

2025 Spring semester

Interaction Lab (1)

week 6

Keunjung Bae

Midterm

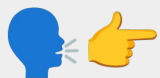
중간과제 발표

중간고사 (4/30) 발표

- 개별 프로젝트
- 작품 ppt발표 (한 사람당 5분 이내 발표) ← **꼭 엄수!**
- 발표 주제: 2가지중 선택
 1. 바우하우스(bauhaus)스타일 인터랙티브 포스터 만들기
 2. 자유주제
- 7주차 까지 배운 프로세싱의 개념 (조건문, 반복문, 함수의 개념까지 포함해서 코드짜기) → 이 개념이 빠지면 안됨.
- 예제코드는 네이버카페에.

❖ 프로세싱 코드 참조사이트

- <http://openprocessing.org>
- <http://www.generative-gestaltung.de/1-archive/code.html>
- <https://processing.org/examples/>
- <https://codepen.io/tag/processing>
- <https://www.instructables.com/search/?q=processing%20arduino&projects=all>
- <https://www.instructables.com/Raincloud-Tree-Watering-Serial-Controller-Game-Wit/>



중간고사 발표 및 제출(20%)

중간과제 발표 당일 (4/30일)까지,

- 발표내용: 뒤의 중간 발표구성을 참조하여 ppt 발표
- 발표당일, 자신의 작품으로 ppt발표
- **개인당 5분 이내 발표.**

중간과제 발표 후 (다음날 5/1일 자정 전까지), 중간과제 블로그 포스팅!

- 5/1일 밤 12까지 중간과제 최종블로그 포스팅(**늦으면 감점!**)
- **포스팅은 개별 포스팅!!!**
 - 중간 과제에 대한 자세한 블로그 포스팅 → 네이버 중간과제 게시판에!
 - 내용: 뒤의 중간 발표 및 최종 블로그 포스팅 구성안 참조



점수반영!

중간발표ppt 및 중간과제 블로그 포스팅에 들어갈 구성

중간과제 프로젝트 작품 이름

중간과제 개요(2~3줄)

- 내가 만들고자 한 것은 무엇인지?
- 어떤 주제나 문제에서 출발했는지?

아이디어 도출 과정

- 어떤 생각에서 출발했는지
- 스케치, 레퍼런스, 키워드 등

작품 소개

- 결과물이 무엇인지 간단히 요약
- 색상, 형태, 인터랙션 구성 설명
- 스크린샷, GIF, 시연 영상 캡처



사용한 기술 및 구현 방식

- Processing에서 쓴 주요 코드 구조
- 코드 분석 및 설명(자기가 쓴 코드의 작동 원리)
- 작동 방식 시연 또는 설명/ 반복 구조나 시각적 효과 설명

📌 문제 해결과 배운 점

- 구현하면서 막혔던 점
- ChatGPT나 검색으로 해결한 과정
- 배운 프로그래밍 개념

📌 앞으로의 이 작업에서의 보완/ 개선사항 (For next..)



마무리 및 느낀점

📌 기존의 비슷한 작업들의 레퍼런스들(참조 사이트나 서적등 참조자료들)

📌 작품 완성물(그림이나 동영상 첨부 및 시연!) - 반드시 첨부!!

📌 프로세싱 전체코드 스케치 반드시 블로그 포스팅에 첨부!!

(프로세싱 코드는 반드시 zip파일로 첨부! 블로그에 직접 쓰지마세요.)

- 프로세싱(프로세싱 메뉴- 도구 - .zip으로 압축하기 클릭 →zip 파일로 저장됨)

*본인이 쓴 코드의 원리를 말로 설명할 줄 알아야!



평가

성적 및 평가

- Lab 실습 포스팅 - 네이버카페 '실습 게시판' 이용 → 20%
- 중간고사 - 자신만의 스위치! 발표 및 제출 (개별진행)→ 20%
- 기말고사 - 최종 결과물(팀별 진행) → 40%
- 출석 → 10%
- 참여도 → 10%

중간고사 평가부분(20%)

- 중간고사 플랜부터~ 발표까지 프로세스 진행과정 (발표 와 최종 블로그 포스팅포함)
- 코드에 대한 본인 작품의 이해/아이디어/ 독창성
- 코드에 대한 이해도, chatgpt등 보조수단 활용여부
- 완성도

성적평가 기준

- 상대평가(A = 최대 40% / A+B= 최대 80% / B+ ~ F = 남은 인원)
- 출석으로 간주할 수 있는 결석범위:

직계가족의 사망(증빙서첨부) / 병사관계로 인한 결석(병무청 증빙서첨부) / 정부기관의 요청에 의한 부득이한 행사 참여(학생처장이 발행하는 확인서첨부) / 총학생회의 제업무 및 행사참여 (학생처장이 발행하는 확인서첨부) / 기타 교무처장(교학처장)의 사전승인을 받은 사항에 의한 결석에 한함.(코로나19, 독감확진- 교무처 문의)

- 수업 시작 후 20분 까지는 지각, 이후 지각은 결석.
(결석 : 총점100점에서 -1점감점, 지각 : -0.5점 감점)
- 결석시간이 총 수업시간의 3분의 1을 초과한 학생은 F 처리
- 지각 3회 = 결석 1회

반복문
(loop)

```
int x = 0;
```

```
while ( x < 3 ) {
```

```
    //do
```

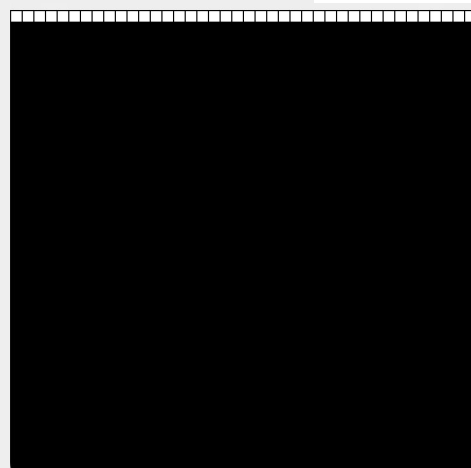
```
    x++;
```

```
}
```

```
for ( ①int x = 0 ; ②x < 3 ; ③x++ ) {  
    //do  
}
```

```
for ( 처음상태 ; 조건; 변화 ) {  
    //조건이 참이면 실행할 코드  
}
```

```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(0);  
  
    for (int x = 0; x < width; x+=10) { //0 에서 width까지 10씩  
증가반복  
        rect(x, 0, 10, 10); //rect 의 x위치에서 10씩증가.  
    }  
  
}
```




```
for ( int x = 0 ; x < width ; x+=10 ) {  
    rect( x, 0, 10, 10 );  
}
```

```
for ( int x = 0 ; x < width ; x+=10 ) {  
    rect( x, 10, 10, 10 );  
}
```

```
for ( int x = 0 ; x < width ; x+=10 ) {  
    rect( x, 20, 10, 10 );  
}
```

```
for ( int y = 0; y < 30 ; y += 10 ) {
```

```
    for ( int x = 0 ; x < width ; x += 10 ) {
```

```
        rect( x, y, 10, 10 );
```

```
    }
```

```
}
```

```
for ( 처음상태 ; 조건; 변화 ) {  
    for ( 처음상태 ; 조건; 변화 ) {  
        조건이 참이면 실행할 코드  
    }  
}
```

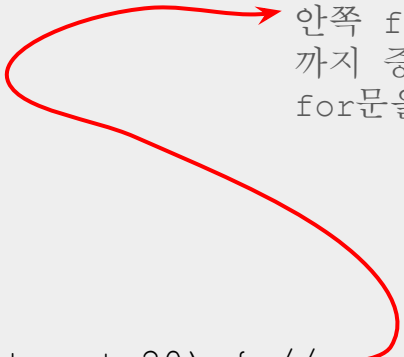
```
int size = 20;
```

```
void setup() {  
    size(400, 400);  
    background(0);  
}
```

```
void draw() {
```

```
    for (int y = 0; y < height; y+=20) { //  
        for (int x = 0; x < width; x+=20) { // 력트가 width까지 20씩 간격으로 반복.  
            rect(x, y, size, size);  
        }  
    }  
}
```

화면을 다 채우는 력트를 만드려면,
안쪽 for문에서 y값으로 값이 20씩 height
까지 증가해야되므로 안쪽 for문 밖에 또
for문을 만들면된다.



연습문제 : 아래의 조건을 만족하는 코드 작성하기.

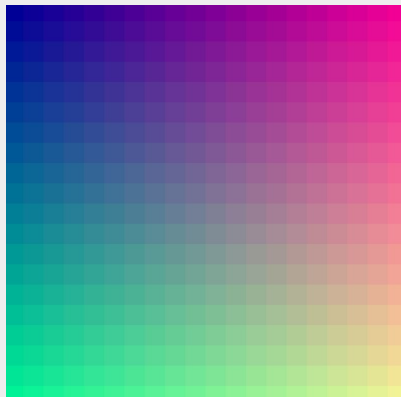
- ❖ 이중 for루프를 사용하여 각 사각형은 `size = 20` 크기를 사용하고, 다음의 색상패턴이 전체 화면에 가득 차도록 배치하세요.

`fill(r, g, b)`와 `rect(x, y, size, size)`를 이용해 색을 지정하고 사각형을 그리세요.

각 사각형들의 색은 다음 기준에 따라 다르게 지정합니다:

- 가로 위치(x)가 증가함에 따라 빨간색(r) 값이 (0 → 255로) 점점 증가 (map함수이용)
- 세로 위치(y)가 증가함에 따라 초록색(g) 값이 (0 → 255로) 점점 증가
- 파란색(b) 값은 고정(예: 150)

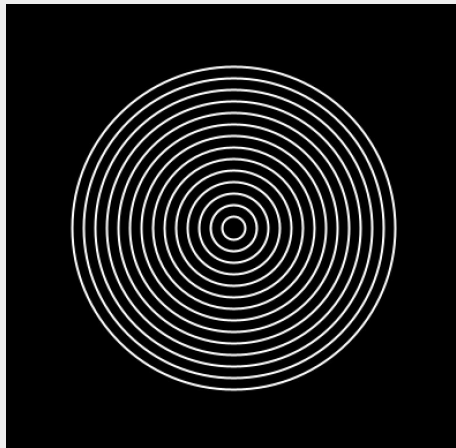
결과적으로 그림과 같은 컬러 그리드가 만들어짐.



연습문제 : 아래의 조건을 만족하는 코드 작성하기.

화면 중앙(width/2, height/2)을 기준으로, 반지름(radius)이 10부터 140까지 10씩 증가하는 **동심원**(겹치는 원들)을 여러 개 그리는 코드를 작성하시오.

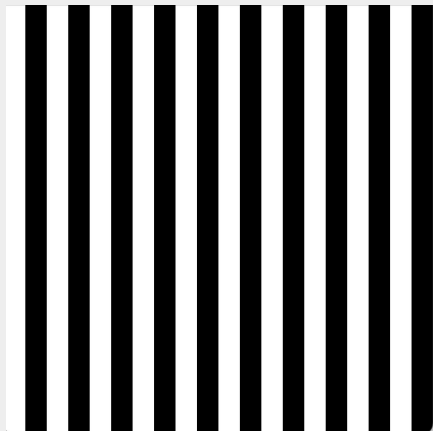
- 배경은 **검정색**, 원의 테두리는 **흰색**, **채우기 없음(noFill)**으로 설정한다.
- 선 두께는 **2픽셀**로 한다.
- **for**문을 사용하여 여러 개의 원을 반복적으로 그릴 것.
- 각 원의 중심은 **화면 중앙**이며, 반지름은 10씩 증가한다.

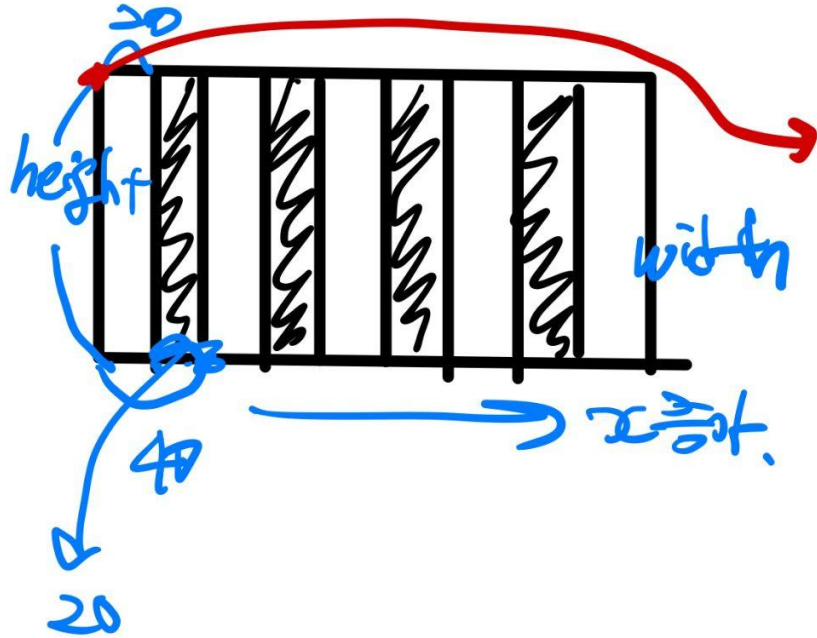


연습문제 : 아래의 조건을 만족하는 코드 작성하기.

다음 조건을 만족하는 **세로 줄무늬 패턴**을 생성하는 Processing 코드를 작성하시오.

- 배경은 **검정색**, 띠는 **흰색**이며 **테두리 없음(noStroke)** 으로 설정한다.
- rect()를 사용하여 줄무늬를 그린다. 줄무늬 띠의 가로너비는 **20픽셀**이고 높이는 창크기인 height이다. 띠 사이에는 같은 크기의 간격(20픽셀)이 있다.
- 즉, 띠 하나 그리고 20픽셀 띄우고 다시 띠 하나, 이런 식으로 전체 화면을 채운다.
- 반복문(for)을 사용하여 줄무늬 띠를 그릴 것.



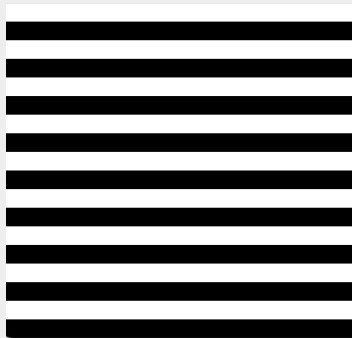


$\text{rect}(0, 0, 20, \text{height})$

↓
2 행만큼이다.

연습문제 : 아래의 조건을 만족하는 코드 작성하기.

다음 조건을 만족하는 가로 줄무늬 패턴을 생성하는 Processing 코드를 작성하시오.

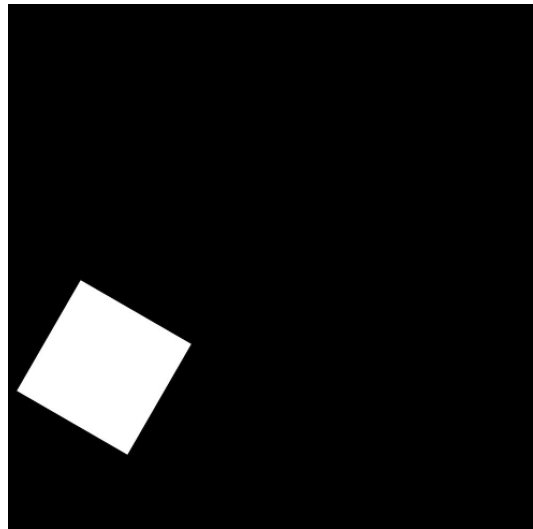


- 배경은 검정색, 띠는 흰색이며 테두리 없음(noStroke) 으로 설정한다.
- rect()를 사용하여 줄무늬를 그린다. 줄무늬 띠의 가로너비는 **20픽셀**이고 높이는 창크기인 height이다. 띠 사이에는 같은 크기의 간격(20픽셀)이 있다.
- 즉, 띠 하나 그리고 20픽셀 띄우고 다시 띠 하나, 이런 식으로 전체 화면을 채운다.
- 반복문(for)을 사용하여 줄무늬 띠를 그릴 것.

변환/이동/회전

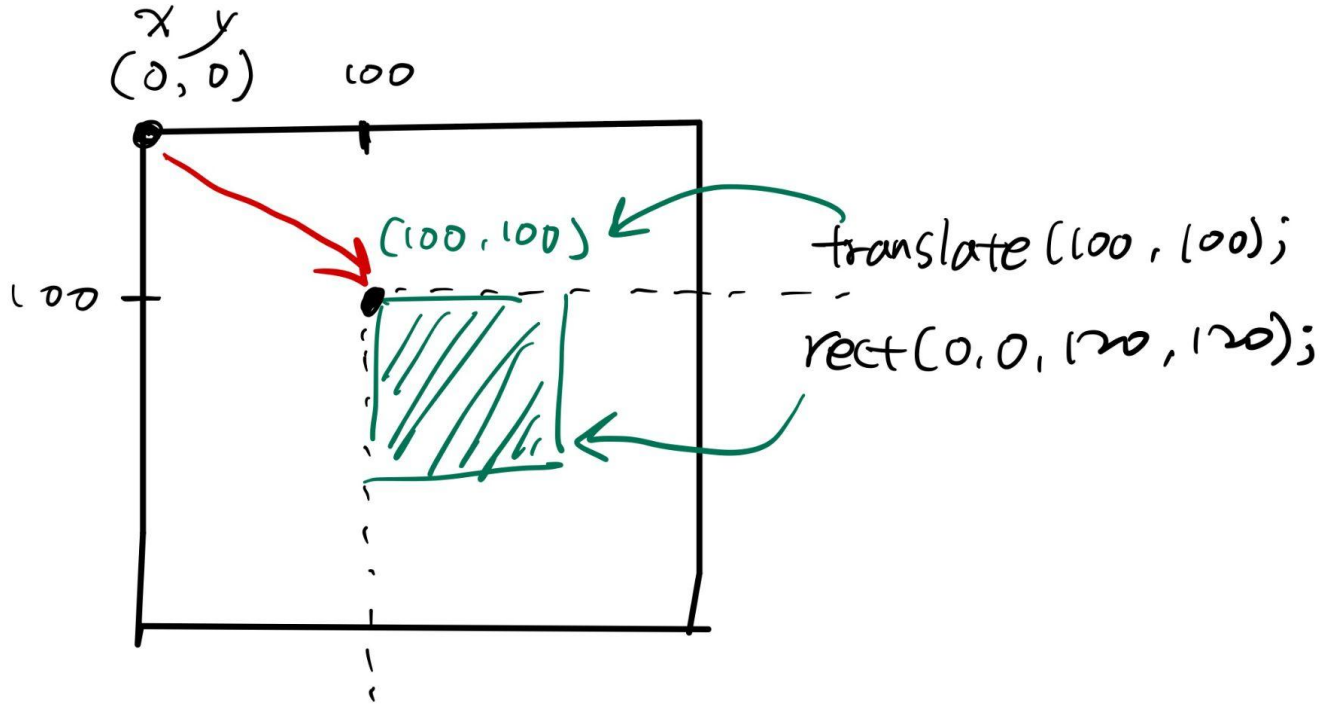
(Transform)

```
void setup() {  
    size(500, 500);  
}  
void draw() {  
    background(0);  
  
    rectMode(CENTER);  
    rotate(PI/6); //30도만큼 회전  
    rect(width/2, height/2, 120, 120);  
}
```

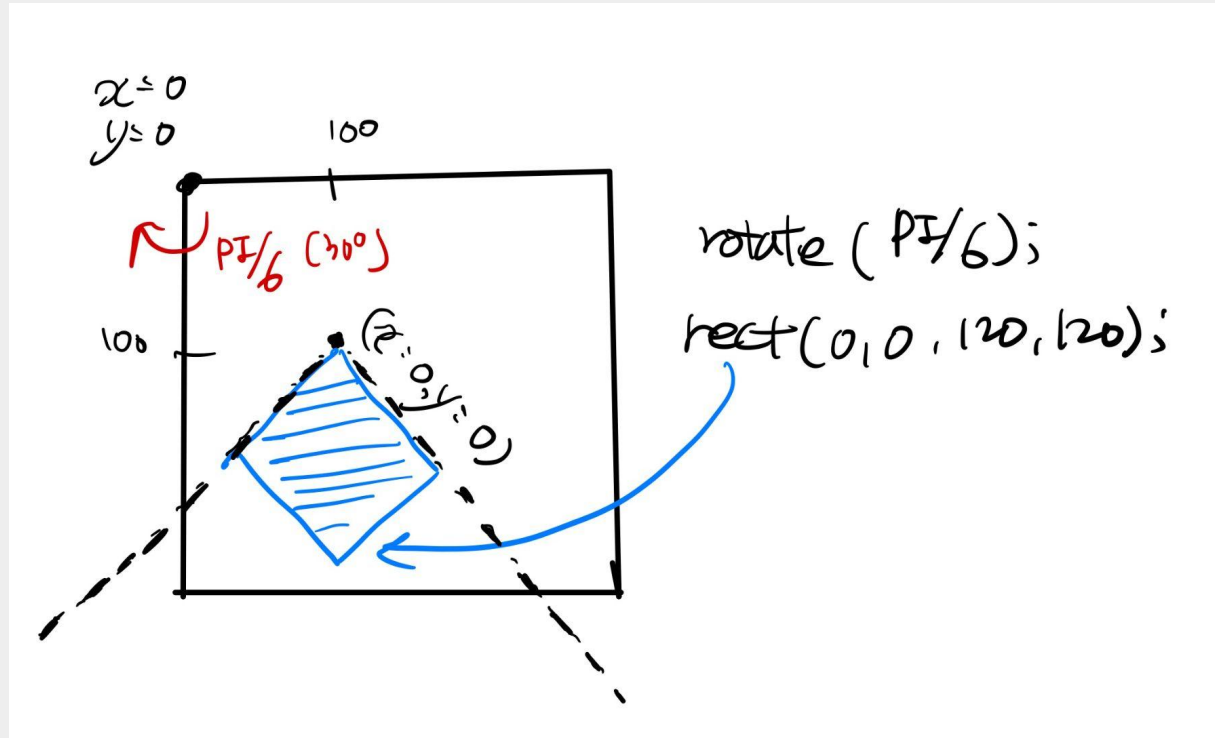


트랜스폼 종류	기능	영향	설명
Translate (이동)	좌표 이동	좌표계가 이동함	x, y (또는 x, y, z) 방향으로 좌표계를 이동
Scale (확대/축소)	크기 조정	좌표계가 확대/축소됨	오브젝트를 확대하거나 축소, 중심 기준으로 배율 조정
Rotate (회전)	회전	좌표계가 회전함	중심을 기준으로 전체 좌표계를 돌림 (오브젝트가 도는 게 아님)

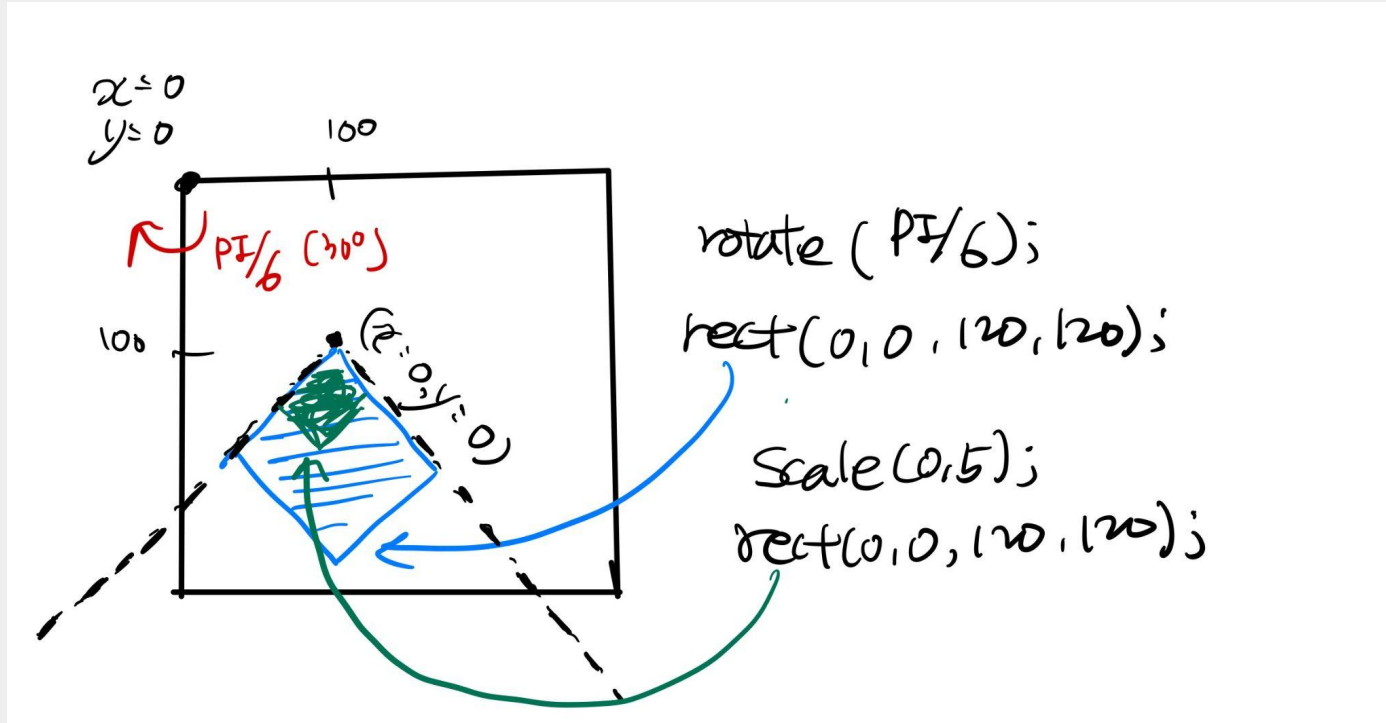
Translate :

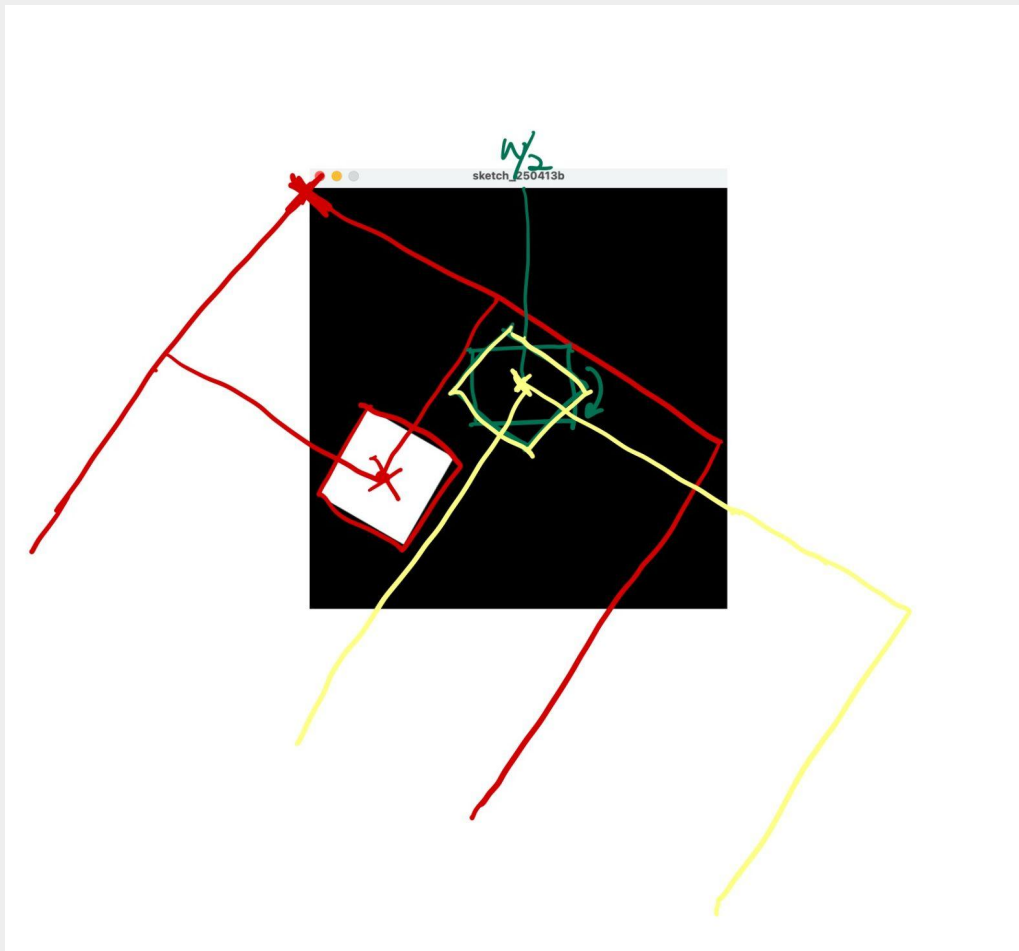


Rotate :

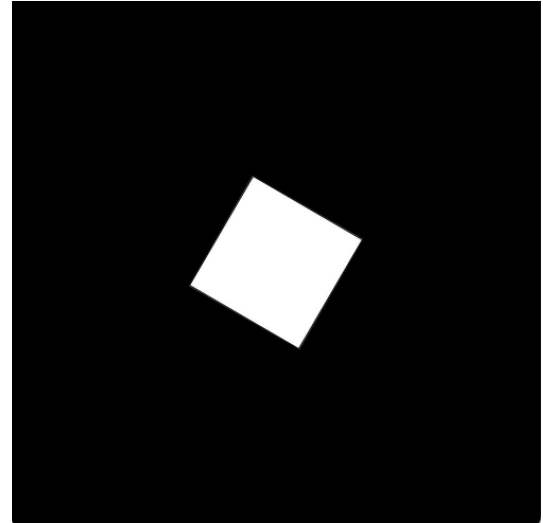


Scale :

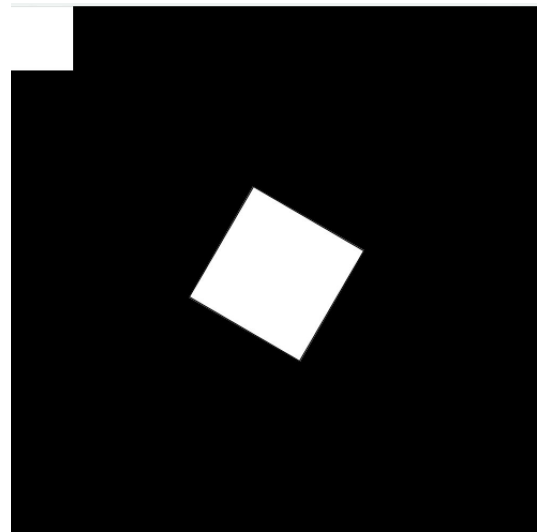




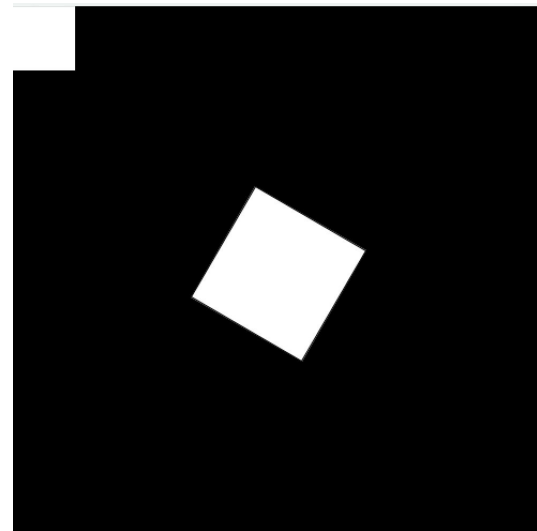
```
void setup() {  
  size(500, 500);  
}  
void draw() {  
  background(0);  
  rectMode(CENTER);  
  translate(width/2, height/2);  
  rotate(PI/6);  
  rect(0,0, 120, 120);  
}
```



```
void setup() {  
    size(500, 500);  
}  
void draw() {  
    background(0);  
    rectMode(CENTER);  
    translate(width/2, height/2); //좌표계를 0,0에서  
    실행창 중앙으로 이동.  
    rotate(PI/6); //30도만큼 좌표계 회전  
    rect(0, 0, 120, 120);  
    rotate(-PI/6); //다시 역으로 -30도만큼 좌표계회전  
    translate(-width/2, -height/2); //역으로 좌표계를  
    0,0으로 이동  
    rect(0, 0, 120, 120); //거기서 렉트 그려라.  
}
```



```
void setup() {  
    size(500, 500);  
}  
void draw() {  
    background(0);  
    pushMatrix(); // pop전까지 좌표계 이동적용됨.  
    rectMode(CENTER);  
    translate(width/2, height/2);  
    rotate(PI/6);  
    rect(0, 0, 120, 120);  
    popMatrix();  
    rect(0, 0, 120, 120);  
}
```



아래의 코드는 pushMatrix()함수를 이용하여 새로운 원점에 대한 정보를 저장 후, popMatrix()함수를 이용하여 저장한 정보를 삭제하는 것이다.
따라서 이후의 도형에는 새로운 원점이 적용되지 않게 된다.

```
pushMatrix();  
translate(50, 50);  
rect(100, 50, 100, 100);  
popMatrix();  
rect(100, 150, 100, 100);
```

새로운 원점이 적용되지 않음

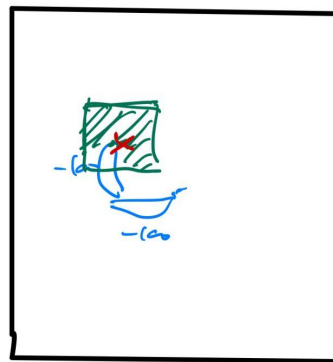
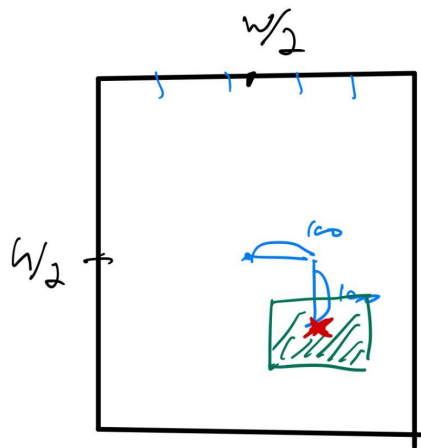
```
float ang =0;
void setup() {
    size(500, 500);
}
void draw() {
    ang += 0.02;
    background(0);
    pushMatrix();
    rectMode(CENTER);
    translate(width/2, height/2);
    rotate(ang); //ang라디안각도만큼 회전해라.
    rect(0, 0, 120, 120);
    popMatrix();
    rect(0, 0, 120, 120);
}
```

```

void setup() {
  size(500, 500);
}

void draw() {
  background(0);
  rectMode(CENTER);
  translate(width/2, height/2); // 중심 이동
  scale(1);
  // scale(-1); // -100,-100으로 그려져서
  위치함.
  rect(100, 100, 120, 120);
}

```

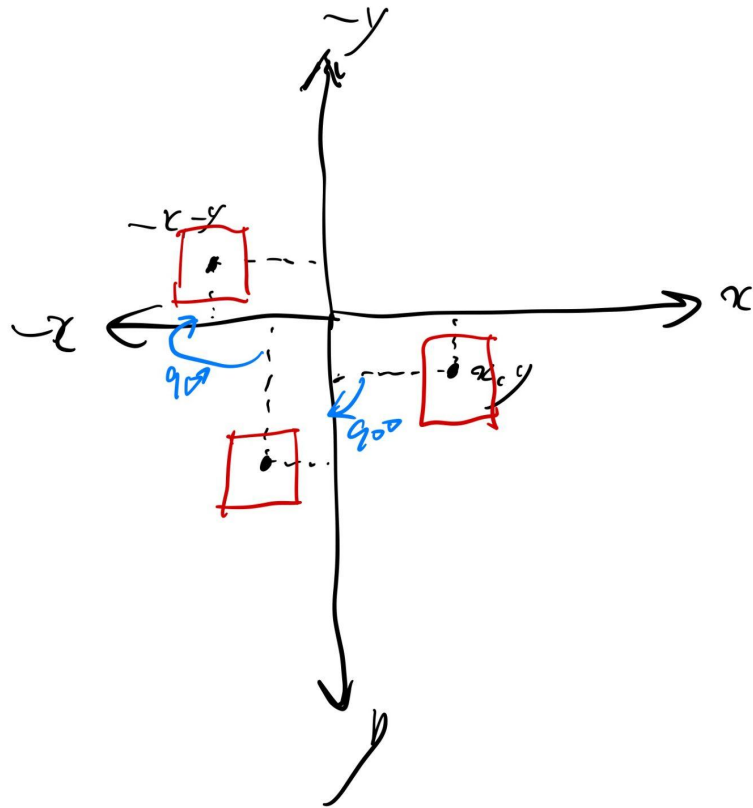


```

float ang =0;
void setup() {
  size(500, 500);
}
void draw() {
  ang += 0.02;

  background(0);
  pushMatrix();
  rectMode(CENTER);
  translate(width/2, height/2);
  rotate(ang); //ang라디안각도만큼
회전해라.
  scale(0.5);
  rect(100,100, 120, 120);
  popMatrix();
  //rect(0, 0, 120, 120);
}

```



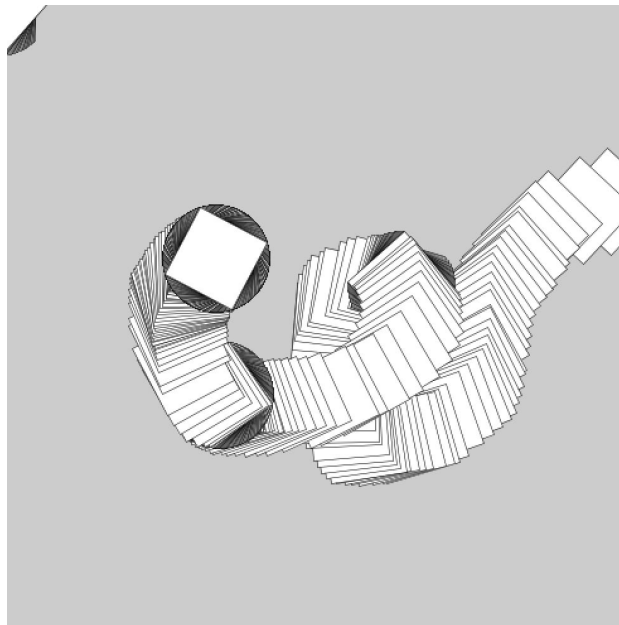

```
float ang =0;

void setup() {
    size(500, 500);
}

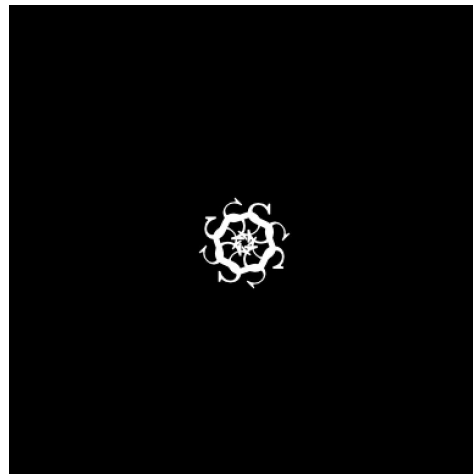
void draw() {

    ang += 0.02;

    //background(0); //지움.
    pushMatrix();
    rectMode(CENTER);
    translate(mouseX, mouseY); //마우스 위치"를
새로운 (0,0)으로 만들어버려서, 거기서 회전하고
스케일이 적용
    rotate(ang);
    scale(0.5);
    rect(0,0, 120, 120);
    popMatrix();
    // rect(0, 0, 120, 120);
}
```

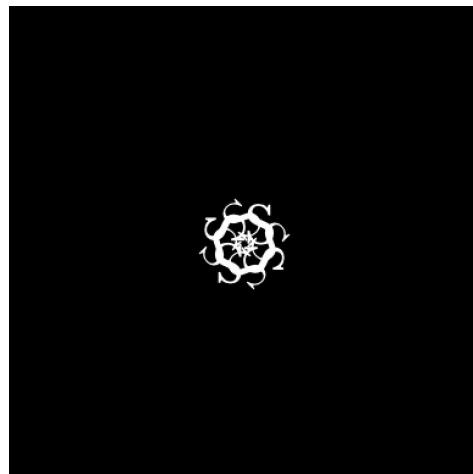


```
void setup() {  
  size(400, 400);  
  myFont = createFont("Baskerville", 48);  
}  
  
void draw() {  
  background(0);  
  textFont(myFont);  
  
  translate(width/2, height/2);  
  
  for (int i =0; i < 8; i++) {  
    pushMatrix();  
    rotate(TWO_PI/8 *i );  
    text("S", 0, 0);  
    popMatrix();  
  }  
}
```



```
void setup() {  
  size(400, 400);  
  myFont = createFont("Baskerville", 48);  
}
```

```
void draw() {  
  background(0);  
  textFont(myFont);  
  
  translate(width/2, height/2);  
  
  for (int i =0; i < 8; i++) {  
    pushMatrix();  
    rotate(TWO_PI/8 *i );  
    text("S", 0, 0);  
    popMatrix();  
  }  
}
```



연습문제 : 아래의 조건을 만족하는 코드 작성하기.

translate(), rotate(), scale, push, popMatrix등의 개념을 사용해서 다음 조건을 모두 만족하는 "꽃잎 패턴"을 만드세요.

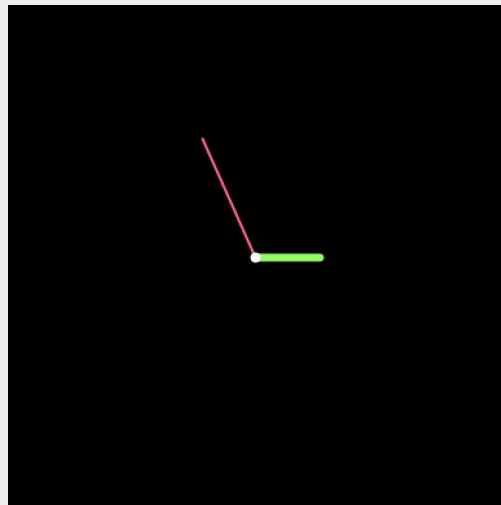
- 중심을 기준으로 꽃잎이 **8개 방사형으로** 배치
- 꽃잎은 **핑크색** 타원 형태이며, 크기는 대략 80×80
- 꽃잎은 **세로 방향으로 납작하게(scale)** 늘린 형태여야 한다. (scale이용)
- 꽃잎은 **좌표계 회전**을 이용해 각각 균등한 간격으로 배치되어야 한다.
- 반복문(**for**)을 사용해서 **코드의 반복을 최소화**할 것.
- 꽃 중심에는 **노란색 원**을 그려 꽃의 중심을 표현할 것.



연습문제 : 아래의 조건을 만족하는 코드 작성하기.

초침 + 시침 아날로그 시계 만들기

- 시계의 중심은 화면 가운데에 위치하도록 `translate()`를 사용할 것.
- `hour()`과 `second()` 내장 함수를 사용하여
- 시침: `int hr = hour() % 12; // 0~23를 반환. 12를 나눈 나머지값이 1~12가 됨.`
- 초침: `int sc = second(); // 0에서 59반환.`
- `map()` 함수를 활용하여 시각을 0~TWO_PI 라디안 각도로 변환할 것.
- `rotate()`를 이용해 각각 시침과 초침이 회전하도록 만들 것.
- 각 바늘은 라인함수로 그리되, `pushMatrix()` / `popMatrix()`를 사용할 것.
- 바늘의 중심에는 ****하얀 점(point)****을 표시할 것.



시스템변수

시스템 상태나 입력과 관계된 변수

: keyPressed, mousePressed, key, keyCode, mouseX, mouseY 등

변수 이름	변수 타입	의미
width	float	현재 스케치 창의 너비(픽셀)
height	float	현재 스케치 창의 높이(픽셀)
displayWidth	float	컴퓨터 화면(디스플레이)의 전체 너비
displayHeight	float	컴퓨터 화면(디스플레이)의 전체 높이
frameRate	float	초당 몇 프레임을 그릴지 설정된 목표 FPS
frameCount	int	현재까지 그려진 프레임 수 (자동 증가)

변수 이름	변수 타입	의미
<code>mouseX</code>	<code>float</code>	마우스의 현재 X 좌표 (픽셀)
<code>mouseY</code>	<code>float</code>	마우스의 현재 Y 좌표 (픽셀)
<code>pmouseX</code>	<code>float</code>	이전 프레임에서의 마우스 X 좌표
<code>pmouseY</code>	<code>float</code>	이전 프레임에서의 마우스 Y 좌표
<code>mousePressed</code>	<code>boolean</code>	마우스 버튼이 눌러 있는지 (<code>true/false</code>)
<code>mouseButton</code>	<code>int</code>	어떤 버튼이 눌렸는지 (LEFT, RIGHT, CENTER)

변수 이름	변수 타입	의미
keyPressed	boolean	키보드가 눌러 있는 상태인지 여부 (true/false)
key	char	눌린 문자 키(일반 키). 예: 'A', '1', '%' 등 (알파벳, 숫자,부호) (방향키나 특수키일 경우 CODED 값이 들어옴)
keyCode	int	눌린 키의 값. 주로 방향키, Shift, Ctrl 등 특수키 구분용

```
void setup() {  
  size(400, 400);  
}  
  
void draw() {  
  background(0);  
  
  textSize(50); // 추가  
  text("Hi!", width-100, height-200);  
}
```

A black square with the text "Hi!" in white, positioned in the bottom right corner of the square.

Hi!

```
void setup() {  
    size(400, 400);  
}  
  
void draw() {  
    background(0);  
  
    if (mousePressed == true) {  
        fill(255, 255, 0);  
    } else {  
        fill(255);  
    }  
  
    textSize(50);  
    text("Hi!", width-100, height-200);  
}
```

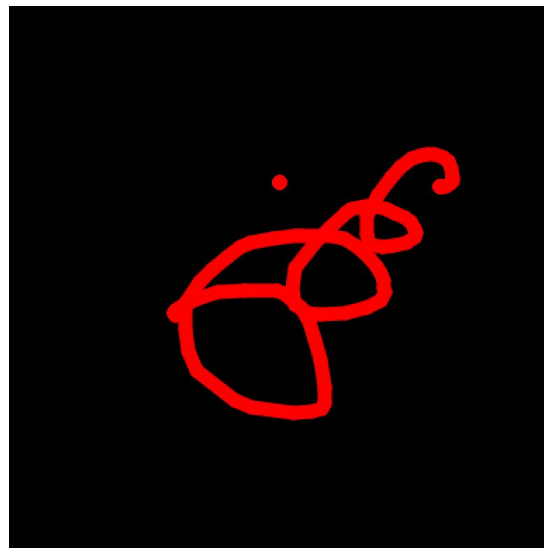
```
void setup() {  
    //size(displayWidth, displayHeight  
);  
    fullScreen();//전체화면  
}  
  
void draw() {  
    background(0);  
  
    if (mousePressed == true) {  
        fill(255, 255, 0);  
    } else {  
        fill(255);  
    }  
  
    textSize(50);  
    text("Hi!", width-100, height-200);  
}
```

```
void setup() {  
  size(400, 400 );  
  //fullScreen();  
}  
  
void draw() {  
  background(0);  
  
  if (mousePressed == true) {  
    fill(255, 255, 0);  
  } else {  
    fill(255);  
  }  
  
  textSize(50);  
  text(frameCount, width-100, height-300);  
  text(frameRate, width-100, height-200);  
}
```

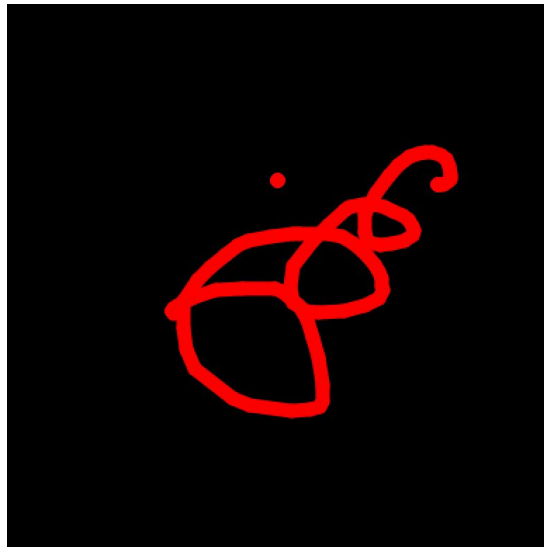
6773

59.7

```
void setup() {  
  size(400, 400);  
  frameRate(60);  
  //fullScreen();  
  background(0);  
}  
  
void draw() {  
  //background(0);  
  
  //noStroke();  
  //fill(255, 255, 0);  
  //ellipse(mouseX, mouseY, 100, 100);  
  
  if(mousePressed == true){  
    stroke(255, 0, 0);  
    strokeWeight(10);  
    line(pmouseX, pmouseY, mouseX, mouseY);  
  }  
  
}
```



```
void setup() {  
  size(400, 400);  
  frameRate(60);  
  //fullScreen();  
  background(0);  
}  
  
void draw() {  
  //background(0);  
  
  //noStroke();  
  //fill(255, 255, 0);  
  //ellipse(mouseX, mouseY, 100, 100);  
  
  if (mousePressed == true) {  
    if (mouseButton == LEFT) { //왼쪽버튼 : 화면그리기  
      stroke(255, 0, 0);  
      strokeWeight(10);  
      line(pmouseX, pmouseY, mouseX, mouseY);  
    } else if (mouseButton == RIGHT) {  
      background(0); //오른쪽버튼 : 화면지우기  
    }  
  }  
}
```



```
float xpos, ypos;

void setup() {
  size(400, 400);
  xpos  = width/2;
  ypos  = height/2;
}

void draw() {
  background(0);

  if ( keyCode  == LEFT) {
    xpos--;
  }
  if (keyCode == RIGHT) {
    xpos++;
  }

  textSize(150);
  text(key, xpos, ypos);
}
```



```
float xpos, ypos;
char currentKey; // 현재 글자 저장

void setup() {
    size(400, 400);
    xpos = width / 2;
    ypos = height / 2;
}

void draw() {
    background(0);

    if (keyPressed) { //지금 키보드가 눌러있는 상태인지.

        // 방향키일 때 이동만 하고 글자 안 바뀜
        if (key == CODED) { //방향키같은 특수키일때
            if (keyCode == LEFT) {
                xpos--;
            }
            if (keyCode == RIGHT) {
                xpos++;
            }
        }
    }
}
```

```
//일반 키를 눌렀을 땐 글자 저장 (위치는 그대로)

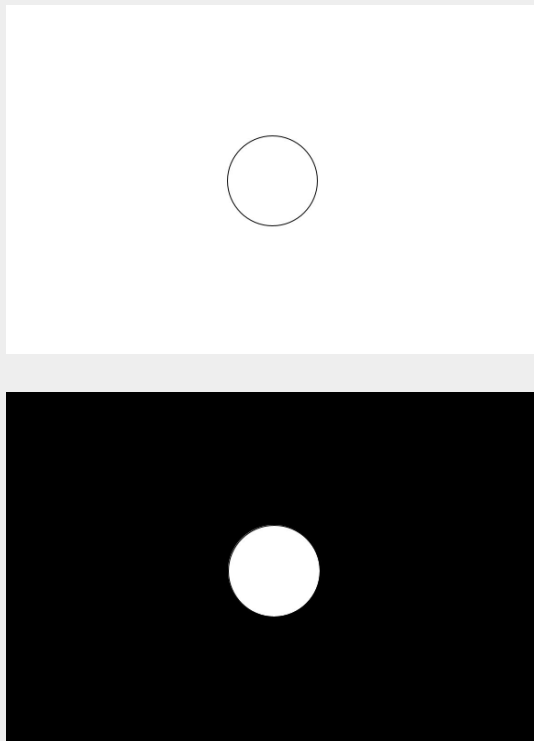
else {
    currentKey = key;
}

textSize(150);
text(currentKey, xpos, ypos);
}
```

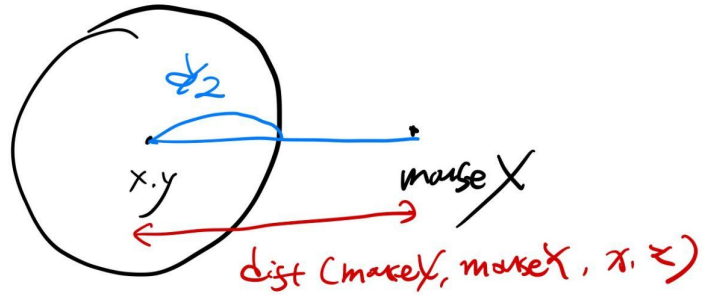
연습문제 : 아래의 조건을 만족하는 코드 작성하기.

마우스가 원안에 들어가면 배경색 바꾸기!

- 화면 중앙에 반지름이 50인 원을 그린다.
- 마우스가 원 안으로 들어가면 배경색이 검정색, 원 바깥에 있으면 배경색이 하얀색이 되도록 한다.
- **dist()** 함수를 사용하여 마우스와 원 중심 사이의 거리를 계산할 것.
- 조건문 (**if**) 을 사용하여 배경색을 바꾼다.



$\text{dist}(\text{mouseX}, \text{mouseY}, x, y)$
 $\text{ellipse}(x, y, d, d)$



연습문제 : 아래의 조건을 만족하는 코드 작성하기.

아래 조건에 맞는 **슬라이더 인터페이스**를 Processing으로 구현하시오. (mousePressed 이용)

- 슬라이더 막대는 $y = 100$ 위치에 가로로 길게 표시되며,
시작 x좌표는 50, 끝 x좌표는 350이다. (라인함수로)
- 사용자가 슬라이더 막대 범위 안을 마우스로 클릭하고 (mousePressed) 드래그 (mouseX) 하면, 마우스의 x위치에 따라 슬라이더 값이 0에서 100 사이로 실시간 변한다.
- 슬라이더 손잡이 (작은 원)는 슬라이더 값에 따라 막대 위를 따라 움직인다.

