

교육용프로그래밍언어기초(스크래치)

4주차 - 2교시

# 병렬 처리의 개념과 구현



 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

 병렬 처리 개념 구현하기





# 학습목표

① 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩 개념을 설명할 수 있다.

② 병렬 처리 방식을 이해하고 이벤트 블록을 활용할 수 있다.



1

# 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념





# 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

### 1 멀티 태스킹

#### 1 태스크

##### 태스크

실행 중인 프로그램을 의미  
(‘프로세스’와 거의 동일한 개념)



# 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

### 1 멀티 태스킹

#### 2 멀티 태스킹

- 여러 개의 프로그램들을 동시에 실행시키는 개념

## 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 1 멀티 태스킹

## 3 멀티 태스킹의 예



웹 브라우저로  
웹 서핑



채팅 프로그램으로  
채팅



음악 플레이어로  
음악감상

## 1

# 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 2 멀티 스레딩의 개념

## 1

### 스레드

#### 스레드

프로그램의 기능 (문장들)을 실행하는  
프로그램의 실행 주체를 의미

- 프로그램은 여러 개의 스레드를 생성해 자신 기능들을 병렬적으로 수행 가능



## 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 2 멀티 스레딩의 개념

## 2 멀티 스레딩

- 하나의 프로그램이 여러 개의 기능들을 동시에 실행시키는 개념

## 3 멀티 스레딩의 예

채팅 프로그램의 채팅  
기능으로 친구들과 채팅



채팅 프로그램의 파일 전송  
기능으로 친구에게 파일 전송

## ③ 멀티 스레딩의 유의점

- 1 여러 스레드들 중 **하나만 오류**를 범해도 프로그램 **전체가 종료**될 수 있음
  - 예를 들어, 채팅 프로그램에서 채팅 중 파일 전송에 오류가 발생하면 채팅 프로그램 전체가 종료될 수 있음

## 1

# 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 3 멀티 스레딩의 유의점

2 스레드들은 한 프로그램의 자원을 공유하기 때문에  
예기치 않은 상황이 발생할 수 있음

- 예를 들어, 두 스레드가 하나의 변수를 공유할 때 하나의 스레드가 그 값을 읽어 계산하는 도중 다른 스레드가 그 변수를 다른 값으로 변경해 버릴 수 있음
- 이런 경우 처음의 스레드는 변수의 최신 값이 아닌 예전 값으로 계산하기 때문에 계산 결과가 무의미할 수 있음

3 실행 중인 프로그램은 자신이 생성한  
모든 스레드들이 종료될 때까지 종료되지 않음

## 1

# 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 4 스크래치의 병렬 처리 기능

- 1 스프라이트와 무대는 이벤트가 발생할 때 이에 대응하는 행동을 수행하도록 프로그래밍 함

### 이벤트

어떤 동작을 실행시키는 사건



## 4 스크래치의 병렬 처리 기능

- 1 스프라이트와 무대는 이벤트가 발생할 때 이에 대응하는 행동을 수행하도록 프로그래밍 함

### 스크래치의 대표적 이벤트

깃발 메뉴 클릭, 키보드의 특정 키 누름, 마우스 클릭 등

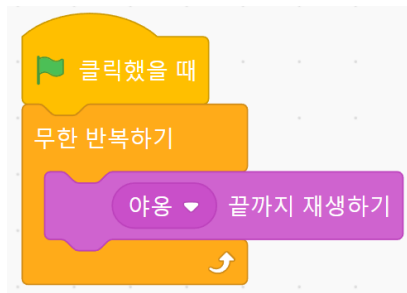
- 스크래치는 다양한 이벤트들에 대해서 이를 처리할 기능을 조립하도록 이벤트 블록들을 제공함

## 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 4 스크래치의 병렬 처리 기능

## 2 이벤트 블록을 활용한 고양이 스프라이트 행동의 예



- <실행을 클릭하는 이벤트가 발생하면 '야옹' 소리를 무한히 반복하기 시작한다.>

# 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

### 4 스크래치의 병렬 처리 기능

#### 2 이벤트 블록을 활용한 고양이 스프라이트 행동의 예



- 〈스페이스 바를 누르는 이벤트가 발생하면 마우스 포인터를 향해 무한히 따라가기 시작한다.〉

## 1

# 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

## 4 스크래치의 병렬 처리 기능

### 3 앞 장의 프로그램의 실행 모습

- 깃발 메뉴를 클릭하면, 고양이 스프라이트가 야옹 소리를 무한히 반복하여 출력함
- 스페이스 바를 누르면, 고양이 스프라이트가 마우스 포인터를 향해 움직임
- 고양이 스프라이트는 위 두 가지 행동이 **병렬적으로** 수행됨





# 1

## 병렬 처리를 위한 멀티 태스킹과 멀티 스레딩의 개념

### 4 스크래치의 병렬 처리 기능

#### 4 스크래치의 병렬 처리 기능

- 위의 프로그램 실행에서 알 수 있듯이, 스크래치의 병렬 처리는 이벤트에 반응하는 스프라이트와 무대의 행동들마다 스레드를 생성하기 **멀티 스레딩 방식**을 사용함

#### Point

프로그램에 사용된 모든 이벤트 블록들이 멀티 스레딩 방식으로 병렬 처리 됨

## 4 스크래치의 병렬 처리 기능



(실행) 클릭



(실행) 클릭  
이벤트 처리를 위한  
스레드 생성

스페이스바 누름

스페이스 바 누름  
이벤트 처리를 위한  
스레드 생성

‘야옹’ 소리를 무한히 내는 기능 수행

마우스 포인터를 향해 무한히 움직이는 기능 수행

‘야옹’ 소리를 무한히 내는 기능과 마우스 포인터를  
향해 무한히 움직이는 기능이 병렬적으로 실행



## 2

# 병렬 처리 개념 구현하기



## 2

## 병렬 처리 개념 구현하기

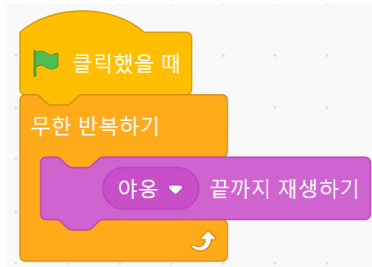
### 1 프로그램의 개요

- 1 프로그램을 실행하면 고양이 스프라이트와 개 스프라이트는 제자리에서 각각 야옹 소리와 멍멍 소리를 반복
- 2 스페이스 바를 누르면 고양이 스프라이트는 마우스 포인터를 쫓아가고 개 스프라이트는 고양이 스프라이트를 쫓아감

## 2

## 병렬 처리 개념 구현하기

## 2 고양이의 스크립트



- 〈실행을 클릭하는 이벤트가 발생하면 ‘야옹’ 소리를 무한히 반복하기 시작한다.〉

## 2

## 병렬 처리 개념 구현하기

## 2 고양이의 스크립트



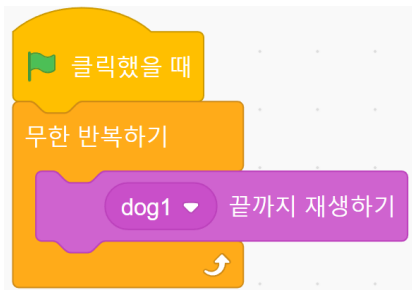
- 〈스페이스 바를 누르는 이벤트가 발생하면 마우스 포인터를 향해 무한히 따라가기 시작한다.〉

## 2

## 병렬 처리 개념 구현하기

## ③ 개의 스크립트

- [스프라이트 고르기]에서 “Dog1”을 추가하고 아래와 같이 프로그래밍함



- <실행을 클릭하는 이벤트가 발생하면 ‘멍멍’ 소리를 무한히 반복하기 시작>

## 2

## 병렬 처리 개념 구현하기

## ③ 개의 스크립트

- [스프라이트 고르기]에서 “Dog1”을 추가하고 아래와 같이 프로그래밍함

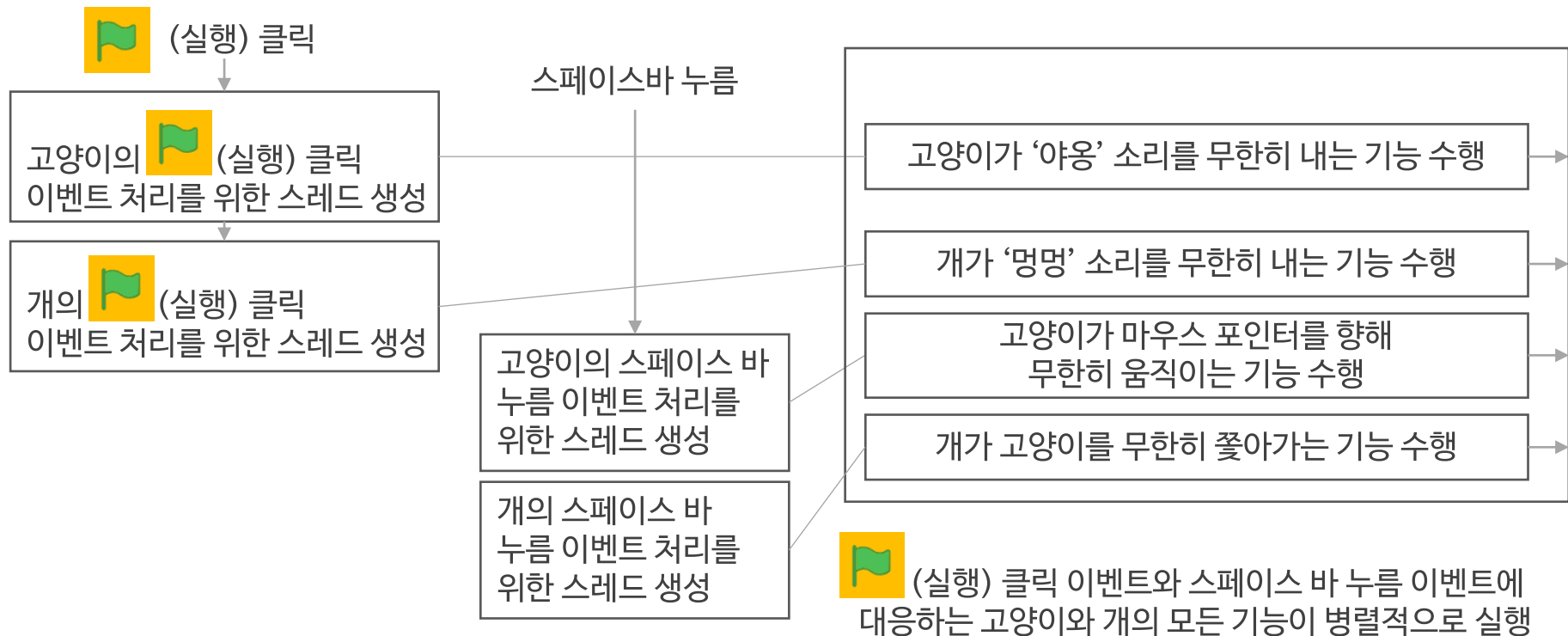


- 〈스페이스 바를 누르는 이벤트가 발생하면 고양이를 향해 무한히 따라가기 시작〉



## 2 병렬 처리 개념 구현하기

### 4 프로그램의 동작 구조



## 2

## 병렬 처리 개념 구현하기

## 5 &lt;개가 고양이를 쫓게 하기&gt;

## 1 프로그램의 개요

- ① 프로그램을 실행하면 고양이는 “야옹” 소리를 무한히 반복합니다.
- ② 프로그램을 실행하면 개는 “멍멍” 소리를 무한히 반복합니다.
- ③ 키보드의 방향 키를 누르면 고양이는 방향 키가 가리키는 방향을 바라보며 5만큼 이동합니다.
- ④ 개는 0.2초마다 2만큼씩 이동하며 고양이를 쫓아갑니다.



## 2

# 병렬 처리 개념 구현하기

## 5 <개가 고양이를 쫓게 하기>

### 2 스크래치를 실행하여 스프라이트 준비

- 고양이 스프라이트
- 개 스프라이트 ([스프라이트 고르기]에서 “Dog1”)

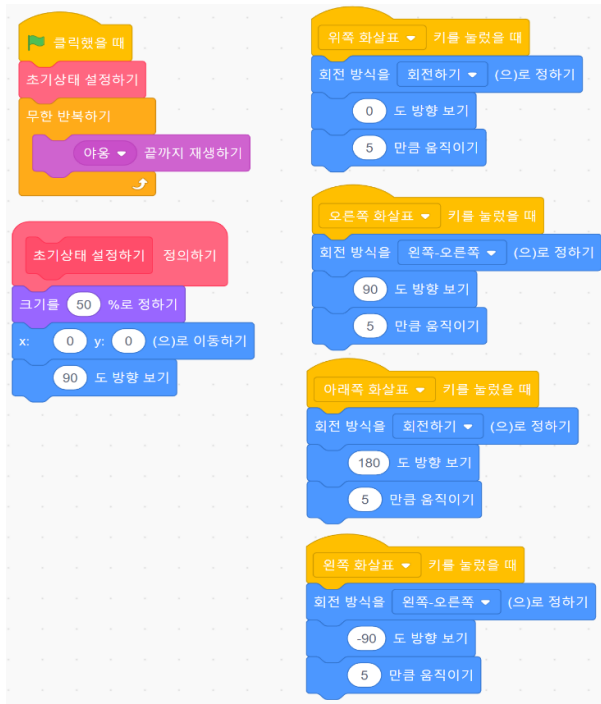
## 2

## 병렬 처리 개념 구현하기

## 5 &lt;개가 고양이를 쫓게 하기&gt;

## 3

## 고양이의 스크립트



The image displays a Scratch script for a cat character, organized into two columns. The script is designed to make the cat move and respond to clicks.

**Left Column Script:**

- When Clicked:** A yellow block labeled "클릭했을 때" (When Clicked) triggers the following actions:
  - Set Initial State:** A pink block labeled "초기상태 설정하기" (Set Initial State) followed by a "정의하기" (Define) block.
  - Repeat:** An orange "무한 반복하기" (Repeat Forever) block containing a purple "아옹" (Aong) block and a "끝까지 재생하기" (Play to End) block.
  - Size and Position:** A pink "크기를 50%로 정하기" (Set Size to 50%) block, followed by a blue "x: 0 y: 0 (으)로 이동하기" (Move to x: 0 y: 0) block, and a blue "90도 방향 보기" (Turn 90 degrees) block.

**Right Column Script:**

- When Key Pressed (Up):** A yellow block labeled "위쪽 화살표 키를 눌렀을 때" (When Up Arrow Key is Pressed) triggers:
  - Turn:** A blue "회전 방식을 회전하기 (으)로 정하기" (Set Turn Method to Turn) block.
  - Direction and Movement:** Blue blocks for "0도 방향 보기" (Turn 0 degrees) and "5만큼 움직이기" (Move 5 units).
- When Key Pressed (Right):** A yellow block labeled "오른쪽 화살표 키를 눌렀을 때" (When Right Arrow Key is Pressed) triggers:
  - Turn:** A blue "회전 방식을 왼쪽-오른쪽 (으)로 정하기" (Set Turn Method to Left-Right) block.
  - Direction and Movement:** Blue blocks for "90도 방향 보기" (Turn 90 degrees) and "5만큼 움직이기" (Move 5 units).
- When Key Pressed (Down):** A yellow block labeled "아래쪽 화살표 키를 눌렀을 때" (When Down Arrow Key is Pressed) triggers:
  - Turn:** A blue "회전 방식을 회전하기 (으)로 정하기" (Set Turn Method to Turn) block.
  - Direction and Movement:** Blue blocks for "180도 방향 보기" (Turn 180 degrees) and "5만큼 움직이기" (Move 5 units).
- When Key Pressed (Left):** A yellow block labeled "왼쪽 화살표 키를 눌렀을 때" (When Left Arrow Key is Pressed) triggers:
  - Turn:** A blue "회전 방식을 왼쪽-오른쪽 (으)로 정하기" (Set Turn Method to Left-Right) block.
  - Direction and Movement:** Blue blocks for "-90도 방향 보기" (Turn -90 degrees) and "5만큼 움직이기" (Move 5 units).

## 5 <개가 고양이를 쫓게 하기>

### 4 개의 스크립트



교육용프로그래밍언어기초(스크래치)

Next

# 프로시저와 병렬 처리를 사용한 프로그램 구현

