# 창의 컴퓨팅 입문

## Week 13 : Code Literacy - Read & Write

# 목차

- 지난시간 리뷰
- 씨앗 읽기
- 씨앗 확장하기
- 리믹스
- 읽고 쓴다는 것의 의미 : 리터러시

# 지난시간 리뷰

- 어떤 신체 장치의 시작
- 좀 더 움직여보기
- 인터랙티브 바디 인터페이스
- 확장하기

# 씨앗 읽기

# 리터러시란?

- 읽고 쓰기
- 코드도 일정의 언어로서, 읽고 쓰기를 할 수 있다.
- 여러분은 이제 스크래치의 블록 언어를 읽고 쓸 수 있기에,
- 다른 사람의 의도나 프로그램의 동작에 대해 이해할 수 있습니다.

# 팀 빌딩

- 2명이 1팀!
- 노트북, 컴퓨팅 보드
- A4용지 1장, 필기도구

# 씨앗 읽기

- 6종류의 귀여운 프로젝트가 있어요.
- 규칙
  - 씨앗을 하나씩 실행해보고, 코드를 읽고, 동작을 이해해 봅시다.
  - 각 씨앗으로 어떤 작품을 만들 수 있을지 아이디어를 종이에 기록해 봅시다.

# 동작을 이해하기 위해서는,

- 코드 블록에 다양한 변화를 줘봅시다.
  - 블록을 빼거나 순서를 바꿔보거나,
  - 변수의 값을 바꿔 보거나,
  - 조건문의 조건을 바꿔봅시다.

# 함께 읽기!

- 가장 이해하기 어려운 프로젝트는 무엇인가요?
- 시간이 조금 걸리지만, 다같이 읽어 봅시다.

# 씨앗 확장하기

# 씨앗 확장하기

- 규칙
  - 마음에 드는 씨앗 1~2개를 정하고,
  - 새로운 프로젝트로 확장해 봅시다.
  - 그리고, 컴퓨팅 보드로 새로운 인터페이스를 추가해 봅시다.
    - 되도록, 이전에 사용하지 않은 재료를 사용해 주세요!

# 씨앗 나누기

- 규칙
  - 드라이버는 우리 팀의 씨앗을 소개합니다.
  - 내비게이터는 다른 팀의 씨앗을 살펴봅니다.
  - 내비게이터는 다틀 팀의 씨앗 중 우리 팀의 씨앗에 적용할 내용을 찾습니다.
  - 팀으로 돌아온 뒤, 가장 인상깊은 다른 팀의 씨앗을 기억해 주세요. (팀원 이름)

# 리믹스

# 확장에 또 확장하기

- 규칙
  - 새로운 씨앗을 추가하거나, 다른 팀의 씨앗을 섞어봅시다.
  - 참고한 씨앗 이름 또는 팀원의 이름을 프로젝트 문서에 기록해 주세요.
  - 다음의 구조가 각각 1개 이상 포함되도록 만들어 보세요.
    - 반복문, 난수(랜덤), 변수, 조건문 중 1개 이상
    - 배경, 묻고 기다리기, 리스트, 확장 기능 중 1개 이상

# 읽고 쓴다는 것의 의미 : 리터러시

# 디지털 시대의 리터러시란?

- 지금은 너무나 일상화된 (문자) 언어의 읽고 쓰기
- 고전적인 리터러시가 지금 시대에는 어떤 점이 달라졌을까?
- 오늘날 읽고 쓰는 정보는 (문자) 언어만 있을까?
- 내가 감각하고, 인지하고, 사유하는 공간은 어떻게 동작할까?
- 만약, 그 동작을 일으키는 원리를 읽고 쓸 수 있다면 어떻게 될까?

# 디지털 시대의 리터러시

- 프로그램 코드를 읽고 쓴다는 것은 어떤 의미일까?
- 다른 사람의 코드를 써도 될까?
- 다른 사람의 코드와 내가 작성한 코드가 섞인 결과물, 이 결과물은 누구의 것일까?

# 리믹스: 스크래치 저장소

# scratch.mit.edu

## 커뮤니티: 아이디어 (코드) 저장소

# 프로젝트 리믹스

# [활동] 다시, 코드 리터러시!

- 씨앗 스튜디오
- 규칙
  - 여러개의 씨앗 중에서 관심이 가는 씨앗 **2개**를 고르기
  - 각 씨앗별로 동작(코드) 이해하여 기록하기
    - 반복문은 어떻게 쓰였지?
    - 변수/난수는 어떤 용도로 사용되고 있을까?
    - 조건문은 어떤 논리를 나타낼까?
    - 어떻게 확장할 수 있을까?

# 함께 읽어보기

- 씨앗들을 함께 살펴봅시다.

# [활동] 리믹스 하기

- 씨앗 2개를 섞어 봅시다.
- 규칙
  - 첫 번째 씨앗을 리믹스하기
  - 두 번째 씨앗의 아이디어 또는 스프라이트 중 마음에 드는 것을 골라 가져오기
  - 단순히 추가하는 것이 아닌, 아이디어를 가져와서 다른 형태, 의미, 과정, 결과로 가져와 주세요.

# [활동] 리믹스 하기

- 규칙 (이어서)
  - 다음의 컴퓨팅 구조가 반드시 포함되도록 만듭니다.
    - 난수 / 변수(또는 리스트), 반복문 / 조건문
  - 프로젝트 문서 작성
  - CCL 추가하기

# 라이선스와 오픈소스

# 저작권? 라이선스?

- 저작권 : 창작자가 자기 창작물에 행사할 수 있는 '법적인 권리'
  - 카피라이트 copyright 로고로 표기
- 라이선스 : 창작물에 대한 '사용권'의 개념
  - 사용범위(포함할 콘텐츠 종류, 사용방식 등)를 정하는 것
  - 글, 그림, 음악, 소스코드 등 다양한 형태의 창작물에 부여할 수 있다.
- 자신이 만든 콘텐츠에 라이선스를 부여하는 것은,
  곧 '저작권을 행사 하는 것' 이다.

# 오픈소스

- 오픈소스란, '원시 코드'에 접근하여 라이선스의 범위 안에서 변형(가공)/공유(배포)할 수 있게 한 SW/HW의 사용권
- 오픈소스 소프트웨어 라이선스 종류
- 생활코딩 저작권
- 다양한 라이선스가 있으며, 자기가 라이선스를 직접 만들 수도 있다. (예: 비어웨어)

# 라이선스 부여하기

- 자신이 창작한 콘텐츠의 특성에 맞는 라이선스 부여하기
- CCL 라이선스 부여하기
- 스크래치 프로젝트 리믹스와 복사

# 디지털 시대의 공유

# 복붙 or 공유

- 거의 모든 컨텐츠가 ctrl+c / v 로 간단히 복제가 가능
  원본과 복제본의 차이를 구별 할 수 있을까?
- 하늘 아래 새로운 것은 없다.
  우리가 만드는 컨텐츠는 대부분 어느 누군가가 만들어 놓은
  토대 위에 만들어 가는 것.
  (문자에 대해 누군가가 저작권을 행사하지 않듯이)

## 거인의 어깨 위에 올라서는 행운

# 복붙 or 공유

- 그렇다면 오픈소스와 상용 소프트웨어는 서로 적대적일까?
- 현명하게 디지털 콘텐츠를 공유하는 방법은?
  - 사진을 무료로 공개하고 나를 알리자. unsplash.com

# 기말놀이

- 중간놀이와 동일하게 진행됩니다.
- 수업시간에 여러분들이 만들었던 코드들을 참고로 할 수 있습니다.
- 평가는 마찬가지로 동료 간 평가와 교수자 평가가 이뤄집니다.
- 가벼운 마음으로 평소 수업시간 하던 대로 하면 됩니다.

# Thanks! 🎉

수업 관련하여 궁금한 사항은
이메일, 수톡, 이클래스 쪽지 등으로 연락주세요.