2023 코딩거점학교
– p5.js –

디지털 대전환 시대로의 돌입

© World Economic Forum

마이크로소프트 창업자 빌 게이츠는 인공지능은 적은 노동력으로
더 많은 생산과 서비스를 가능하게 하는 최신 기술이며,
인간의 친구가 될 수 있다고 언급하며 제대로 된 준비가 필요함을 지적[1)]

1) 연합뉴스(2018). 빌 게이츠 'AI 종말론' 일축…"AI는 인간의 친구", 연합뉴스(2018.02.17.).

© World Economic Forum

세계적 역사학자인 유발 하라리는 디지털 기술의 빠른 발전은
일자리, 자유 평등과 같은 인간 삶의 기본 구조를 위협할 수도 있음을
경고하며 적극적인 대응이 필요함을 지적[2)]

2) 연합뉴스(2017). 유발 하라리 "무지·공포 아닌 이해에 기반한 AI 규제 필요". 연합뉴스(2017.07.13.).

© Yuichi Sakuraba

미래학자 레이 커즈와일은 디지털 기술이 인간의 능력을 넘는 순간을
'특이점(singularity)'이라고 부르며, 미래 기술 빠른 변화의 속도와
영향력으로 인해 사회의 모든 개념에 극적인 변화가 일어날 것을 예상,
이에 대한 체계적인 준비가 필요함을 지적[3)]

3) 동아사이언스(2020). 미래학자 레이 커즈와일 "2030년 사람 뇌와 AI 잇는 인터페이스 나온다". 동아사이언스(2020.09.18.).

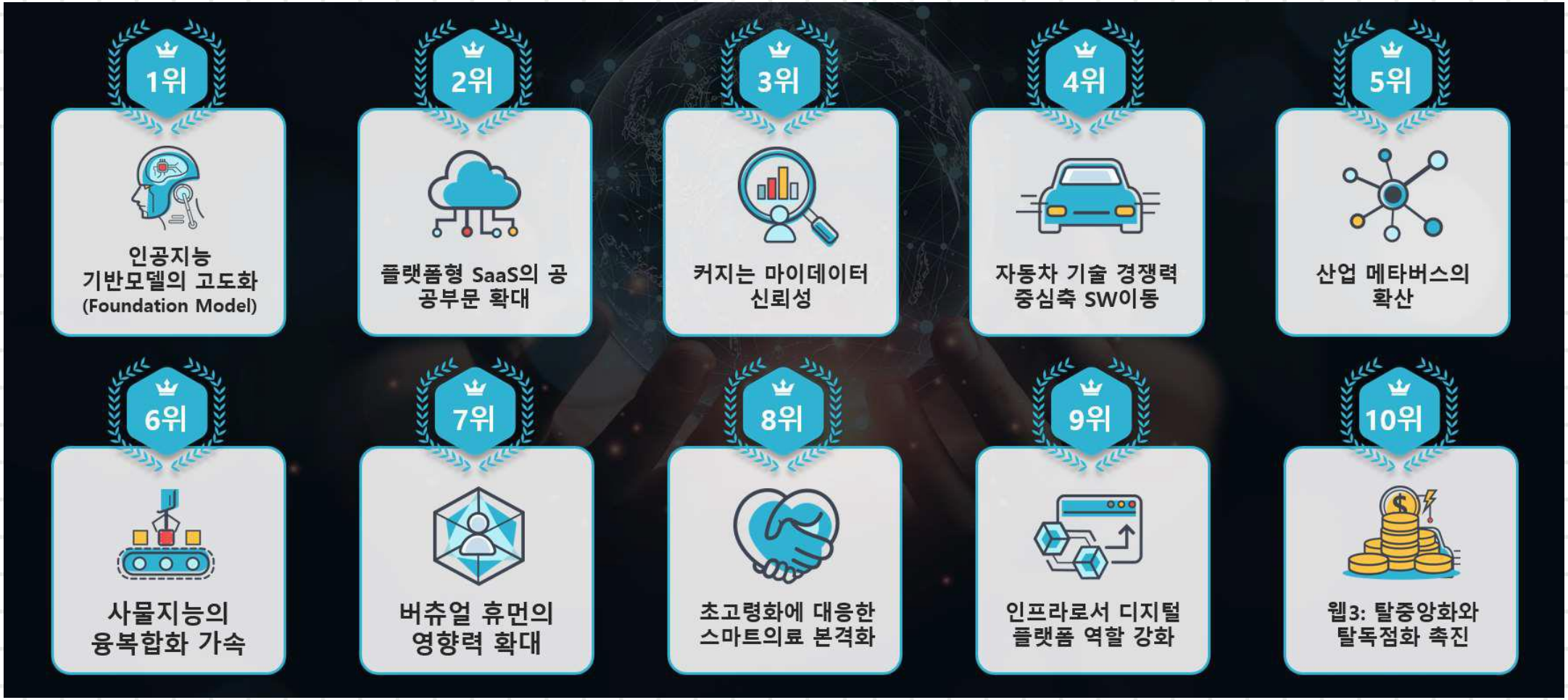
* 참고: 구글 검색(검색일, 2021.05.17)

출처: 소프트웨어정책연구소(2021)

디지털 대전환 시대로의 돌입

출처: 소프트웨어정책연구소(2021)

2023년 SW산업 10대 이슈전망



출처: 소프트웨어정책연구소(2023)

디지털 리터리시

디지털리터러시는 디지털 사회 구성원으로 살아가기 위한 기초 소양으로서, ICT 도구 활용할 수 있는 기초 능력(ICT 활용)에서 컴퓨팅파워(추상화와 자동화)를 활용한 문제해결력 능력(컴퓨팅사고력)까지 포함

출처: 소프트웨어정책연구소(2023)

비버챌린지란?

<컴퓨팅 사고력을 바탕으로 하는 정보과학(Informatics)를 경험할 수 있는 전 세계인의 축제!>

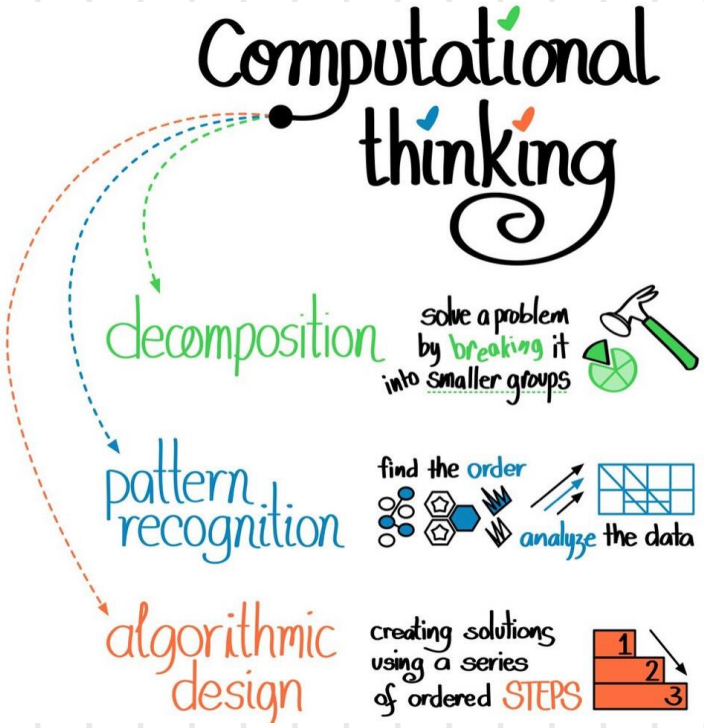
비버챌린지는 모든 연령대의 학생에게 정보과학(정보, SW) 및 컴퓨팅사고력을 알려주기 위한 국제 행사입니다.

비버챌린지는 특별한 사전 지식 없이도 누구나 도전할 수 있는 과제로 이루어져있습니다. 참가자는 정보과학을 지도할 수 있는 교사와 함께 과제를 해결합니다.

귀여운 비버가 등장하는 즐거운 정보과학의 축제로 여러분을 초대합니다!

비버챌린지란?

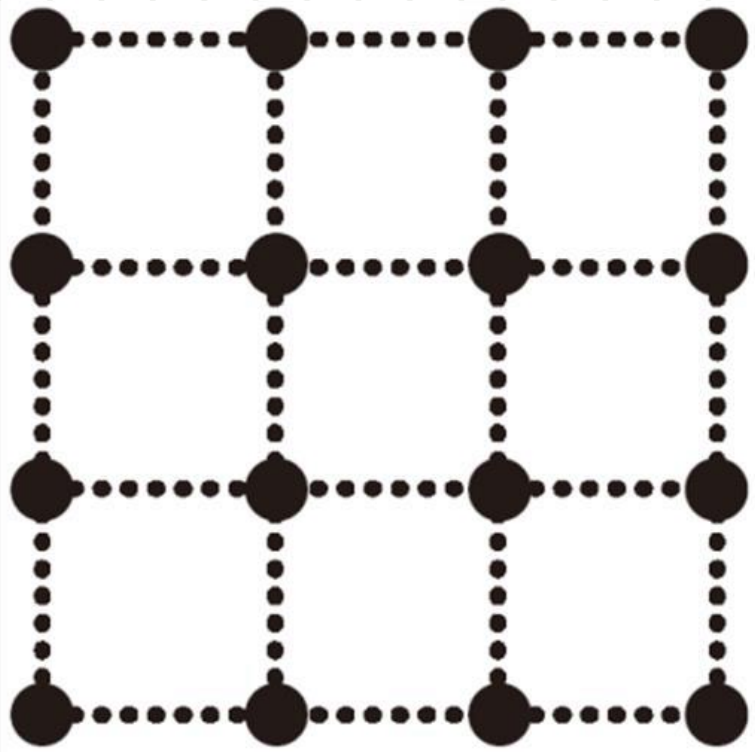
'컴퓨터과학의 기본 개념을 토대로 문제를 해결하고, 시스템을 설계하고, 인간의 행동을 이해하는 능력(Wing, 2006)'



컴퓨팅사고력은 문제를 작은 단위로 분해(Decomposition)하고 핵심적인 패턴을 인식(Pattern Recognition)한 뒤, 알고리즘 설계(Algorithmic design)를 하는 추상화 단계를 거쳐 프로그래밍을 통한 자동화(Automation) 단계로 이어지는 개념으로 설명 가능

정사각형의 수 구하기

4개의 점을 연결하여 만들 수 있는 서로 다른 정사각형의 수를 구하시오.

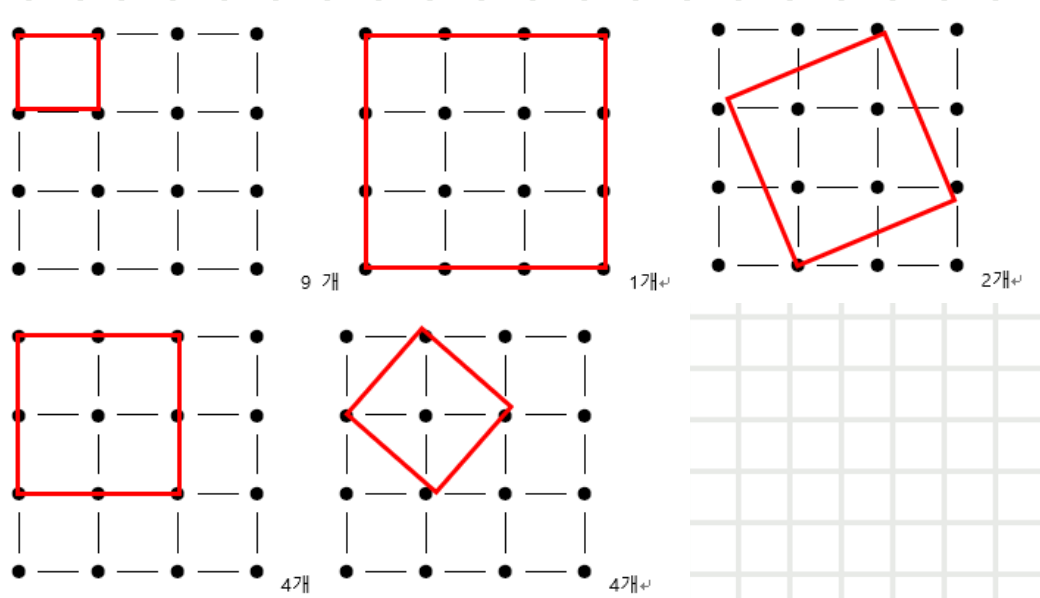


정사각형의 수 구하기

4개의 점을 연결하여 만들 수 있는 서로 다른 정사각형의 수를 구하시오.

(문제 분해)
변의 길이로 접근하기

1) 길이: 1 -> 9
2) 길이: 루트 2 -> 4
3) 길이: 2 -> 4
4) 길이: 루트 5 -> 2
5) 길이: 3 -> 1

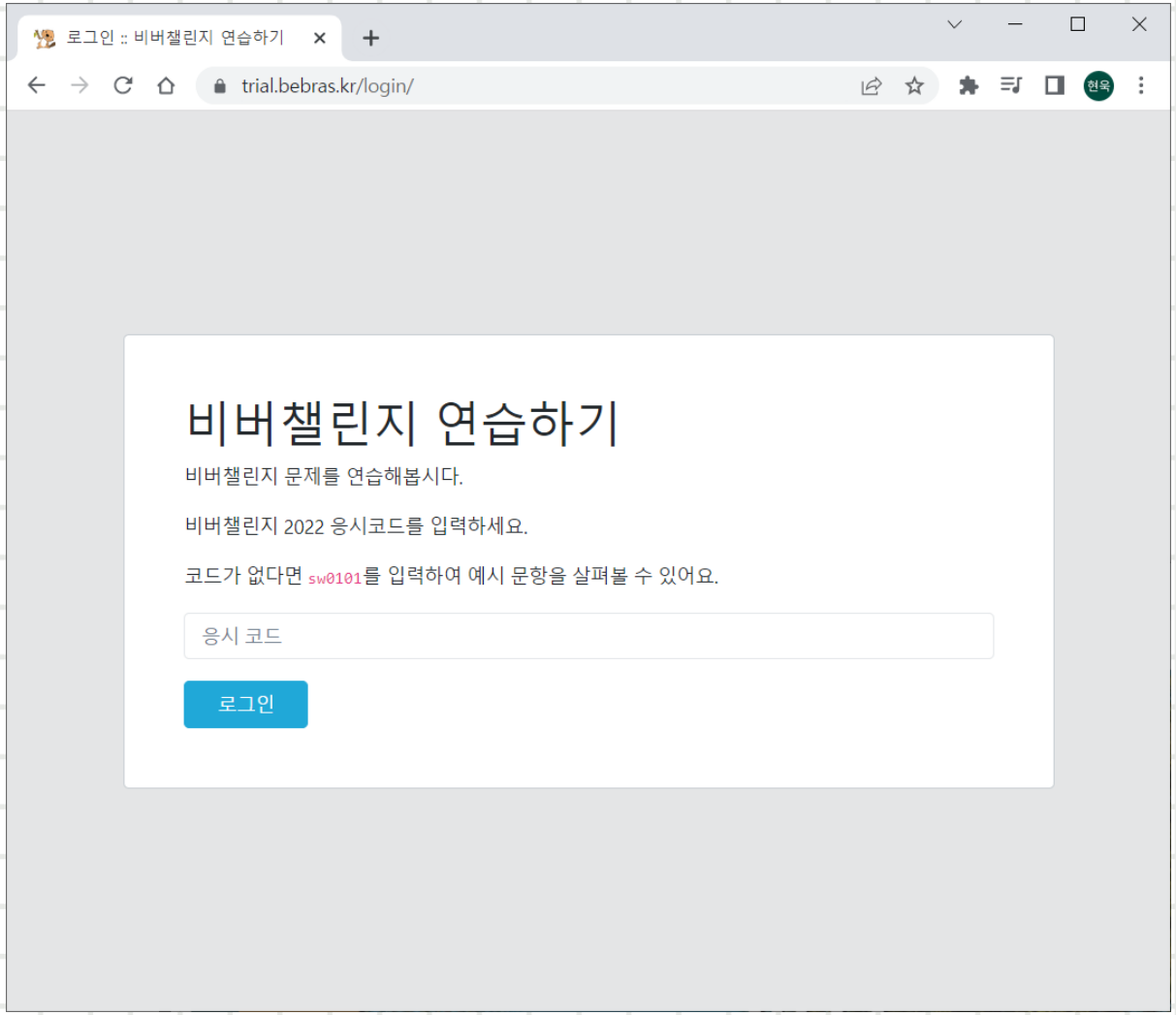


답: 20개

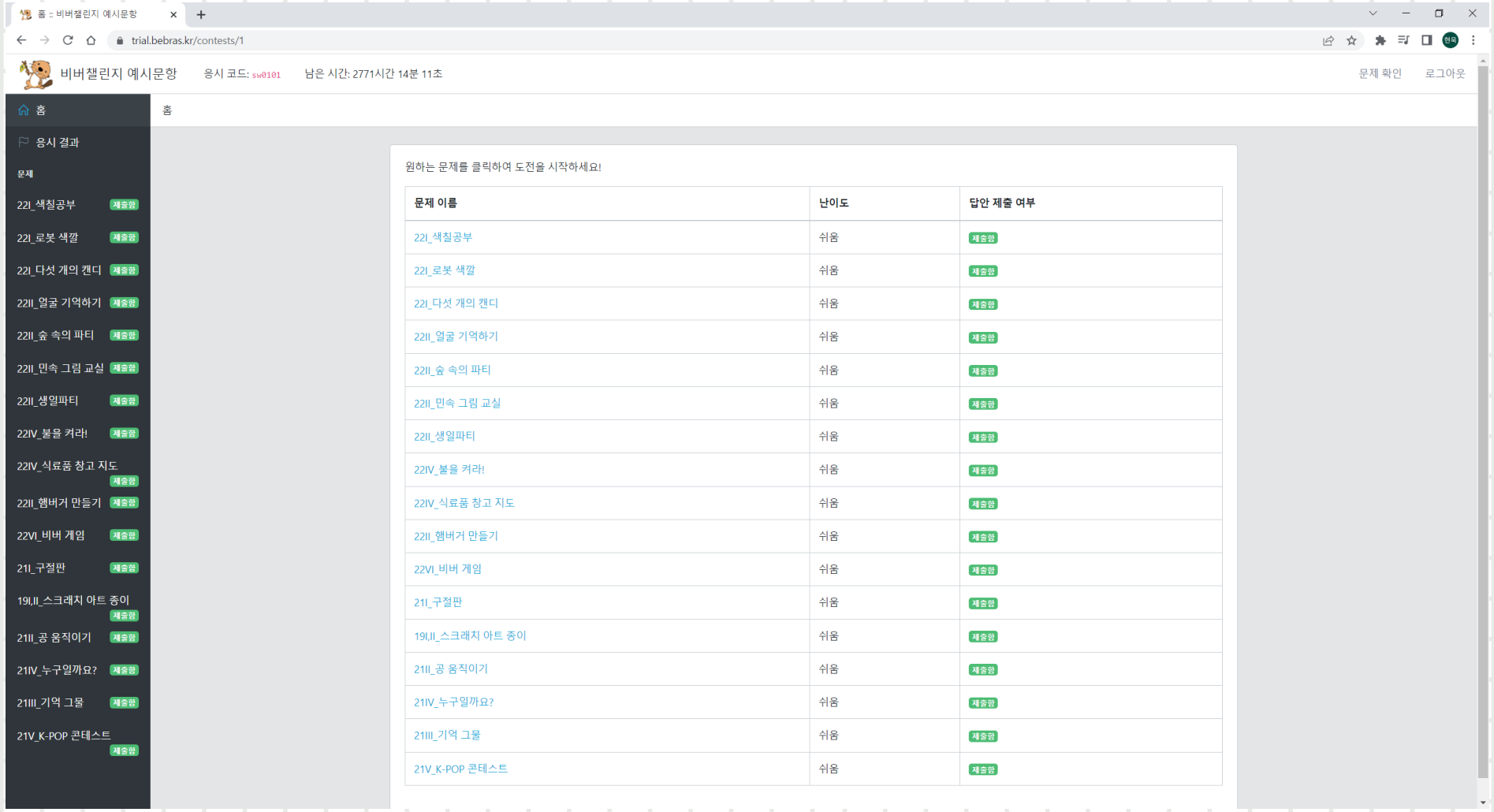
도전! 비버챌린지

도전하기

1. 비버챌린지 홈페이지 접속
   https://www.bebras.kr/
2. 메뉴 '참여하기' - '연습하기' 접속
   https://trial.bebras.kr/login/
3. 코드 sw0101 입력



도전하기



출처: 한국비버챌린지 연습하기 문항

도전하기

문제의 배경

비버가 색칠 공부를 하려고 한다. 오른쪽 3가지 색 중에 하나를 클릭하고 왼쪽 그림을 클릭하면 클릭한 칸이 색칠된다.

문제/도전

이웃한 칸이 서로 다른 색이 되도록 색칠하여 그림을 완성하시오.

모든 칸을 색칠해야 합니다.

다시 하기

출처: 한국비버챌린지 연습하기 문항

도전하기

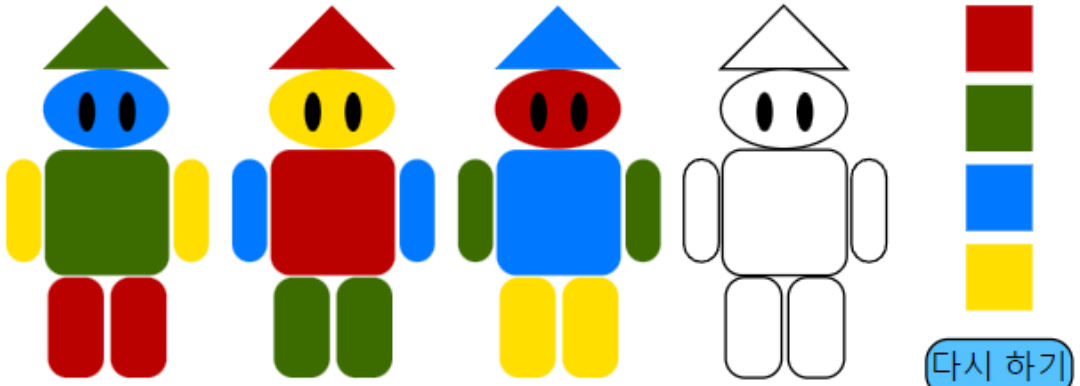
문제의 배경

타워은 4가지 색(빨강, 파랑, 초록, 노랑) 로봇 네 대를 가지고 있다.
각 로봇의 모자, 얼굴, 팔, 몸, 다리 색깔이 모두 달라 구분된다.

문제/도전

로봇 3대의 색깔이 다음과 같을 때 나머지 로봇 1대를 색칠하시오.

모든 칸을 색칠해야 합니다.



다시 하기

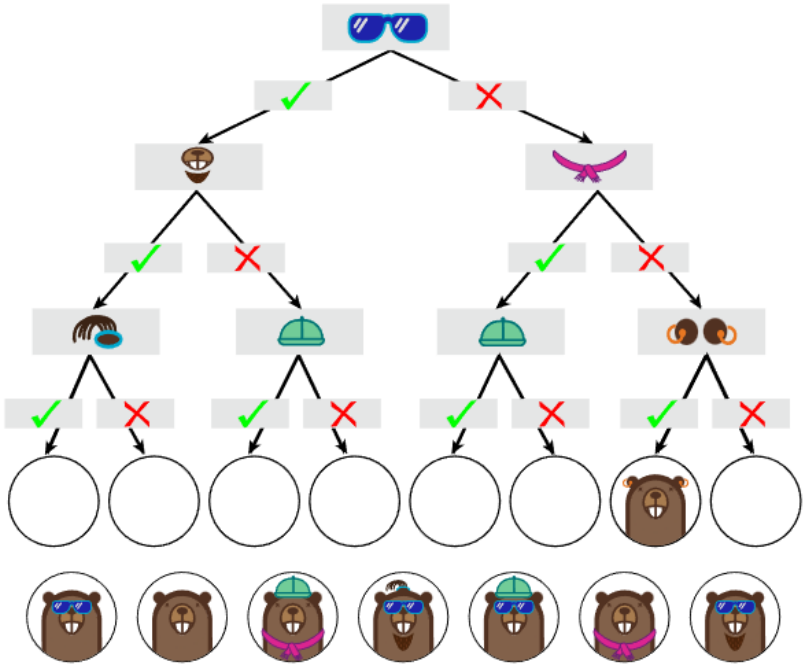
도전하기

문제의 배경

새 친구들이 전학 왔다. 전학 온 친구를 알아보기 위한 사진은 다음과 같이 정리 되어있다.

문제/도전

친구의 사진을 올바른 위치에 놓으시오.



친구의 사진을 마우스로 끌어서 놓아 주세요.

다시하기

출처: 한국비버챌린지 연습하기 문항

도전하기

문제의 배경

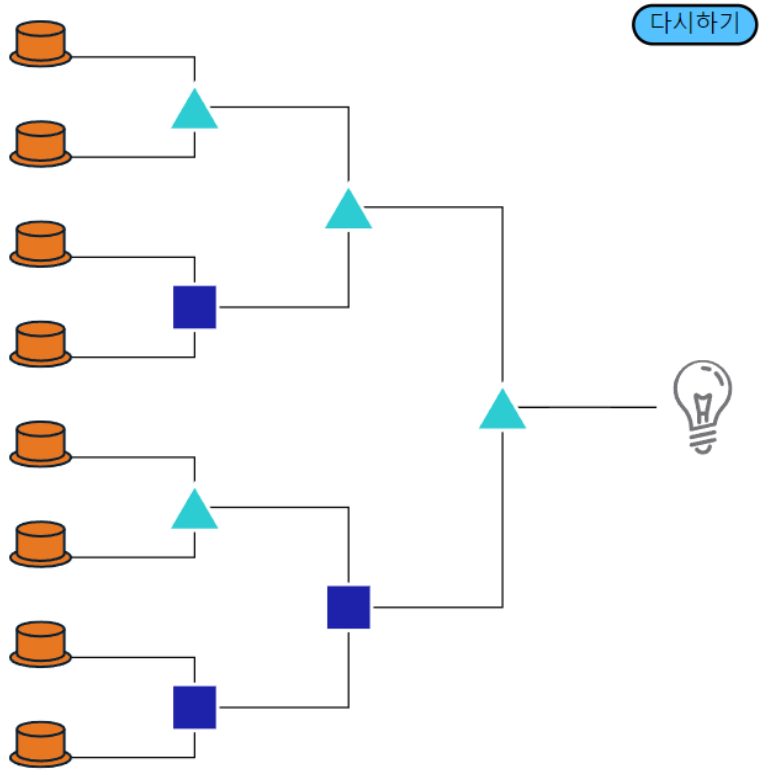
비버 소피아(Sofia)와 친구들이 오락 게임장에서 게임을 하고 있다.
소피아와 친구들은 버튼( )이 8개 있는 'Light ON'이라는 게임을 해보기로 했다.
버튼 8개와 연결된 전기선은 삼각형( ) 또는 사각형( ) 모양의 상자를 통해 최종적으로 전구( )와 연결된

- 삼각형( ) 상자는 왼쪽 2개 모두 전기가 연결되어야, 전구 방향으로 전기가 통한다.

- 사각형( ) 상자는 왼쪽 2개 중 1개만 전기가 연결되어야, 전구 방향으로 전기가 통한다.

문제/도전

전구에 불( )이 켜지도록 버튼을 누르시오.



출처: 한국비버챌린지 연습하기 문항

도전하기

문제의 배경

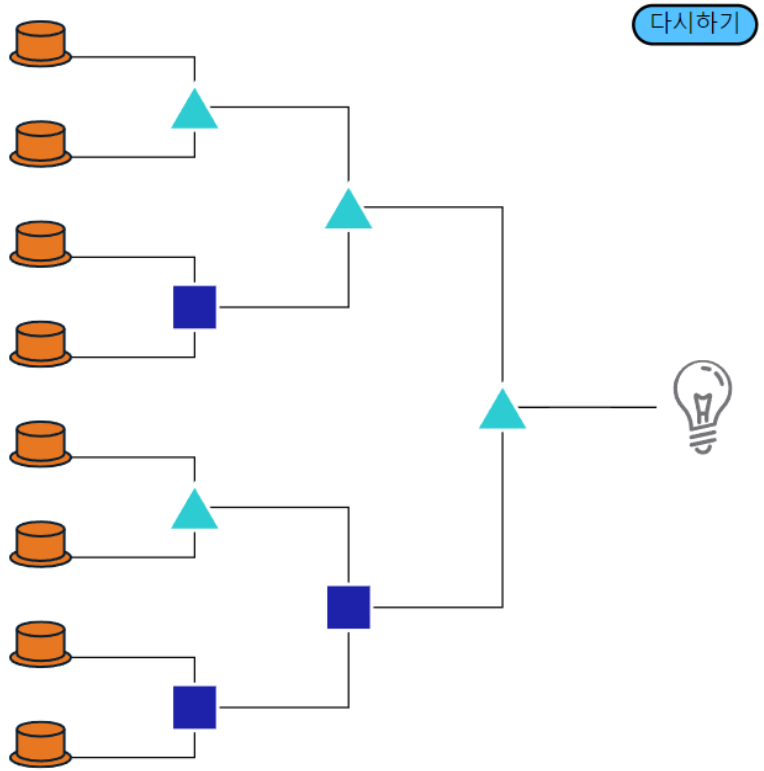
비버 소피아(Sofia)와 친구들이 오락 게임장에서 게임을 하고 있다.
소피아와 친구들은 버튼( )이 8개 있는 'Light ON'이라는 게임을 해보기로 했다.
버튼 8개와 연결된 전기선은 삼각형( ) 또는 사각형( ) 모양의 상자를 통해 최종적으로 전구( )와 연결된

- 삼각형( ) 상자는 왼쪽 2개 모두 전기가 연결되어야, 전구 방향으로 전기가 통한다.

- 사각형( ) 상자는 왼쪽 2개 중 1개만 전기가 연결되어야, 전구 방향으로 전기가 통한다.

문제/도전

전구에 불( )이 켜지도록 버튼을 누르시오.



출처: 한국비버챌린지 연습하기 문항

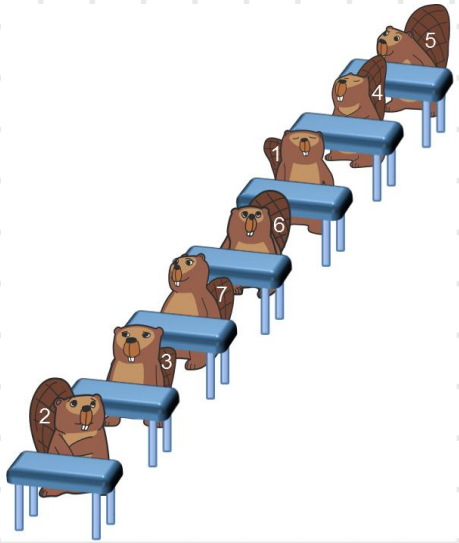
도전하기(그룹Ⅵ_비버챌린지2021)

문제의 배경

비버 학교의 한 교실에는 7명의 비버가 있고, 각 비버에는 숫자가 적힌 표시가 있다. 그들은 일렬로 한 줄로 앉아 있다.

처음에는 그림과 같이 비버가 무작위로 앉아 있다. 비버반 선생님은 앞자리에 1번부터 뒷자리의 7번 순으로 자리를 재배치하려고 한다.

자리 재배치는 두 비버 간의 자리 교환을 통해서만 가능하다.



문제/도전

비버선생님이 원하는 대로 1번부터 7번까지 자리를 배치하기 위해서는 최소한 몇 번의 교환을 해야 할까?

- ○ 3번
- ○ 4번
- ○ 5번
- ○ 6번

출처: 한국비버챌린지 2021 그룹Ⅵ 문항

도전하기(그룹Ⅵ_비버챌린지2021)

문제의 배경

다음의 패턴은 복잡해 보이지만 사실 동일한 타일 1장을 이어붙인 것이다. 이와 같이 서로 인접한 타일의 방향과 관계 없이 여러 방향으로 서로 연결되어 패턴을 만들어내는 타일을 트루셰 타일이라고 한다.

문제/도전

트루셰 타일이 아닌 것은 무엇인가?

○ A)
○ B)
○ C)
○ D)

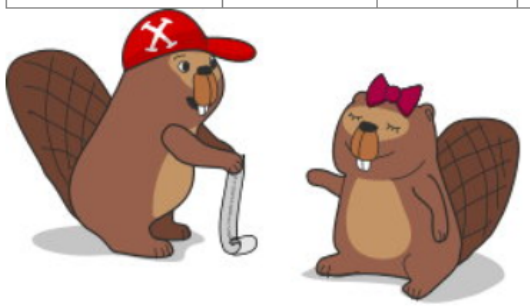
출처: 한국비버챌린지 2021 그룹Ⅵ 문항

도전하기(그룹Ⅵ_비버챌린지2021)

문제의 배경

비버 자비에르(Xavier)는 이진 코드로 메시지를 작성하는 중이다. 아래 표에는 각 문자와 변환될 코드가 적혀있다. T와 E처럼 자주 사용하는 문자는 코드 길이가 비교적 짧다.

문자	T	E	A	K	C	R
코드	1	00	0010	0110	1010	1110



비버 자비에르는 비버 이본(Yvonne)에게 다음과 같은 코드를 보냈다.

1001001100010100010111000

비버 이본은 이 메시지의 끝 글자가 E임을 알고 있다.

문제/도전

비버 자비에르가 적은 메시지의 의미는?(글자의 형식으로 작성하시오.)

정답 입력:

출처: 한국비버챌린지 2021 그룹Ⅵ 문항

p5.js

윈도우 관련 유용한 단축키

- 화면 분할은 'Win + 좌우방향키'
- 캡쳐는 'Win + Shift + s' 또는 'PrtSc'
- 전체 선택은 'Ctrl + a'
- 복사는 'Ctrl + c'
- 잘라내기는 'Ctrl + x'
- 붙여넣기는 'Ctrl + v'
- 작업 취소는 ' Ctrl + z '
- 작업 복구는 'Ctrl + y' 또는 'Ctrl + Shift + z'
- 수정모드/삽입모드는 'insert'

프로세싱이란?

프로세싱은 유연한 소프트웨어 스케치북이자
시각 예술의 맥락 내에서 코딩하는 방법을
배우는 언어

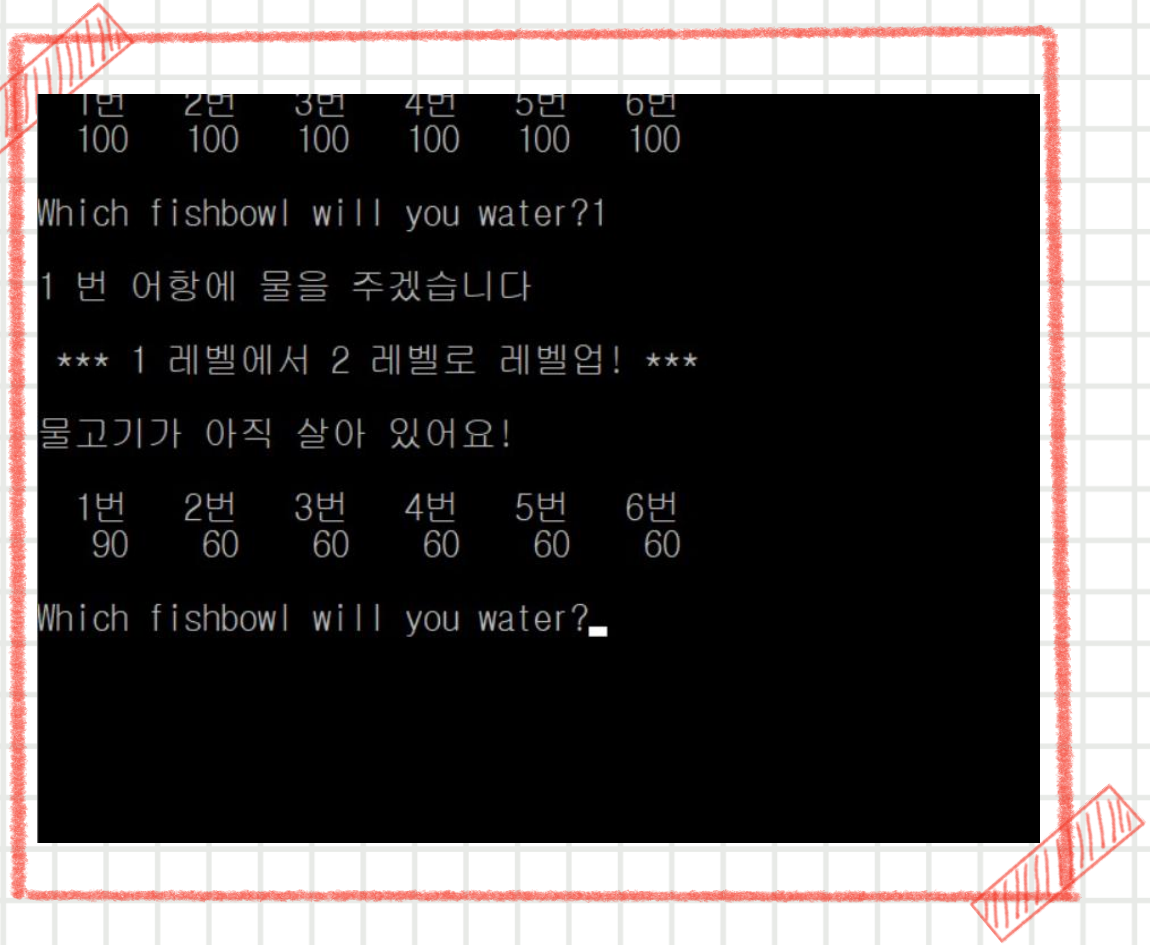
(JAVA 기반이나 아두이노처럼 간소화된 문법사용)

프로세싱 특징

C언어 등 기본적인 텍스트 기반의 프로그래밍에 비해 시각적인 효과로 흥미로움.

프로젝트를 진행하는데 있어서도 시각적인 효과로 인해 만족도가 높은 프로그램을 제작할 수 있음.

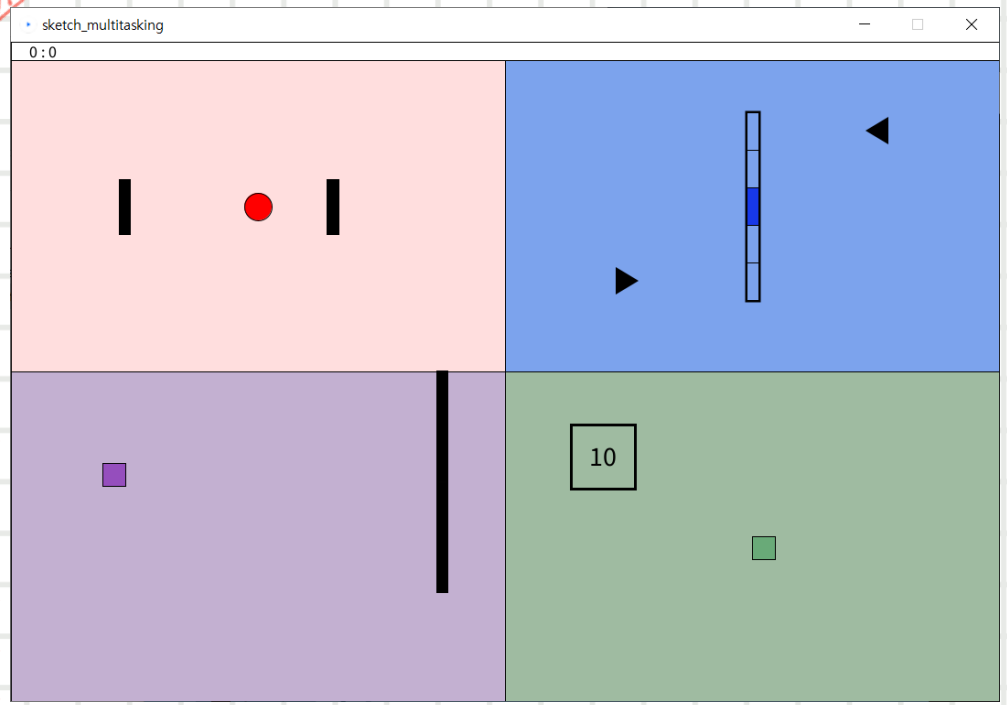
C언어 프로젝트



C언어

- 초급자는 텍스트 기반의 프로젝트 진행

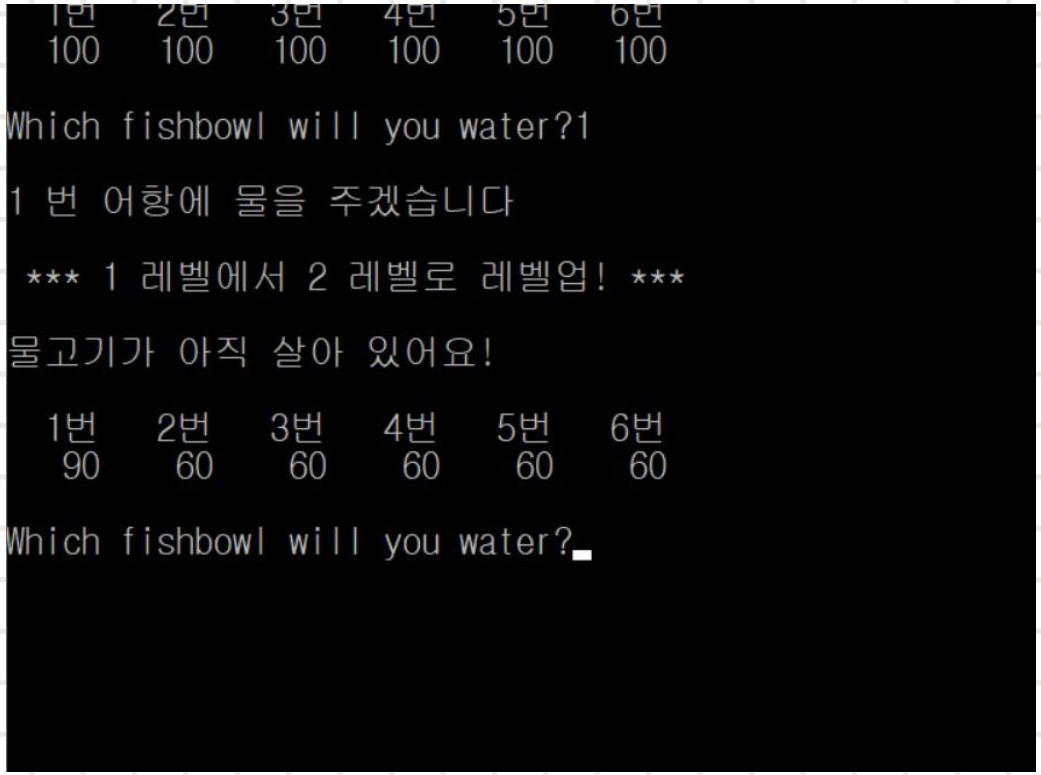
프로세싱 프로젝트

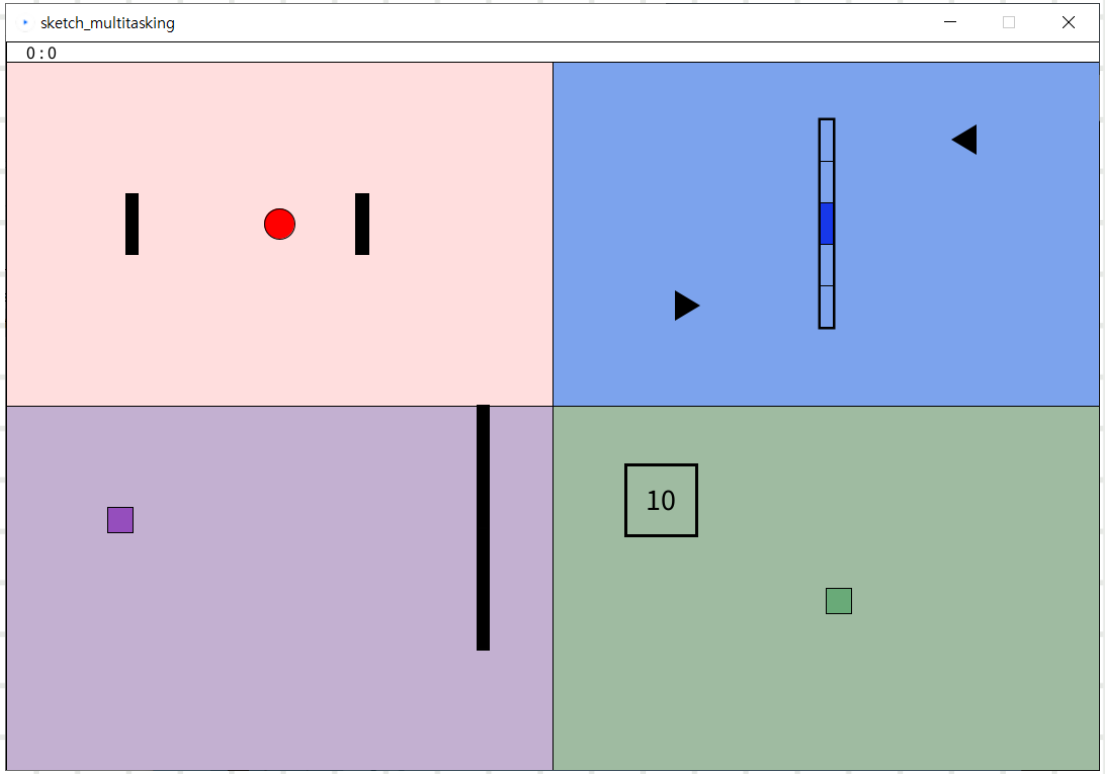


프로세싱

- 초급자도 그래픽 기반의 프로젝트 진행

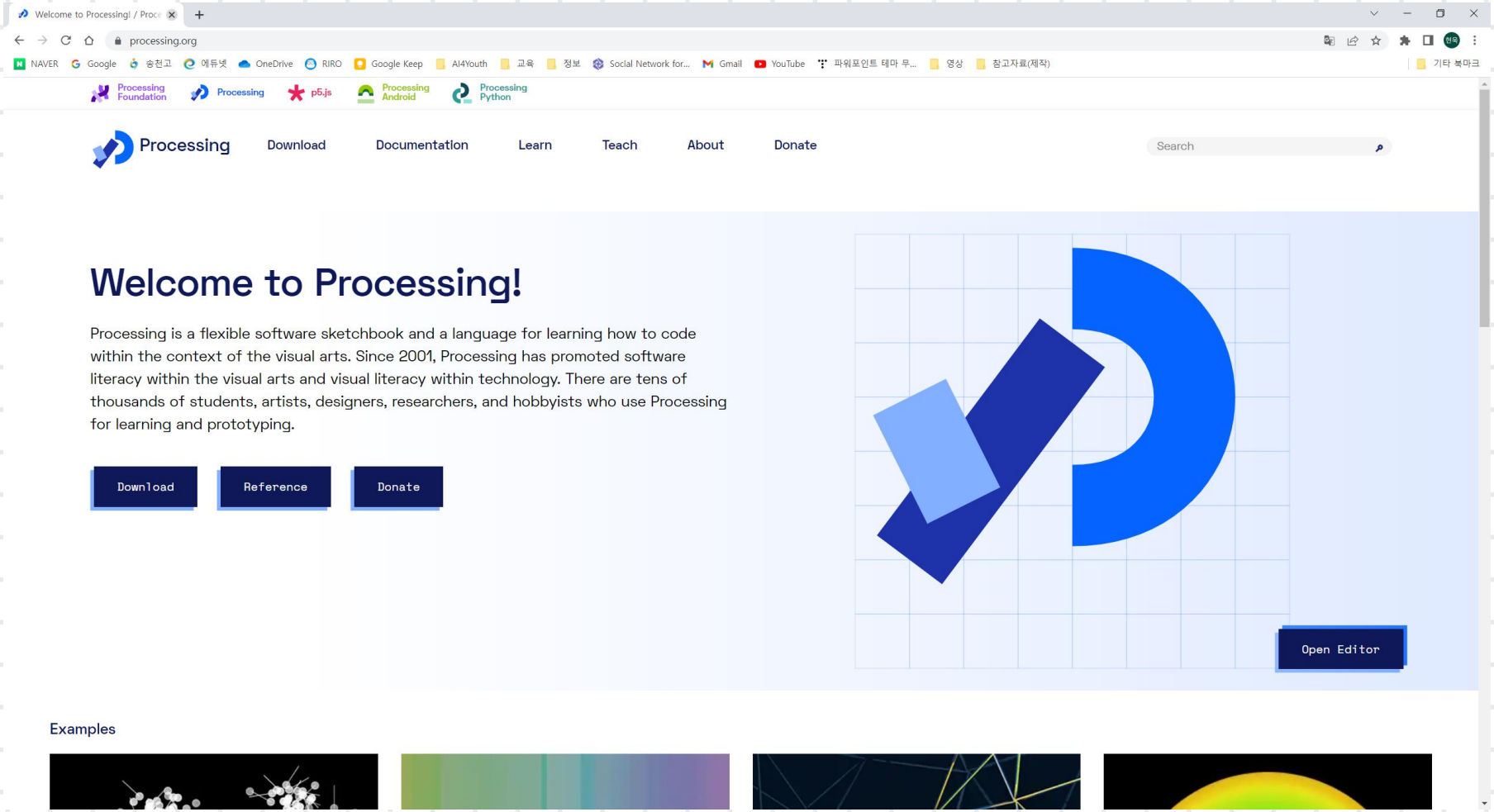
프로세싱 특징



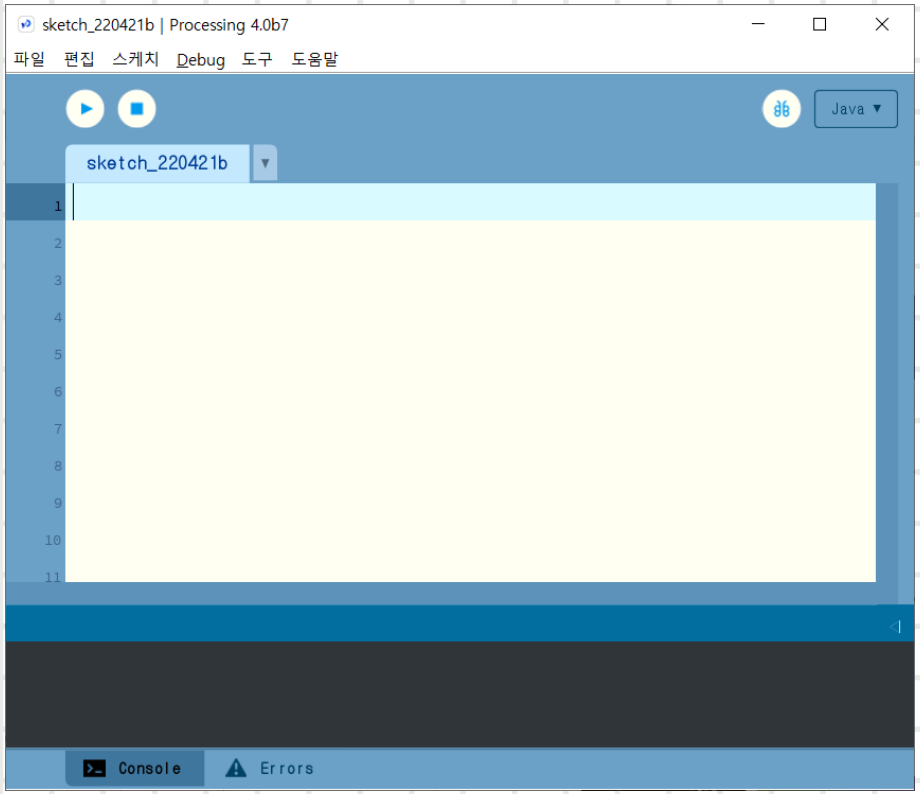


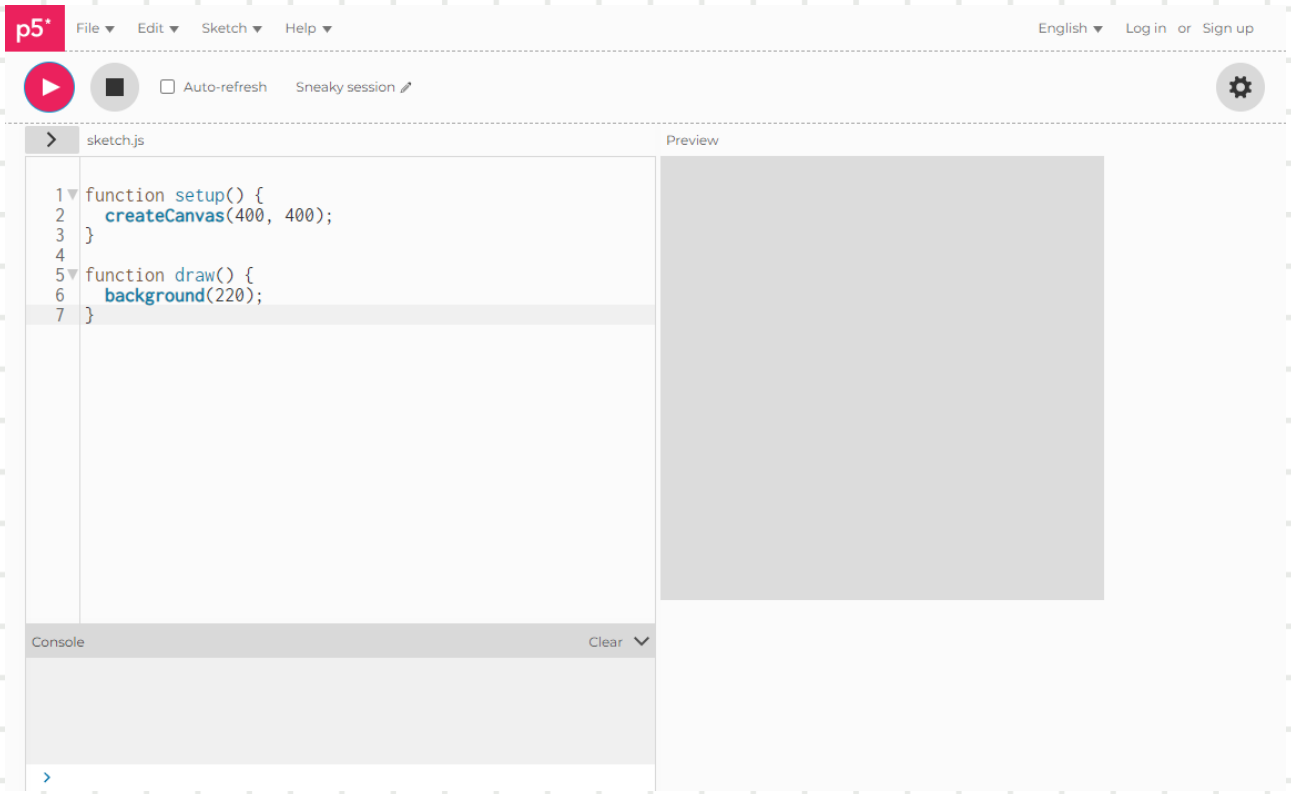
- C언어 프로젝트
- 프로세싱 프로젝트

프로세싱 접속


Welcome to Processing! / Proce
processing.org
NAVER
Google
송천고
에듀넷
OneDrive
RIRO
Google Keep
AI4Youth
교육
정보
Social Network for...
Gmail
YouTube
파워포인트 테마 무...
영상
참고자료(제작)
기타 북마크
Processing Foundation
Processing
p5.js
Processing Android
Processing Python
Processing
Download
Documentation
Learn
Teach
About
Donate
Search
Welcome to Processing!
Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.
Download
Reference
Donate
Open Editor
Examples

프로세싱 특징

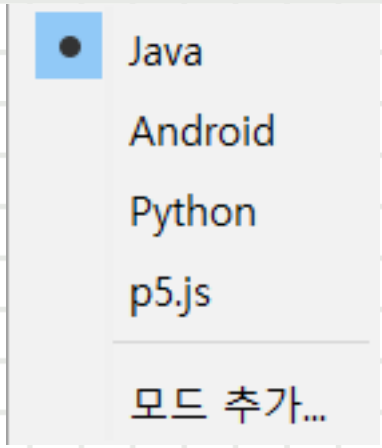




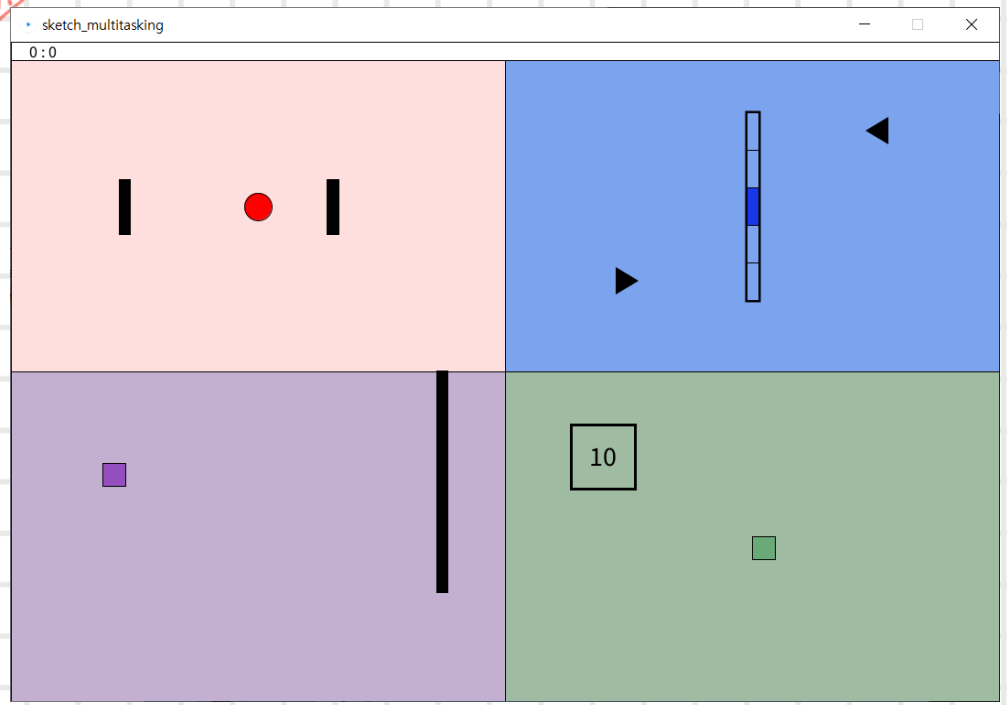
- IDE(JAVA, Python, JS 등)
- Web(p5js전용)
  https://editor.p5js.org/

프로세싱으로 사용 가능한 언어 종류

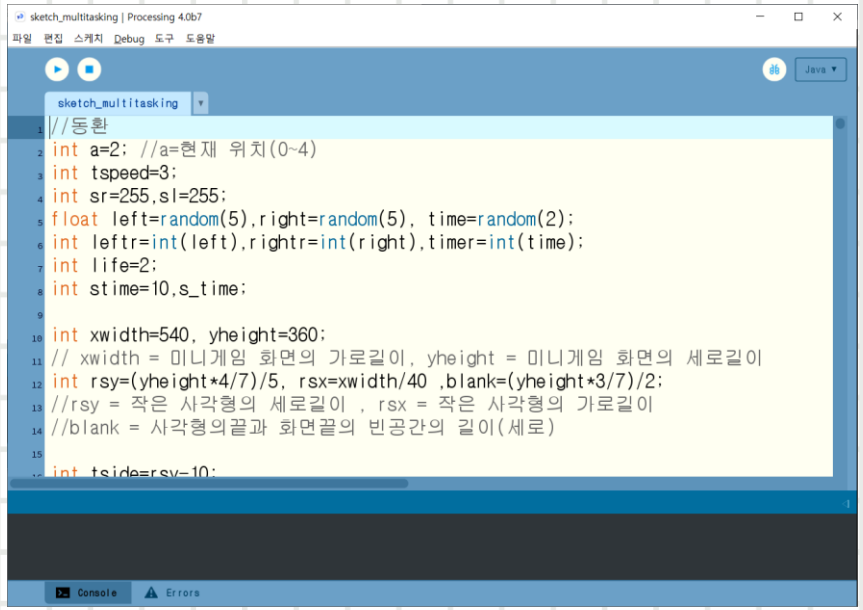
- ▷ 자바 (IDE)
- ▷ 파이썬 (IDE)
- ▷ 자바스크립트 (IDE, Web) - p5js
- ▷ 안드로이드 앱 생성(IDE, APDE)



프로세싱 프로젝트

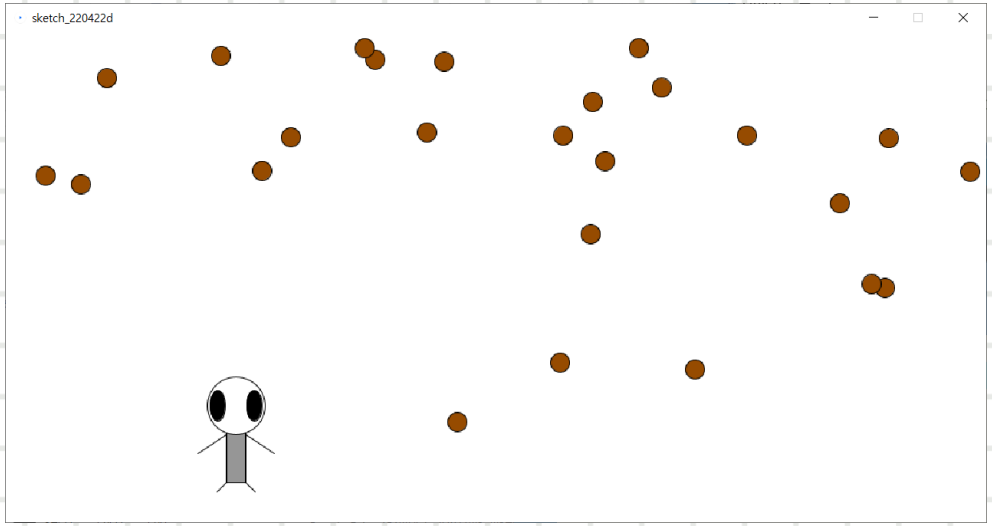


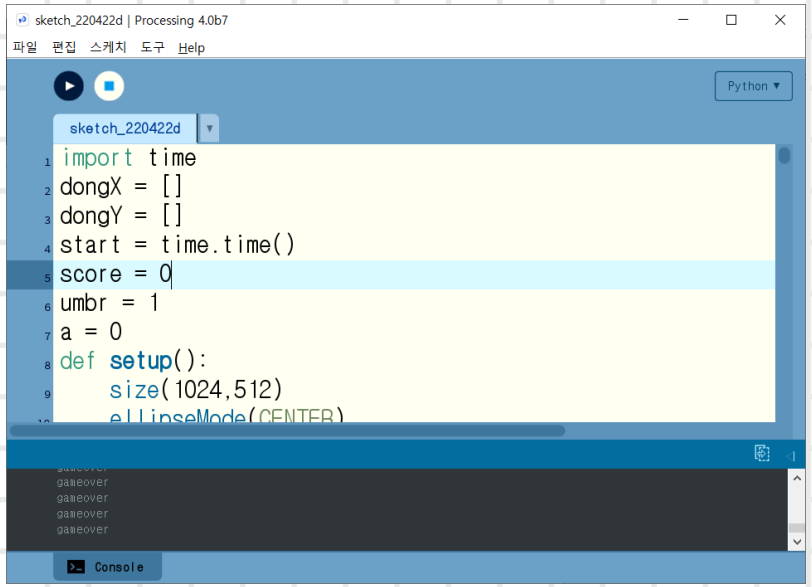
학생작품1(2학년 , 2021)



프로세싱 프로젝트

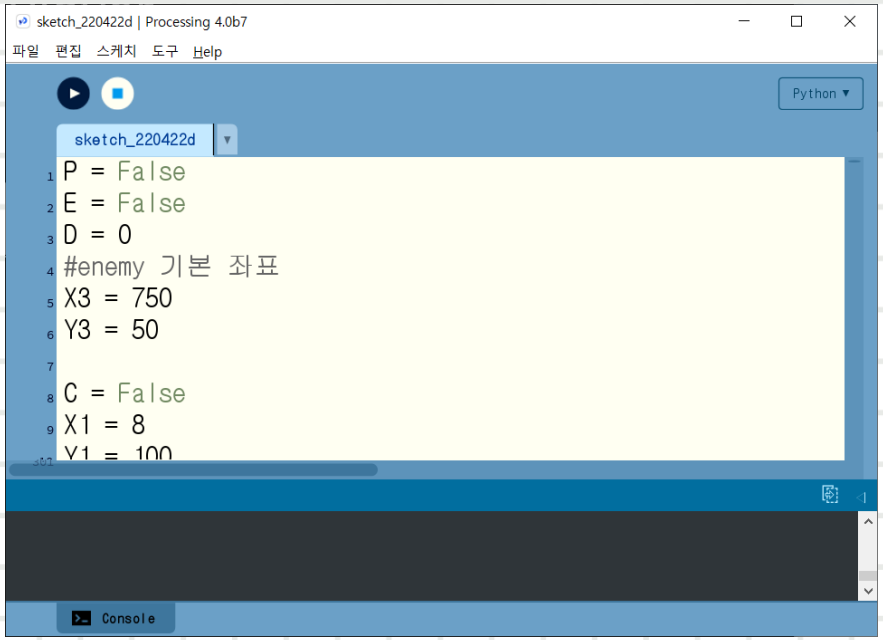
학생작품2(1학년 , 2021)



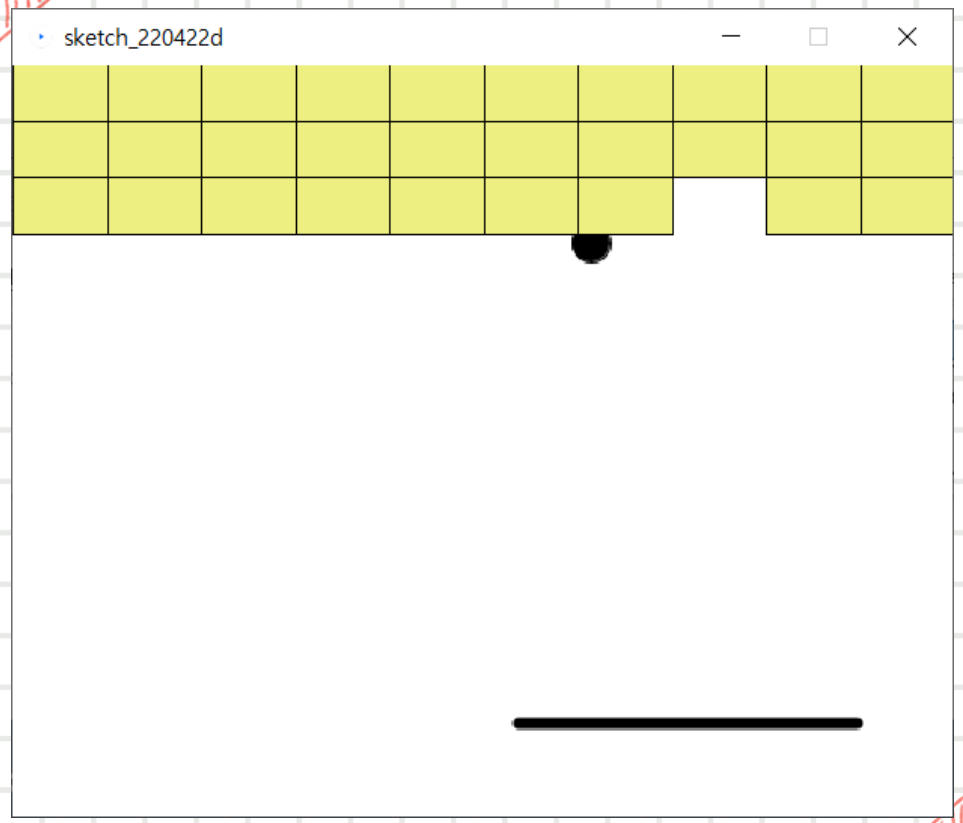


프로세싱 프로젝트

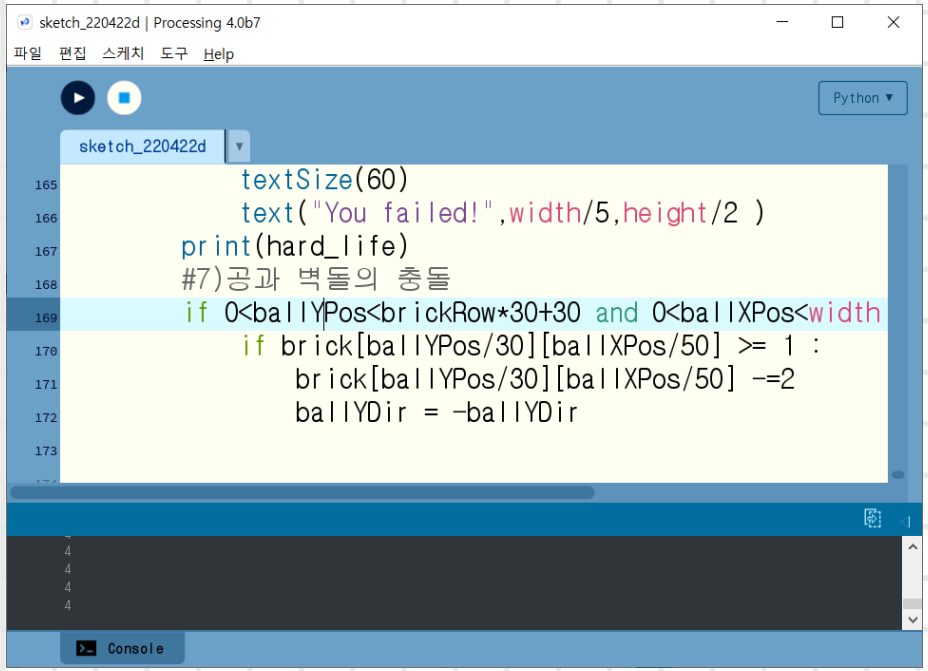
학생작품3(1학년 , 2021)



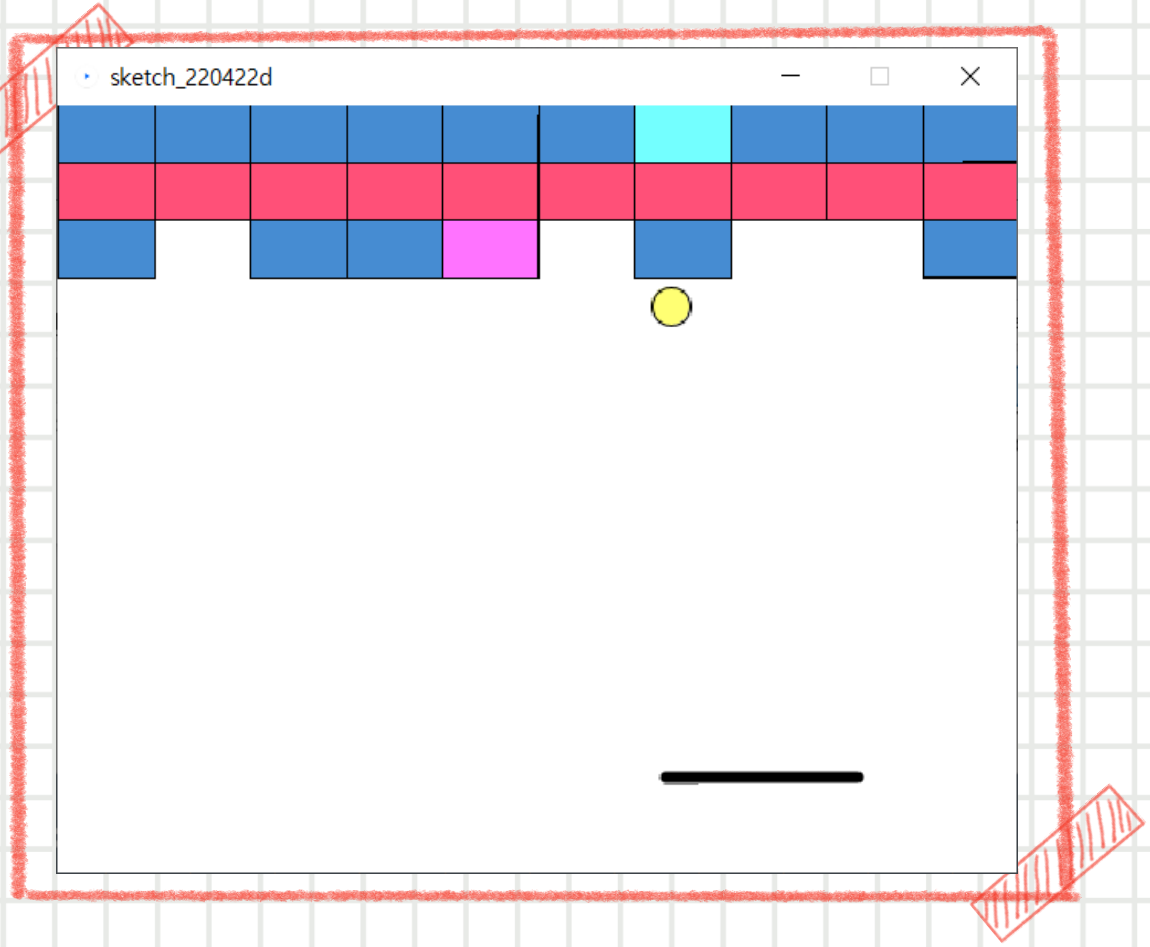
프로세싱 프로젝트



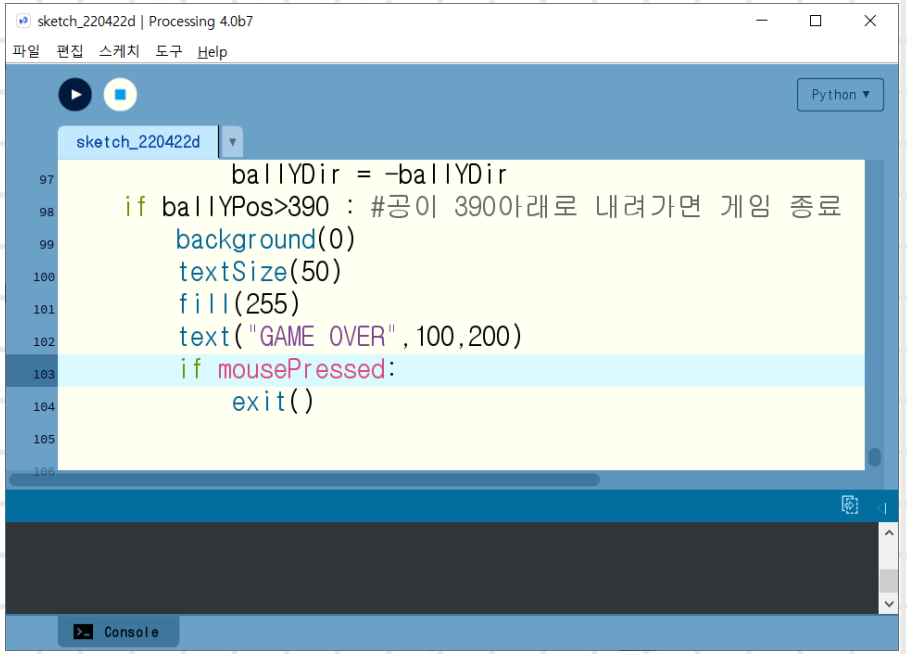
학생작품4(1학년 , 2021)



프로세싱 프로젝트

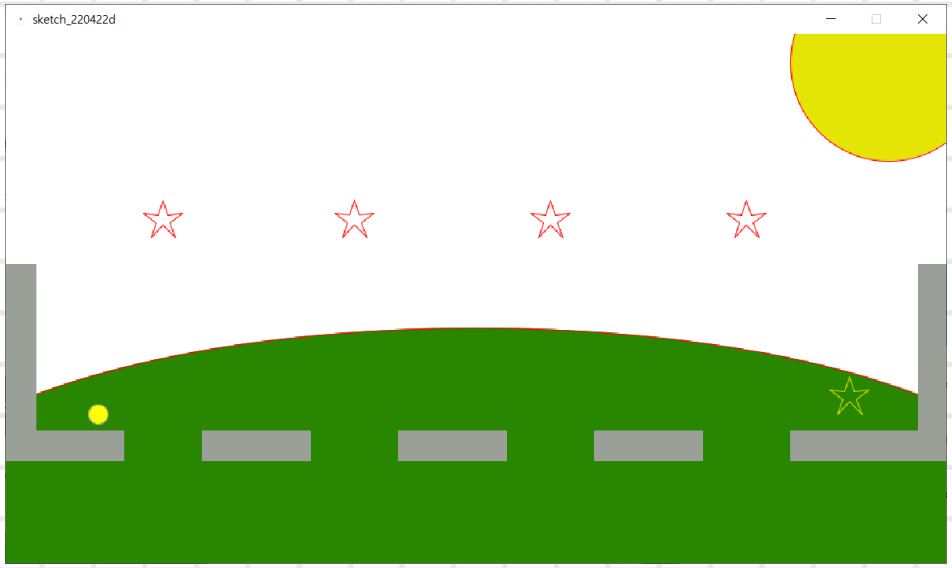


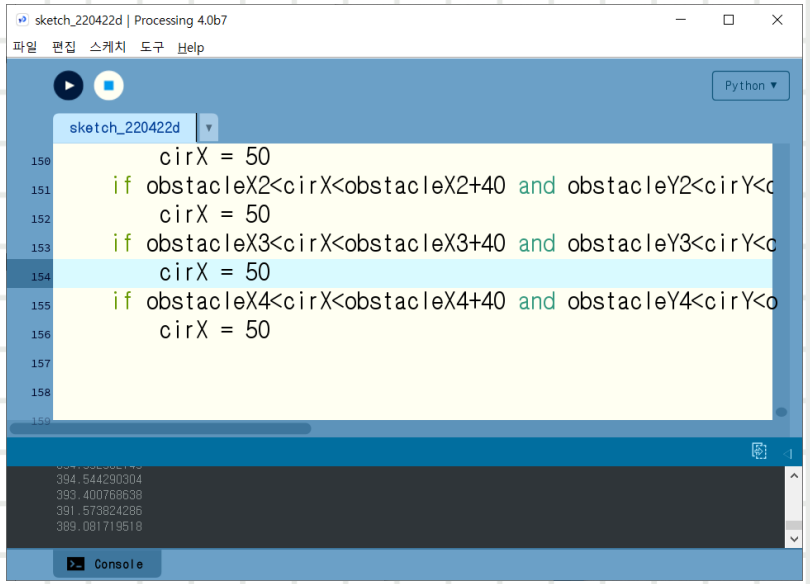
학생작품5(1학년 , 2021)



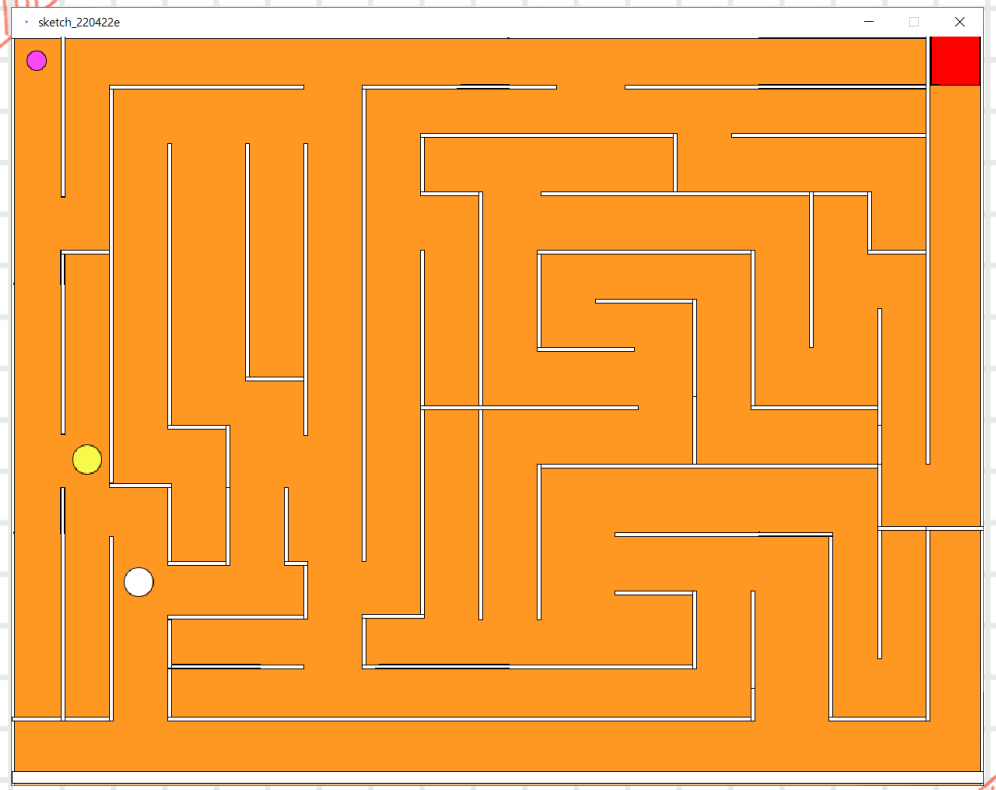
프로세싱 프로젝트

학생작품6(1학년 , 2021)

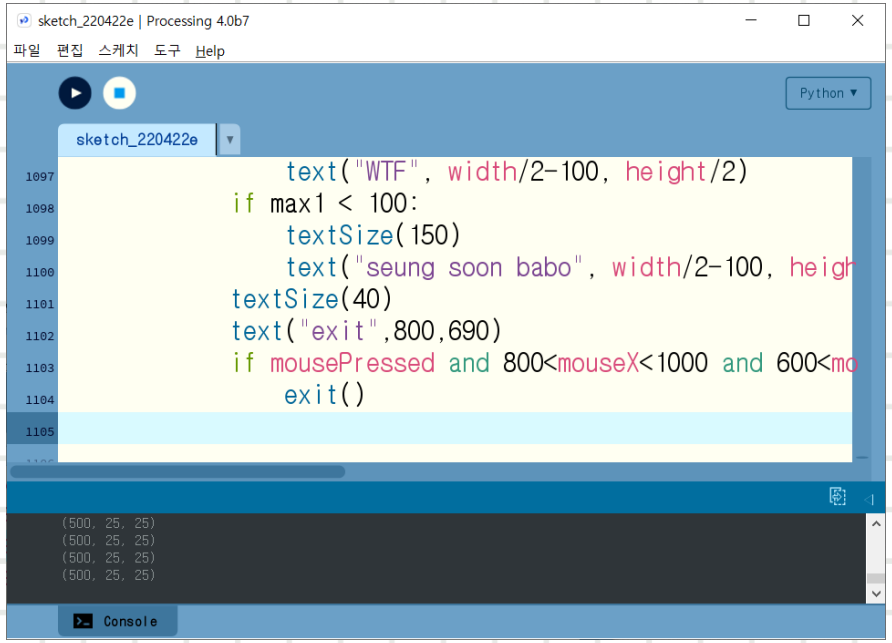




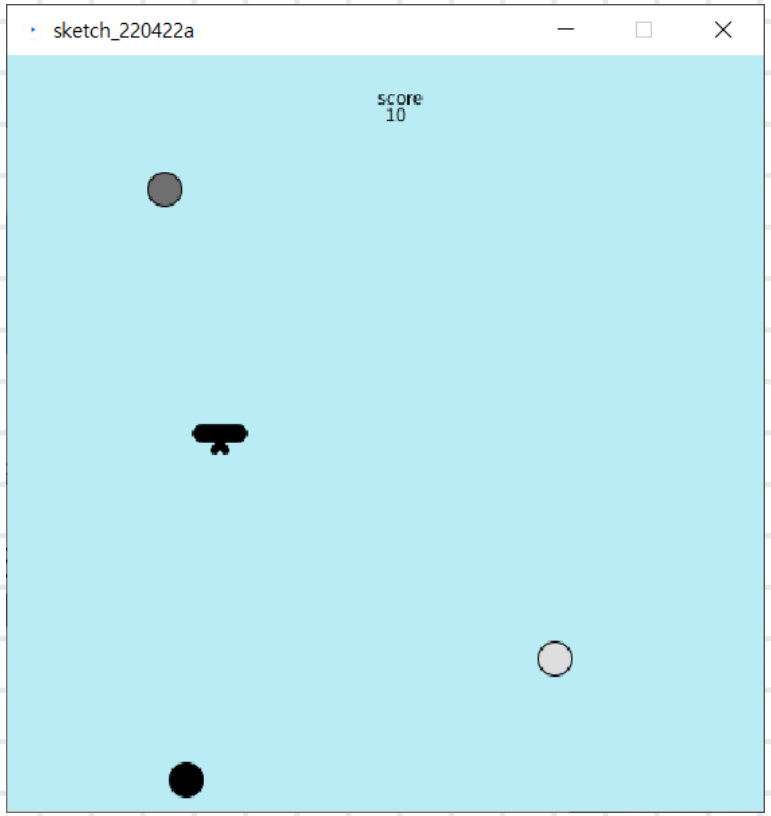
프로세싱 프로젝트



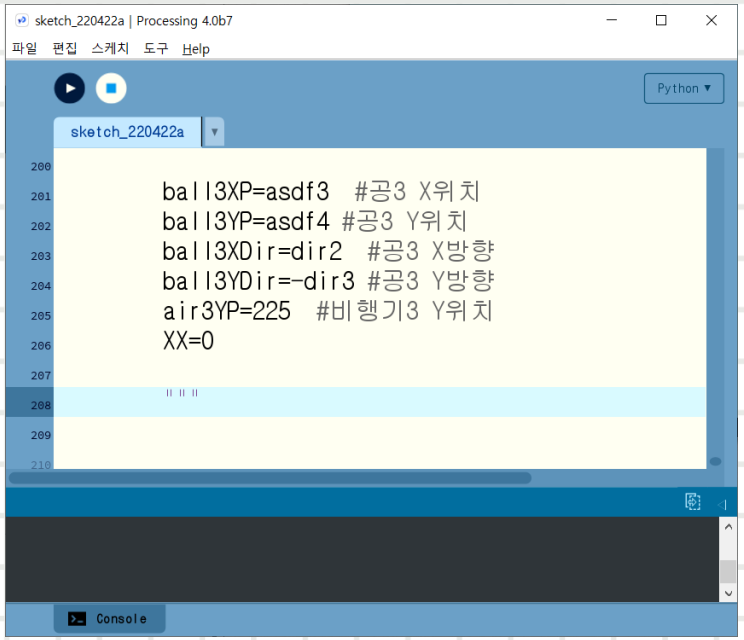
학생작품7(1학년 , 2021)



프로세싱 프로젝트

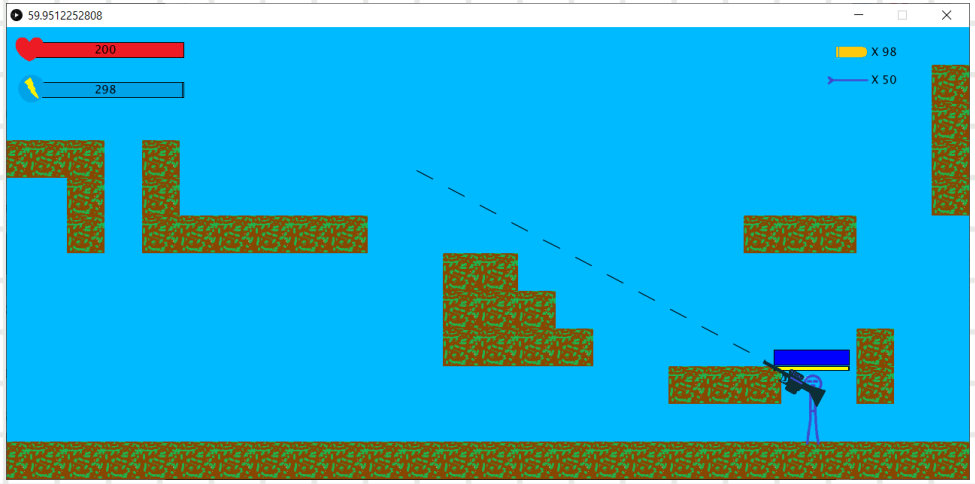


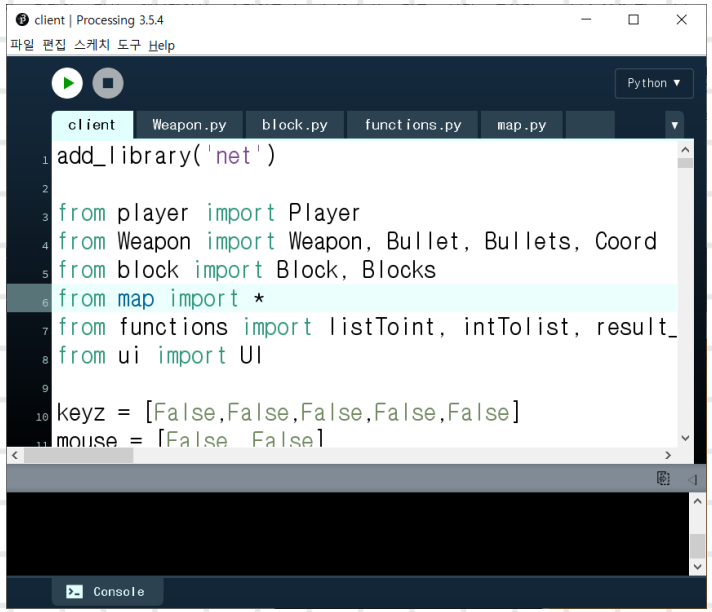
학생작품8(1학년 , 2021)

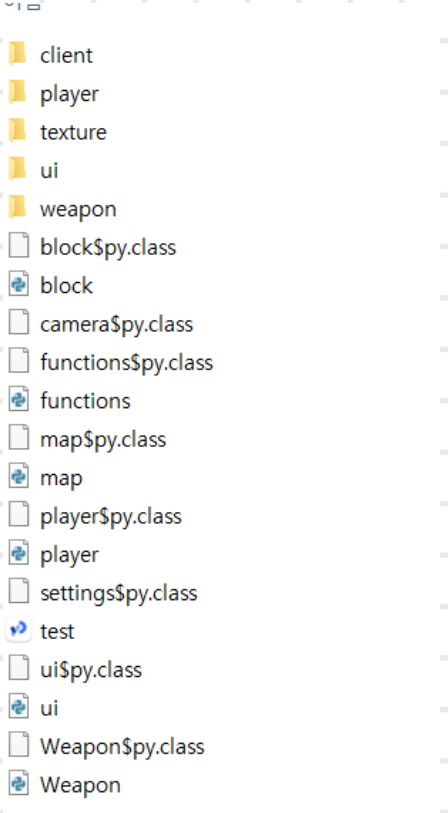


프로세싱 프로젝트

학생작품9(1학년, 2021)

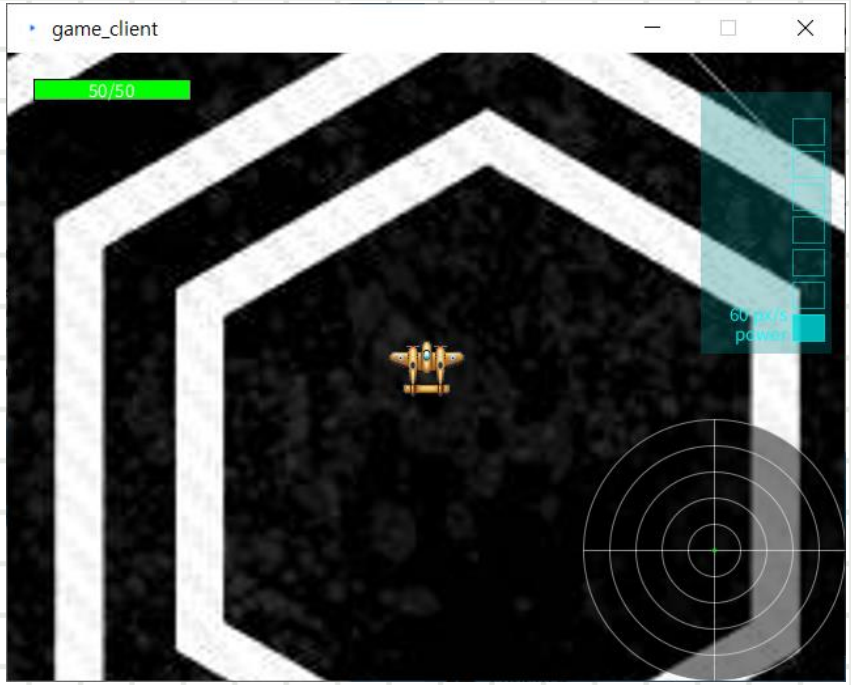


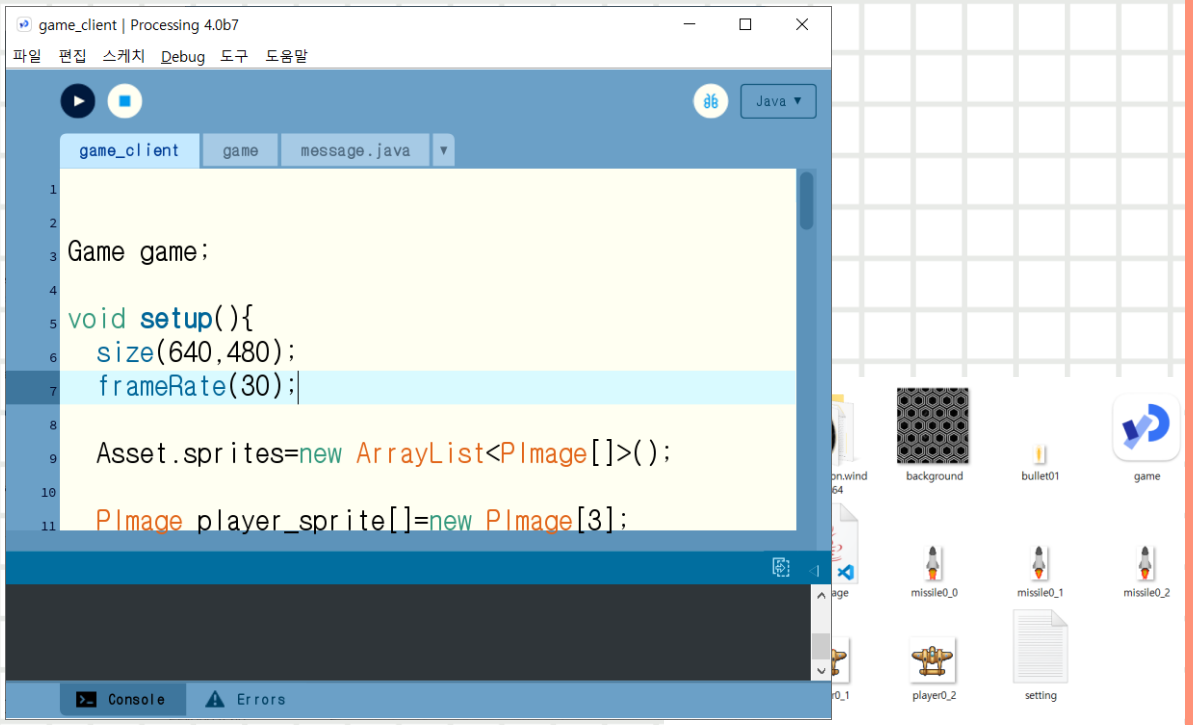




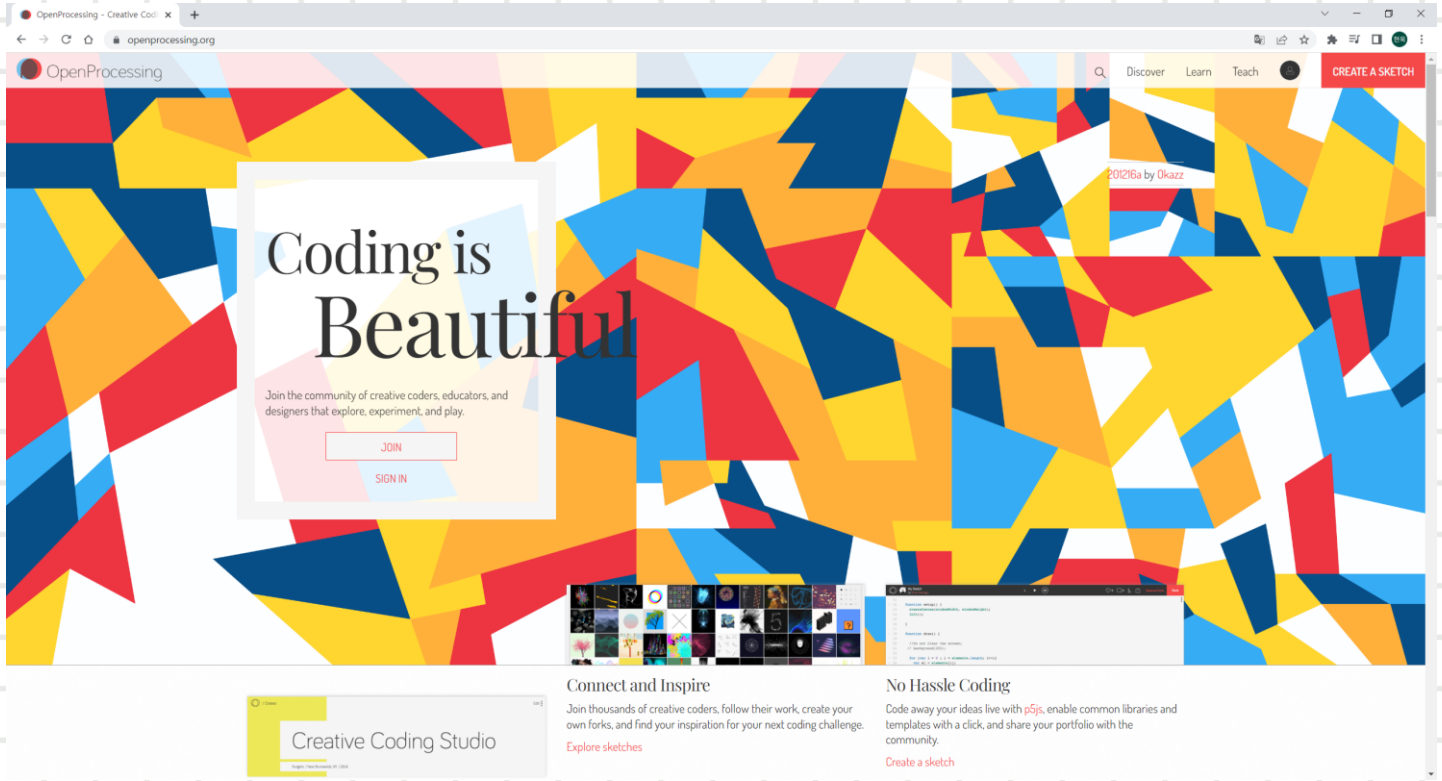
프로세싱 프로젝트

학생작품10(2학년, 2019)





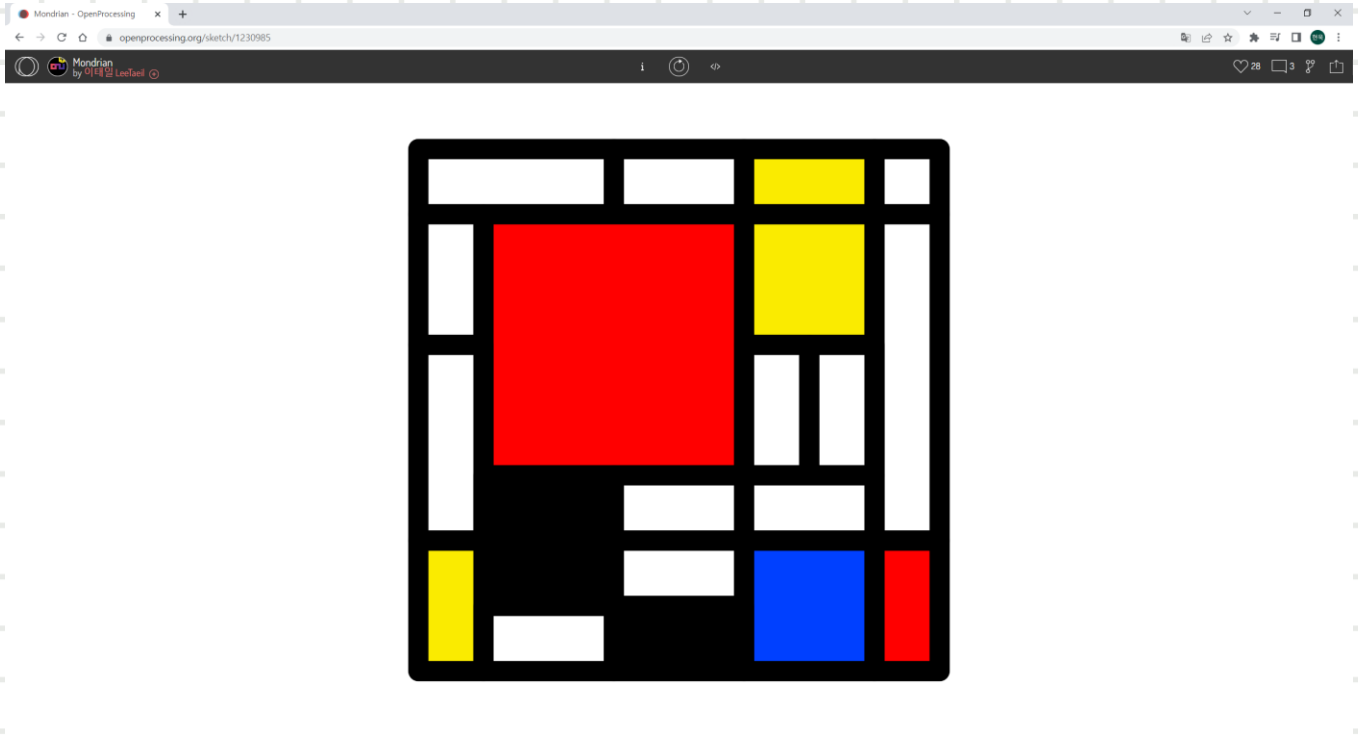
프로세싱 경험하기



오픈 프로세싱에 접속하여 작품 체험하기

https://openprocessing.org/

프로세싱 경험하기



실행, 정보, 코드 확인해보기

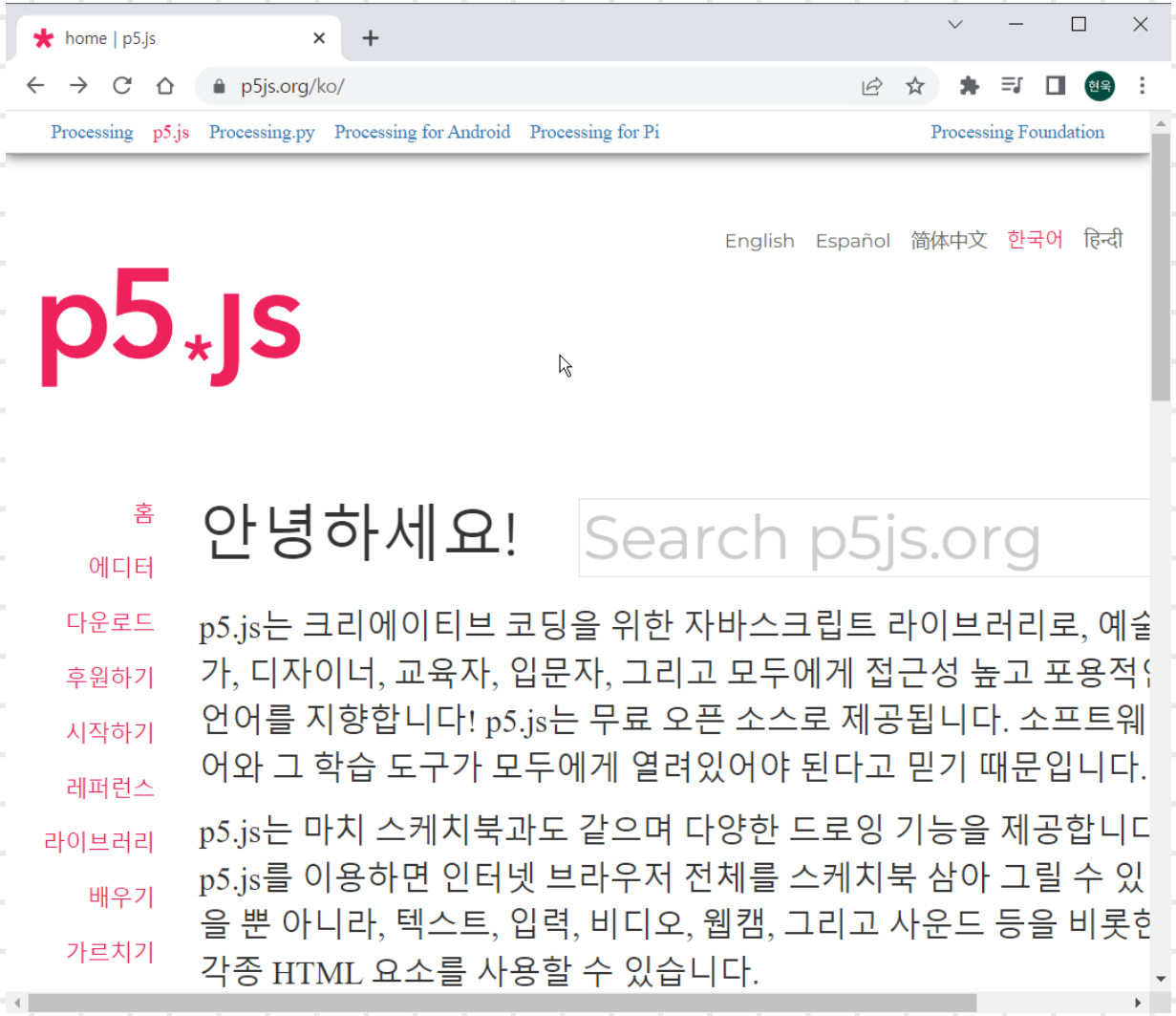
프로세싱 경험하기

오픈 프로세싱 체험 (20분)
관심가는 작품 1개 선정 및 근거 작성

프로젝트 선정	선정 근거
(여기에 작성하세요)	(여기에 작성하세요)

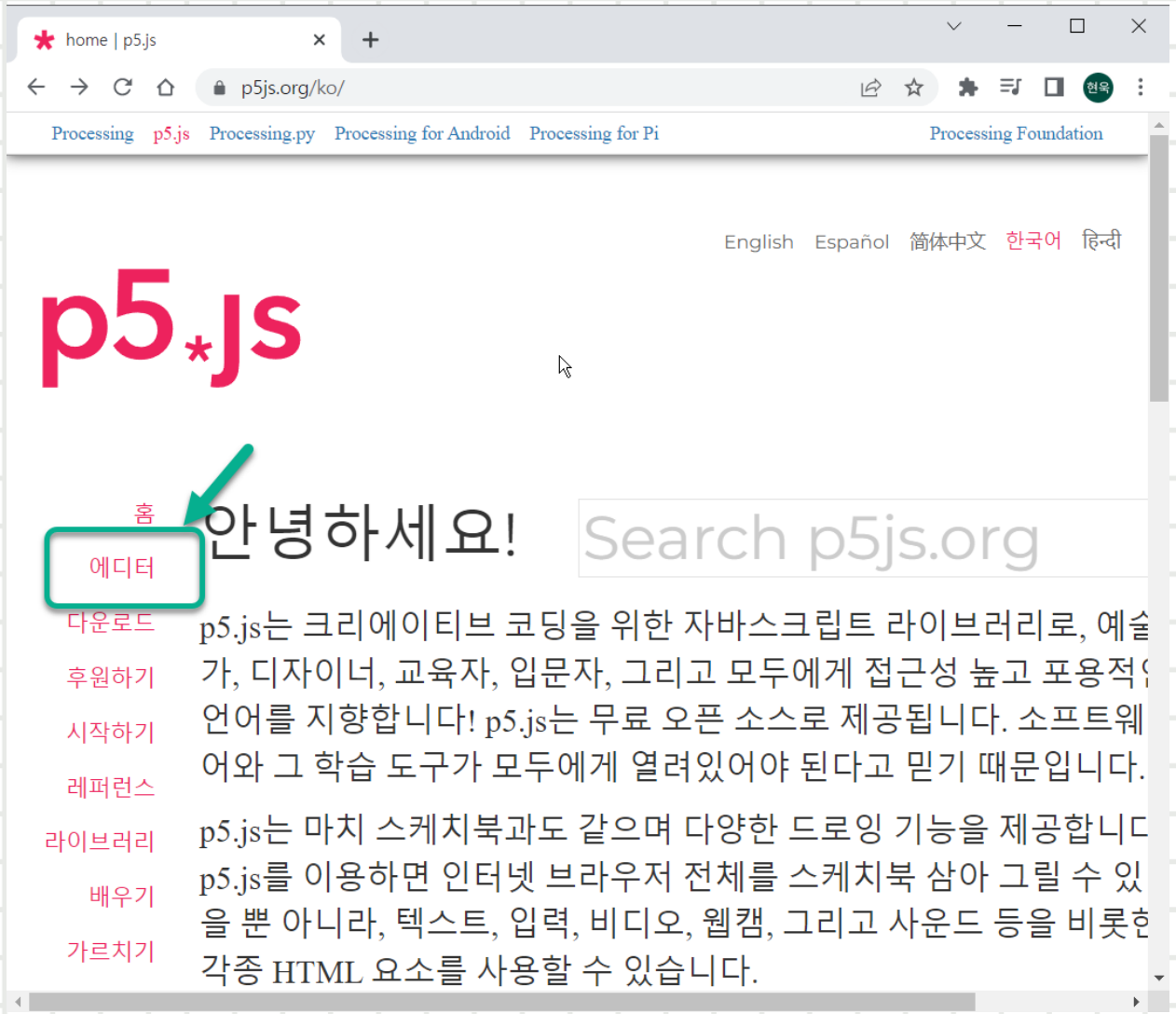
p5.js 에디터 접속

- 프로세싱에 접속
- https://p5js.org/



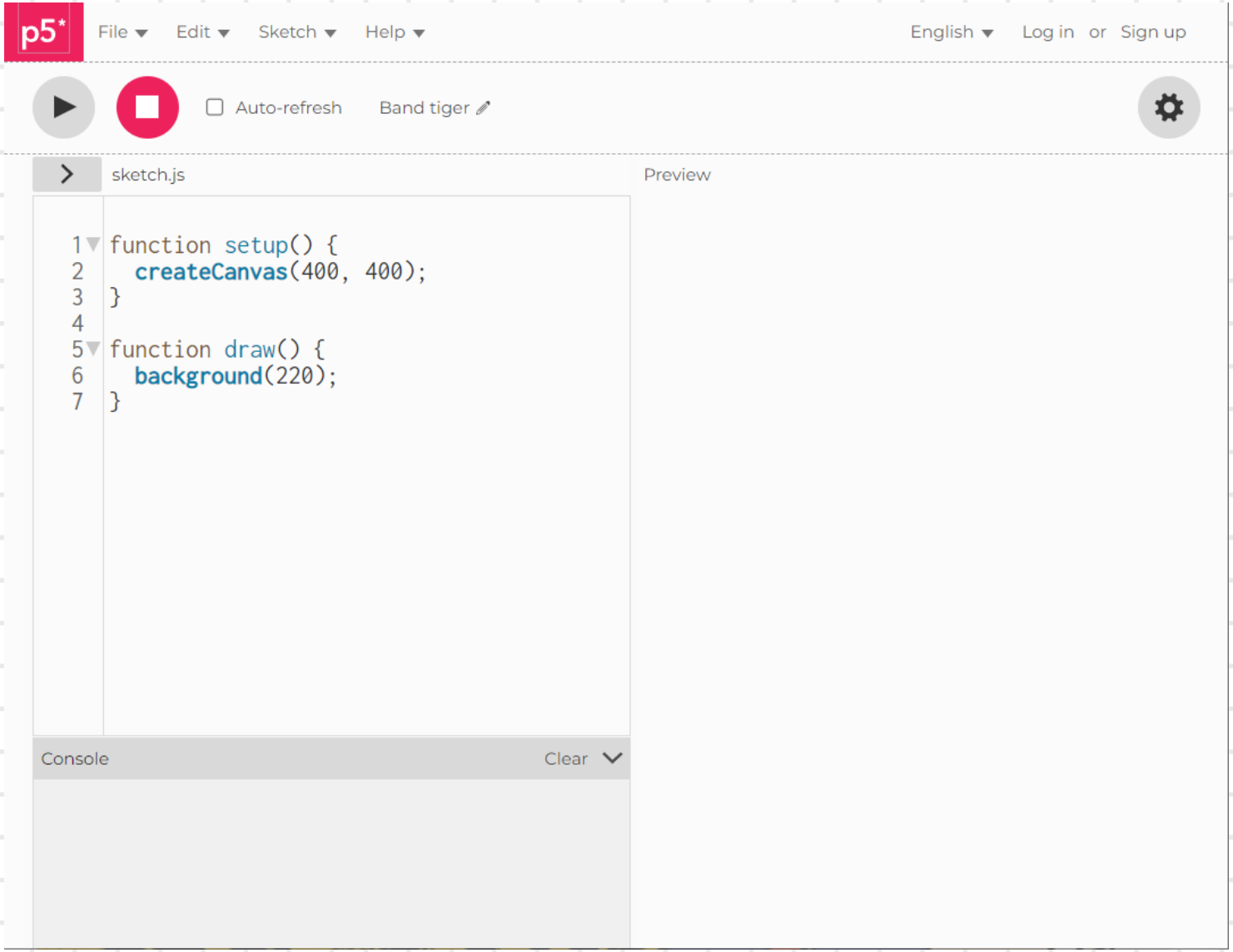
p5.js 에디터 접속

- 프로세싱에 접속
- https://p5js.org/



p5.js

- 프로세싱에 접속
- https://p5js.org/

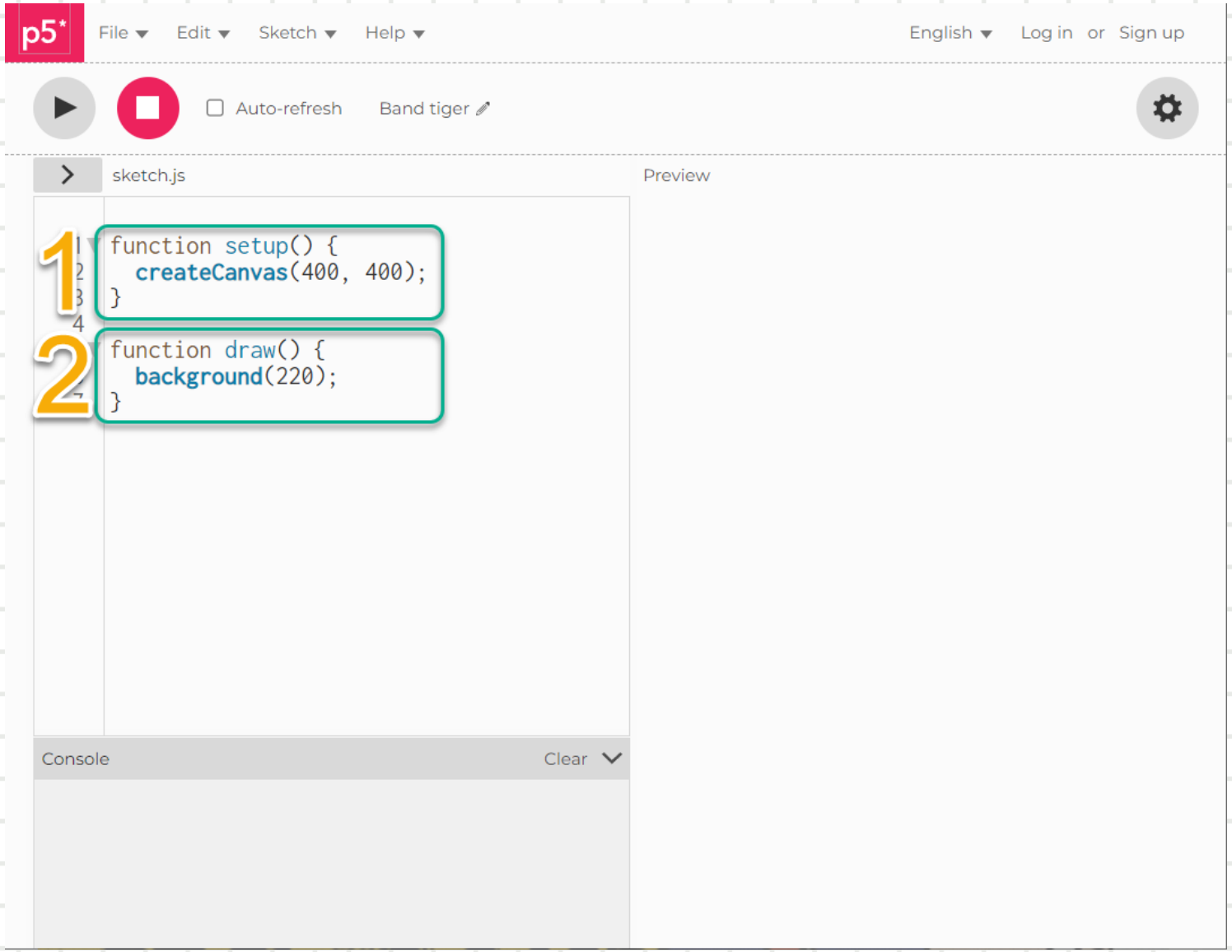


p5.js

- Code 영역
  코드를 편집하는 공간
- Preview 영역
  실행결과를 확인하는 공간
- Console 영역
  출력함수를 통해 문구 출력
  또는 에러 메시지 출력

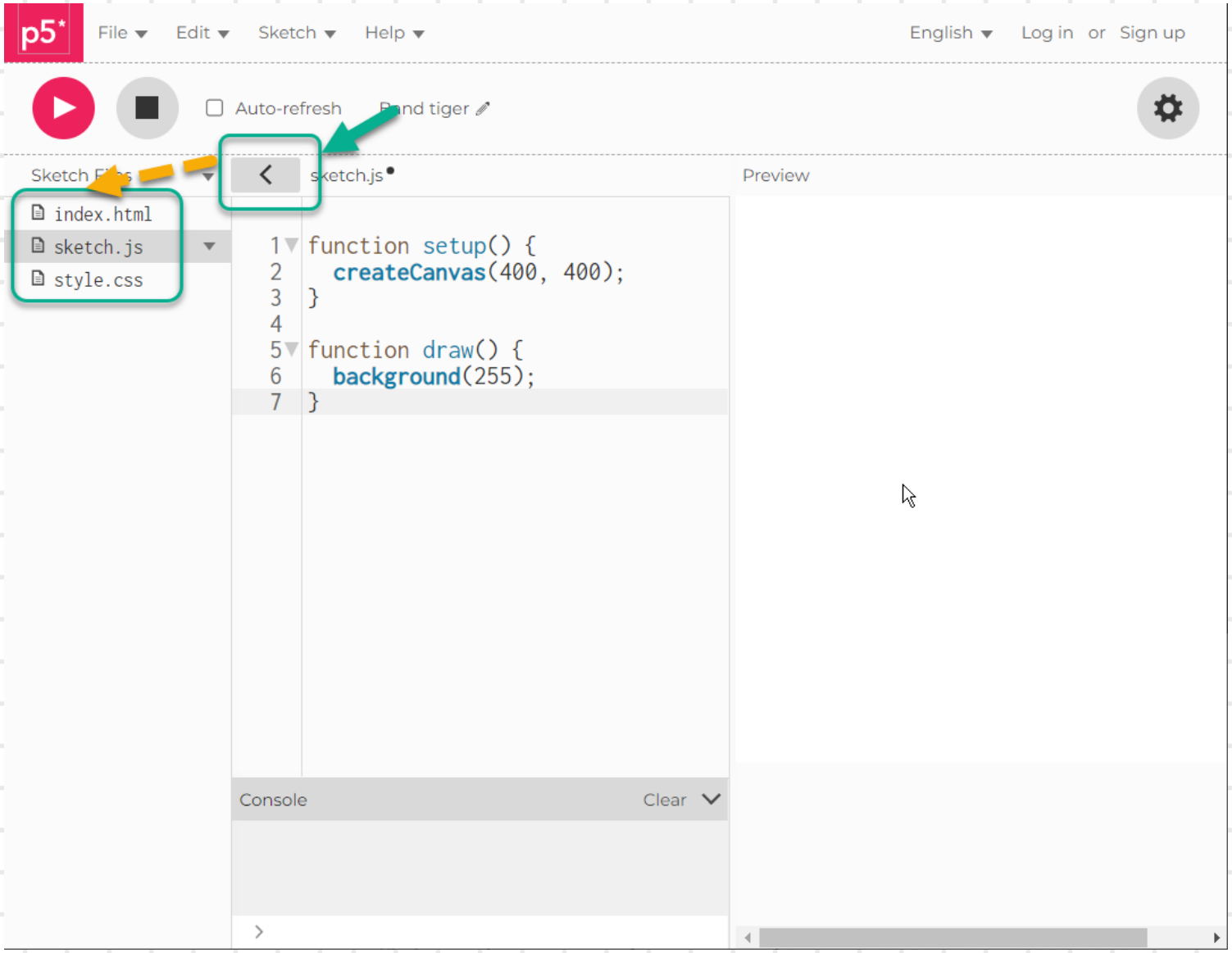
p5.js

- 1) setup() 함수
  프로그램의 시작 영역
- 2) draw() 함수
  프로그램의 실행 영역으로
  setup() 함수 이후 코드가
  반복하여 실행되는 부분



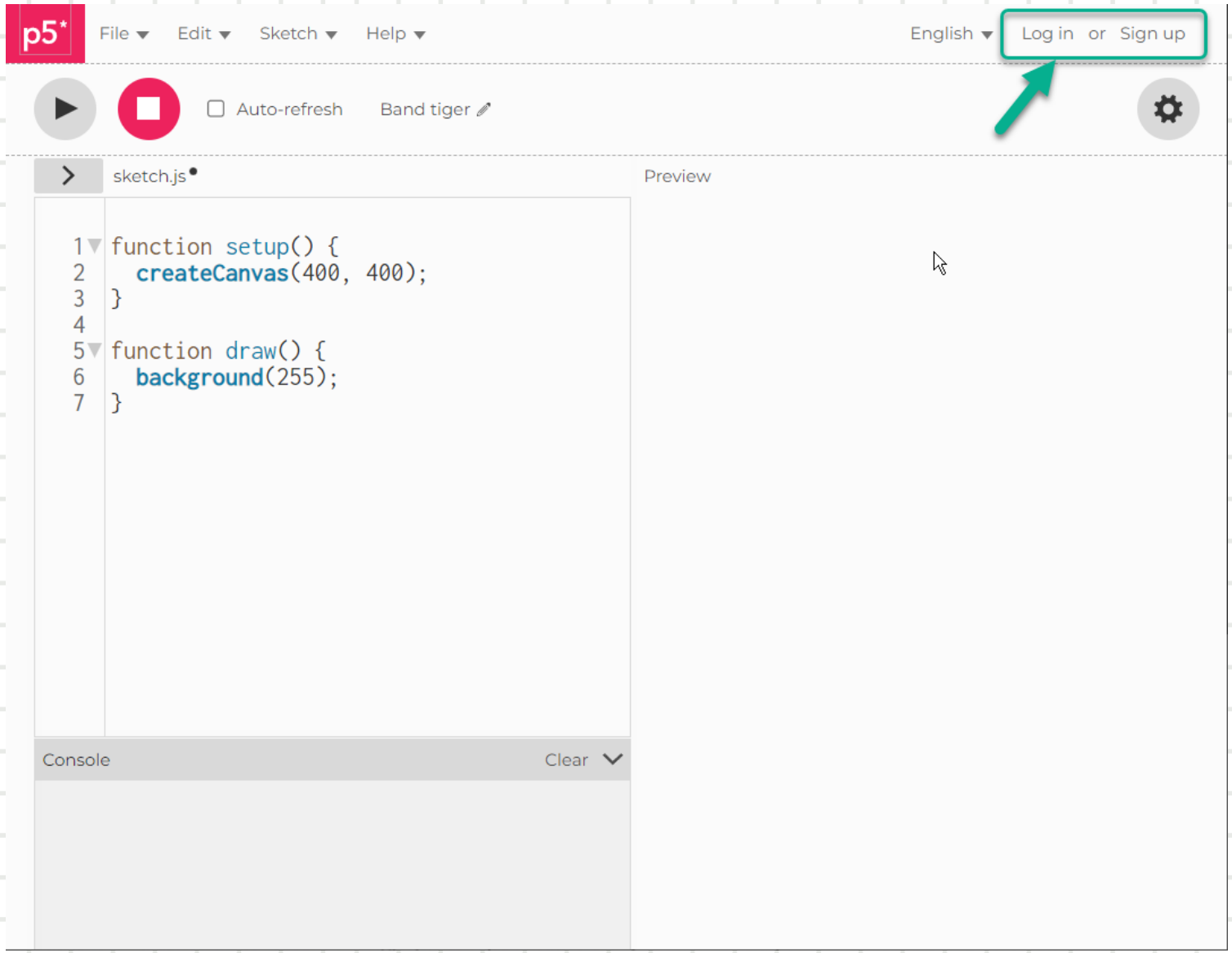
p5.js

- p5js는 자바스크립트이다.
  따라서 HTML기반,
  즉 웹에서 작동한다
- 웹 페이지는
  html, css, js로 구성이 됨
- html은 페이지 문서
  css는 디자인 설정
  js는 코드



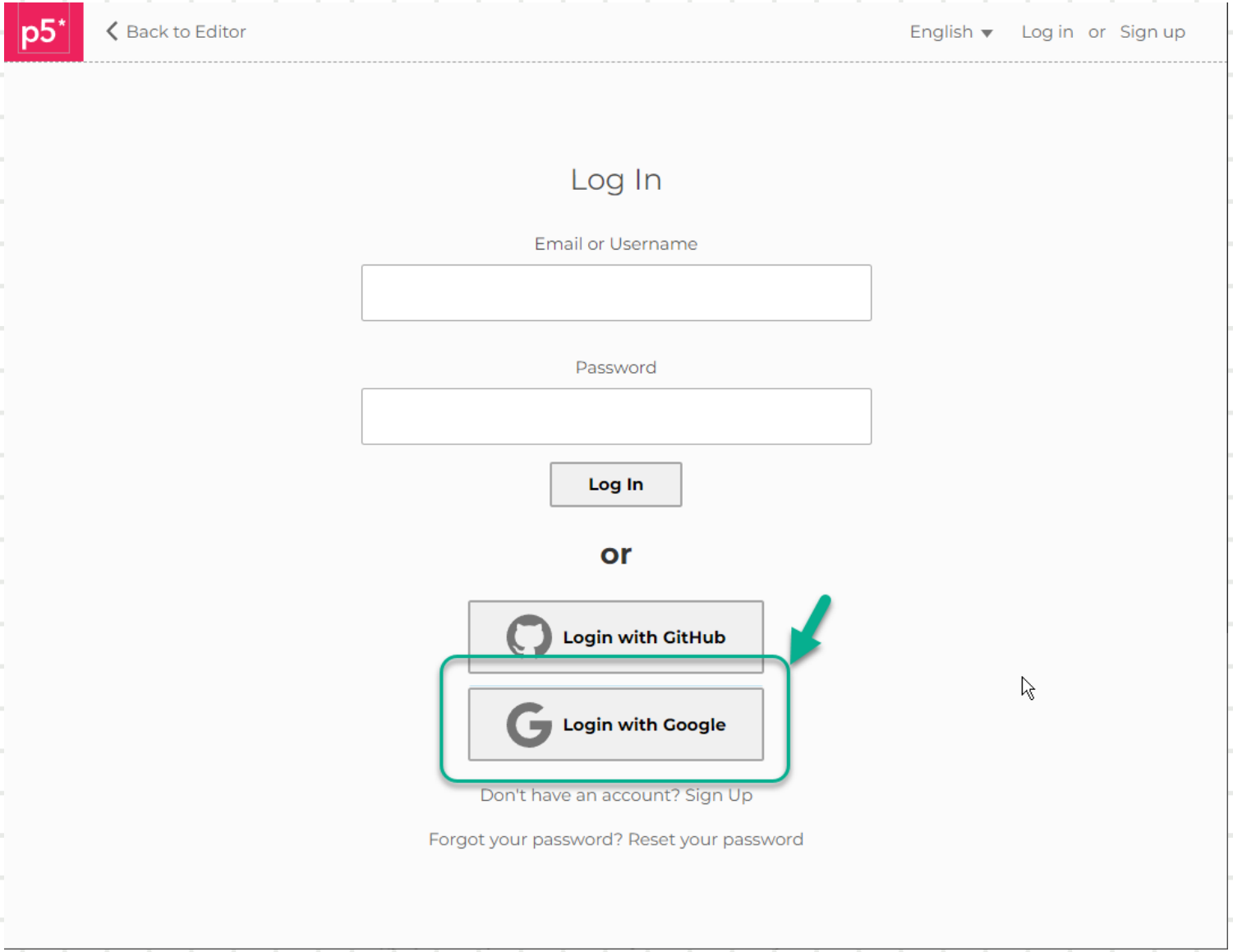
p5.js

- 회원 가입하기
- *개별 아이디로 회원가입이 가능하나 구글 계정으로도 로그인가능
- 코드를 웹상에서 관리



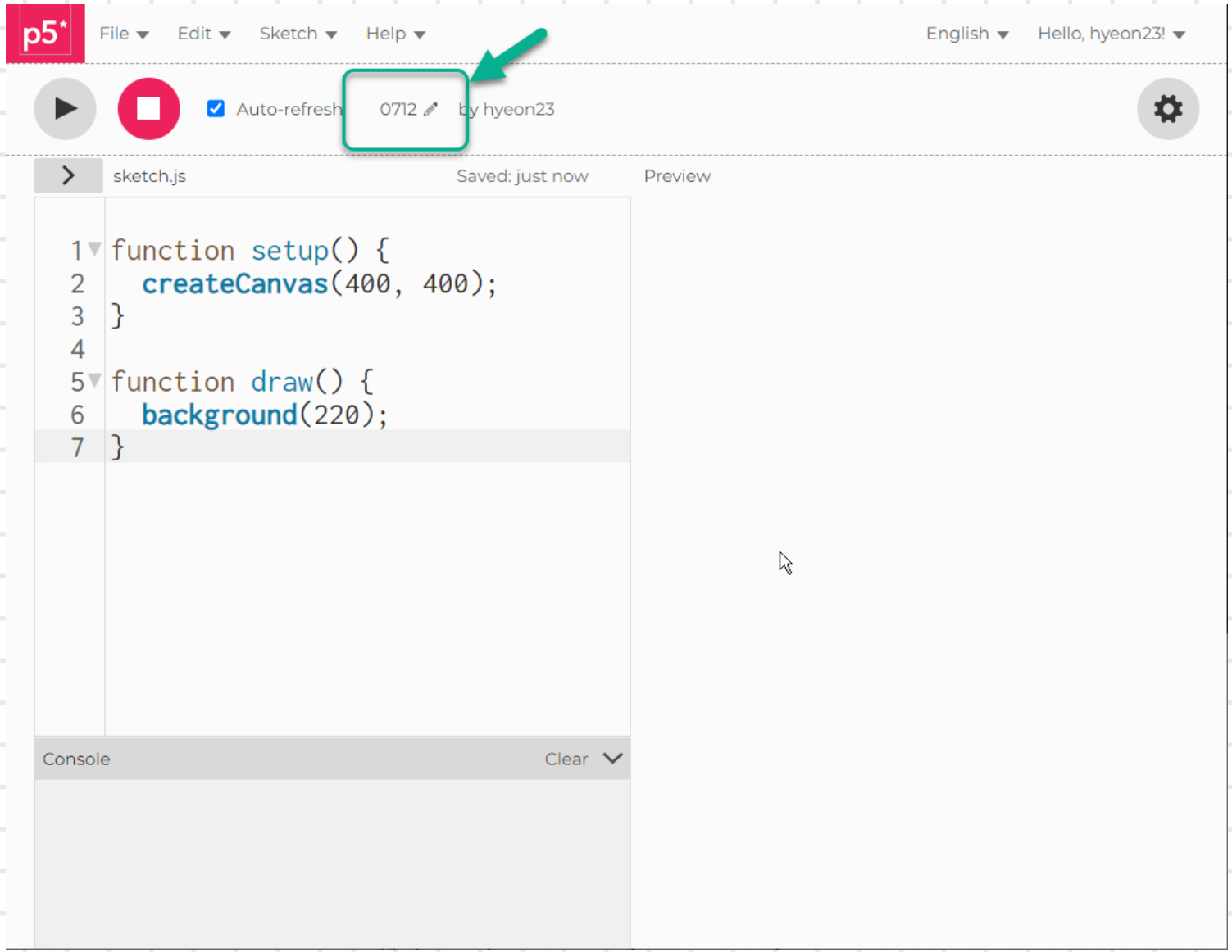
p5.js

- 회원 가입하기
- *개별 아이디로 회원가입이 가능하나 구글 계정으로도 로그인가능
- 코드를 웹상에서 관리



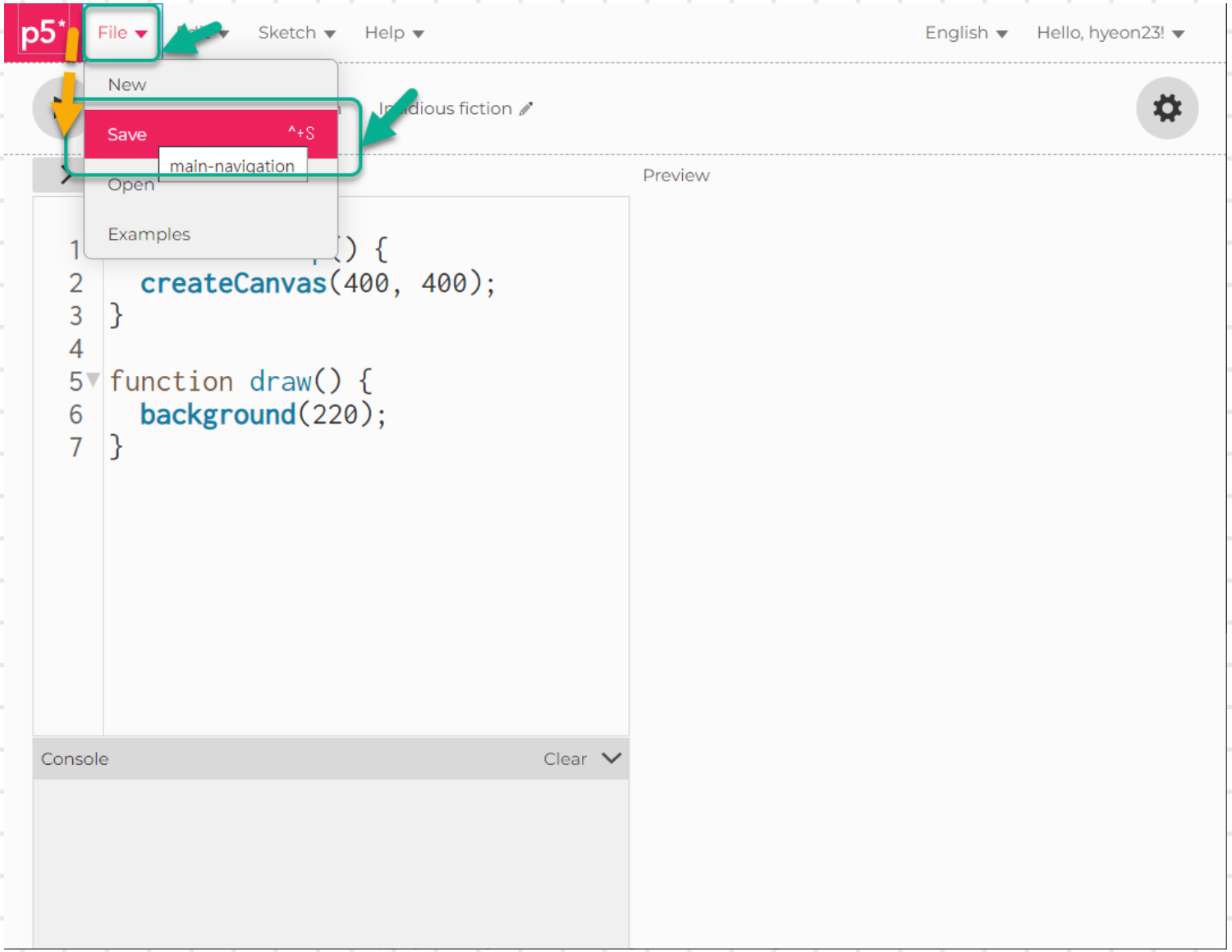
p5.js

- 코드 이름 변경하기



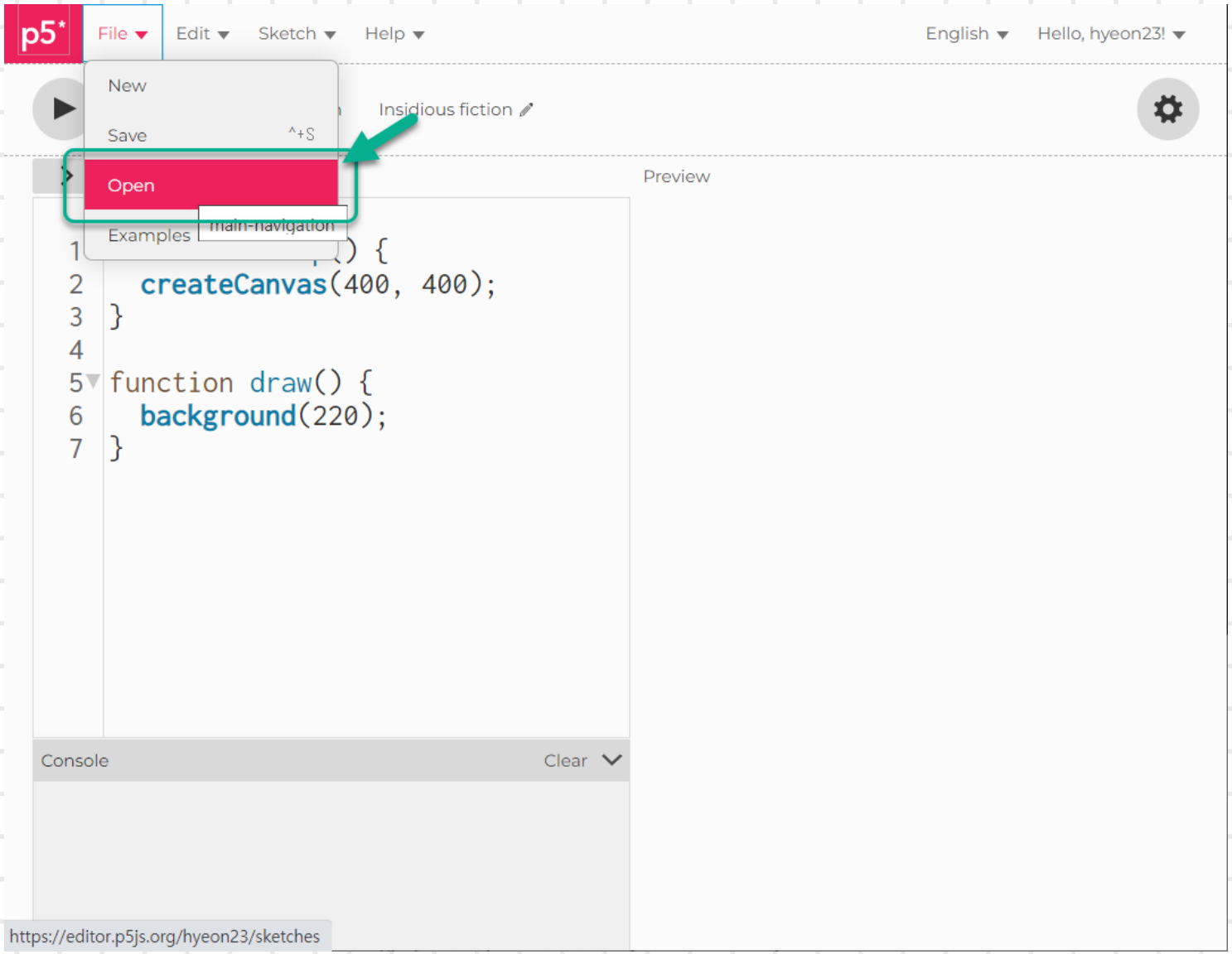
p5.js

- 코드 저장하기


p5*
File
Sketch
Help
English
Hello, hyeon23!
New
Save
^+S
main-navigation
Open
Examples
Preview
createCanvas(400, 400);
function draw() {
background(220);
Console
Clear

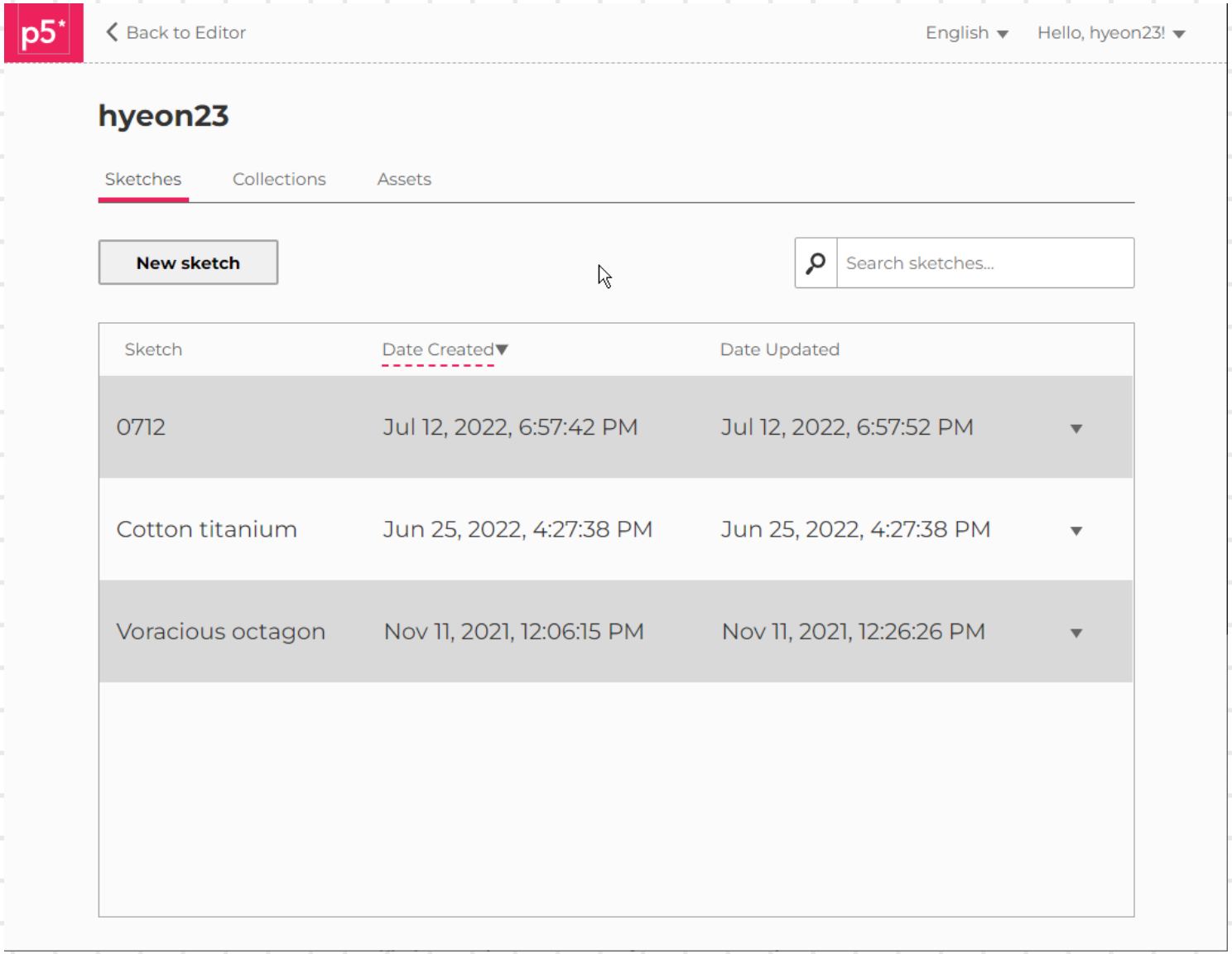
p5.js

- 코드 불러오기


p5*
File
Edit
Sketch
Help
English
Hello, hyeon23!
New
Save ^+S
Open
Examples
main-navigation
Insidious fiction
Preview
1) {
2 createCanvas(400, 400);
3 }
4
5 function draw() {
6 background(220);
7 }
Console
Clear
https://editor.p5js.org/hyeon23/sketches

p5.js

- 코드 불러오기
- 지난 코드 목록을 확인하고 불러올 수 있음



좌표체계

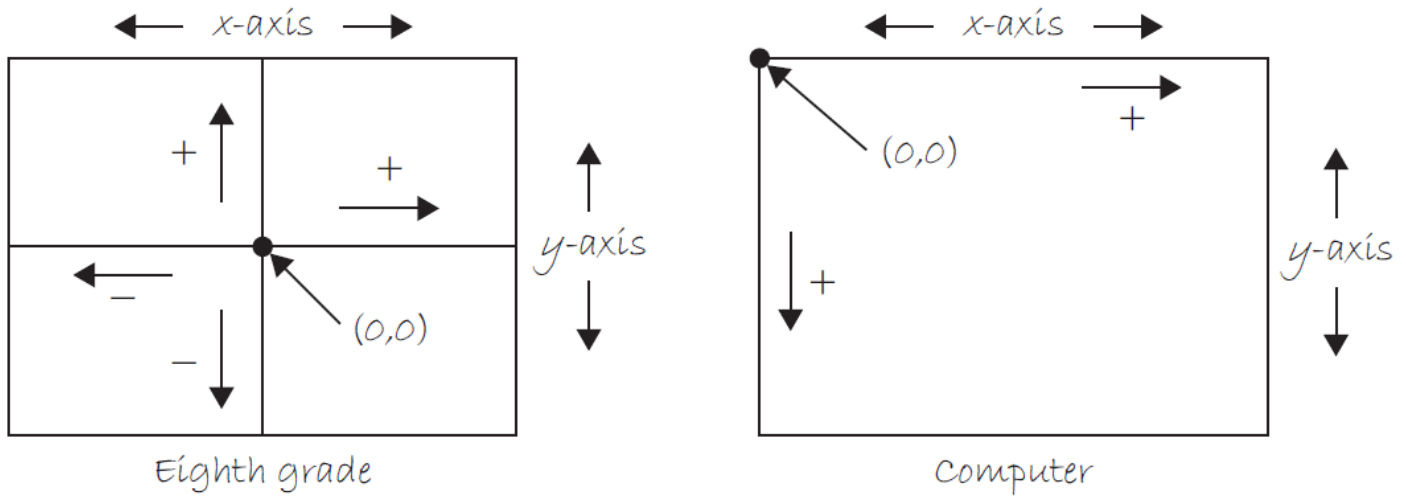
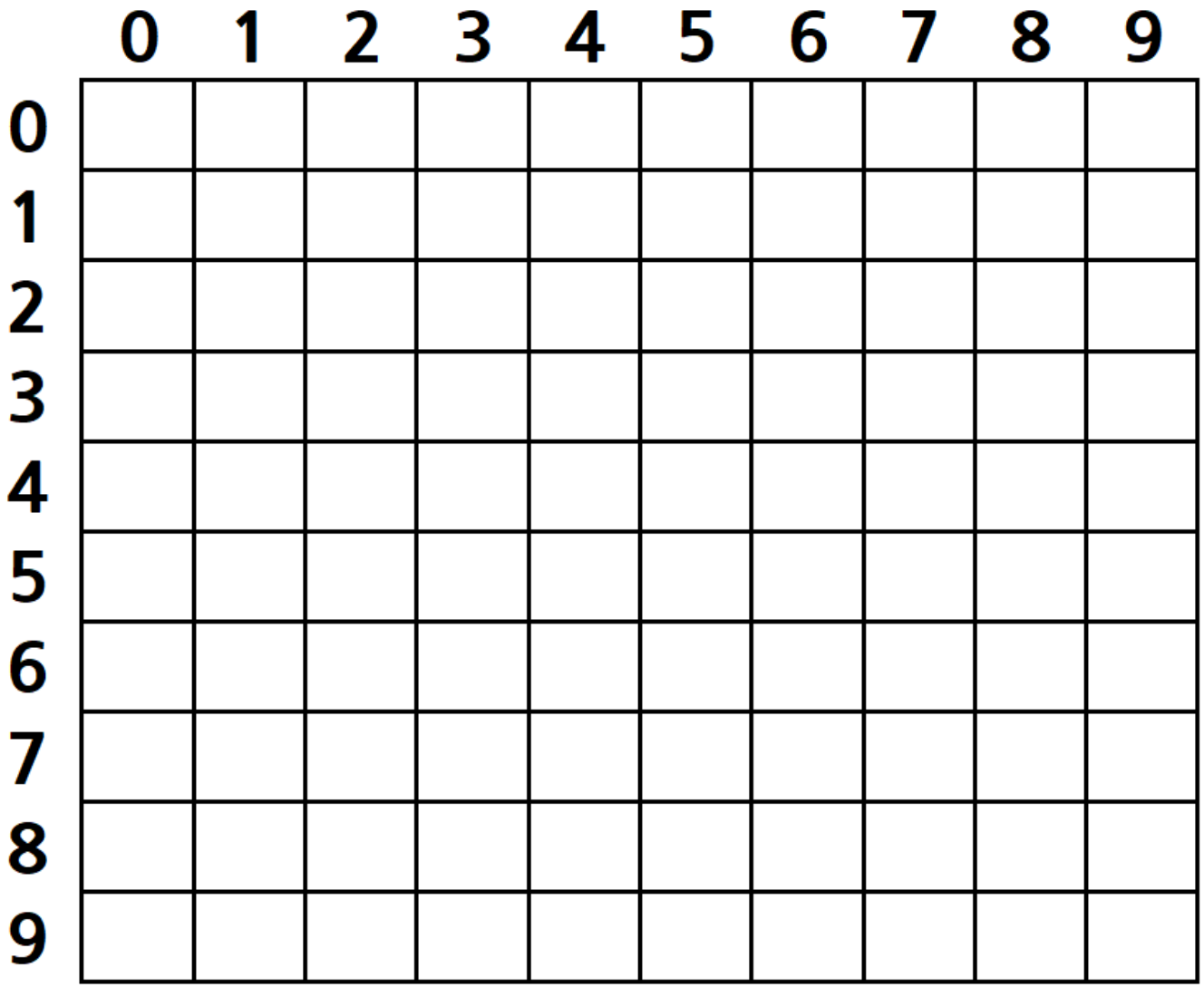


fig. 1.3

- 그래프 용지(직교좌표계)
  -원점이 정중앙에 배치
  -위쪽, 오른쪽이 양수

- 컴퓨터 좌표체계
  -원점이 좌측상단에 배치
  -아래쪽, 오른쪽이 양수

좌표체계



기본 도형

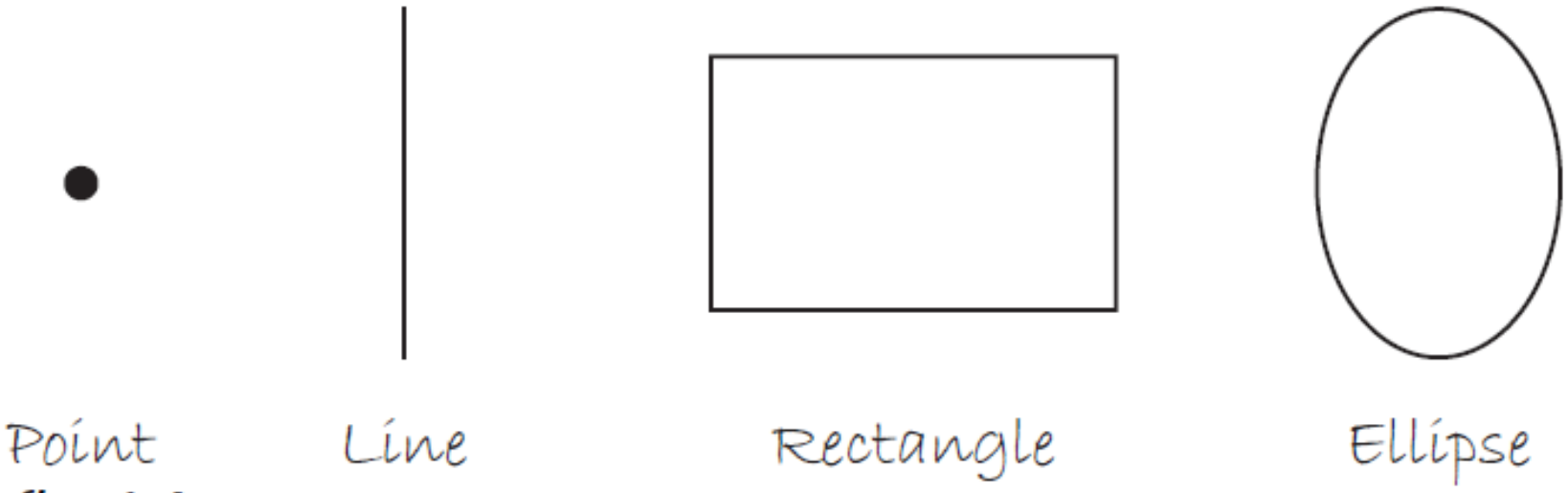
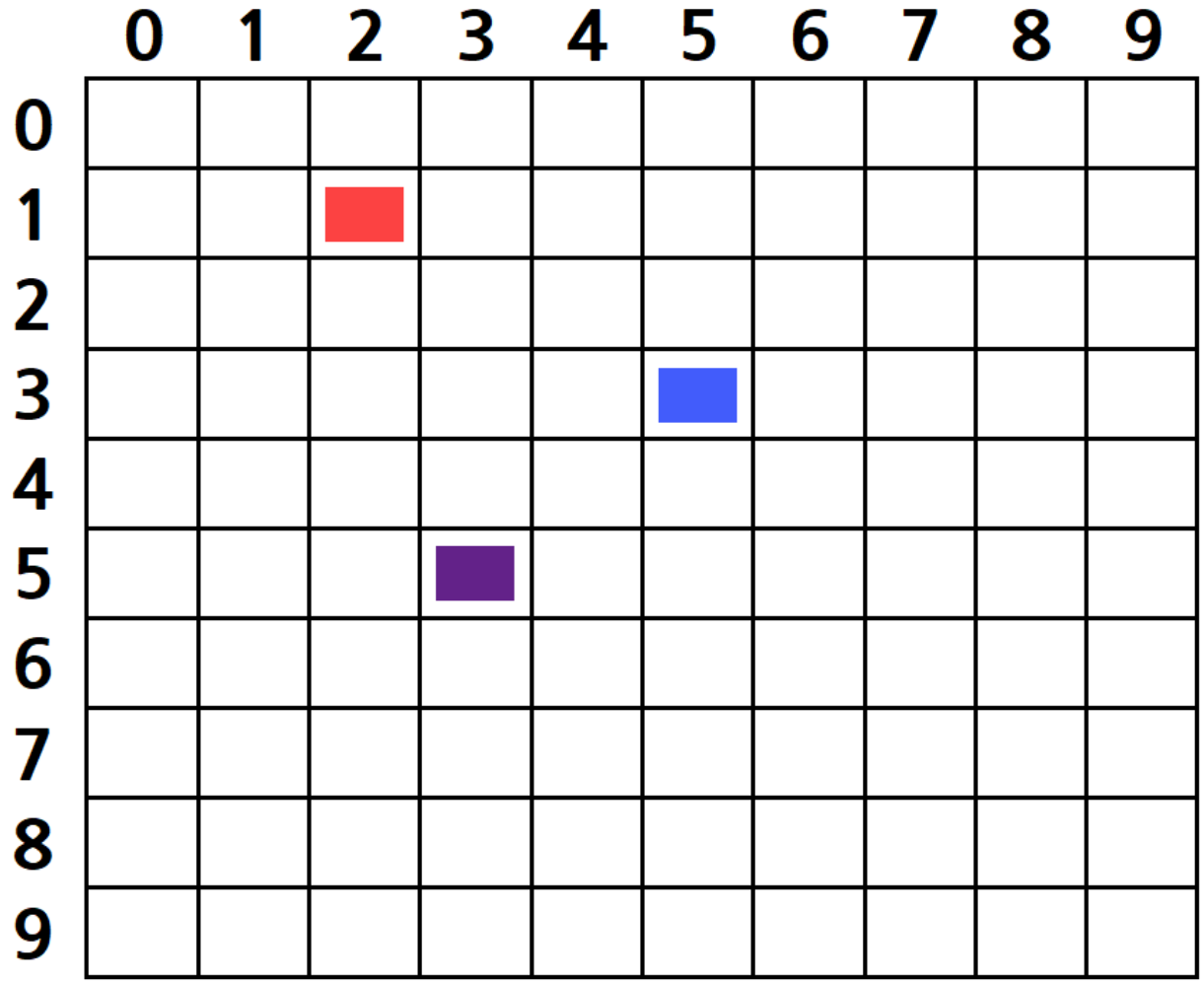


fig. 1.4

- 점: point()
- 선: line()
- 사각형: rect()
- 타원형: ellipse()

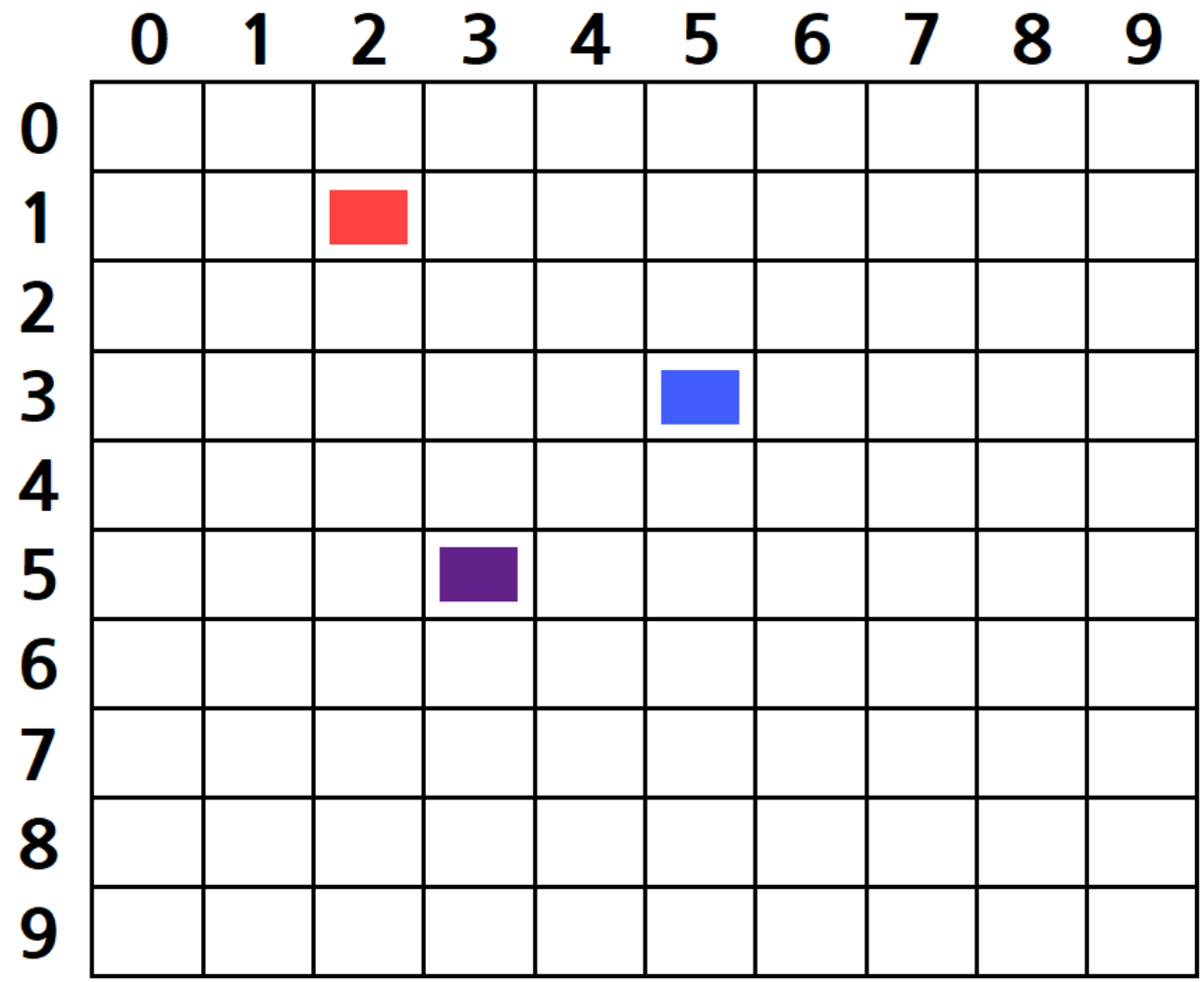
점: point()

- 1) 빨간색 점의 표현은?
- 2) 파란색 점의 표현은?
- 3) 보라색 점의 표현은?

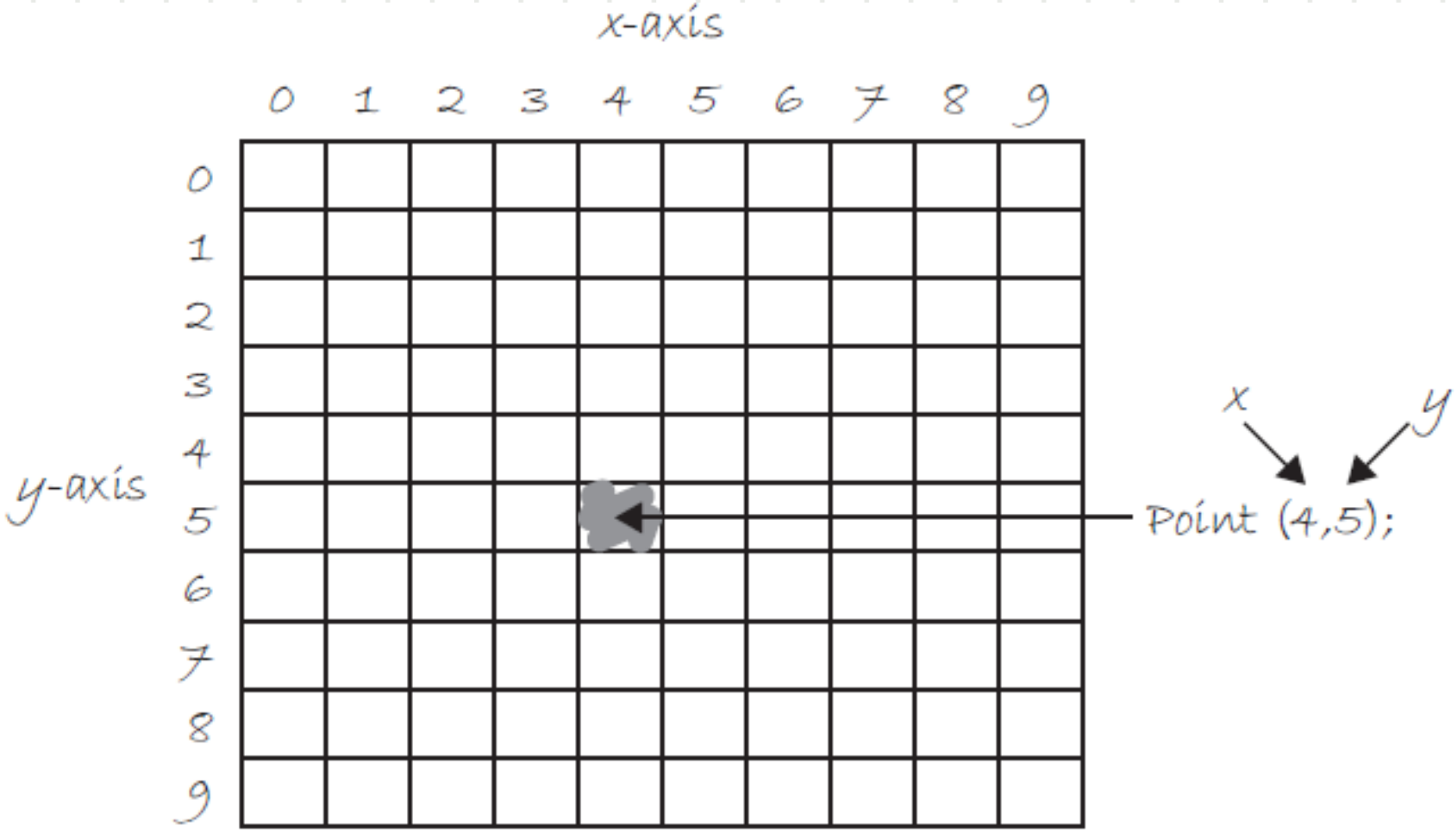


점: point()

- 1) 빨간색 점의 표현은?
  (2,1)
- 2) 파란색 점의 표현은?
  (5,3)
- 3) 보라색 점의 표현은?
  (3,5)

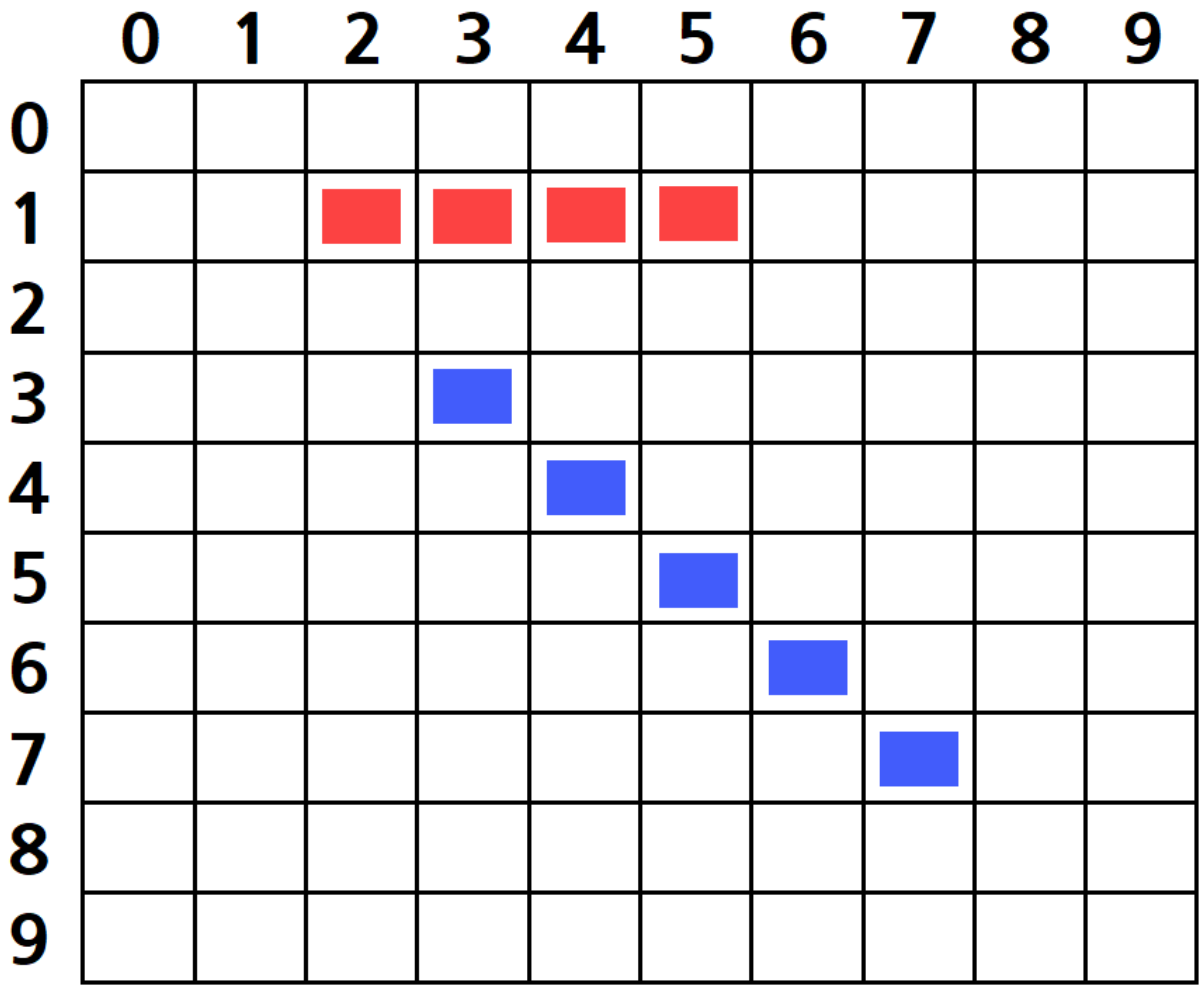


점: point(x, y)



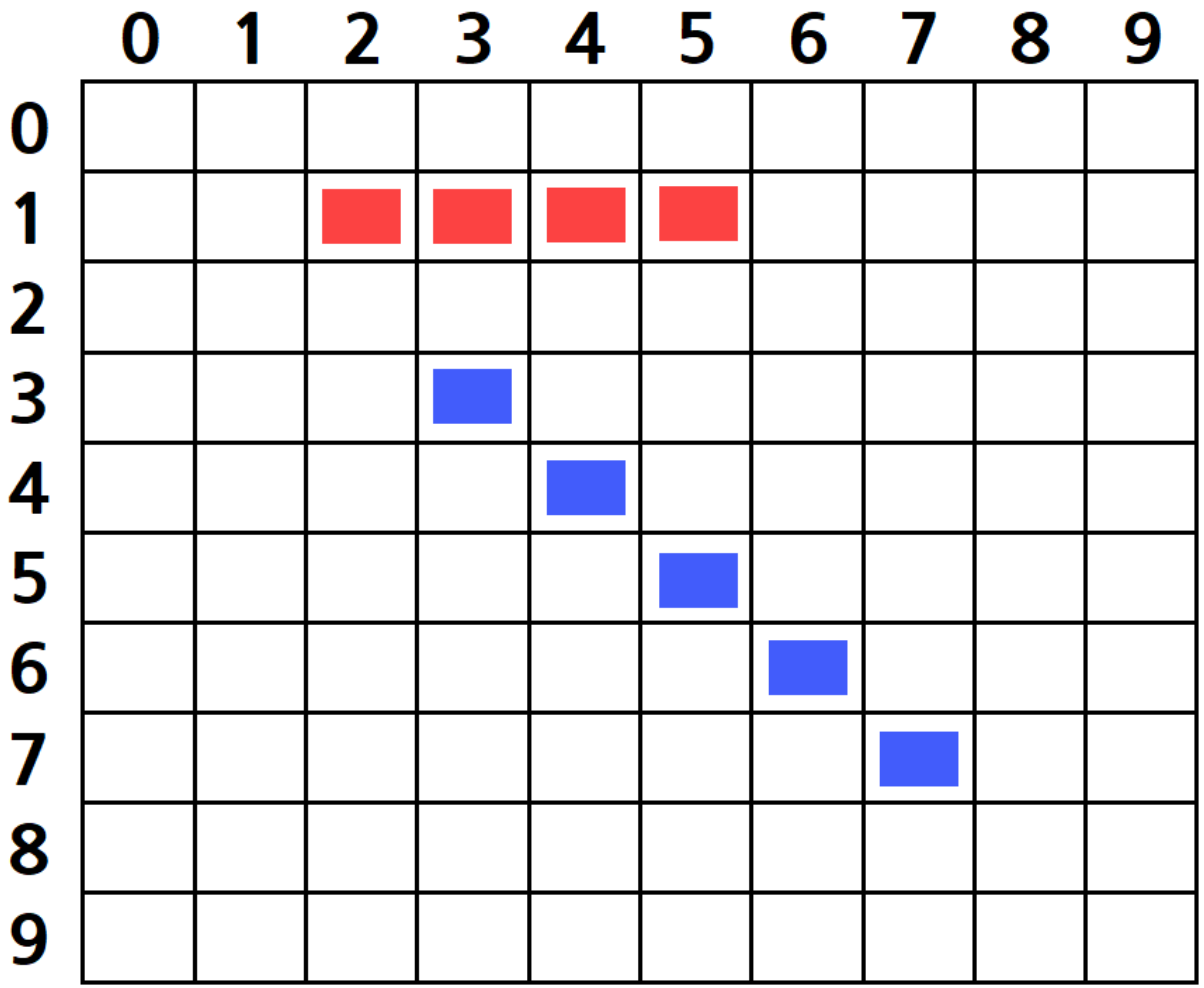
선: line()

- 1) 빨간색 선의 표현은?
- 2) 파란색 선의 표현은?

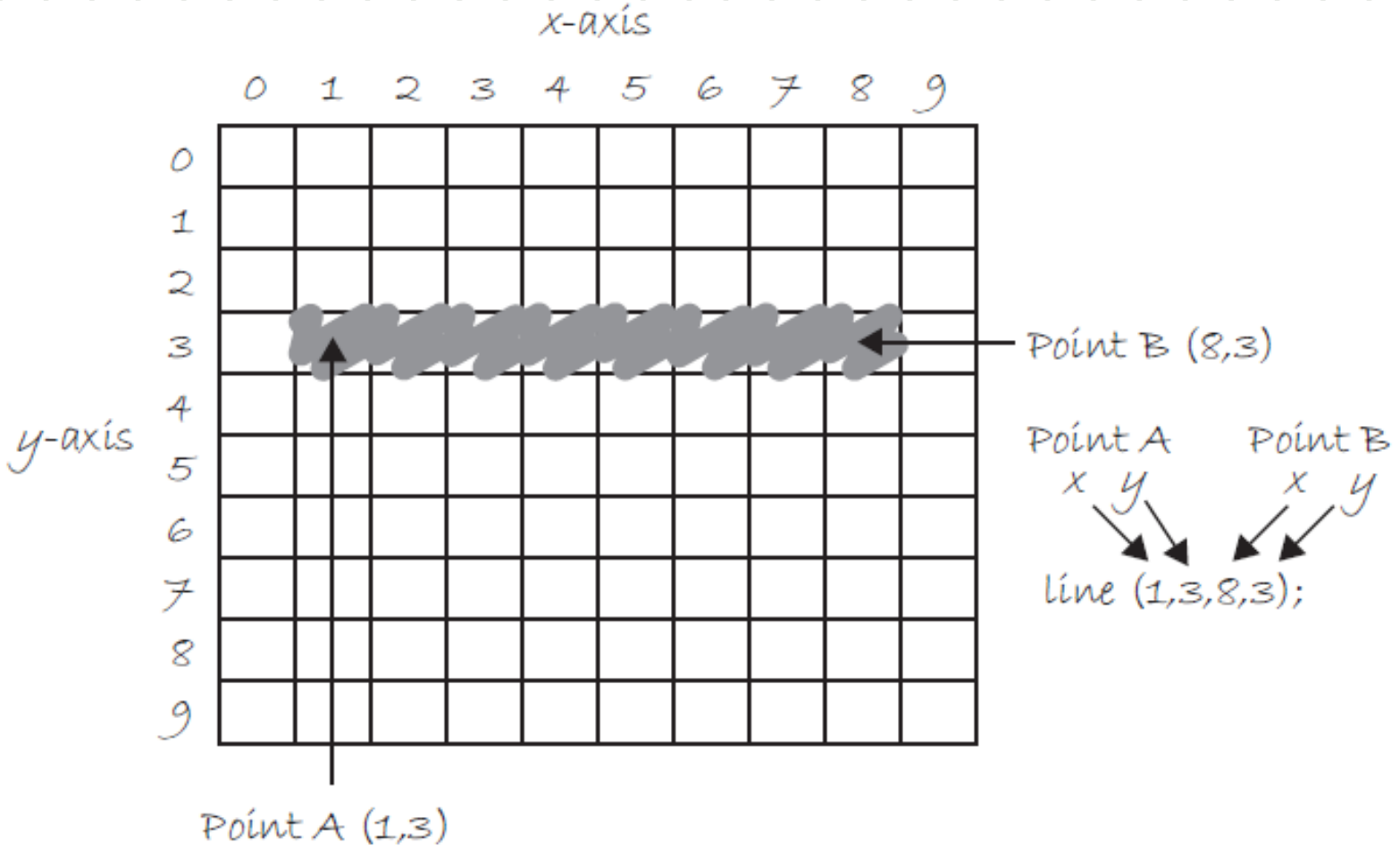


선: line()

- 1) 빨간색 선의 표현은?
  (2,1), (5,1)
- 2) 파란색 선의 표현은?
  (3,3), (7,7)

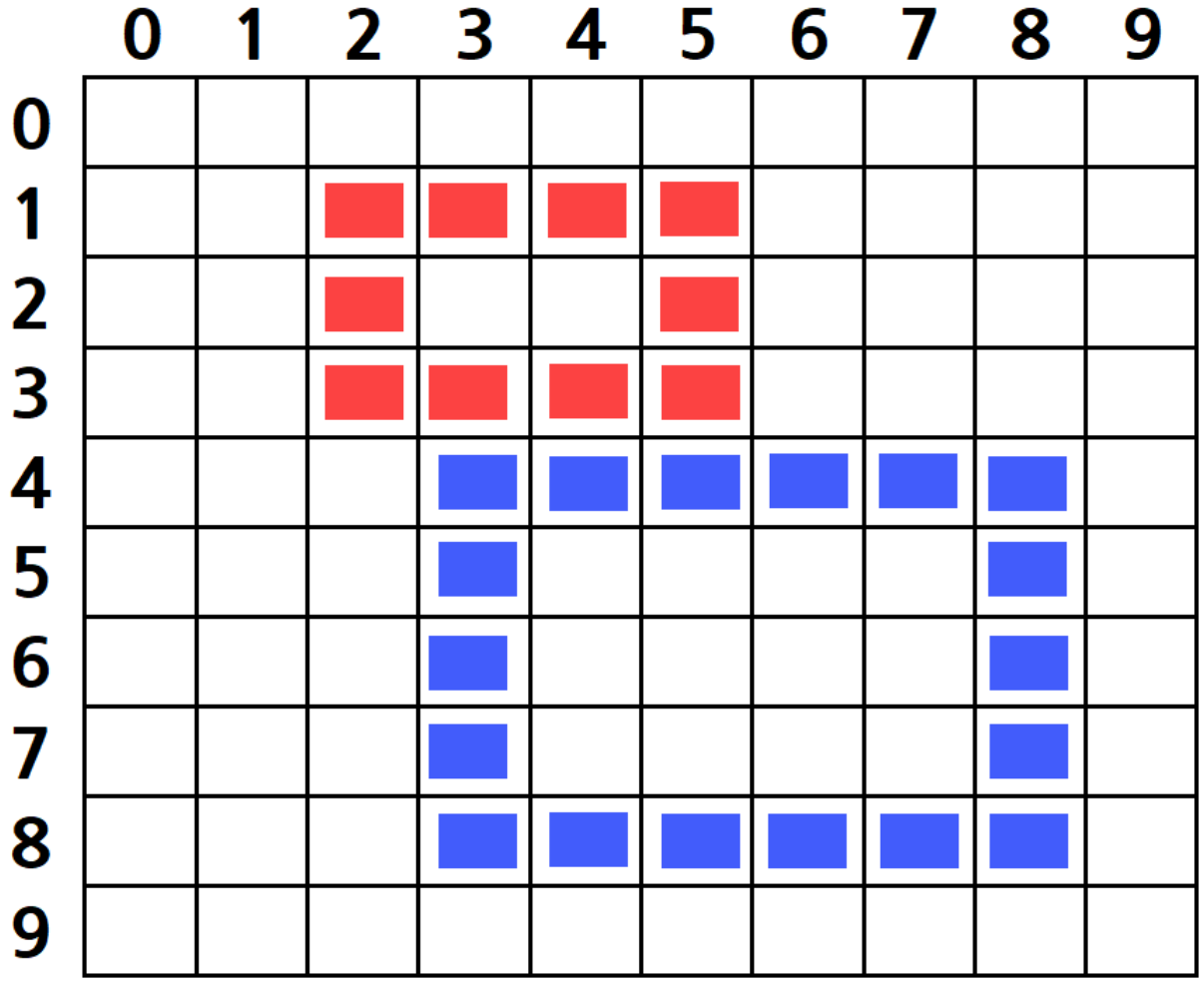


선: line(x1, y1, x2, y2)



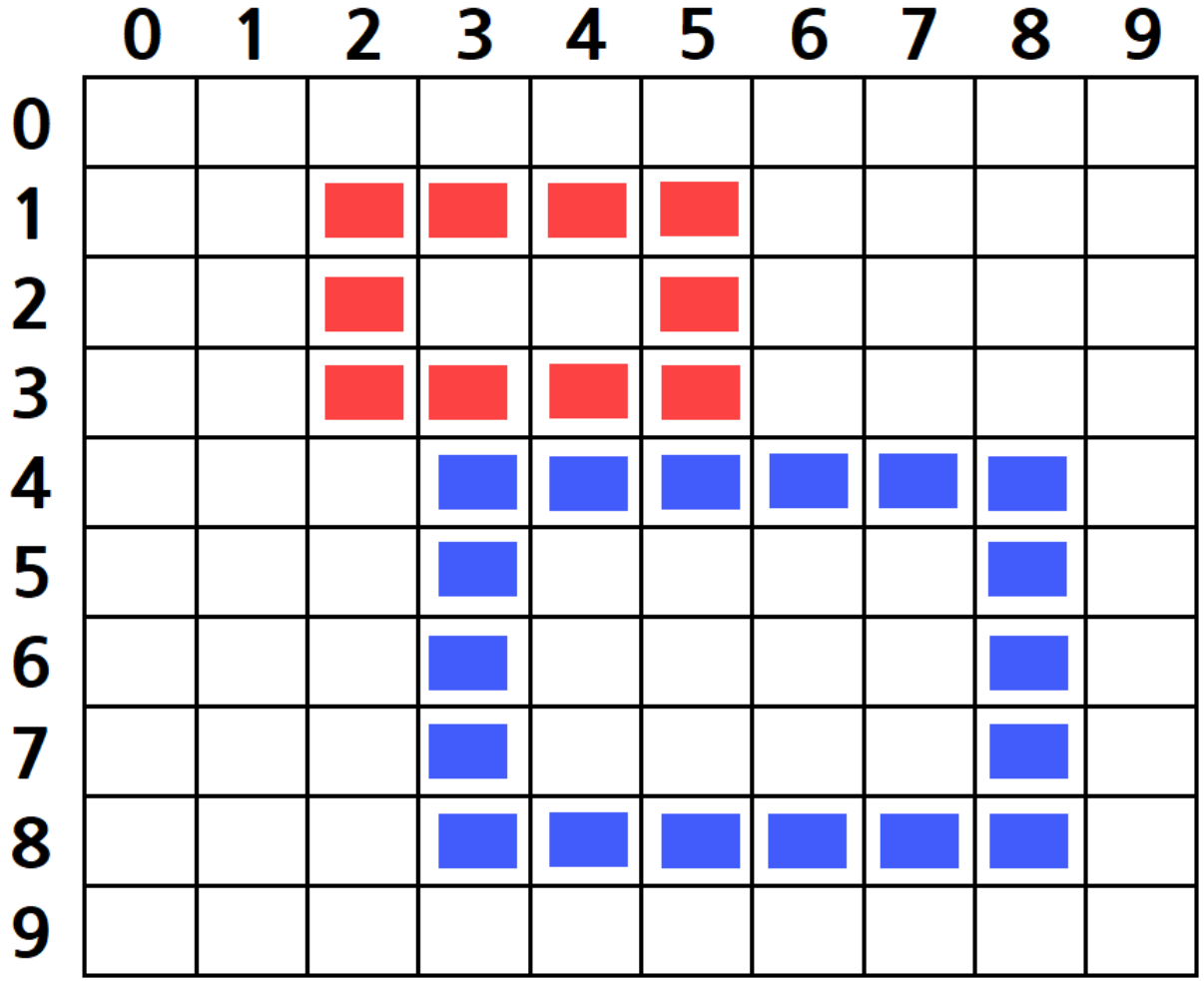
직사각형 : rect()

- 1) 빨간색 사각형의 표현은?
- 2) 파란색 사각형의 표현은?

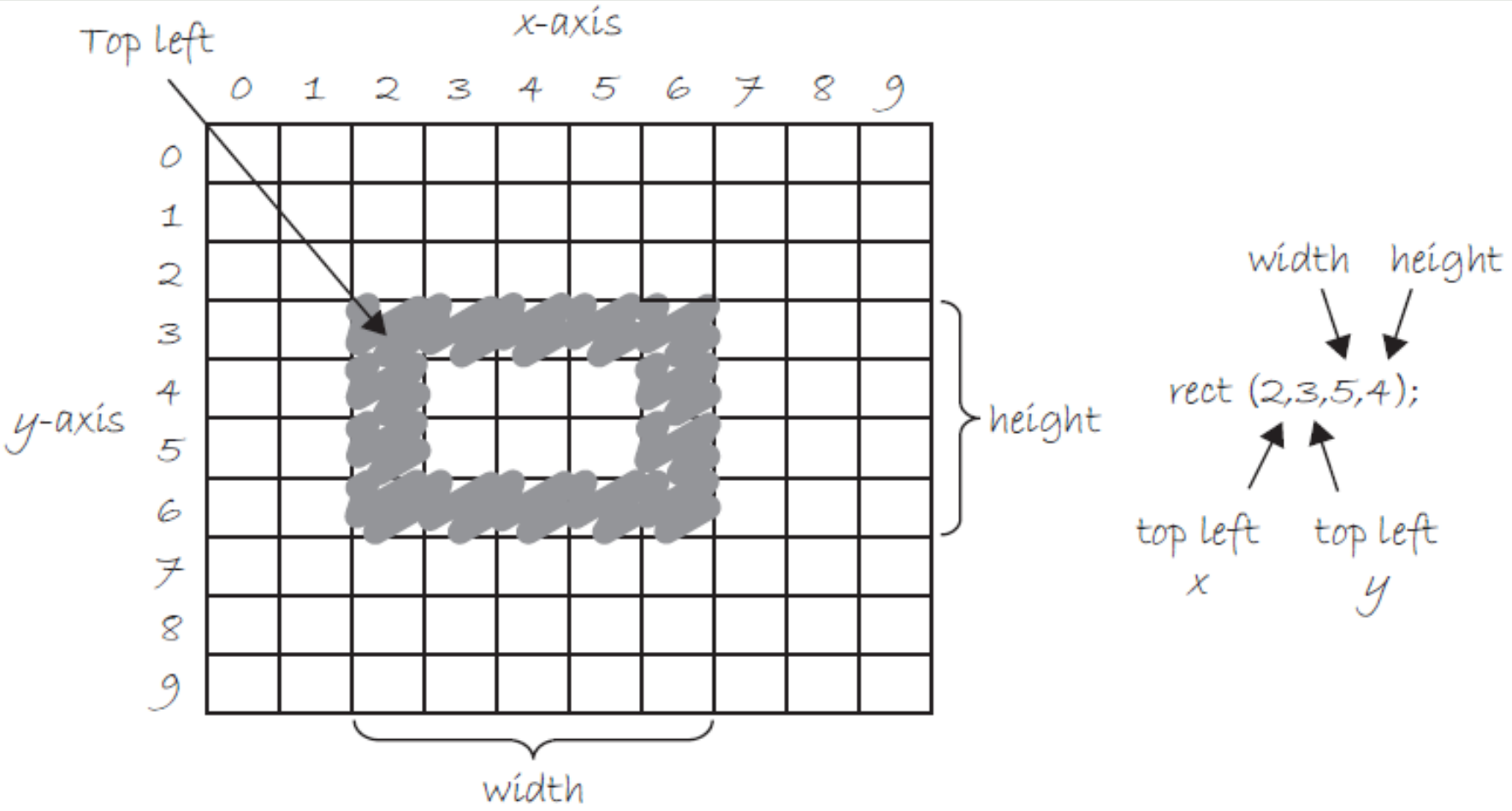


직사각형 : rect()

- 1) 빨간색 사각형의 표현은?
  -기준 (2,1), 너비 4, 높이 3
  -기준 (3.5, 2), 너비 4, 높이 3
  -(2,1), (5,3)
- 2) 파란색 사각형의 표현은?
  -기준 (3,4), 너비 6, 높이 5
  -기준 (5.5,6), 너비 6, 높이 5
  -(3,4), (8,8)

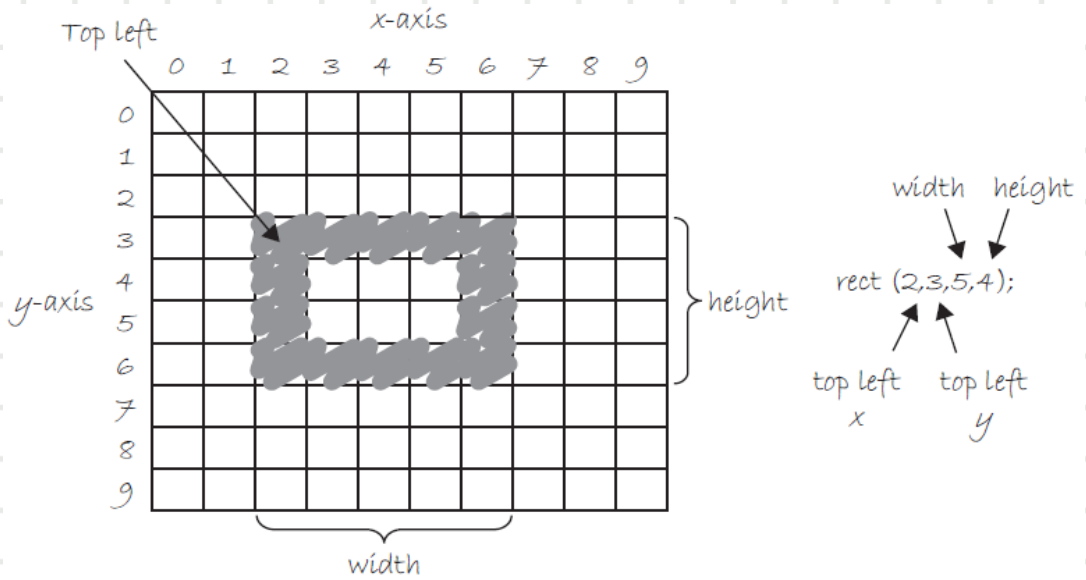


직사각형: rect(x, y, width, height)



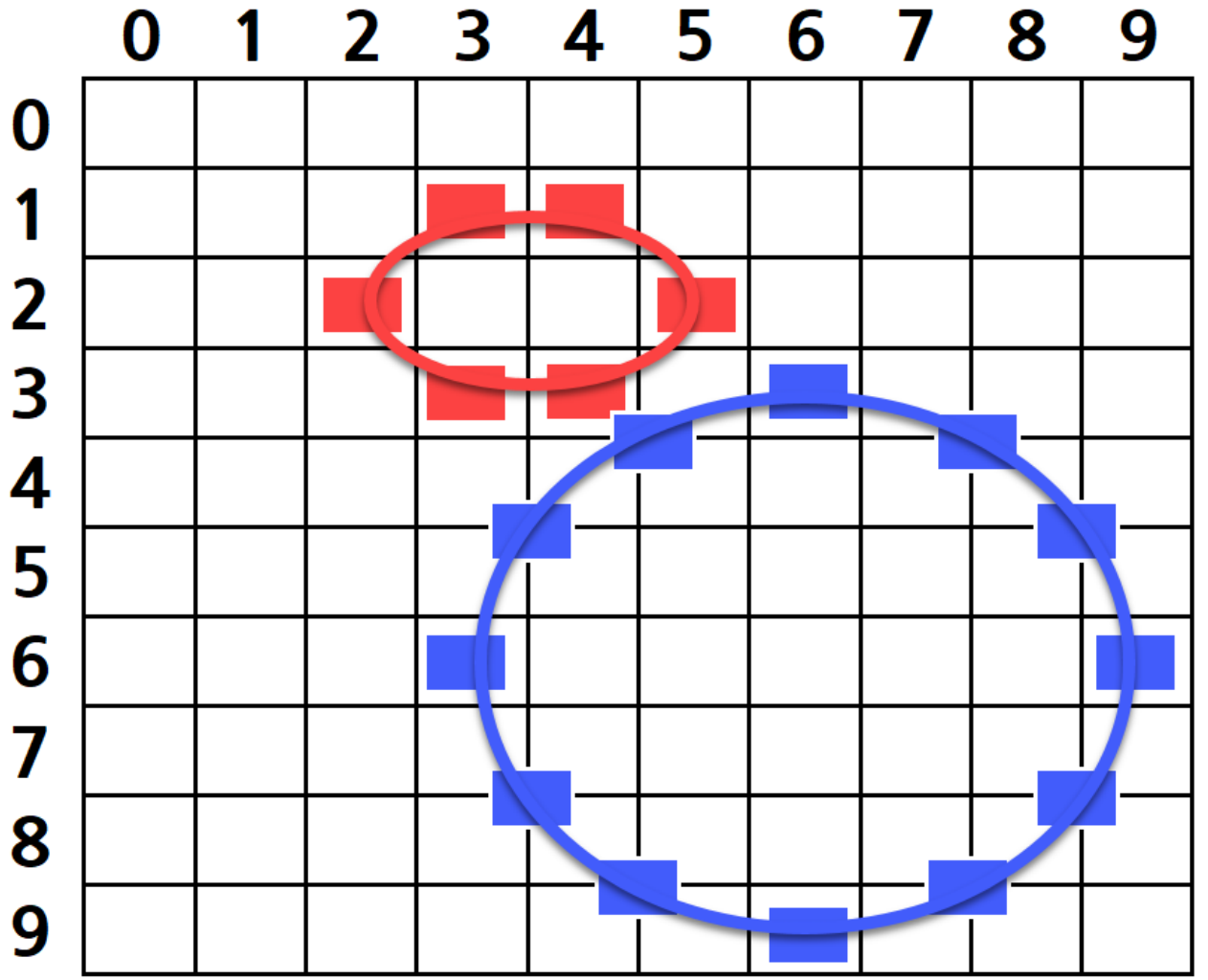
직사각형: rect(x, y, width, height)

- x: x좌표, y: 좌표,
width: 너비, height: 높이
- rectMode(): 사각형 그리는 방법
① CORNER: 좌측상단, 너비, 높이
② CENTER: 중심, 너비, 높이
③ CORNERS: 좌측상단, 우측하단
*기본값(default): CORNER



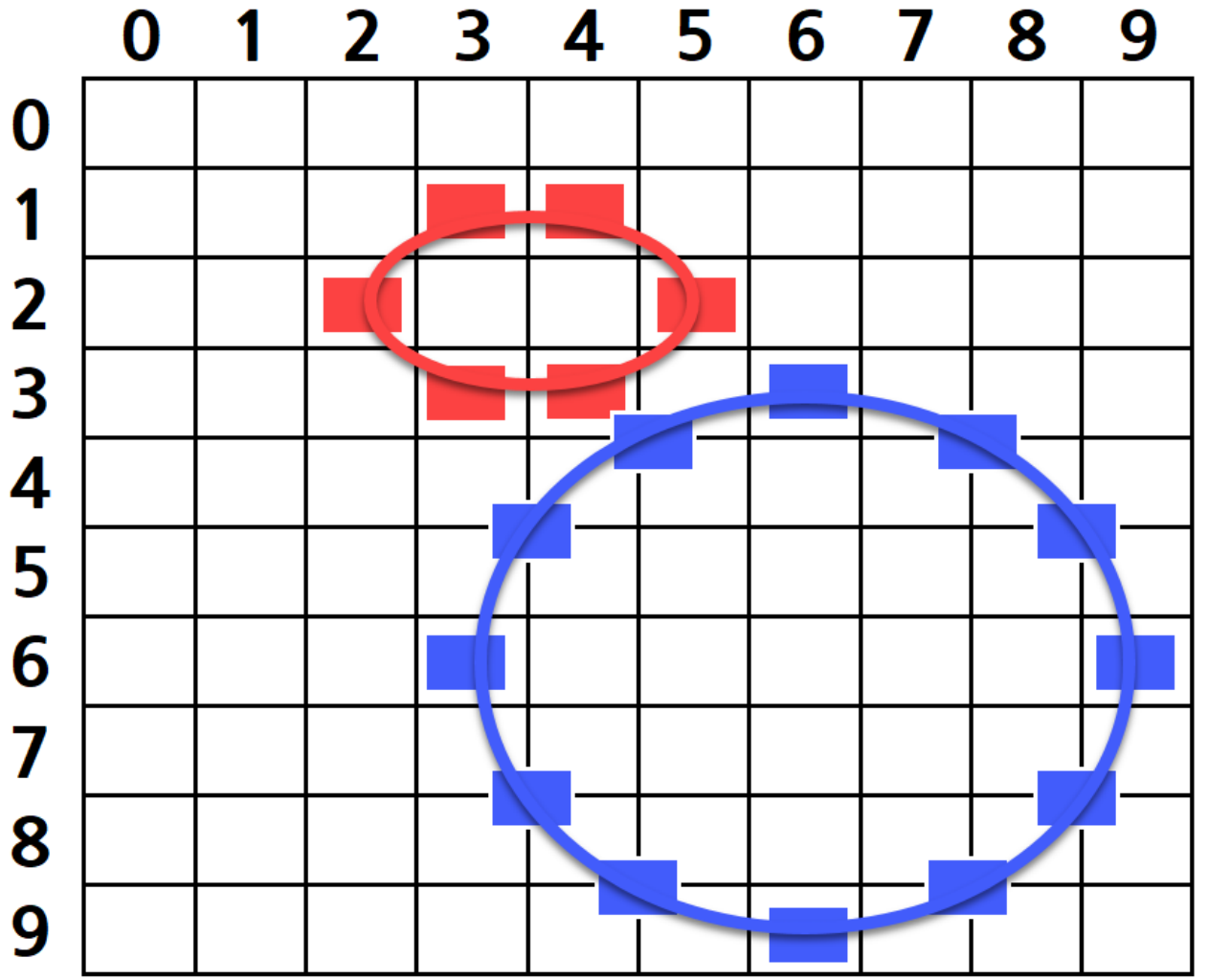
타원: ellipse()

- 1) 빨간색 타원의 표현은?
- 2) 파란색 타원의 표현은?

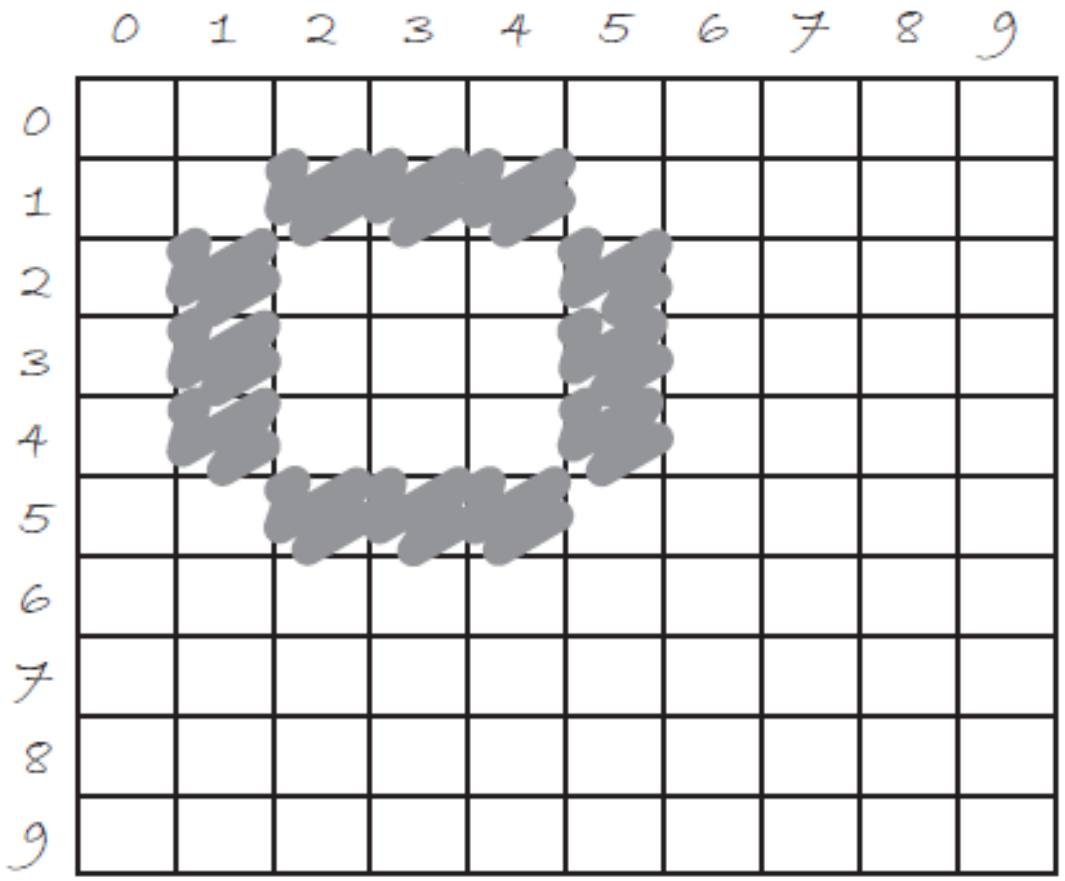


타원: ellipse()

- 1) 빨간색 타원의 표현은?
  -기준 (2,1), 너비 4, 높이 3
  -기준 (3.5, 2), 너비 4, 높이 3
  -(2,1), (5,3)
- 2) 파란색 타원의 표현은?
  -기준 (3,4), 너비 7, 높이 7
  -기준 (6,6), 너비 7, 높이 7
  -(3,4), (8,8)



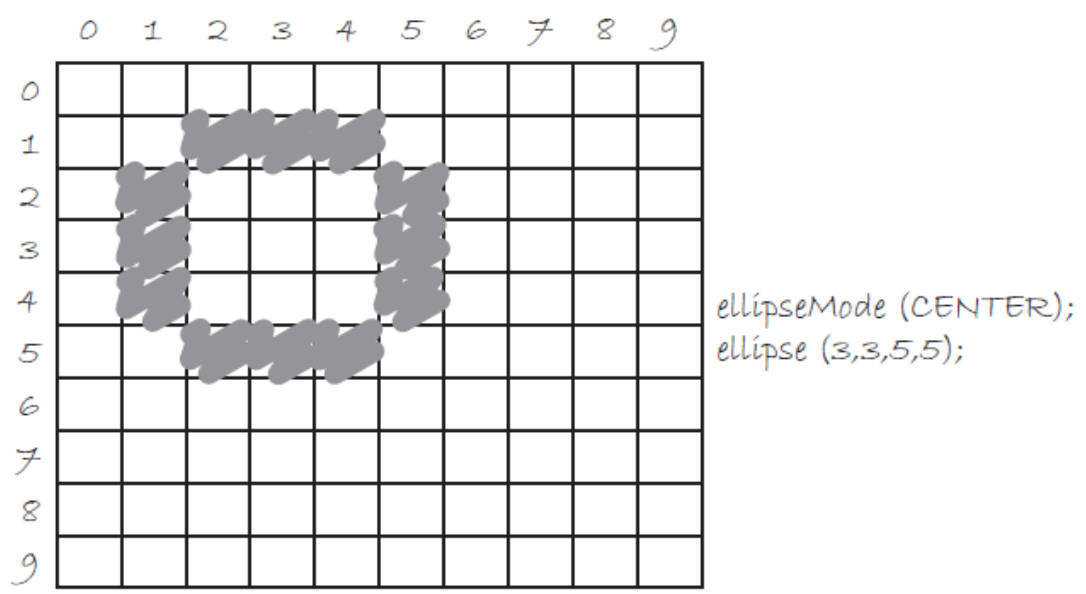
타원: ellipse(x, y, width, height)



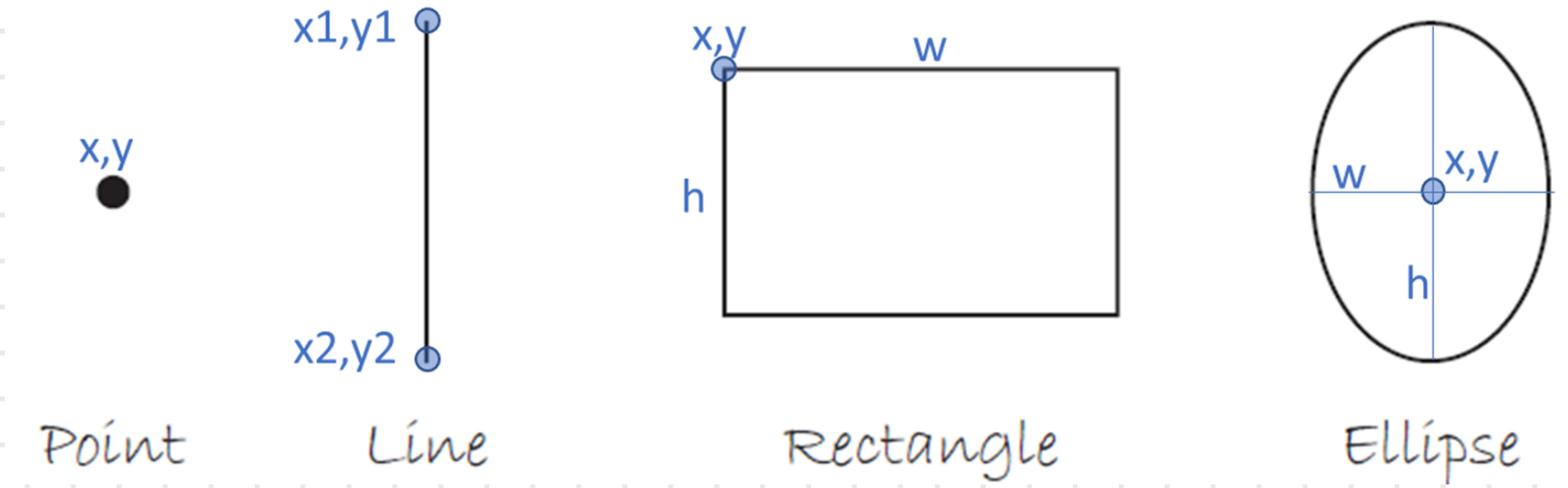

```
ellipseMode (CENTER);
ellipse (3,3,5,5);
```

타원: ellipse(x, y, width, height)

- x: x좌표, y: 좌표,
width: 너비, height: 높이
- ellipesMode(): 사각형 그리는 방법
① CORNER: 좌측상단, 너비, 높이
② CENTER: 중심, 너비, 높이
③ CORNERS: 좌측상단, 우측하단
*기본값(default): CENTER



기본 도형

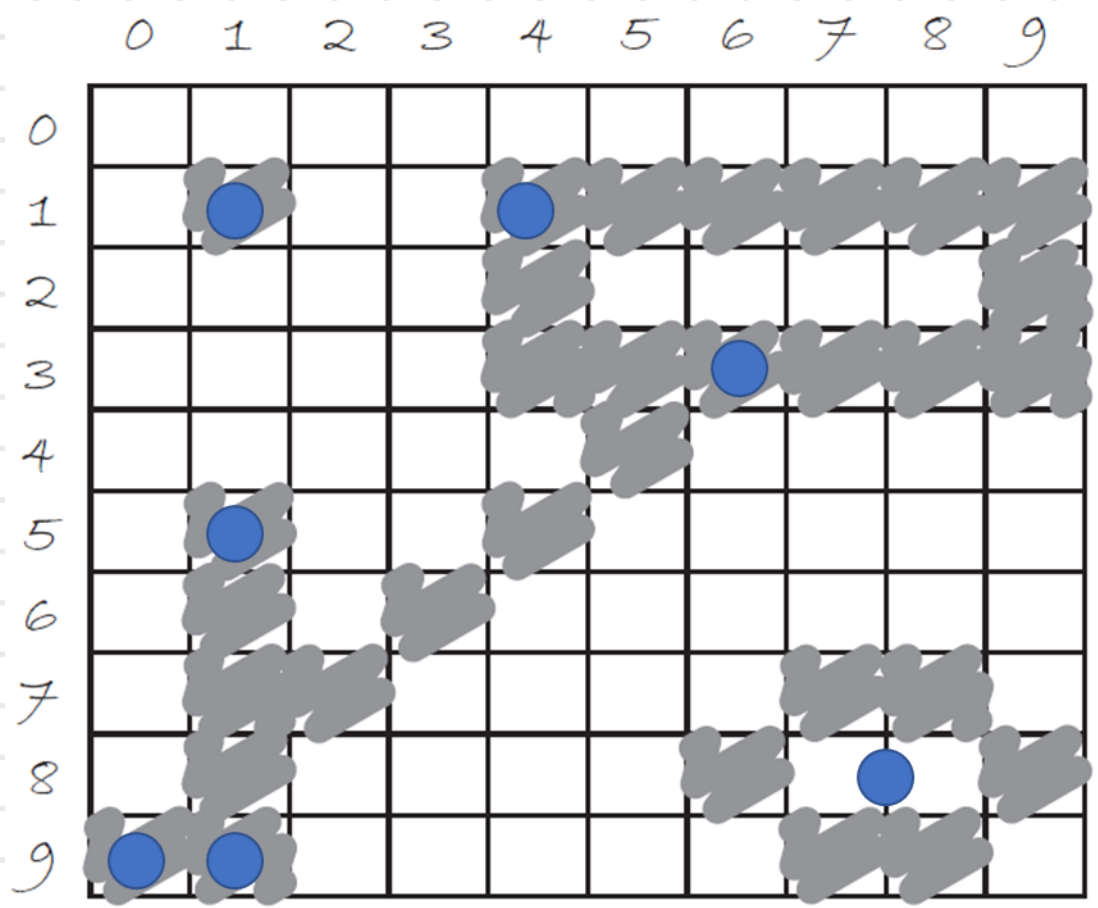


점: point(x, y);
선: line(x1, y1, x2, y2);

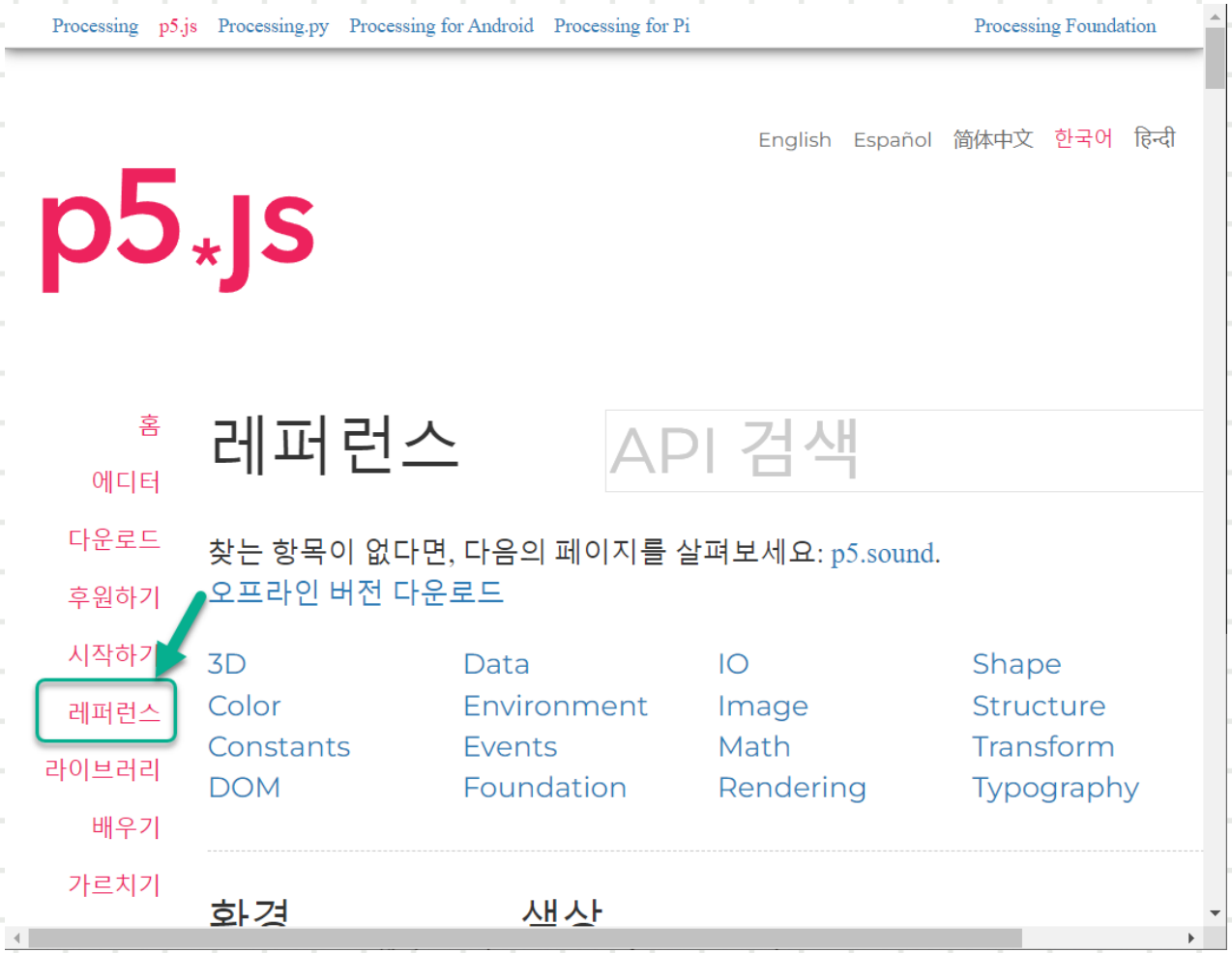
사각형: rect(x, y, width, height);
타원형: ellipse(x, y, width, height);

도형 연습

```
function setup() {
 createCanvas(400, 400);
}
function draw() {
 background(220);
 point(□, □);
 line(□, □, □, □);
 line(□, □, □, □);
 rect(□, □, □, □);
 ellipse(□, □, □, □);
}
```

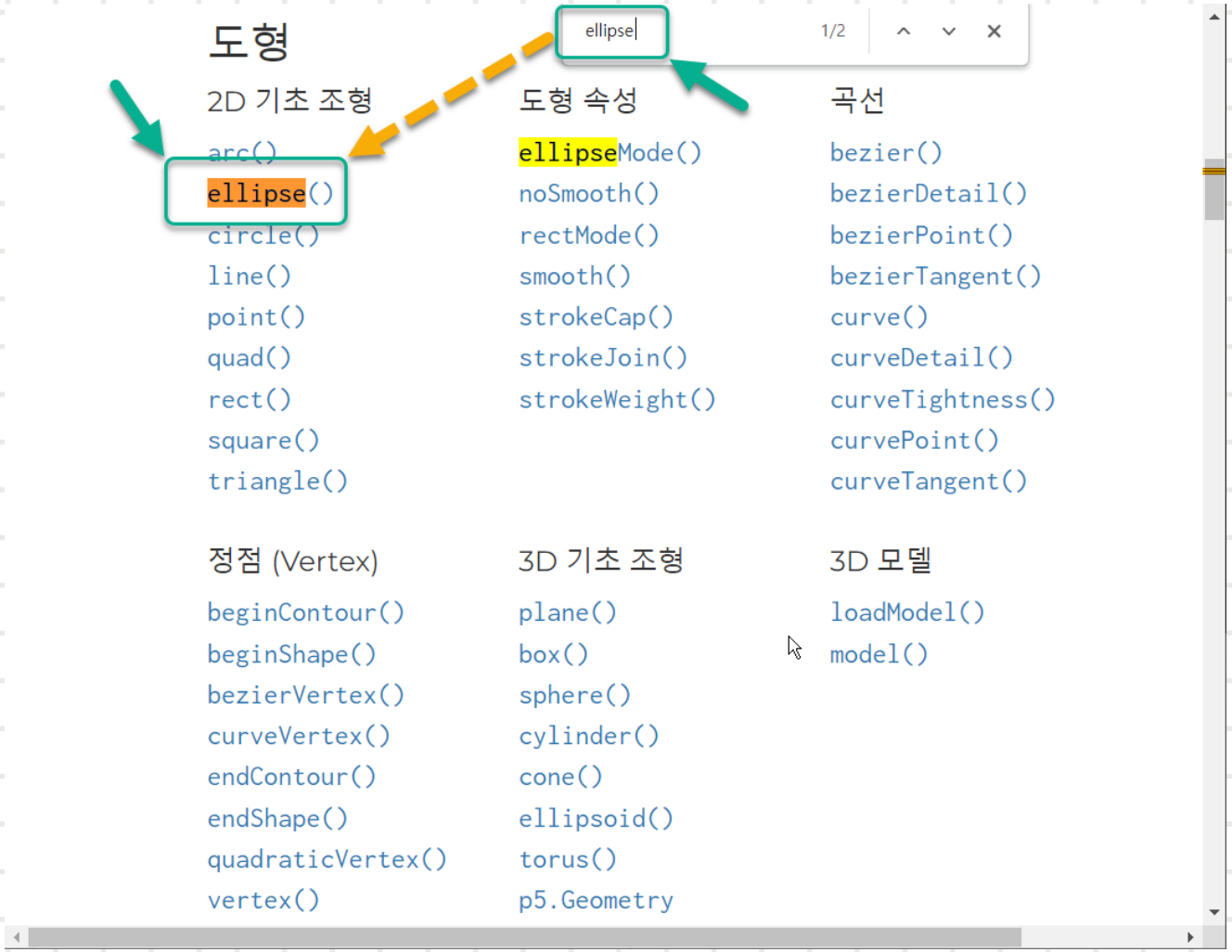


레퍼런스 사용


Processing p5.js Processing.py Processing for Android Processing for Pi Processing Foundation
English Español 简体中文 한국어 हिन्दी
p5.js
홈
에디터
다운로드
후원하기
시작하기
레퍼런스
라이브러리
배우기
가르치기
레퍼런스
API 검색
찾는 항목이 없다면, 다음의 페이지를 살펴보세요: p5.sound.
오프라인 버전 다운로드
3D
Color
Constants
DOM
Data
Environment
Events
Foundation
IO
Image
Math
Rendering
Shape
Structure
Transform
Typography
환경
생산

레퍼런스 사용

- 'ctrl+f' 키로
  검색창을 열고
  원하는 명령어를 입력하면
  쉽게 찾을 수 있다.



레퍼런스 사용

- 예제, 설명, 문법, 매개변수, 반환값 등에 대해 작성이 되어 있음.

레퍼런스 사용

- 문법 부분이 명령어를 사용하는 방법이며 매개변수는 사용되는 문자의 의미이다.

GitHub
Twitter
Instagram
Discord

ellipseMode() 함수를 이용하면 타원의 시작점을 원의 중심으로 지

정할 지의 여부를 결정할 수 있습니다.

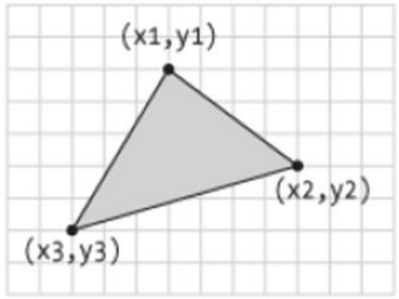
문법

```
ellipse(x, y, w, [h])

ellipse(x, y, w, h, [detail])
```

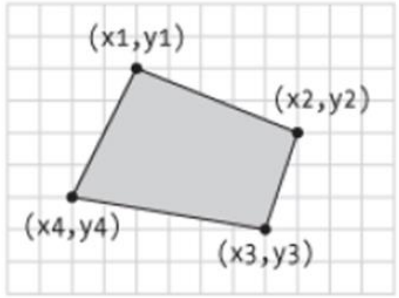
매개변수

x	숫자: 타원의 x좌표
y	숫자: 타원의 y좌표값
w	숫자: 타원의 너비값
h	숫자: (선택사항) 타원의 높이값
detail	정수: (선택사항) 타원을 몇 개의 부분으로 나누어 그릴 것인지 지정 (WebGL 모드용)

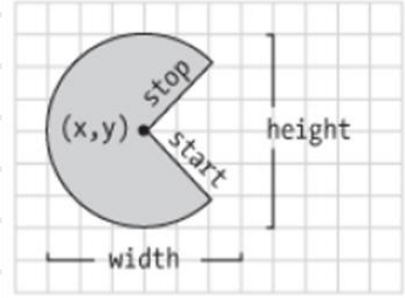
추가 도형



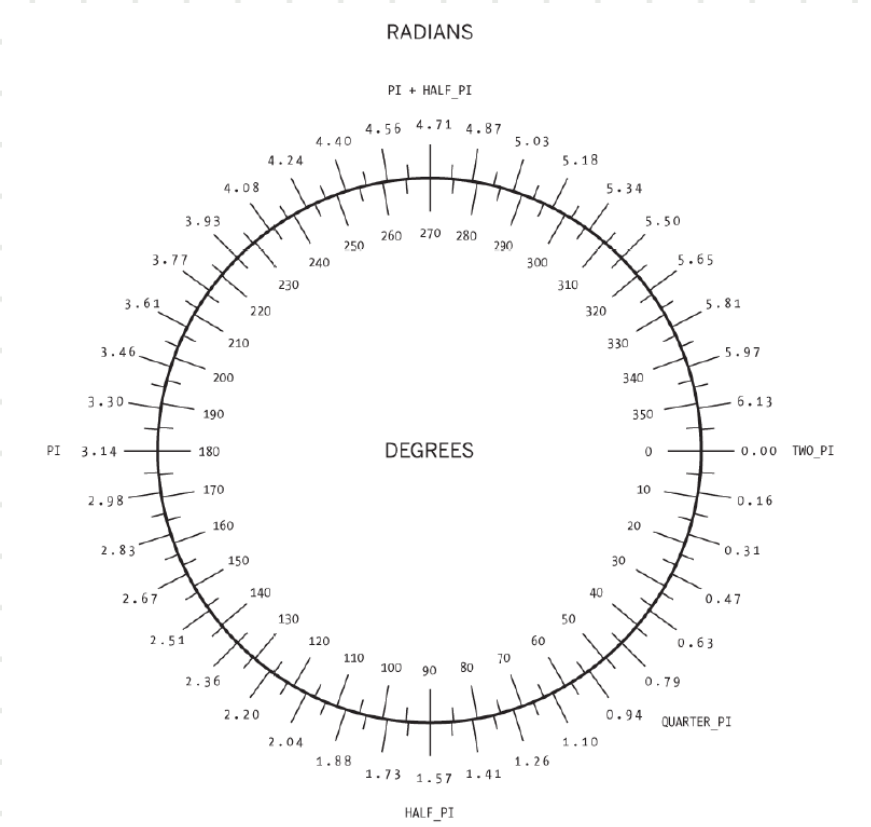

```
triangle(x1, y1, x2, y2, x3, y3)
```




```
quad(x1, y1, x2, y2, x3, y3, x4, y4)
```




```
arc(x, y, width, height, start, stop)
```



추가 도형

edit reset copy

```
quad(38, 31, 86, 20, 69, 63, 30, 76);
describe('irregular white quadrilateral with black outline');
```

edit reset copy

```
triangle(30, 75, 58, 20, 86, 75);
describe('white triangle with black outline in mid-right of canvas');
```

edit reset copy

```
arc(50, 50, 80, 80, 0, PI + QUARTER_PI);
describe('white ellipse with top right quarter missing');
```

edit reset copy

```
arc(50, 50, 80, 80, 0, PI + QUARTER_PI, OPEN);
describe('white ellipse with black outline with top right missing');
```

edit reset copy

```
arc(50, 50, 80, 80, 0, PI + QUARTER_PI, CHORD);
describe('white open arc with black outline with top right missing');
```

추가 도형

edit reset copy

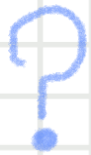
```
strokeWeight(3);
beginShape(POINTS);
vertex(30, 20);
vertex(85, 20);
vertex(85, 75);
vertex(30, 75);
endShape();
```

edit reset copy

```
createCanvas(100, 100, WEBGL);
background(240, 240, 240);
fill(237, 34, 93);
noStroke();
beginShape();
vertex(0, 35);
vertex(35, 0);
vertex(0, -35);
vertex(-35, 0);
endShape();
```

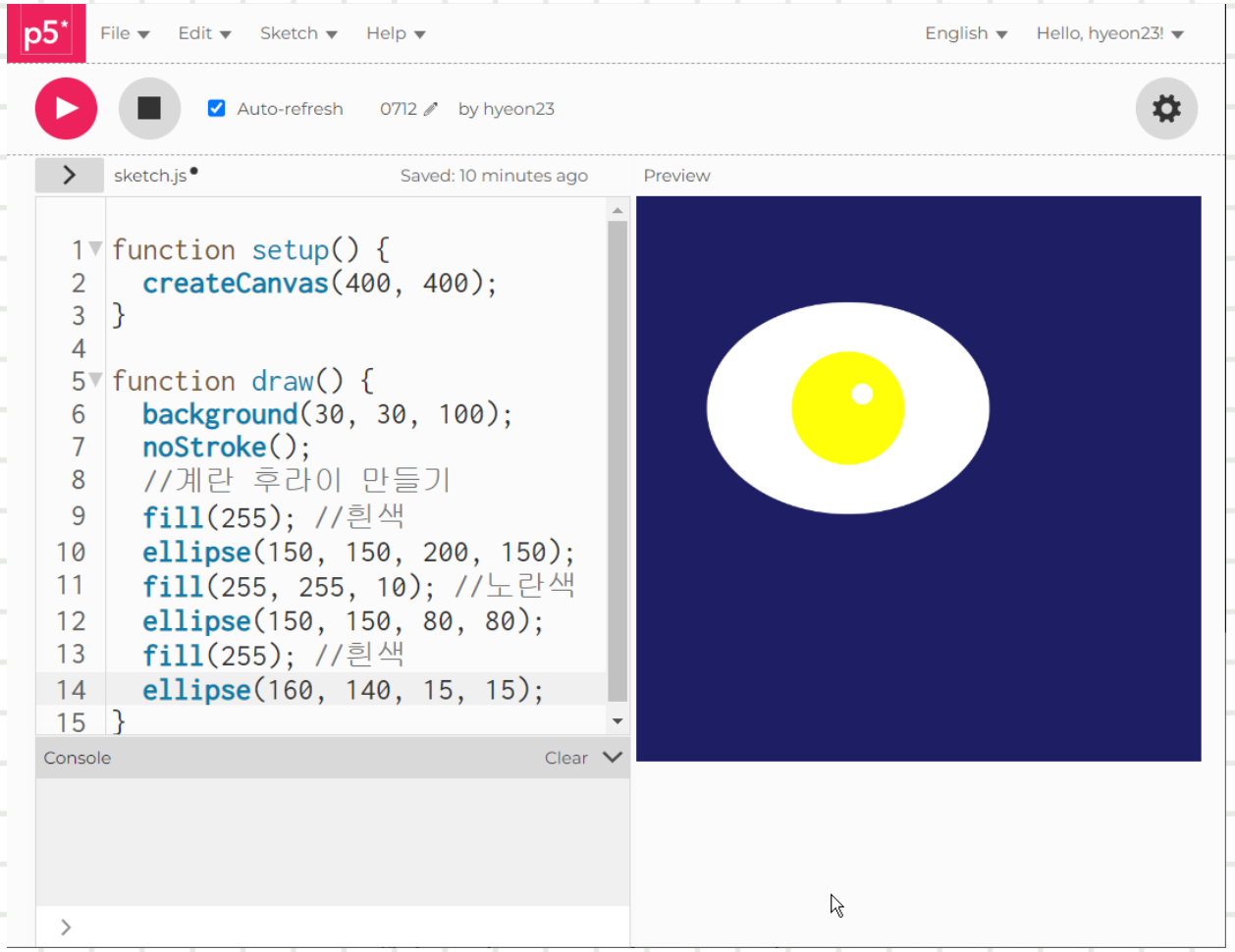
edit reset copy

```
createCanvas(100, 100, WEBGL);
background(240, 240, 240);
fill(237, 34, 93);
noStroke();
beginShape();
vertex(-10, 10);
vertex(0, 35);
vertex(10, 10);
vertex(35, 0);
vertex(10, -8);
vertex(0, -35);
vertex(-10, -8);
vertex(-35, 0);
endShape();
```



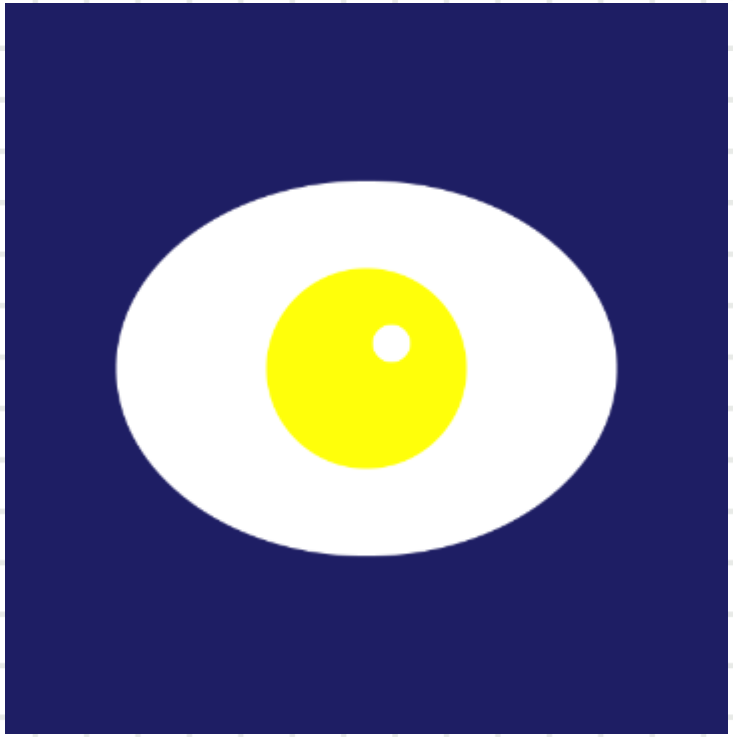
질문해 주세요.

색상 알아보기

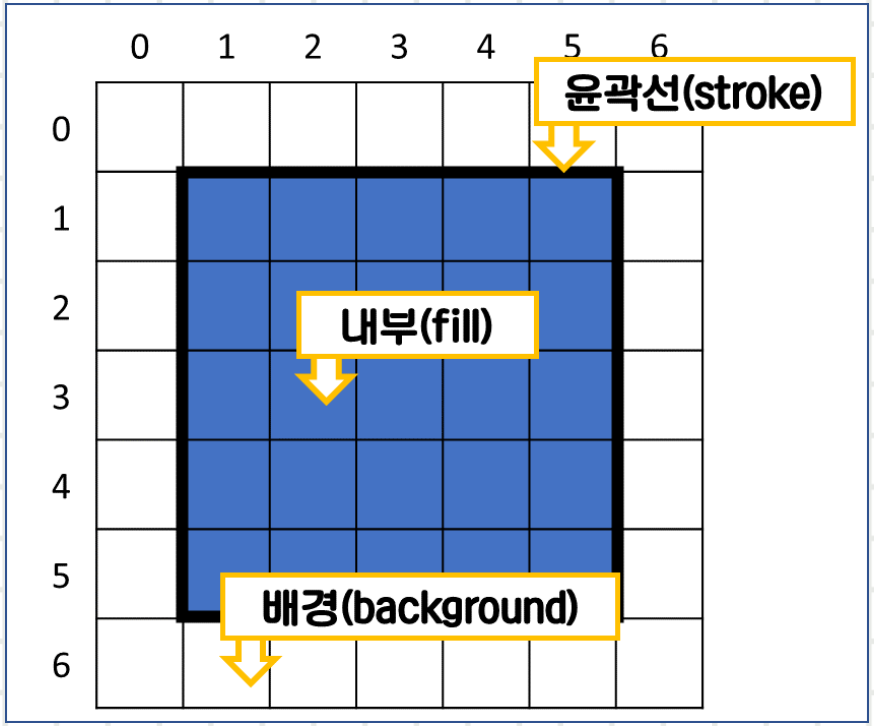

p5*
File
Edit
Sketch
Help
English
Hello, hyeon23!
Auto-refresh
0712
by hyeon23
sketch.js
Saved: 10 minutes ago
Preview
1 function setup() {
2 createCanvas(400, 400);
3 }
4
5 function draw() {
6 background(30, 30, 100);
7 noStroke();
8 //계란 후라이 만들기
9 fill(255); //흰색
10 ellipse(150, 150, 200, 150);
11 fill(255, 255, 10); //노란색
12 ellipse(150, 150, 80, 80);
13 fill(255); //흰색
14 ellipse(160, 140, 15, 15);
15 }
Console
Clear

색상 알아보기

```
function draw() {
  background(30, 30, 100);
  noStroke();
  //계란 후라이 만들기
  fill(255); //흰자
  ellipse(150, 150, 200, 150);
  fill(255, 255, 10); //노른자
  ellipse(150, 150, 80, 80);
  fill(255); //빛반사
  ellipse(160, 140, 15, 15);
}
```

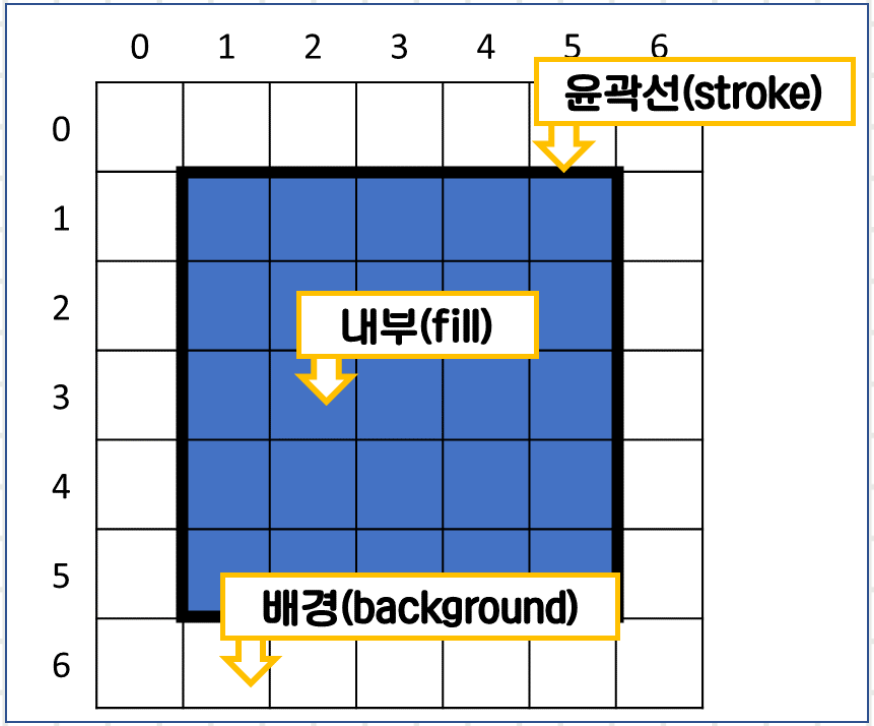


색상 관련 명령어



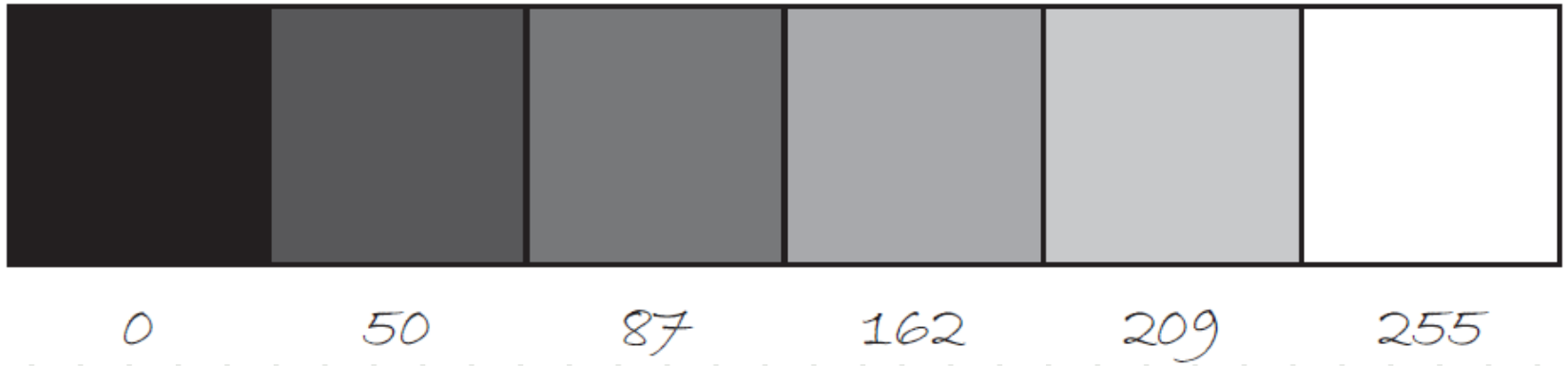
- 윤곽선 색상: stroke(); //기본값: 0(검정)
- 도형 내부 색상: fill(); //기본값: 255(흰)
- 배경 색상: background();

색상 관련 명령어



- 윤곽선 색상 제거: noStorke();
- 도형 내부 색상 제거: noFill();

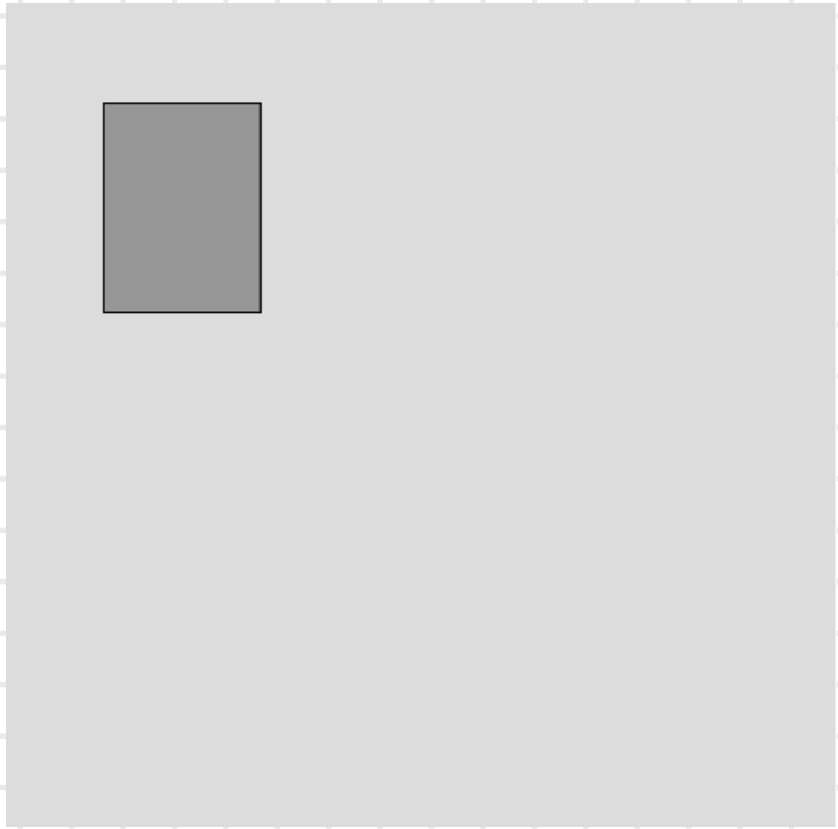
색상 - 그레이스케일(회색조)



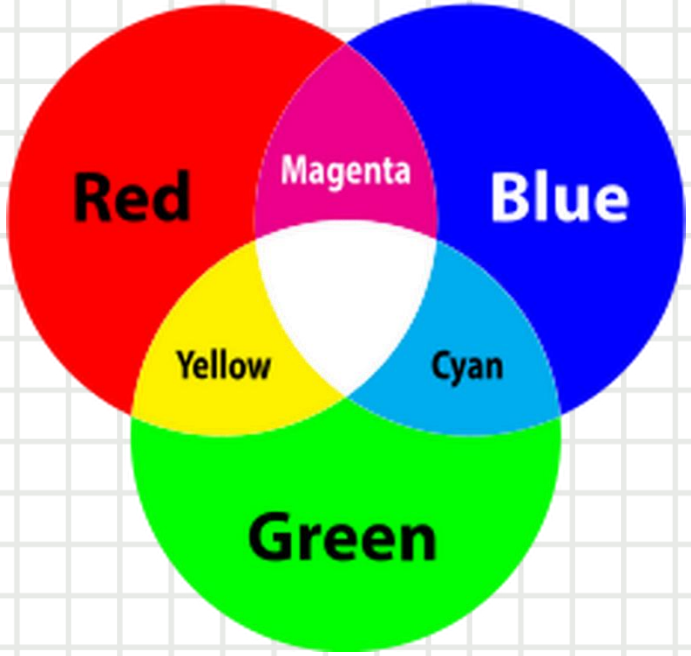
- 0(검정) ~ 255(흰)
- 윤곽선 색상: stroke(); //기본값: 0(검정)
- 도형 내부 색상: fill(); //기본값: 255(흰)
- 색상 제거: noStroke();, noFill();
- 배경 색상: background();

그레이스케일 연습

```
function draw() {
  background(220);
  stroke(0);
  fill(150);
  rect(50, 50, 75, 100);
}
```



색상 - RGB 색상



빨강(Red), 초록(Green), 파랑(Blue)

→ 빛의 3원색

- 각각 0(없음)~255(밝음) 값을 가짐

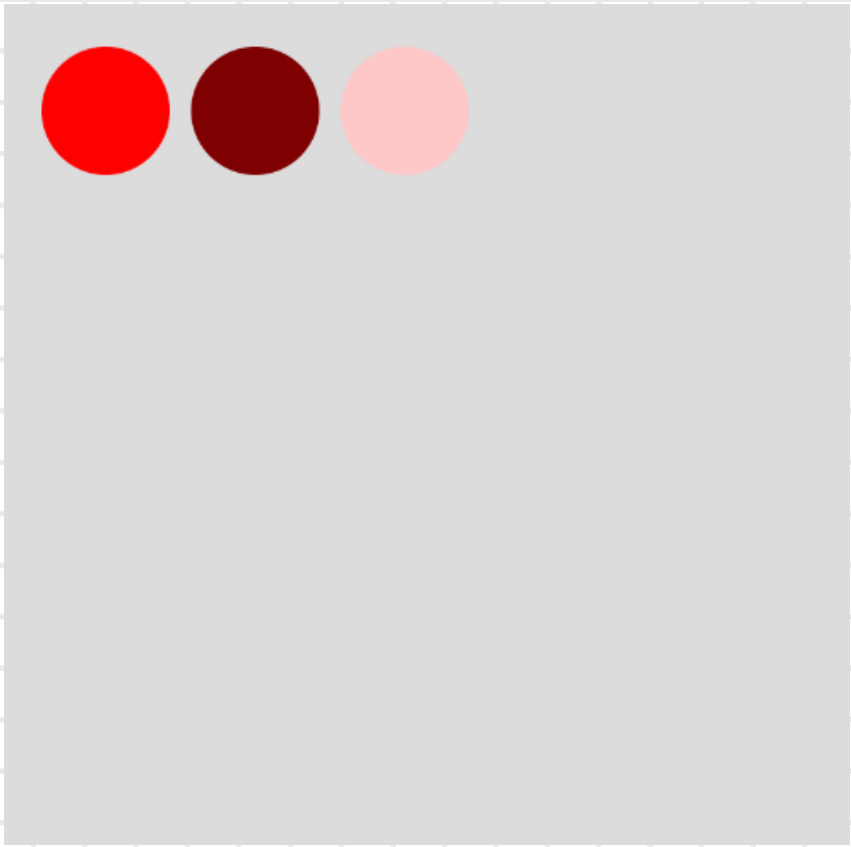
색상 - RGB 색상

Color	Color RGB
	rgb(0,0,0)
	rgb(255,0,0)
	rgb(0,255,0)
	rgb(0,0,255)
	rgb(255,255,0)
	rgb(0,255,255)
	rgb(255,0,255)
	rgb(192,192,192)
	rgb(255,255,255)

Lime #A4C400 RGB(164, 196, 0)	Green #60A917 RGB(96, 169, 23)	Emerald #008A00 RGB(0, 138, 0)	Teal #00ABA9 RGB(0, 171, 169)
Cyan #1BA1E2 RGB(27, 161, 226)	Cobalt #0050EF RGB(0, 80, 239)	Indigo #6A00FF RGB(106, 0, 255)	Violet #AA00FF RGB(170, 0, 255)
Pink #F472D0 RGB(244, 114, 208)	Magenta #D80073 RGB(216, 0, 115)	Crimson #A20025 RGB(162, 0, 37)	Red #E51400 RGB(229, 20, 0)
Orange #FA6800 RGB(250, 104, 0)	Amber #F0A30A RGB(240, 163, 10)	Yellow #E3C800 RGB(227, 200, 0)	Brown #825A2C RGB(130, 90, 44)
Olive #6D8764 RGB(109, 135, 100)	Steel #647687 RGB(100, 118, 135)	Mauve #76608A RGB(118, 96, 138)	Taupe #87794E RGB(135, 121, 78)

RGB색 연습

```
function draw() {
  background(220);
  noStroke();
  //첫번째 원
  fill(255, 0, 0);
  ellipse(50, 50, 60, 60);
  //두번째 원
  fill(127, 0, 0);
  ellipse(120, 50, 60, 60);
  //세번째 원
  fill(255, 200, 200);
  ellipse(190, 50, 60, 60); }
```



연습코드

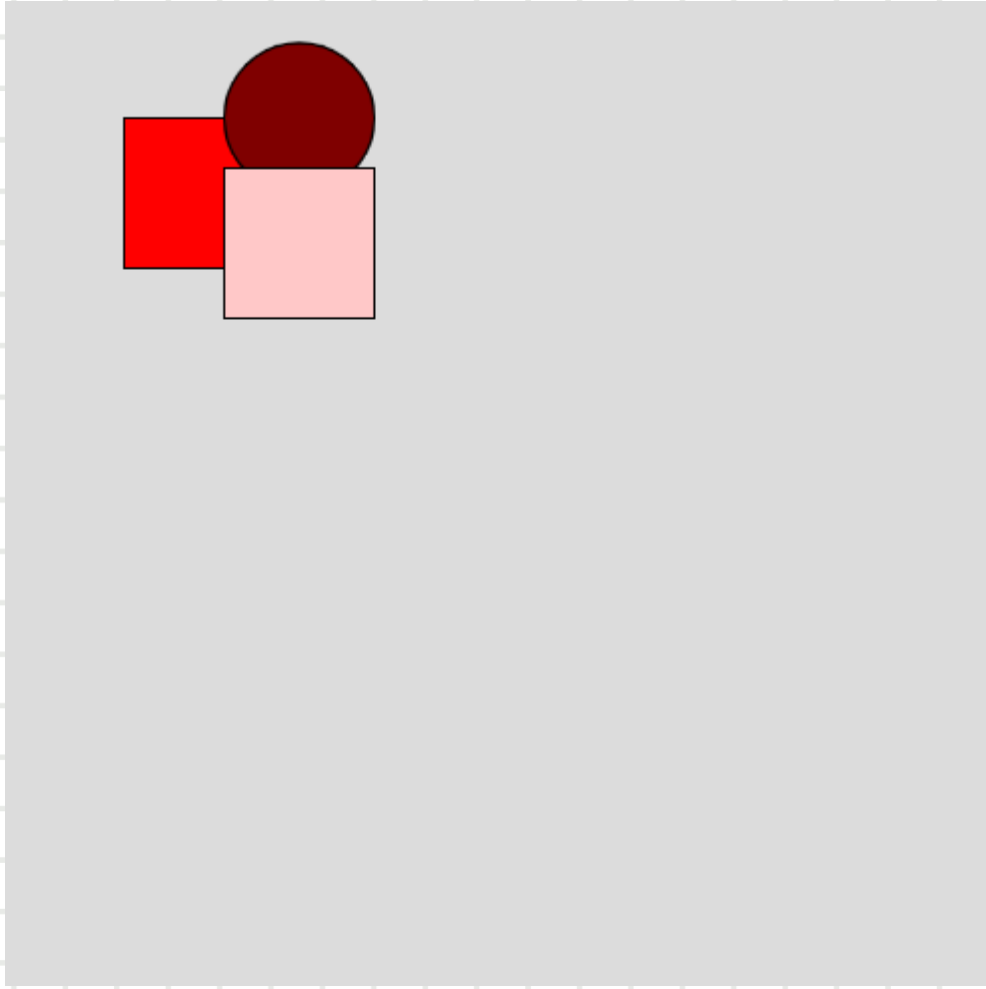
- 다음 코드를 작성하여
실행해보고
어떤 특징이 있는지
설명해보세요.

(여기에 작성하세요)

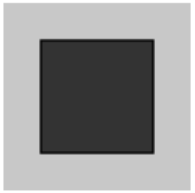
```
function draw() {
  background(220);
  fill(255, 0, 0);
  rect(50, 50, 60, 60);
  fill(127, 0, 0);
  ellipse(120, 50, 60, 60);
  fill(255, 200, 200);
  rect(90, 70, 60, 60);
}
```

연습코드

```
function draw() {
  background(220);
  fill(255, 0, 0);
  rect(50, 50, 60, 60);
  fill(127, 0, 0);
  ellipse(120, 50, 60, 60);
  fill(255, 200, 200);
  rect(90, 70, 60, 60);
}
```

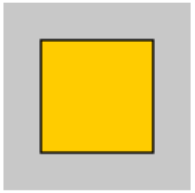


다양한 컬러모드



edit reset copy

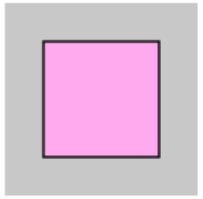
```
// Grayscale integer value
fill(51);
rect(20, 20, 60, 60);
describe('dark charcoal grey rect with black outline
in center of canvas');
```



edit reset copy

```
// R, G & B integer values
fill(255, 204, 0);
rect(20, 20, 60, 60);
describe('yellow rect with black outline in center of
canvas');
```

다양한 컬러모드



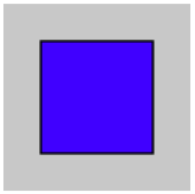
edit reset copy

```
// three-digit hexadecimal RGB notation
fill('#fae');
rect(20, 20, 60, 60);
describe('pink rect with black outline in center of
canvas');
```

edit reset copy

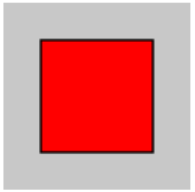
```
// six-digit hexadecimal RGB notation
fill('#222222');
rect(20, 20, 60, 60);
describe('black rect with black outline in center of
canvas');
```

다양한 컬러모드



edit reset copy

```
// H, S & B integer values
colorMode(HSB);
fill(255, 204, 100);
rect(20, 20, 60, 60);
describe('royal blue rect with black outline in center
of canvas');
```



edit reset copy

```
// Named SVG/CSS color string
fill('red');
rect(20, 20, 60, 60);
describe('red rect with black outline in center of
canvas');
```

윤곽선 관련 명령어

edit reset copy

```
// Example of different stroke weights
strokeWeight(1); // Default
line(20, 20, 80, 20);
strokeWeight(4); // Thicker
line(20, 40, 80, 40);
strokeWeight(10); // Beastly
line(20, 70, 80, 70);
```

edit reset copy

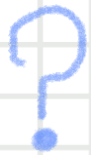
```
//Example of stroke weights
//after transformations
strokeWeight(1); // Default
line(20, 20, 80, 20);
scale(5); // Adding scale transformation
strokeWeight(1); // Resulting strokeweight is 5
line(4, 8, 16, 8); // Coordinates adjusted for scaling
```

edit reset copy

```
// Example of MITER type of joints
noFill();
strokeWeight(10.0);
strokeJoin(MITER);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(35, 80);
endShape();
```

edit reset copy

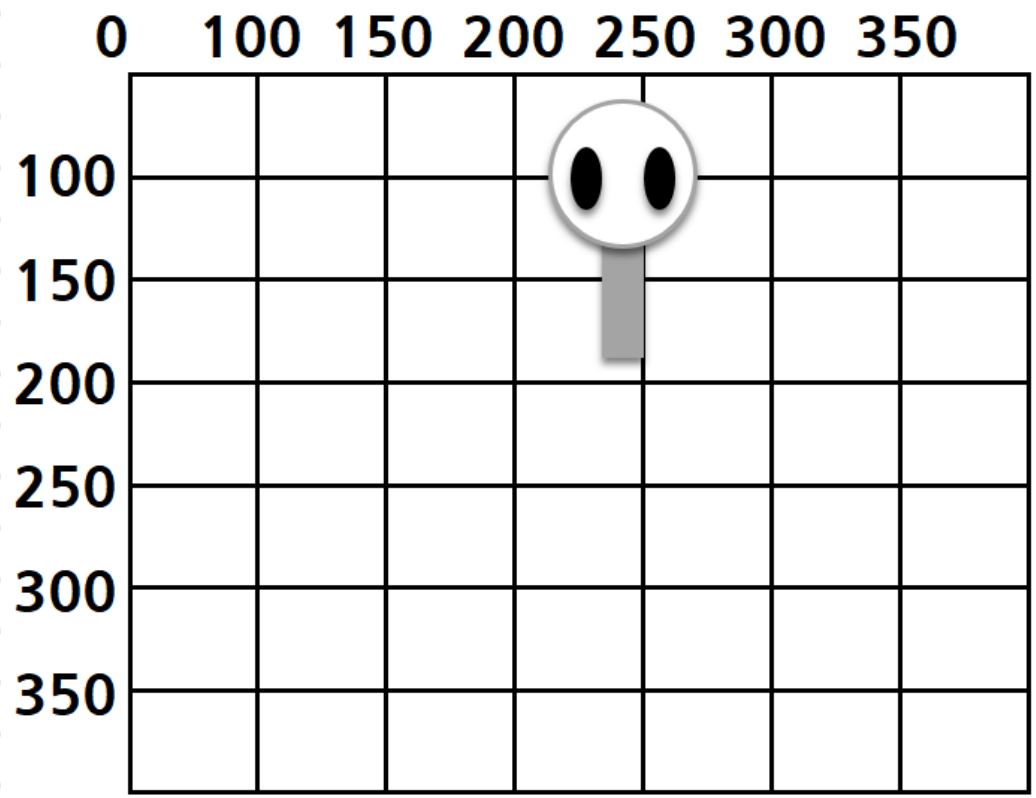
```
// Example of BEVEL type of joints
noFill();
strokeWeight(10.0);
strokeJoin(BEVEL);
beginShape();
vertex(35, 20);
vertex(65, 50);
vertex(35, 80);
endShape();
```



질문해 주세요.

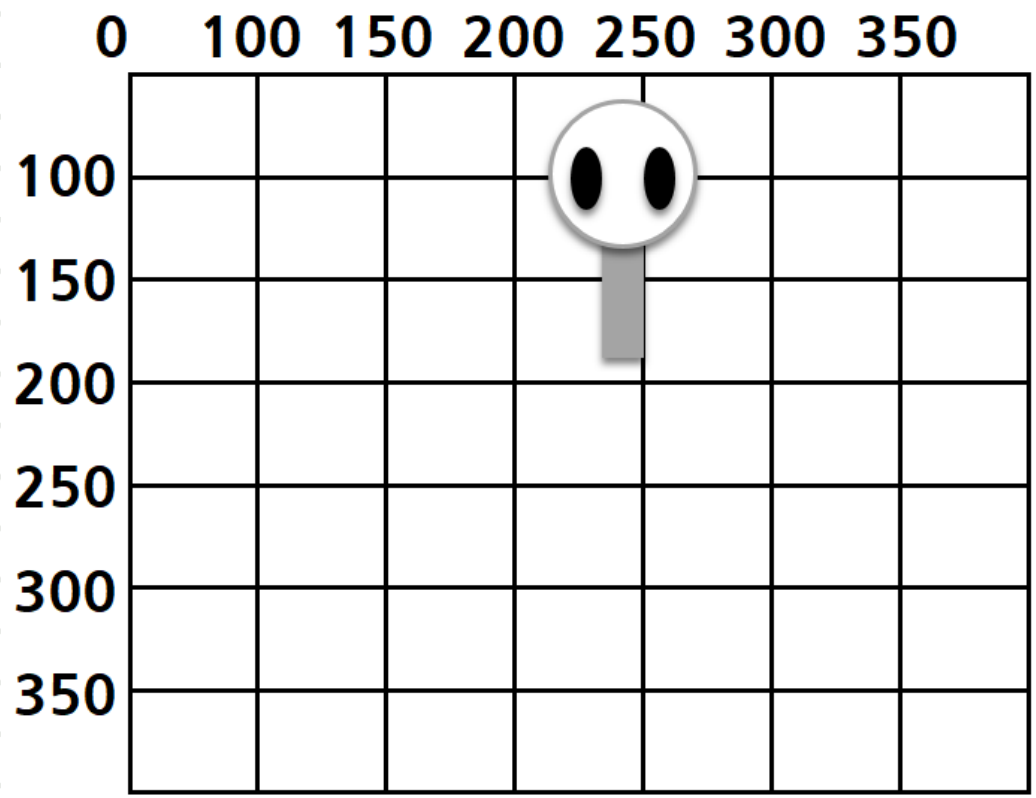
Zoog 그리기

```
// Draw Zoog
background(255);
rectMode(CENTER);

// Zoog body (rect 20*100)

// Zoog head (ellipse 60*60)

// Zoog eyes  (ellipse 16*32)

// Zoog legs (line)
```



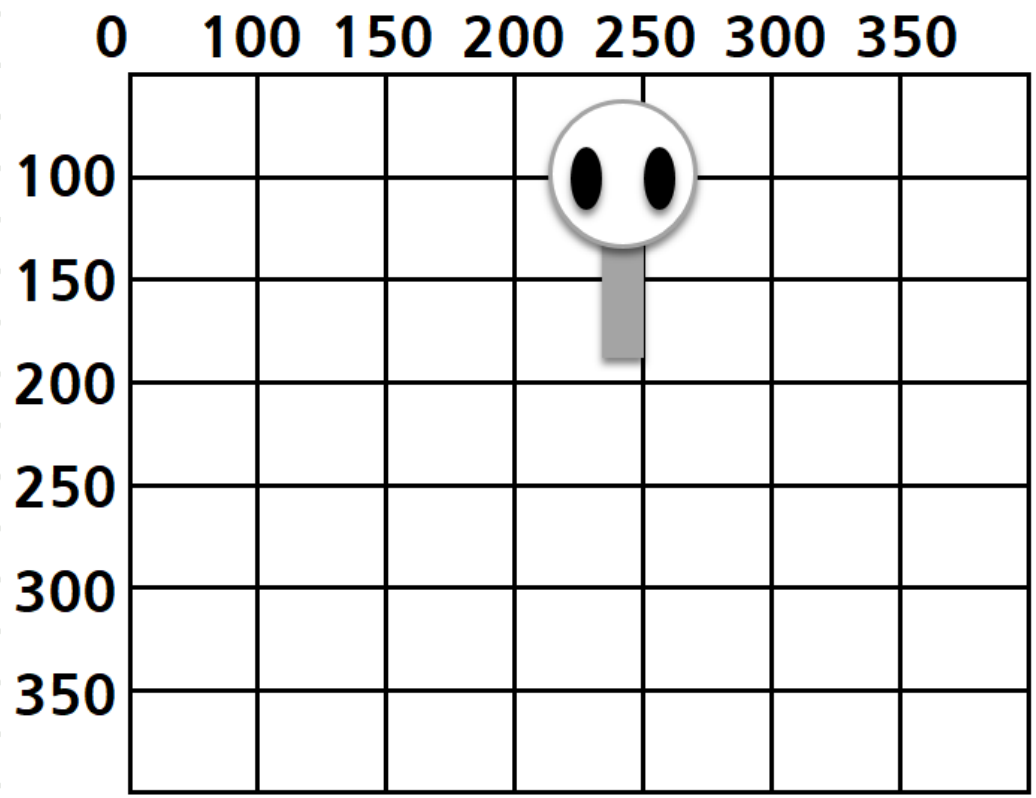
Zoog 그리기

```
// Draw Zoog
background(255);
rectMode(CENTER);
// Zoog body (rect 20*100)
stroke(0);
fill(150);
rect(240, 145, 20, 100);
// Zoog head (ellipse 60*60)
// Zoog eyes  (ellipse 16*32)
// Zoog legs (line)
```



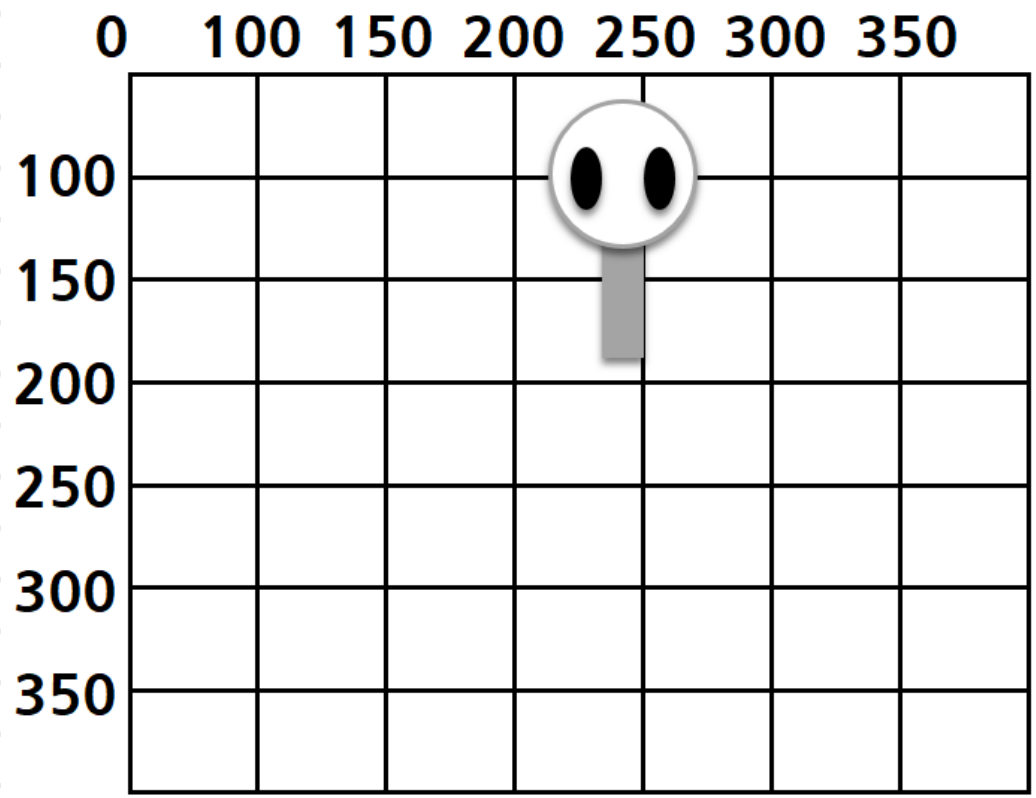
Zoog 그리기

```
// Draw Zoog
background(255);
rectMode(CENTER);
// Zoog body (rect 20*100)
// Zoog head (ellipse 60*60)
fill(255);
ellipse(240, 115, 60, 60);
// Zoog eyes  (ellipse 16*32)
// Zoog legs (line)
```



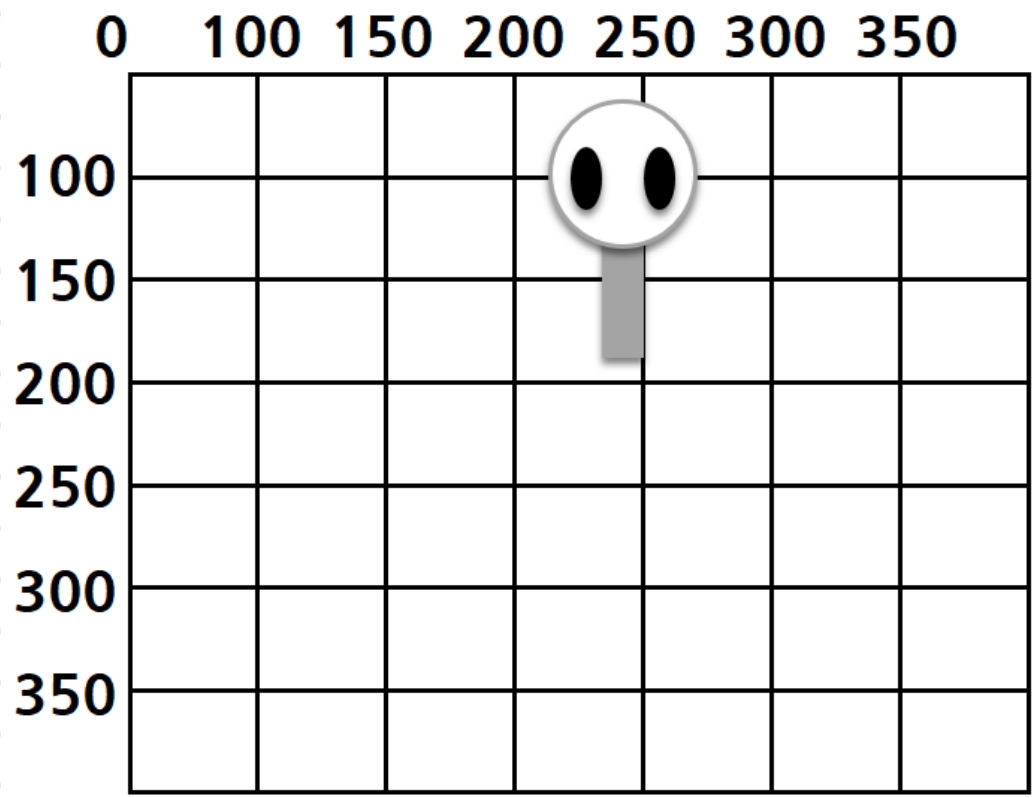
Zoog 그리기

```
// Draw Zoog
background(255);
rectMode(CENTER);
// Zoog body (rect 20*100)
// Zoog head (ellipse 60*60)
// Zoog eyes  (ellipse 16*32)
fill(0);
ellipse(221, 115, 16, 32);
ellipse(259, 115, 16, 32);
// Zoog legs (line)
```



Zoog 그리기

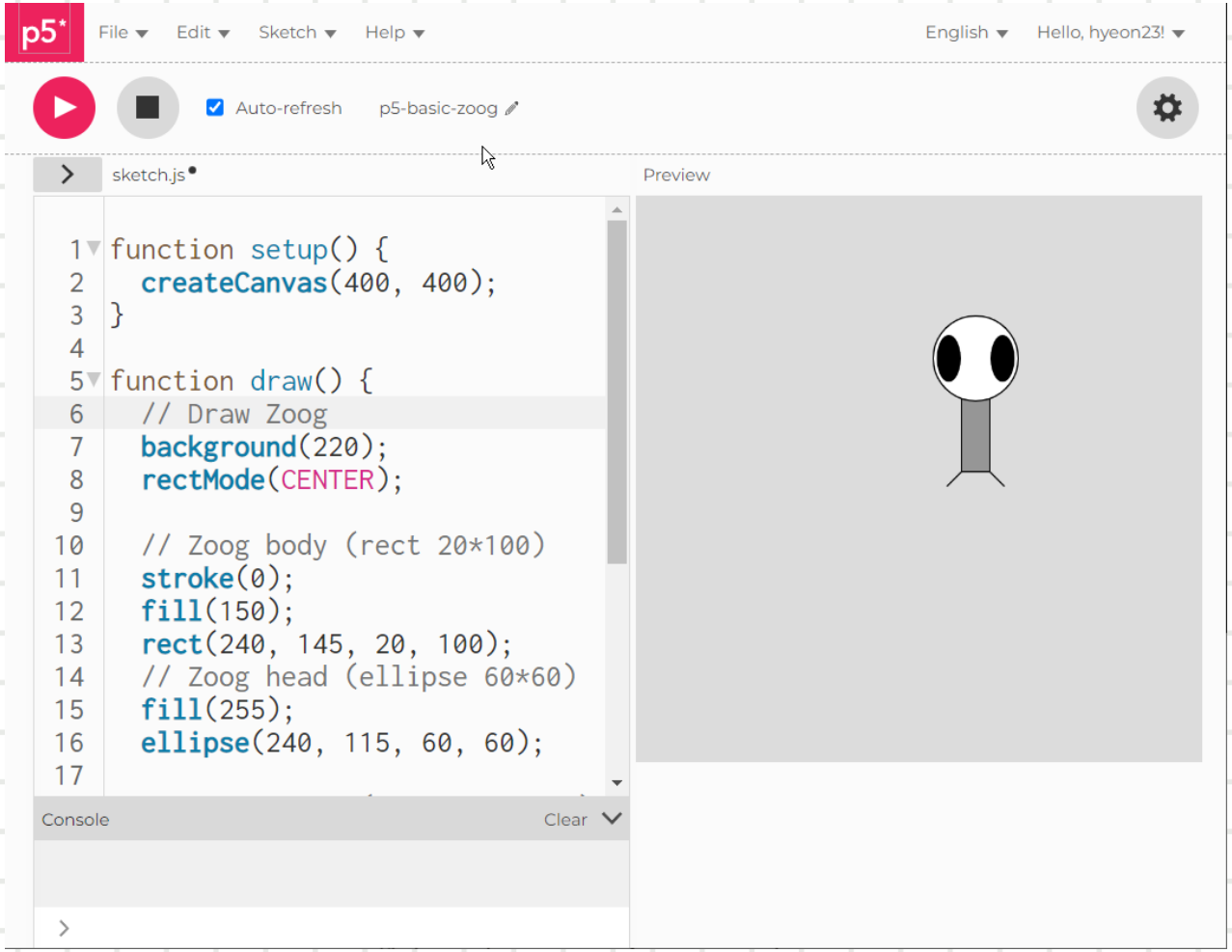
```
// Draw Zoog
background(255);
rectMode(CENTER);
// Zoog body (rect 20*100)
// Zoog head (ellipse 60*60)
// Zoog eyes  (ellipse 16*32)
// Zoog legs (line)
stroke(0);
line(230, 195, 220, 205);
line(250, 195, 260, 205);
```



Zoog 그리기

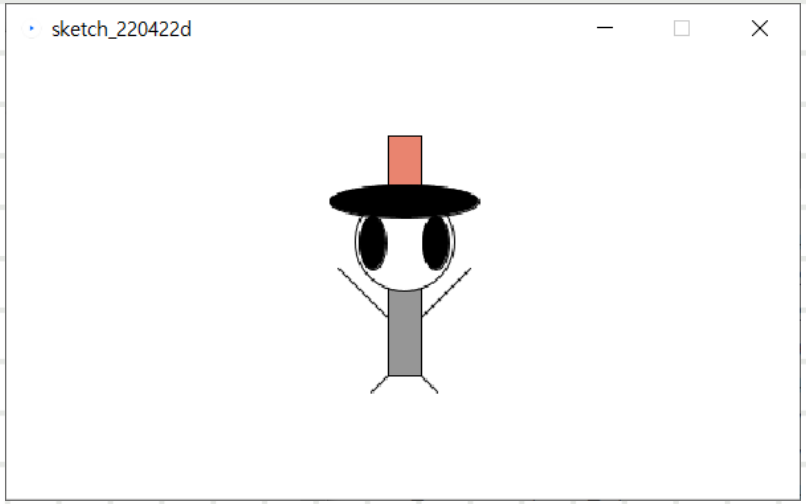
```
function draw() {
// createCanvas(480, 270);
// Draw Zoog
 background(255);
 rectMode(CENTER);

// Zoog body (rect 20*100)
stroke(0);
fill(150);
rect(240, 145, 20, 100);
// Zoog head (ellipse 60*60)
fill(255);
ellipse(240, 115, 60, 60);

// Zoog eyes  (ellipse 16*32)
fill(0);
ellipse(221, 115, 16, 32);
ellipse(259, 115, 16, 32);

// Zoog legs (line)
stroke(0);
line(230, 195, 220, 205);
line(250, 195, 260, 205);  }
```

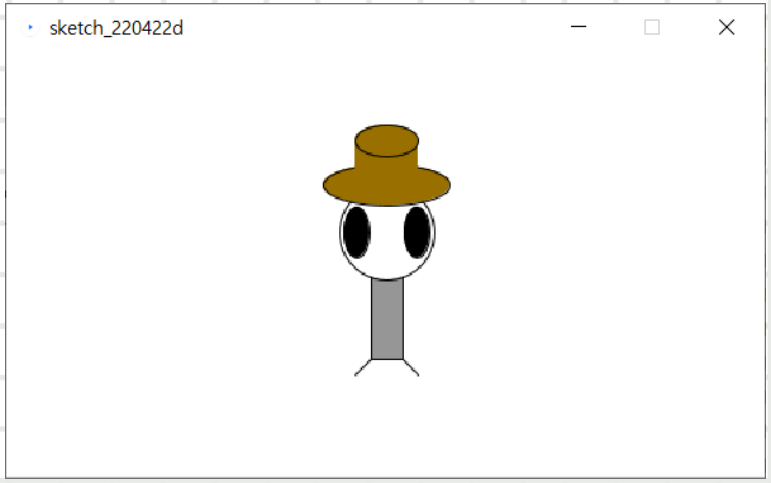
Zoog 그리기

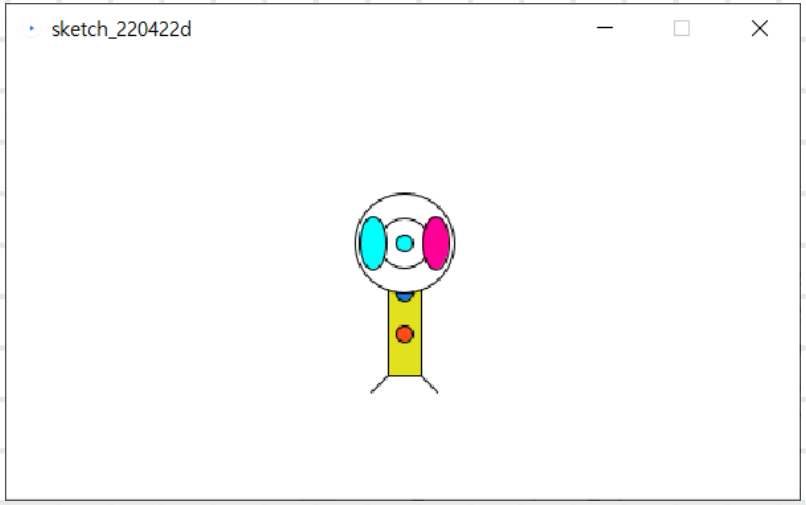

p5*
File
Edit
Sketch
Help
English
Hello, hyeon23!
Auto-refresh
p5-basic-zoog
sketch.js
Preview
1 function setup() {
2 createCanvas(400, 400);
3 }
4
5 function draw() {
6 // Draw Zoog
7 background(220);
8 rectMode(CENTER);
9
10 // Zoog body (rect 20*100)
11 stroke(0);
12 fill(150);
13 rect(240, 145, 20, 100);
14 // Zoog head (ellipse 60*60)
15 fill(255);
16 ellipse(240, 115, 60, 60);
17
Console
Clear

Zoog 꾸미기

- Zoog를 어떻게 꾸미면 좋을지 생각해보고 특징을 작성해봅시다.
- 예1. 모자 만들어주기
  예2. 손 만들어주기
  예3. 신발 만들어주기





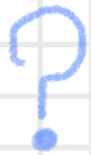


나만의 Zoog 특징 작성하기

(여기에 작성하세요)

나만의 Zoog 꾸미기

나만의 Zoog 발표



질문해 주세요.

p5.js 용어 및 기본구조

- 스케치: 프로세싱 프로그램
- 스케치북: 스케치를 저장하는 폴더
- 코딩하기
  - function setup() : 프로그램 실행 시 가장 먼저 1번 실행됨
  - function draw() : 프로그램 실행 시 여러 번 반복 실행됨
  - 주석처리: 프로그램이 실행될 때 무시하는 텍스트
    //(라인 하나 주석), /*, */(여러 라인 주석)
    한글 주석은 환경설정에서 한글 폰트로 변경

p5.js 용어 및 기본구조

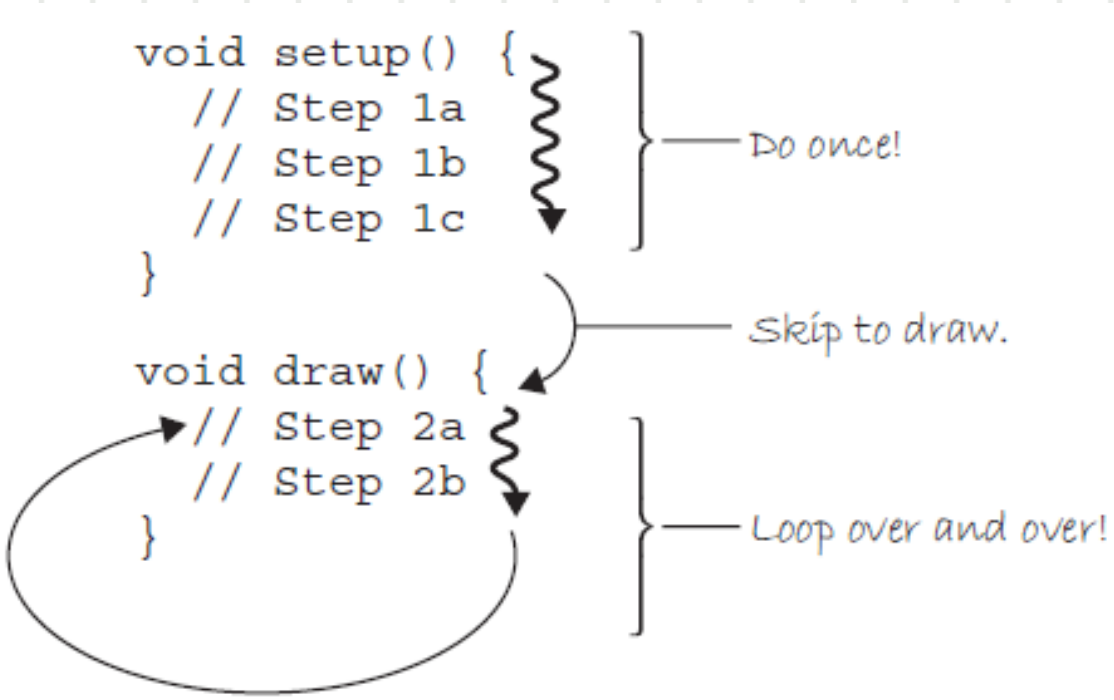
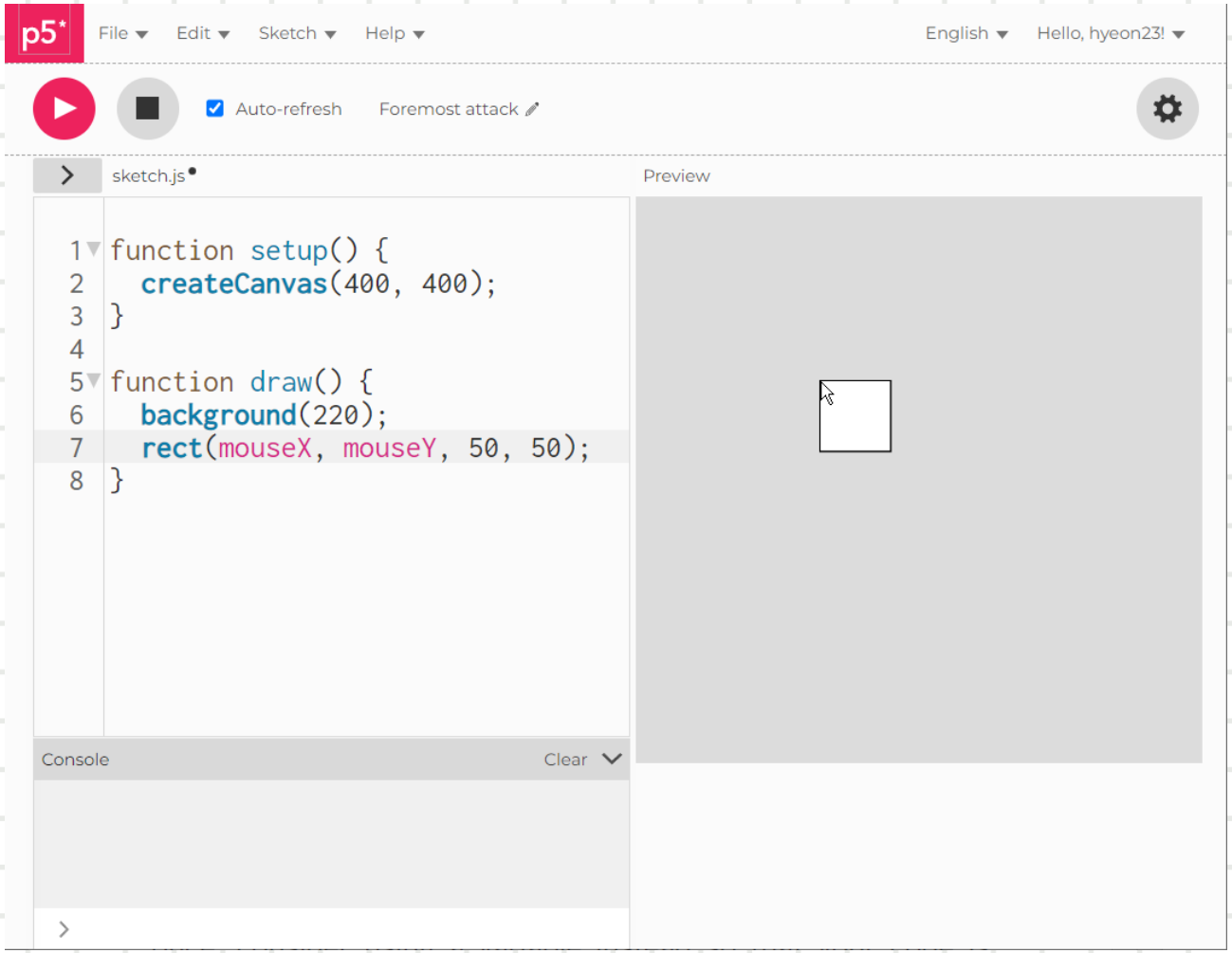


fig. 3.2

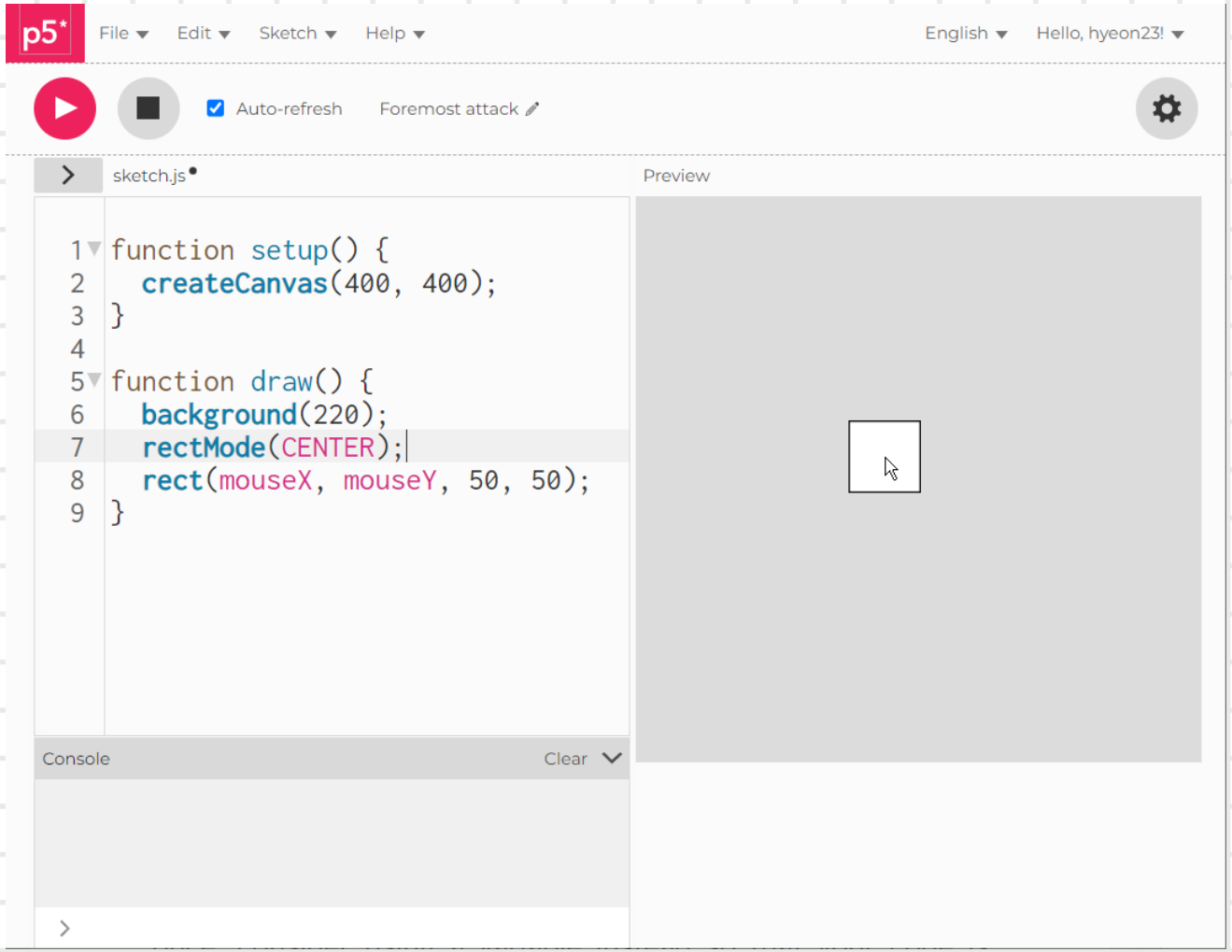
p5.js 키보드&마우스 관련

- 마우스 커서 좌표 관련 시스템 변수
  -mouseX: 현재 마우스 커서의 x 좌표
  -mouseY: 현재 마우스 커서의 y 좌표
- 키보드 입력 관련 시스템 변수
  -key: 아스키 코드 기반의 키 입력 감지
  -keyCode: UP, DOWN, LEFT, RIGHT 등 특수키 입력 감지

p5.js 키보드&마우스 관련1



p5.js 키보드&마우스 관련2



마우스를 따라오는 Zoog 만들기

- 다음 내용에 해당하는 값을 찾아서 작성하세요.
  1) zoog에서 위치 관련된 값 찾기
  2) X좌표 값은 mouseX로, Y좌표 값은 mouseY로 수정
  3) 도형의 상대적인 위치 이해하고 좌표값 보정

p5.js 주석

- 라인 주석: //기호가 시작된 곳부터 해당 라인의 끝까지가 주석문이 된다.

- 블록 주석: /* 가 블록주석의 시작이고 */가 블록주석의 끝이다.

Zoog 기본코드

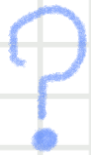
```
function draw() {
// createCanvas(480, 270);
// Draw Zoog
 background(255);
 rectMode(CENTER);

// Zoog body (rect 20*100)
stroke(0);
fill(150);
rect(240, 145, 20, 100);
// Zoog head (ellipse 60*60)
fill(255);
ellipse(240, 115, 60, 60);

// Zoog eyes  (ellipse 16*32)
fill(0);
ellipse(221, 115, 16, 32);
ellipse(259, 115, 16, 32);

// Zoog legs (line)
stroke(0);
line(230, 195, 220, 205);
line(250, 195, 260, 205);  }
```

마우스를 따라오는 Zoog 수정하기

```
// Zoog body (x 240, y 145)
rect(____, ____, 20, 100);
// Zoog head (x 240, y 115)
ellipse(____, ____, 60, 60);
// Zoog eyes (x 221, y 115) (x 259, y 115)
ellipse(____, ____, 16, 32);
ellipse(____, ____, 16, 32);
// Zoog legs (x1 230, y1 195, x2 220, y2 205), (x1 250, y1 195, x2 260, y2 205)
line(____, ____, ____, ____);
line(____, ____, ____, ____);
```

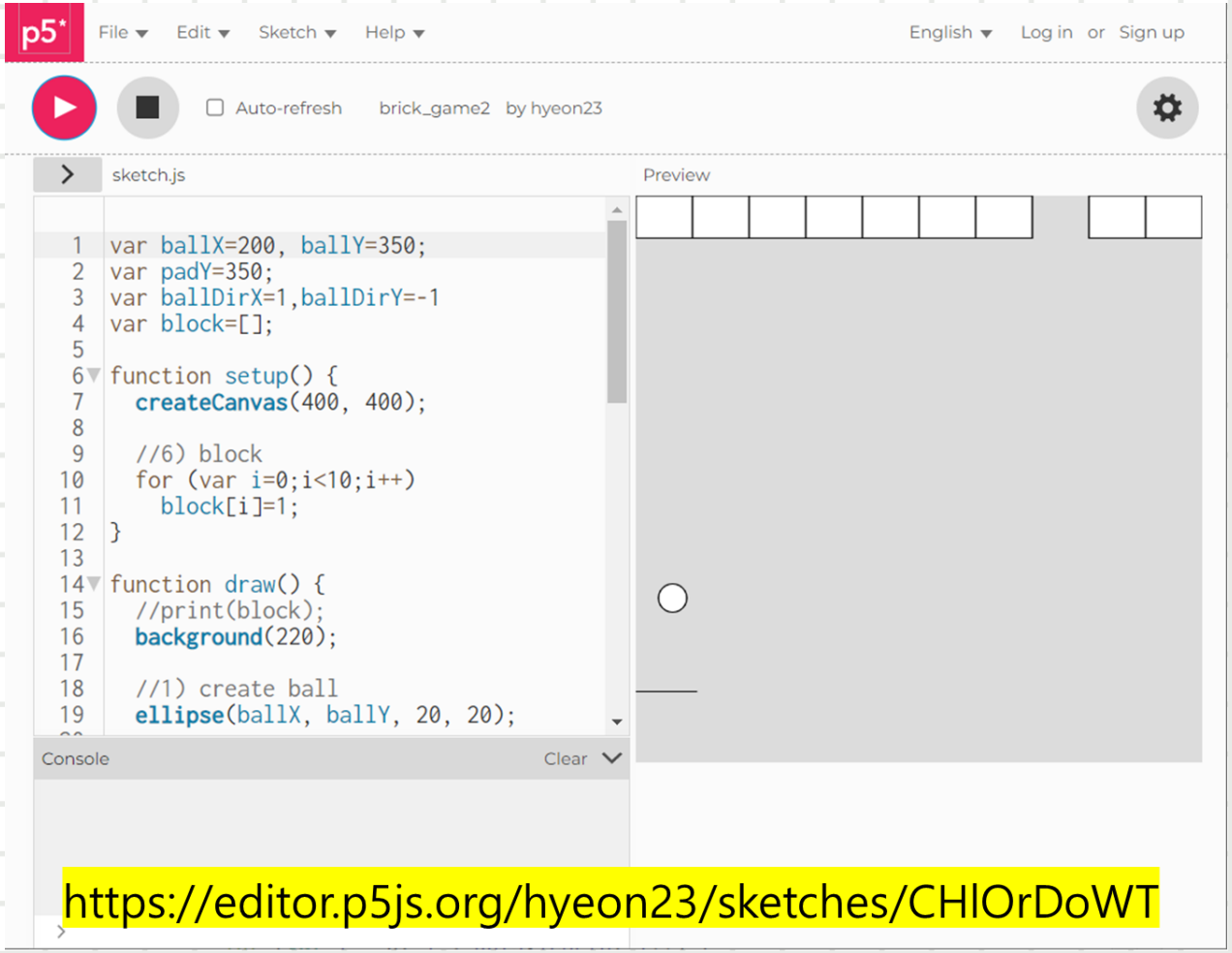
마우스를 따라오는 Zoog

```
function draw() {
// createCanvas(480, 270);
// background(255);
// Draw Zoog
rectMode(CENTER);

// Zoog body (rect 20*100)
stroke(0);
fill(150);
rect(mouseX, 145, 20, 100);
// Zoog head (ellipse 60*60)
fill(255);
ellipse(mouseX, mouseY-30, 60, 60);
// Zoog eyes  (ellipse 16*32)
fill(0);
ellipse(mouseX-19, mouseY-30, 16, 32);
ellipse(mouseX+19, mouseY-30, 16, 32);
// Zoog legs (line)
stroke(0);
line(mouseX-10, mouseY+50, mouseX-20, mouseY+60);
line(mouseX+10, mouseY+50, mouseX+20, mouseY+60);  }
```



질문해 주세요.

벽돌깨기 게임으로 문법 배우기



https://editor.p5js.org/hyeon23/sketches/CHlOrDoWT

https://han.gl/ifnkWi

코드에서 ballX, ballY는 어떤 특징을 가지고 있나요?

코드에서 ballX, ballY는 어떤 특징을 가지고 있나요?

(여기에 작성하세요)

변수

- 데이터를 임시로 저장하는 메모리(공간)이다.
- 이름, 값, 자료형으로 구성이 된다.
  (var a=2 라고 하면 이름 a, 값 2, 자료형 정수)
- 변수 이름은 영문자, 숫자, _(밑줄)을 사용해 만든다.
  (단, 숫자를 첫번째 글자로 사용할 수 없다)
- JS는 var, let 을 통해서 변수를 선언할 수 있다.

자료형

- 데이터의 형태를 의미한다.
- 정수(int), 실수(float), 문자(string) 으로 구분을 한다.
- JS는 숫자형(number type), 문자열(string type) 등으로 구분을 한다.
- 숫자형에는 정수표현과 실수 표현이 존재한다.
- 문자열은 따옴표(" ", ' ')를 사용해서 표현한다.
- typeof()를 사용하면 자료형을 확인할 수 있다.

실습코드

```
function setup() {
  let count = 0;  // 변수명 count, 할당된 값 0
  let letter = 'a';  // 변수명 letter, 할당된 값 'a'
  let d = 132.32;  // 변수명 d, 할당된 값 132.32
  let happy = false;  // 변수명 happy, 할당된 값 false
  var x = 4.0;  // 변수명 x, 할당된 값 4.0
  var y;  // 변수명 y
}
```

실습코드

```
var circleX = 100;
var circleY = 100;

function setup() {
  createCanvas(480, 270);
}

function draw() {
  background(255);
  stroke(0);
  fill(175);
  // 변수 사용하여 타원 만들기
  ellipse(circleX, circleY, 50, 50);
}
```

실습코드

```
var x;
var y;
var r,g,b;

function setup() {
 createCanvas(480, 270);
 background(255);
}
```

```
function draw() {
  r = random(255);
  g = random(255);
  b = random(255);
  x = random(width);
  y = random(height);
  fill(r,g,b);
  ellipse(x, y, 50, 50);
}
```

ballX=ballX+ballDirX는
어떤 특징을 가지고 있나요?

ballX=ballX+ballDirX는 어떤 특징을 가지고 있나요?

(여기에 작성하세요)

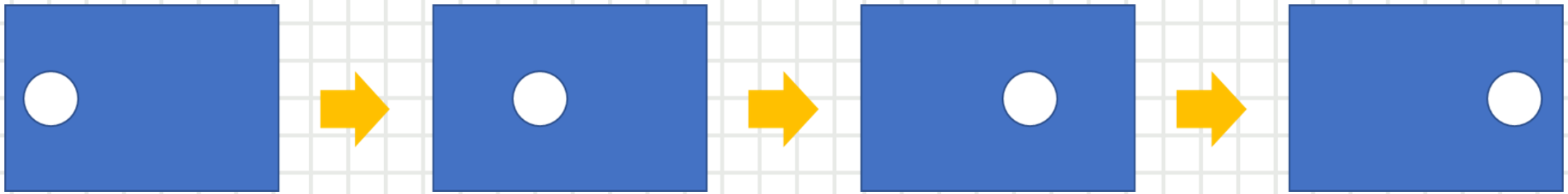
산술연산자

- 연산자는 계산에 사용되는 기호를 의미이다.
- 연산자에는 우선순위와 연산 방향이 있다.
- 숫자 계산과 관련한 연산자를 산술 연산자이다.

+	더하기
-	빼기
*	곱하기
/	나누기
%	나머지

도형의 움직임 만들기

- 원의 움직임(위치값)을 기억하기 위한 변수를 만든다.
- 그 변수에 연산을 통해 값을 변화시킨다

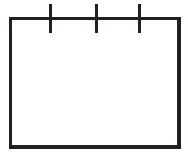


실습코드

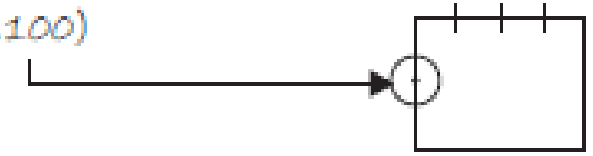
```
var circleX=0;
var circleY=100;

function setup() {
  createCanvas(480, 270);
}

function draw() {
  background(220);
  fill(175);
  ellipse(circleX, circleY, 50, 50);
  circleX = circleX + 1;
}
```

도형의 움직임 만들기

1. Remember circleX = 0 and circleY = 100
2. Run setup(). Open a window 200 X 200

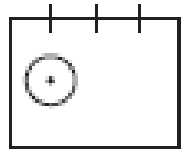


3. Run draw().
   - Draw circle at (circleX, circleY) → (0,100)
   - Add one to circleX



circleX = 0 + 1 = 1

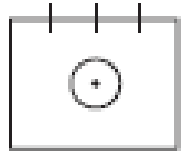
4. Run draw()
   - Draw circle at (circleX, circleY) → (1,100)



   - Add one to circleX

circleX = 1 + 1 = 2

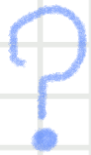
5. Run draw()
   - Draw circle at (circleX, circleY) → (2,100)



   - Add one to circleX

circleX = 2 + 1 = 3

6. And so on and so forth!



질문해 주세요.

if (ballX>width)는
어떤 특징을 가지고 있나요?

if (ballX>width)는 어떤 특징을 가지고 있나요?

(여기에 작성하세요)

조건문

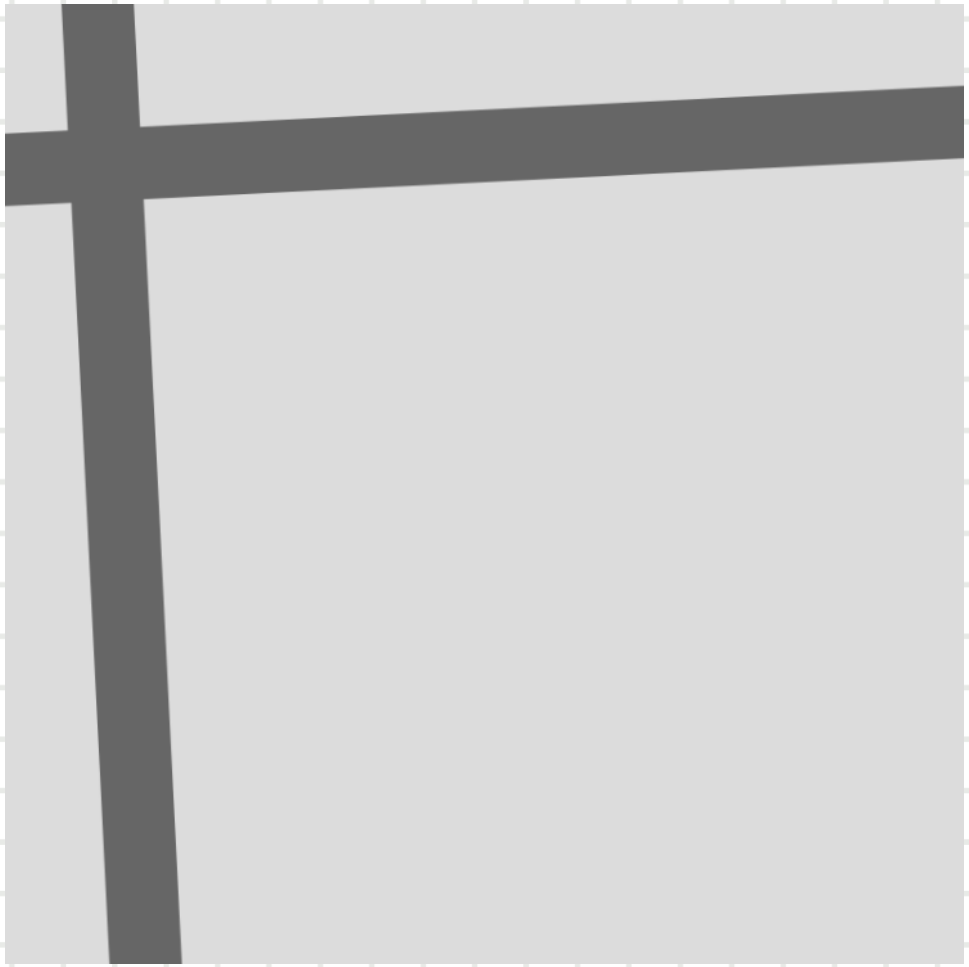
- 조건문: 조건식(True/False)에 따라 코드의 실행여부 결정
- if문: 조건식을 만족(true)하면 구문을 수행함.

```
if (조건식) {
  구문;
}
```

```
var x=10;
if (x<10)
  rect(x, 10, 10, 10);
```

연습코드

```
function draw() {
 background(220);
 strokeWeight(30);
 stroke(102)
 line(40, 0, 60, height);
 if (mouseIsPressed) {
  stroke(0);
 }
 line(0, 70, width, 50);
}
```



조건문

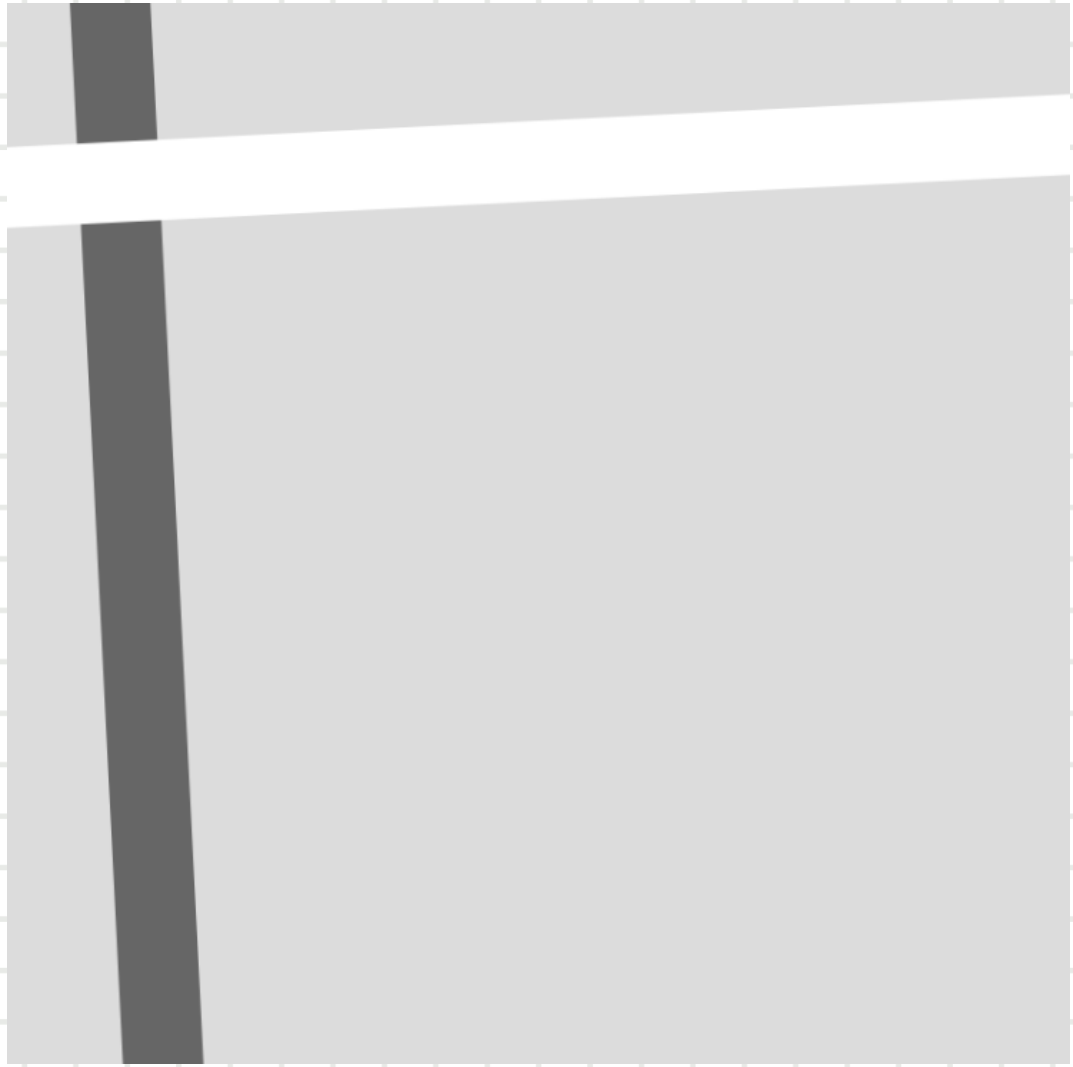
- if-else문: 조건식을 만족하면(true) 구문1을 수행하고,
  만족하지 않으면(false) 구문2를 수행함.

```
if (조건식) {
  구문1;
}
else {
  구문2;
}
```

```
var x=10;
if (x<10)
  rect(x, 10, 10, 10);
else
  rect(10, x, 10, 10);
```

연습코드

```
function draw() {
 background(220);
 strokeWeight(30);
 stroke(102)
 line(40, 0, 60, height);
 if (mouseIsPressed)
   stroke(0);
 else
   stroke(255);
 line(0, 70, width, 50);
}
```



조건문

- if-else if문: 조건식1을 만족하면(true) 구문1을 수행하고,
  만족하지 않으면(false) 조건식2를 평가하여 만족하면 구문2를 수행함.

```
if (조건식1) {
  구문1;
}
else if (조건식2){
  구문2;
}
```

```
var x=10;
if (x<10)
  rect(x, 10, 10, 10);
else if (x<20)
  rect(-x, 10, 10, 10);
```

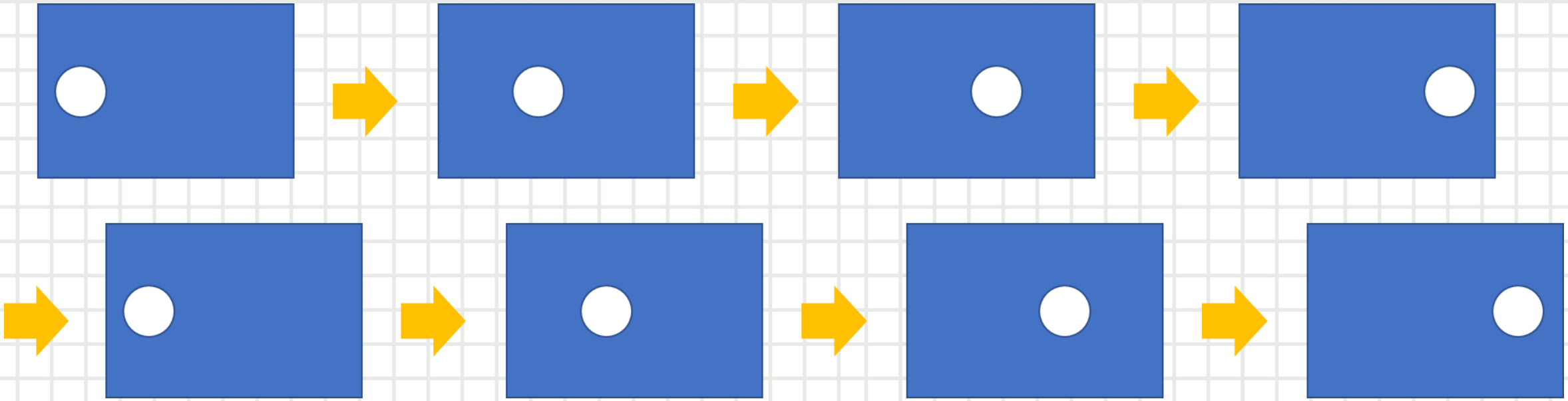
비교연산자 (관계연산자)

- 두 값을 비교하여 true(참)/false(거짓) 결과가 나오는 연산자가 비교연산자이다.

==	같은지
!=	다른지
>	큰지
>=	크거나 같은지
<	작은지
<=	작거나 같은지

도형의 움직임 만들기

- 조건문을 활용하여 공을 처음 위치로 만들기

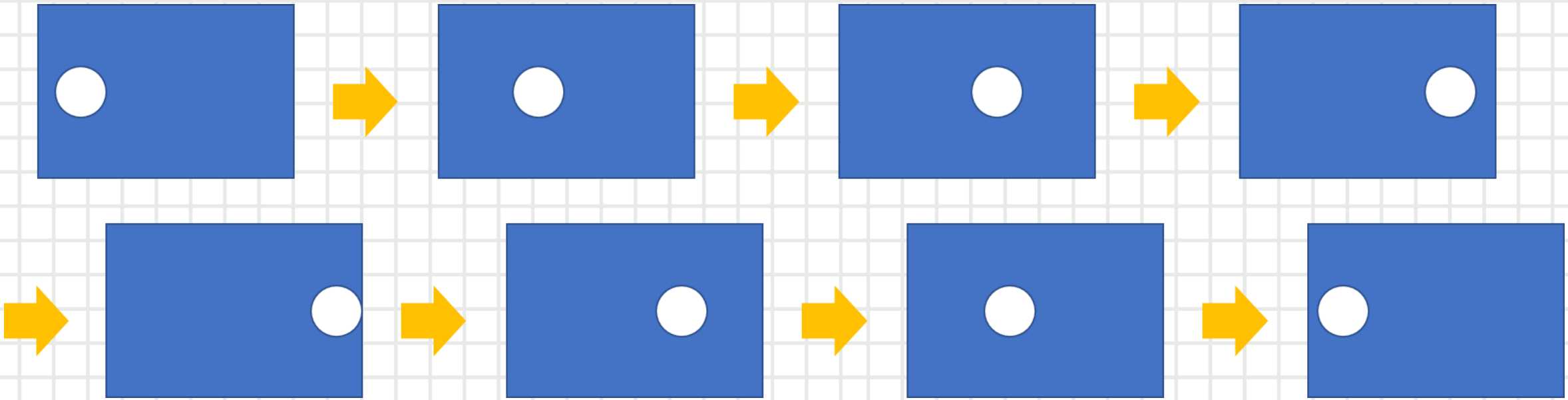


실습코드

```
var circleX=0;
var circleY=100;

function setup() {
  createCanvas(480, 270);
}

function draw() {
  background(220);
  fill(175);
  ellipse(circleX, circleY, 50, 50);
  circleX = circleX + 1;
  if (circleX>width)
    circleX=0;
}
```

도형의 움직임 만들기

- 조건문을 활용하여 공이 벽에 부딪히면 반대로 움직이도록 만들기



실습코드

```
var circleX=0;
var circleY=100;
var dirX=1;

function setup() {
  createCanvas(480, 270);
}

function draw() {
  background(220);
  fill(175);
  ellipse(circleX, circleY, 50, 50);
  circleX=circleX+dirX
  if (circleX>width)
    dirX=-1;
  else if (circleX<0)
    dirX=1;
}
```

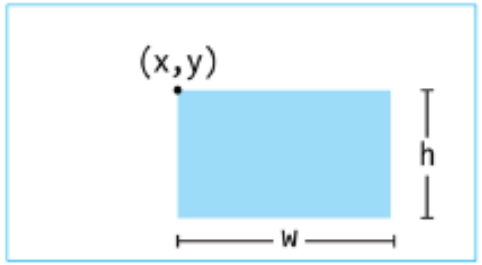
논리연산자

- 두 값을 비교하여 모두 참이거나 하나만 참인 경우 연산자가 비교연산자이다.

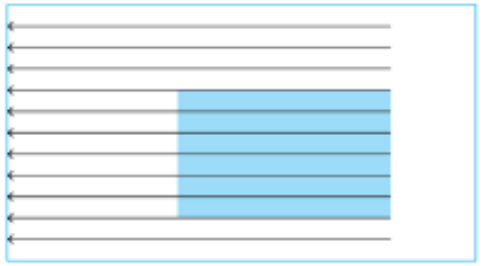
&&	그리고(and)
||	또는(or)
!	아니다(not)

A	B	A && B	A || B
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

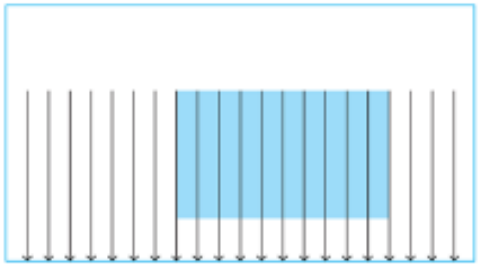
도형의 움직임 만들기



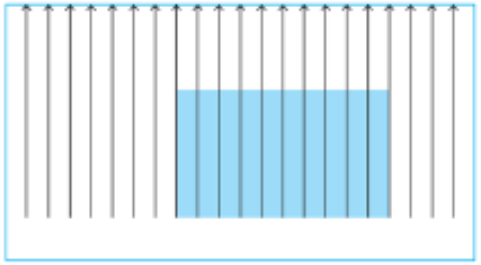
mouseX > x



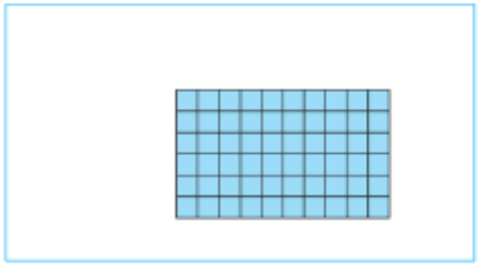
mouseX < x + w



mouseY > y



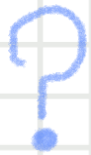
mouseY < y + h



(mouseX > x) and (mouseX < x+w) and
(mouseY > y) and (mouseY < y+h)

실습코드

```
var x=0, y=50;
var w=100, h=75;

function setup() {
  createCanvas(480, 270);
}

function draw() {
  if(button) {
    background(255);
    stroke(0);
  }
  else {
    background(0);
    stroke(255);
  }
  fill(175);
  rect(x,y,w,h);
}

function mouseIsPressed() {
  if (mouseX>x && mouseX<x+w &&
mouseY>y && mouseY<y+h) {
    button = !button;
  }
}
```



질문해 주세요.

```
for (var i=0;i<10;i++)
rect(i*40, 0, width/10, 30);
```

는 어떤 특징을 가지고 있나요?

for (var i=0;i<10;i++)
rect(i*40, 0, width/10, 30);
는 어떤 특징을 가지고 있나요?

(여기에 작성하세요)

반복문 - for문

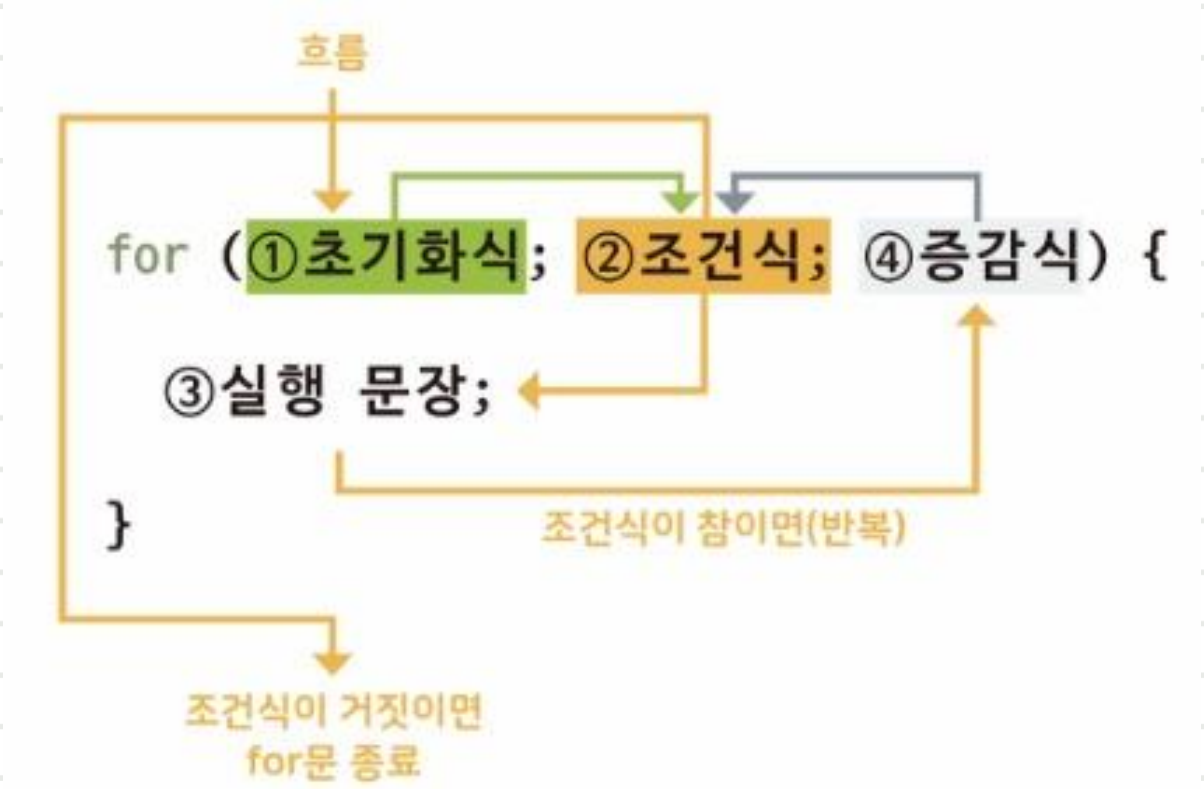
- for문: 초기식을 통해 초기화를 하고, 조건식을 평가하여
만족(true)하면 구문을 수행 후 변화식을 수행함.
조건식 평가하여 '구문->변화식->조건식'을 반복함.

```
for (초기식;조건식;변화식 {
  구문;
}
```

```
for (var i=0;i<10;i++)
  rect(i*5, i*5, 10, 10);
```

- 1) 초기식 → 2) 조건식 → true인 경우 3) 구문 → 4) 변화식 →
2) 조건식 → true인 경우 3) 구문 → 4) 변화식 → (2,3,4 반복)
2) 조건식 → false인 경우 for문 탈출

반복문 - for문



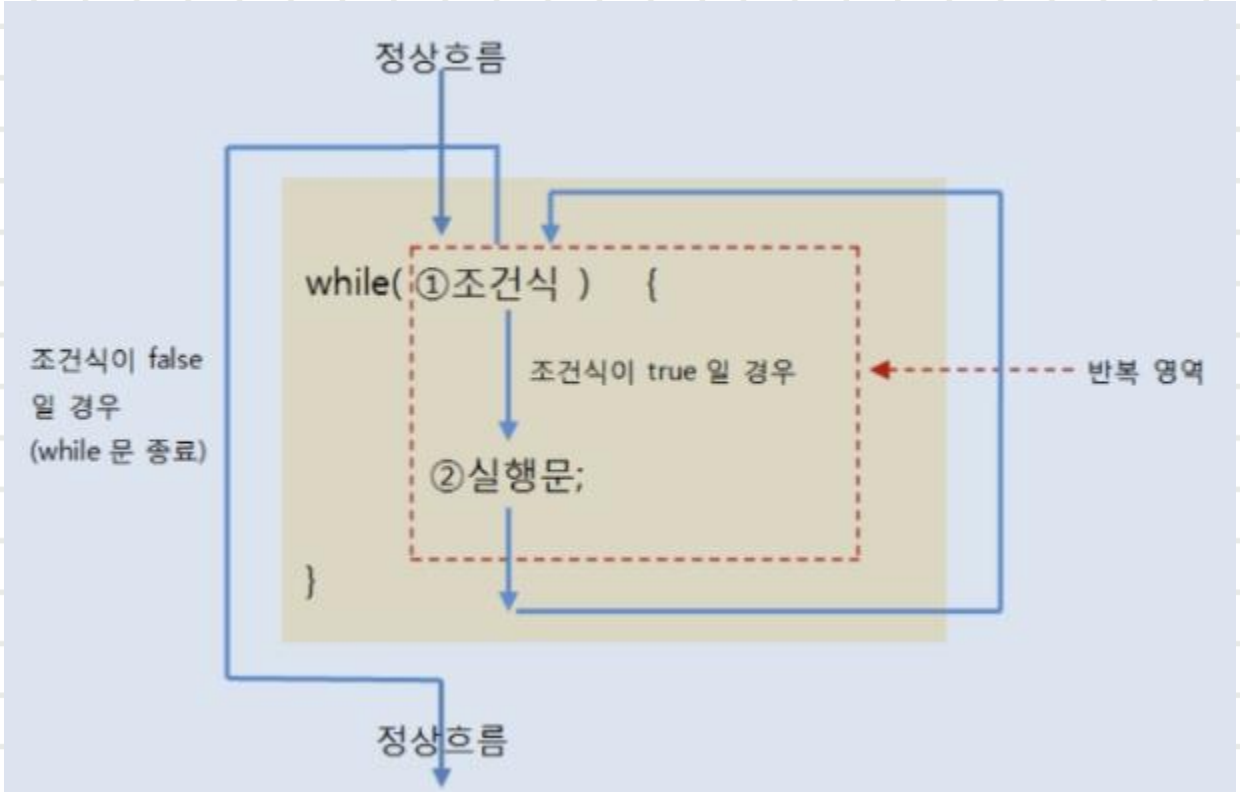
반복문 - while문

- while문: 조건식(True/False)에 따라 코드의 반복 실행여부 결정

```
while (조건식) {
  구문;
}
```

```
var x=10;
while (x>1) {
  rect(x, 10, 10, 10);
  x--; //x를 1씩 감소
}
```

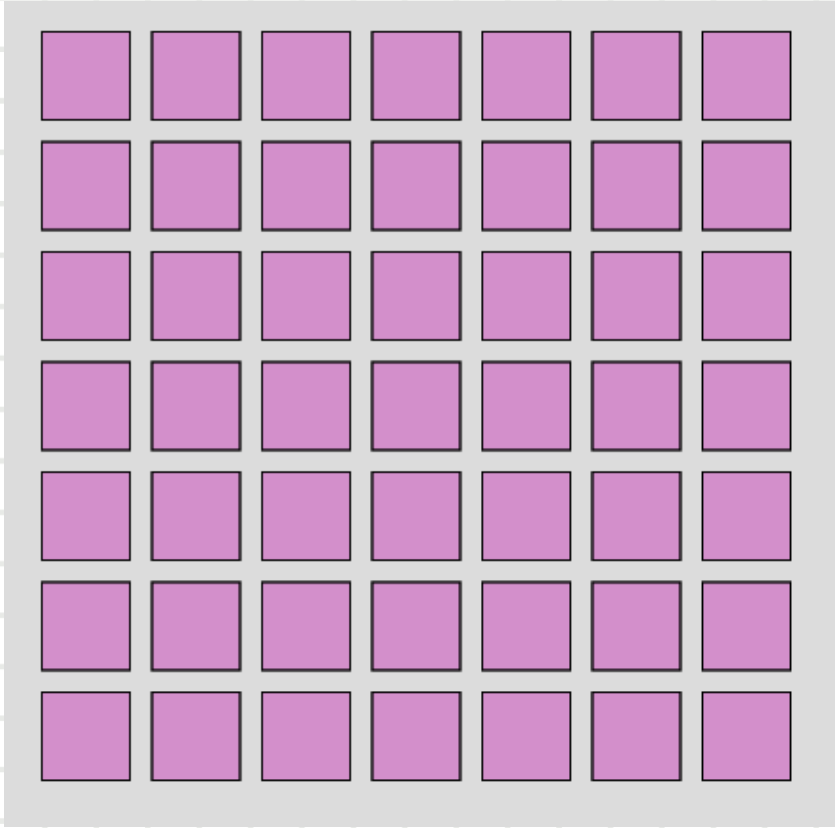
반복문 - while문


정상흐름
while(①조건식) {
조건식이 true 일 경우
②실행문;
}
반복 영역
조건식이 false
일 경우
(while 문 종료)
정상흐름

아래 그림이 나오도록 코드를 작성해봅시다.

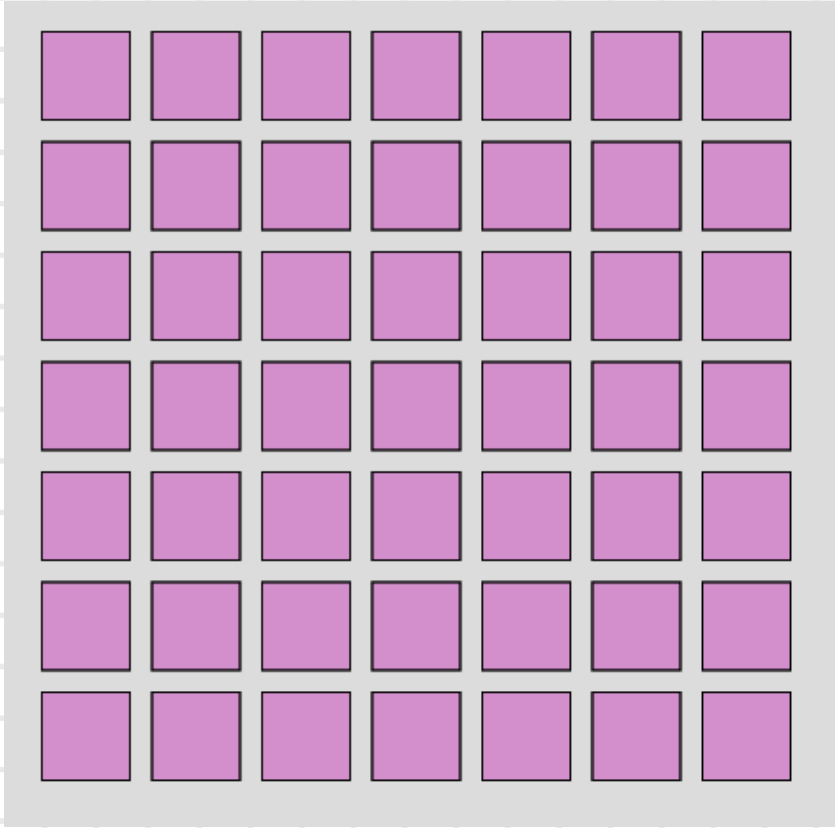
조건1: fill(211, 143, 203)

조건2: 40x40 rect



아래 그림이 나오도록 코드를 작성해봅시다.

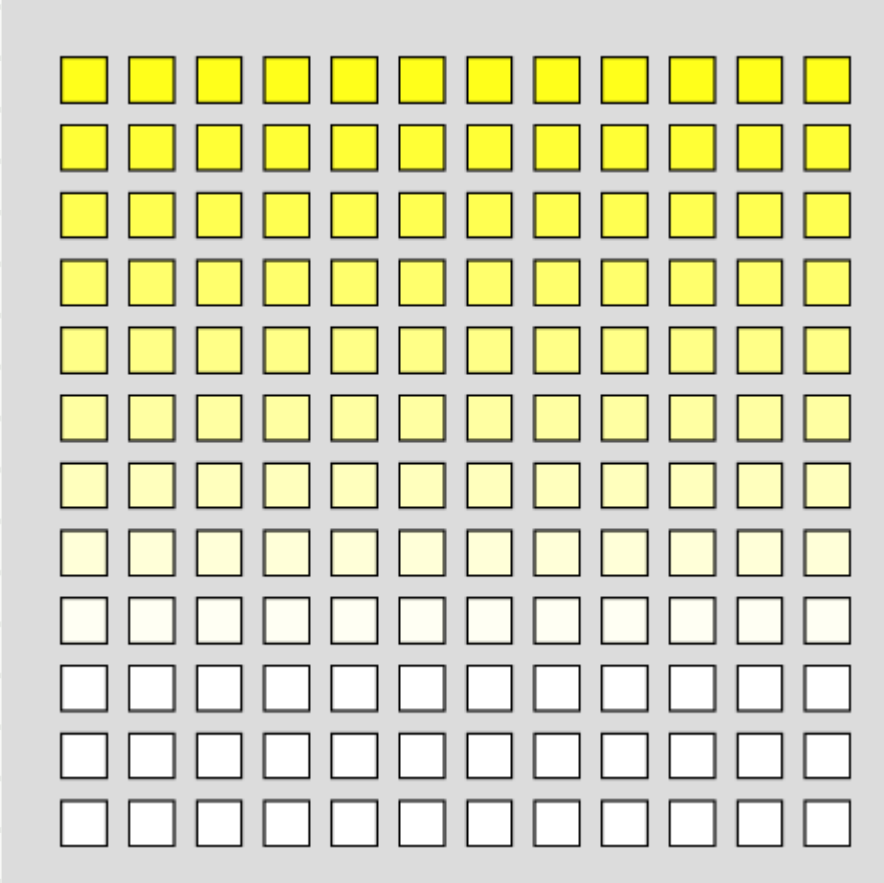
```
function setup() {
  createCanvas(400, 400);
}
function draw() {
  background(220);
  fill(211, 143, 203);
  for (let y=30;y<=350;y+=50) {
    for (let x=30;x<=350;x+=50)
      rect(x, y, 40, 40);
  }
}
```



아래 그림이 나오도록 코드를 작성해봅시다.

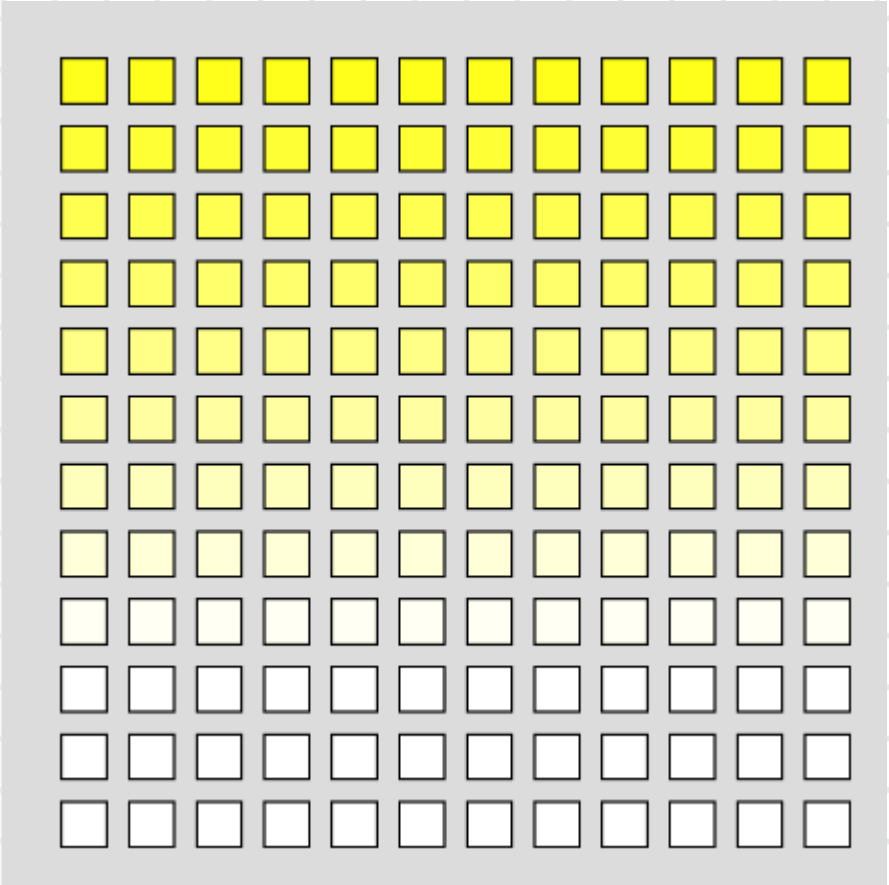
조건: fill(255, 255, B)
0<B<255

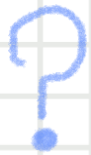
조건1: 20x20 rect



아래 그림이 나오도록 코드를 작성해봅시다.

```
function setup() {
  createCanvas(400, 400);
}
function draw() {
  background(220);
  for (var y=30;y<345;y+=15) {
    for (var x=30;x<345;x+=15) {
      fill(255, 255, y);
      rect(x, y, 10, 10);
  } } }
```





질문해 주세요.

block=[]는
어떤 특징을 가지고 있나요?

block=[]는
어떤 특징을 가지고 있나요?

(여기에 작성하세요)

배열

- 배열은 변수들로 구성된 목록이다.
- 목록 안의 요소(값), 순서(인덱스)가 존재한다.
- 따라서 각 요소는 색인(index)을 갖는다.
- 첫 번째 요소 번호가 0 이다.

SCH101[0]	SCH101[1]	SCH101[2]	SCH101[3]
학생A	학생B	학생C	학생D

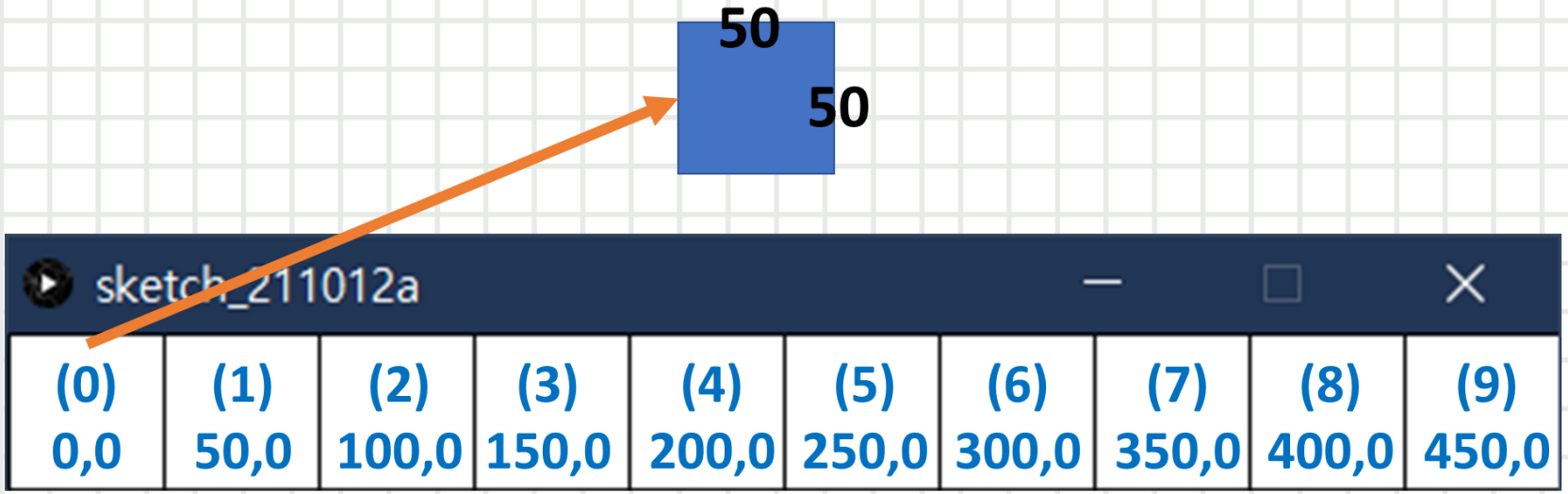
SCH102[0]	SCH102[1]	SCH102[2]	
학생E	학생F	학생G	

배열

- 배열 생성 방법
  1. var arr = [배열요소1, 배열요소2,...]; // 배열 리터럴을 이용하는 방법
  2. var arr = Array(배열요소1, 배열요소2,...); // Array 객체의 생성자를 이용하는 방법
  3. var arr = new Array(배열요소1, 배열요소2,...); // new 연산자를 이용한 Array 객체 생성 방법
- 예제)

```
var arrLit = [1, true, "JavaScript"];
var arrObj = Array(1, true, "JavaScript");
```

배열


50
50
sketch_211012a
(0)
0,0
(1)
50,0
(2)
100,0
(3)
150,0
(4)
200,0
(5)
250,0
(6)
300,0
(7)
350,0
(8)
400,0
(9)
450,0

실행결과를 확인해보세요

```
let arr = [];
let fruits = new Array(10);
let ages = [10, 21, 89, 3, 68];
let misc = ["pumpkin", 10.4, "dog", false, -1];
let more_misc = ["dustpan", "k", misc];
console.log(more_misc[0][2]);
console.log(more_misc[2][1]);
```

실행결과를 확인해보세요

```
let arr = ["mushrooms", "cheerios", "sparkling water"];
for (var i = 0; i < 3; i++) {
  arr[i] = "I love " + arr[i];
}
console.log(arr);
```

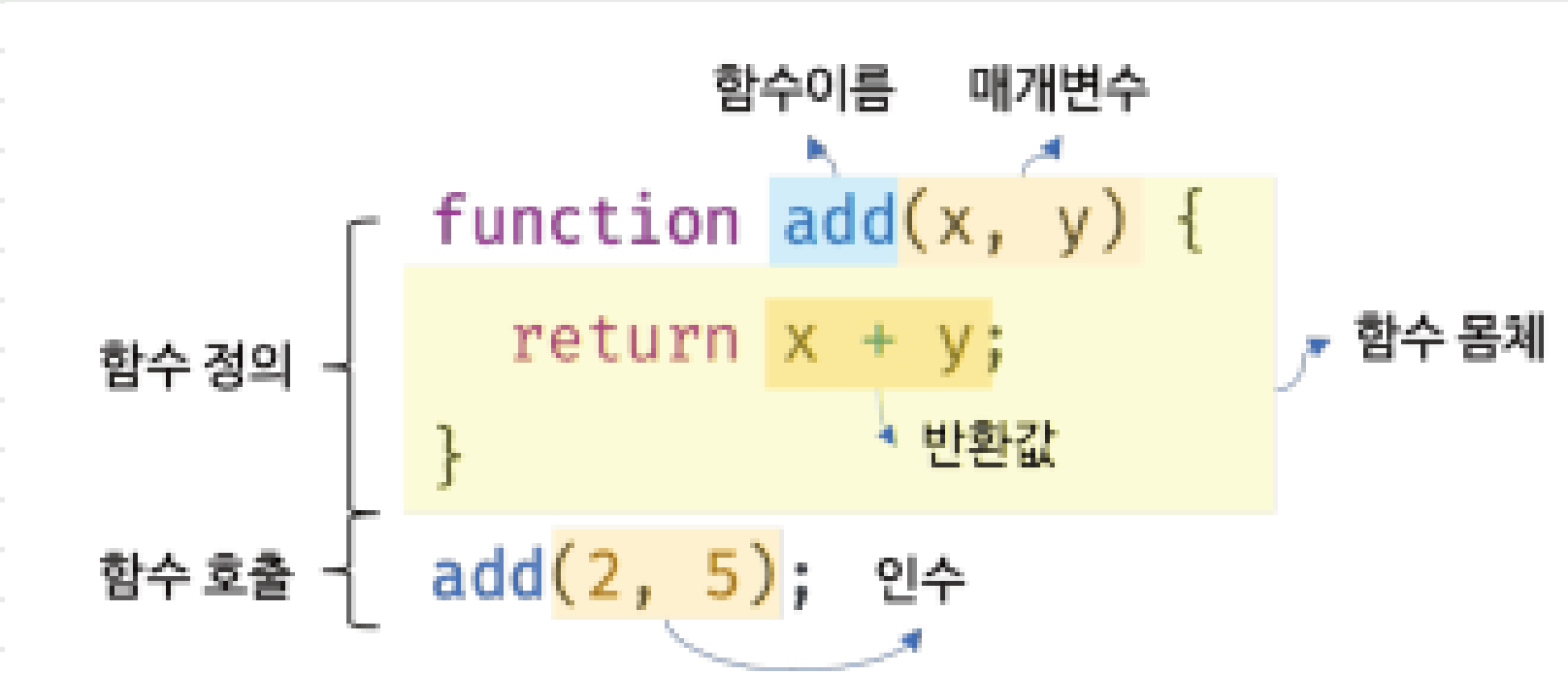
실행결과를 확인해보세요

```
let arr = [3, 5, 19];
for (var i = 0; i < arr.length; i++)
{
  arr[i] *= 2;
}
console.log(arr.length);
console.log(arr);
```

함수

- 특정 기능의 코드를 한 곳에 모아 수행하도록 한 것이다.
- 함수이름, 매개변수(parameter), 반환값(return)이 있다.
- function 함수이름(매개변수1, 매개변수2,...) {
  함수가 호출되었을 때 실행하고자 하는 실행문;
  }
- 예)
  function open_account(name) {
  print(name, “님, 새로운 계좌를 개설합니다.”);
  return 1;
  }
  open_account(“지수”)

함수


함수이름
매개변수
function add(x, y) {
return x + y;
}
반환값
함수 몸체
함수 정의
함수 호출
add(2, 5);
인수

프로젝트 준비하기

- 문제 설정

- 필요한 기능

프로젝트 준비하기

- 프로그램 구조 설계

- 알고리즘 설계

프로젝트 준비하기

- 프로그램 제작

- 프로그램 평가

프로젝트

- 문제 설정

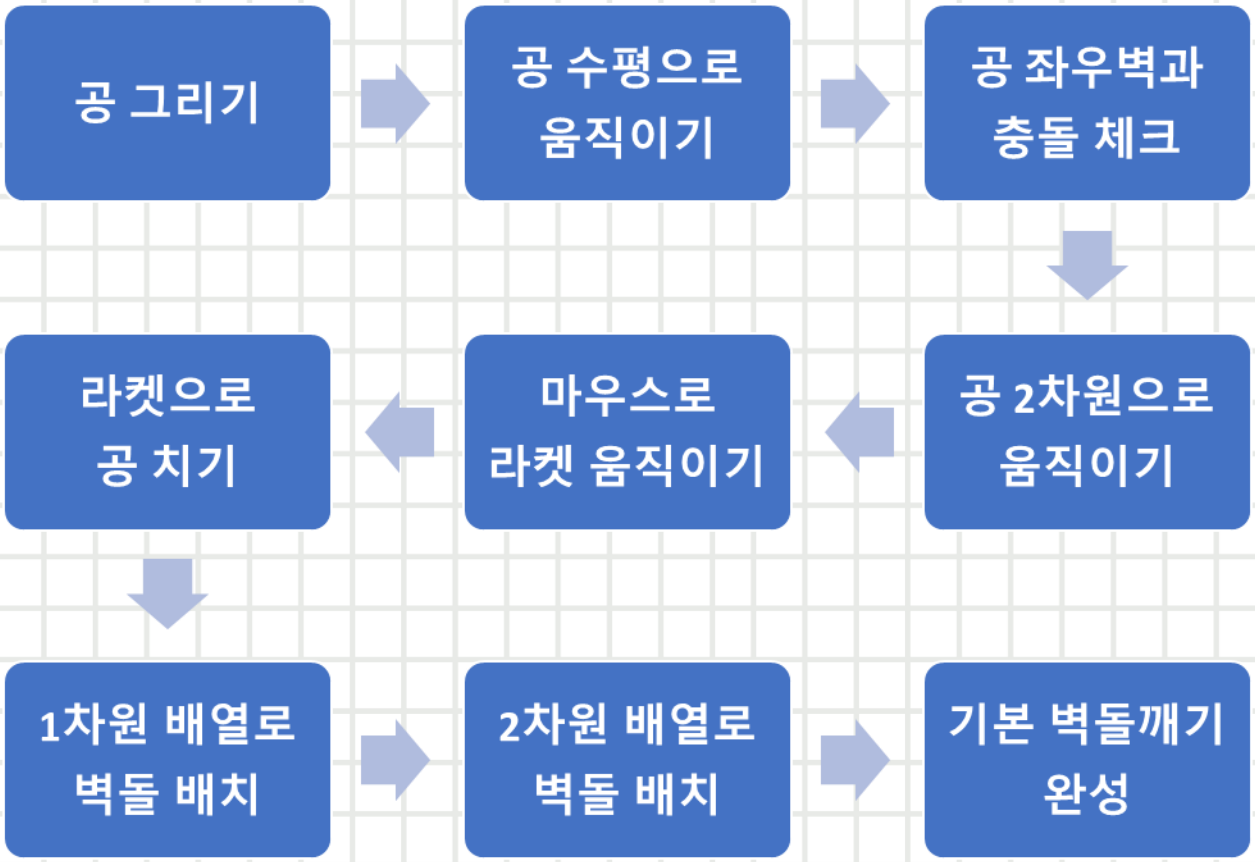
  벽돌깨기 게임 만들기

- 필요한 기능

  공, 공 움직이기
  패드(라켓), 패들 움직이기, 패들 충돌
  블록, 블록 없애기, 블록 충돌
  벽, 벽 충돌
  점수
  미션 클리어, 실패, 다시 실행

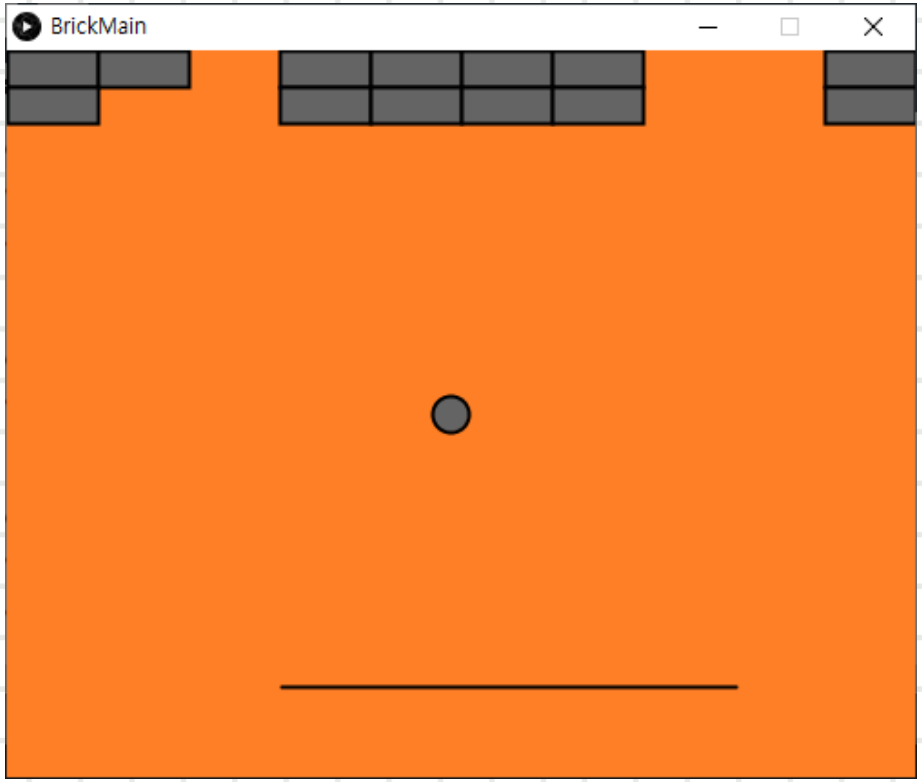
프로젝트

- 프로그램 구조 설계 & 알고리즘 설계



프로젝트

- 프로그램 제작&프로그램 평가



프로젝트

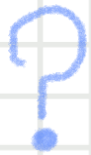
- 프로그램 개발

1) 공 그리기 (ellipse)
2) 공 움직이기 ← 변수의 변화
 (2-1:좌측 이동 2-2:우측 이동 2-3:상하 이동)
3) 공과 벽의 충돌 검사 ← 조건문 사용
 (3-1:좌측 벽 3-2:우측 벽 3-3 상하 벽)
4) 패들 그리기(line) 및 마우스로 조작하기
5) 공과 패들의 충돌 검사

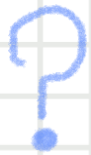
프로젝트

- 프로그램 개발

6) 1줄 벽돌 생성(rect, 배열) ← 배열과 반복 사용
 (6-1 벽돌 그리기 6-2 벽돌과 공 충돌 검사)
7) 3줄 벽돌 생성(2차원 배열)
8) 마무리 작업(게임 클리어/오버 기능, 점수 기능 등)



질문해 주세요.



감사합니다