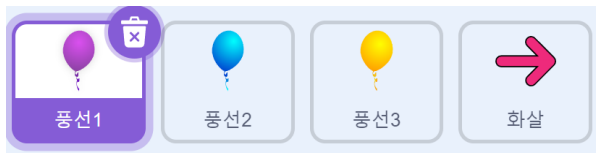
2

난수 생성 기능 연습

③ 난수를 활용한 화살로 풍선 맞추기

2

스프라이트 준비하기



3

변수 준비하기

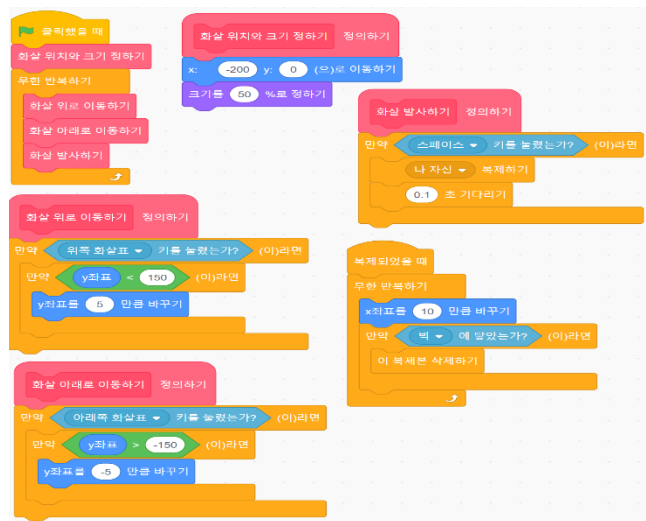
- 무대의 전역 변수 [점수]

난수 생성 기능 연습

③ 난수를 활용한 화살로 풍선 맞추기

4 화살 스프라이트의 행동

- <화살로 풍선 맞히기(기초)>의 화살 스프라이트 행동과 동일함

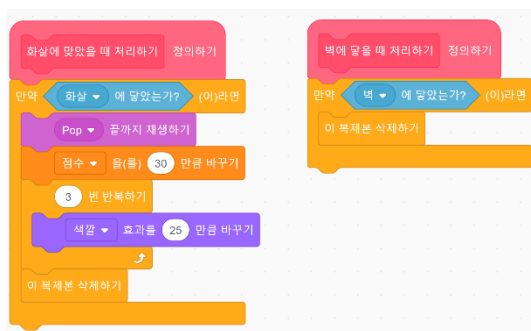
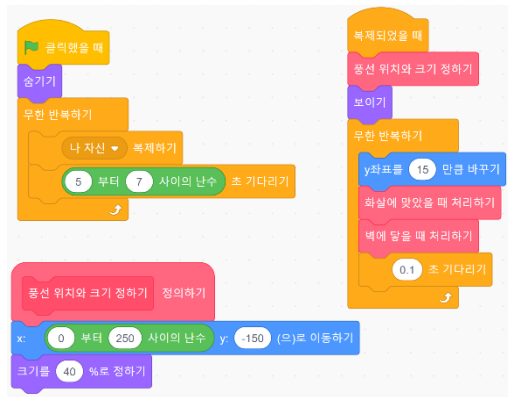


2

난수 생성 기능 연습

3 난수를 활용한 화살로 풍선 맞추기

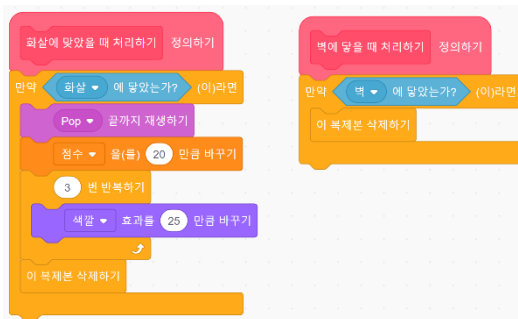
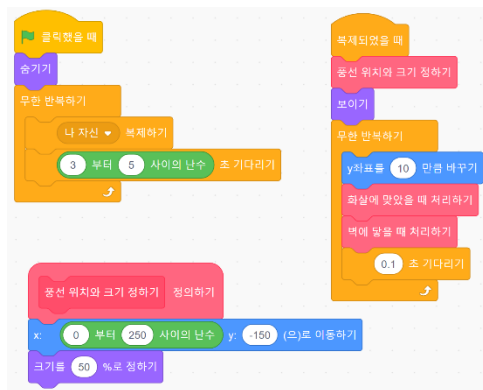
5 풍선1 스프라이트의 행동



2 난수 생성 기능 연습

3 난수를 활용한 화살로 풍선 맞추기

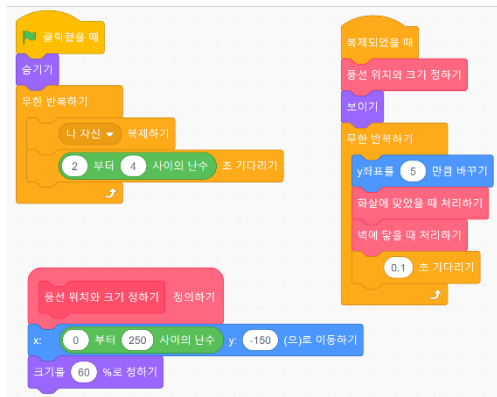
6 풍선2 스프라이트의 행동



2 난수 생성 기능 연습

3 난수를 활용한 화살로 풍선 맞추기

7 풍선3 스프라이트의 행동





3 재귀호출의 개념



3

재귀호출의 개념

1 재귀호출의 이해

1 재귀호출

재귀호출

프로시저가 자기 자신을 호출하는 것



3

재귀호출의 개념

1 재귀호출의 이해

2 재귀 프로시저

재귀 프로시저

재귀호출을 수행하는 프로시저

3

재귀호출의 개념

1 재귀호출의 이해

3 재귀 프로시저 사용 시 유의할 점

- 무한한 호출이 발생하여 프로시저가 종료되지 않을 수 있음
- 재귀 프로시저는 반드시 자신을 호출하는 행동을 종료하는 조건이 존재해야 하고 언젠가 그 조건이 만족되어 재귀호출을 중단해야 함



4

재귀호출을 활용한 알고리즘



4

재귀호출을 활용한 알고리즘

1 재귀 프로시저의 구조

1 재귀 프로시저는 크게 두 부분으로 구성

재귀호출 종료 부분

- 재귀호출을 종료할 조건과 그 조건이 만족될 시 수행되어야 할 작업
- 재귀호출 종료 조건은 프로시저의 매개변수에 의해 결정

재귀호출 수행 부분

- 다시 자신을 호출해야 할 상황에서 수행해야 할 작업과 자기 자신을 호출하는 작업

4

재귀호출을 활용한 알고리즘

1 재귀 프로시저의 구조

1 재귀 프로시저는 크게 두 부분으로 구성

재귀 프로시저 (매개변수)

매개변수를 활용한 <재귀호출 종료 조건>이 만족한다면

- 재귀호출 종료 상황에서 수행되어야 할 작업을 수행함
- 재귀호출을 더 이상 수행하지 않음

<재귀호출 종료 조건>이 만족되지 않는다면

- 최종적인 결과를 위한 일부 작업 또는 중간 과정의 작업을 수행함
- 인자를 변경하여 재귀호출을 수행함

재귀호출할 때마다 매개변수의 값이
변경되어 언젠가는
<재귀호출 종료 조건>을 만족해야 함

4

재귀호출을 활용한 알고리즘

2 재귀호출로 임의의 양의 정수까지 더하기

1 반환값 기능을 지원하는 프로시저의 활용

임의의 양의 정수까지 더하기 (N)

〈N=1〉이 만족되면

1을 반환하고 재귀호출을 더 이상 수행하지 않음

〈N=1〉이 만족되지 않으면

〈N+임의의 양의 정수까지 더하기 (N-1)〉을 반환함

재귀호출할 때마다 인자 값을 1만큼씩 줄이므로 언젠가는 프로시저의 매개변수 값이 1이 되어 재귀호출 종료 조건〈N=1〉을 만족함

반환값 6이 최종 결과값임

↑ 6(=3+3)을 반환

임의의 양의 정수까지 더하기 (3)

(3+임의의 양의 정수까지 더하기(2))를 반환함

재귀호출 ↓

↑ 3(=2+1)을 반환

임의의 양의 정수까지 더하기 (2)

(2+임의의 양의 정수까지 더하기(1))을 반환함

재귀호출 ↓

↑ 1을 반환

임의의 양의 정수까지 더하기 (1)

1을 반환하고 더 이상 재귀호출을 수행하지 않음

4

재귀호출을 활용한 알고리즘

2 재귀호출로 임의의 양의 정수까지 더하기

2 반환값 기능을 지원하지 않는 프로시저의 활용

- 전역 변수 활용 - 재귀호출 때마다 프로시저의 매개변수 값을 전역 변수에 더함

임의의 양의 정수까지 더하기 (N)

〈N=1〉이 만족되면

- 변수 [결과값]에 1을 더함
- 재귀호출을 더 이상 수행하지 않음

〈N=1〉이 만족되지 않으면

- 변수 [결과값]에 N을 더함
- N보다 1만큼 작은 수를 인자로 재귀호출을 수행함

재귀호출할 때마다 인자 값을 1만큼씩 줄이므로 언젠가는 프로시저의 매개변수 값이 1이 되어 재귀호출 종료 조건〈N=1〉을 만족함



4

재귀호출을 활용한 알고리즘

2 재귀호출로 임의의 양의 정수까지 더하기

2 반환값 기능을 지원하지 않는 프로시저의 활용

임의의 양의 정수까지 더하기 (3)

- 변수 [결과값]에 매개변수 3을 더함
- 임의의 양의 정수까지 더하기(2)를 호출함

임의의 양의 정수까지 더하기 (2)

- 변수 [결과값]에 매개변수 2을 더함
- 임의의 양의 정수까지 더하기(1)를 호출함

임의의 양의 정수까지 더하기 (1)

- 변수 [결과값]에 매개변수 1을 더함
- 더 이상 재귀호출을 수행하지 않음

결과값 = 0

결과값 = 3

결과값 = 5

결과값 = 6

호출될 때마다 매개변수의 값을 전역 변수 [결과값]에 더함

재귀호출이 종료될 때 전역 변수 [결과값]은 1부터 3까지 더한 값을 가짐

4

재귀호출을 활용한 알고리즘

2 재귀호출로 임의의 양의 정수까지 더하기

3 스크래치의 프로시저로 구현

- 스�크래치의 프로시저는 반환값 기능이 없으므로 전역 변수 활용





5

스크래치의 재귀 프로시저를 활용한 프로그래밍 연습



1 스크래치의 프로시저로 구현

- 스크래치의 프로시저는 반환값 기능이 없으므로 전역 변수 활용





6

프로그래밍 실습



교육용프로그래밍언어기초(스크래치)

Next

복제, 난수, 재귀 호출을 활용한 응용 프로그래밍

