

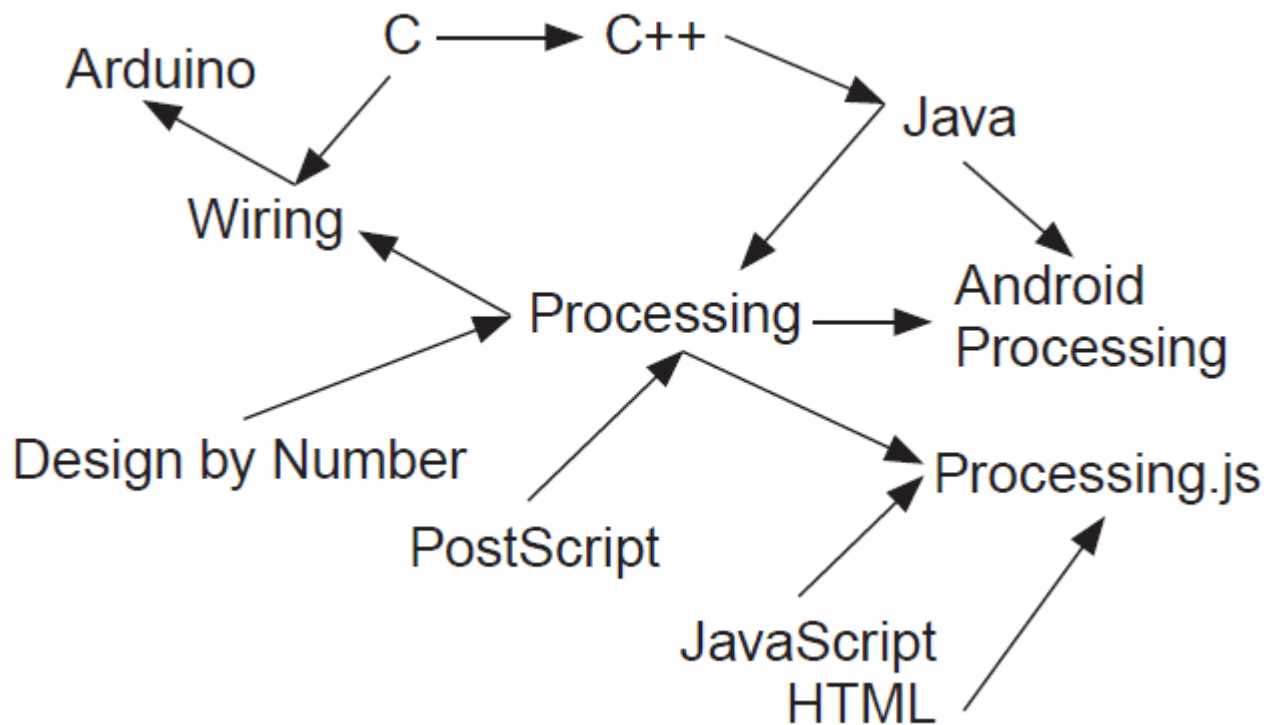
# 재미삼아 프로세싱 1장

프로세싱 소개

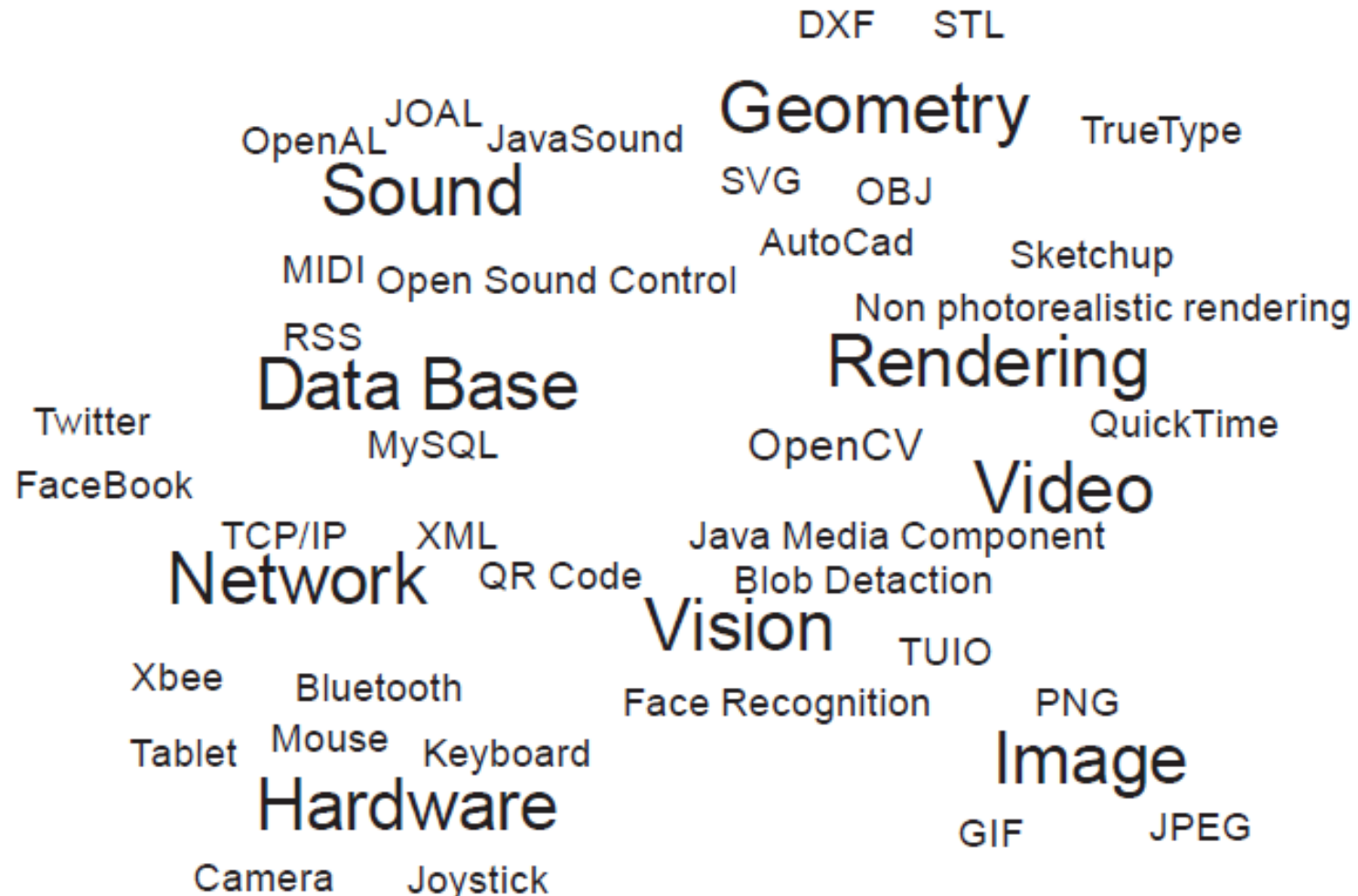
# Processing

- 공식사이트: <http://processing.org>
- 누구를 위한 언어
  - 디자이너, 미디어 아티스트
- 어떤 언어? Java 방언의 일종
- 네트워크 처리
- 2D, 3D, 영상 및 비디오 처리
- PDF, SVG
- 하드웨어 제어

# 프로세싱의 족보



# 프로세싱의 적용 분야



# 프로세싱의 멋진 기능 1/2

- 도형을 그리기 쉽다.
- 3차원을 쉽게 구현할 수 있다.
- PDF로 데이터를 저장할 수 있다.
- 영상처리가 매우 쉽다
- 여러 가지 영상 포맷을 지원한다.
- 사운드 처리가 쉽다
- 카메라를 연결하고 비디오를 바로 처리할 수 있다.

# 프로세싱의 멋진 기능 1/2

- 웹과 네트워크의 연결이 쉽고 자유롭다.
- SVG가 지원된다.
- 동영상 제작이 쉽다.
- 파일 입출력이 자유롭다.
- 아두이노 하드웨어와 쉽게 연동할 수 있다.
- 안드로이드 프로그램이 가능하다.

# Processing Summery

- Free to download and open source
- Interactive programs using 2D, 3D or PDF output
- OpenGL integration for accelerated 3D
- For GNU/Linux, Mac OS X, and Windows
- Projects run online or as double-clickable applications
- Over 100 libraries extend the software into sound, video, computer vision, and more...

# 재미삼아 프로세싱 2장

프로세싱 시작하기



# 프로세싱 설치

- 다운로드 받기
  - <http://processing.org>
- 압축풀기
- 프로세싱 실행

1.5.1 | 15 May 2011

↓ Linux

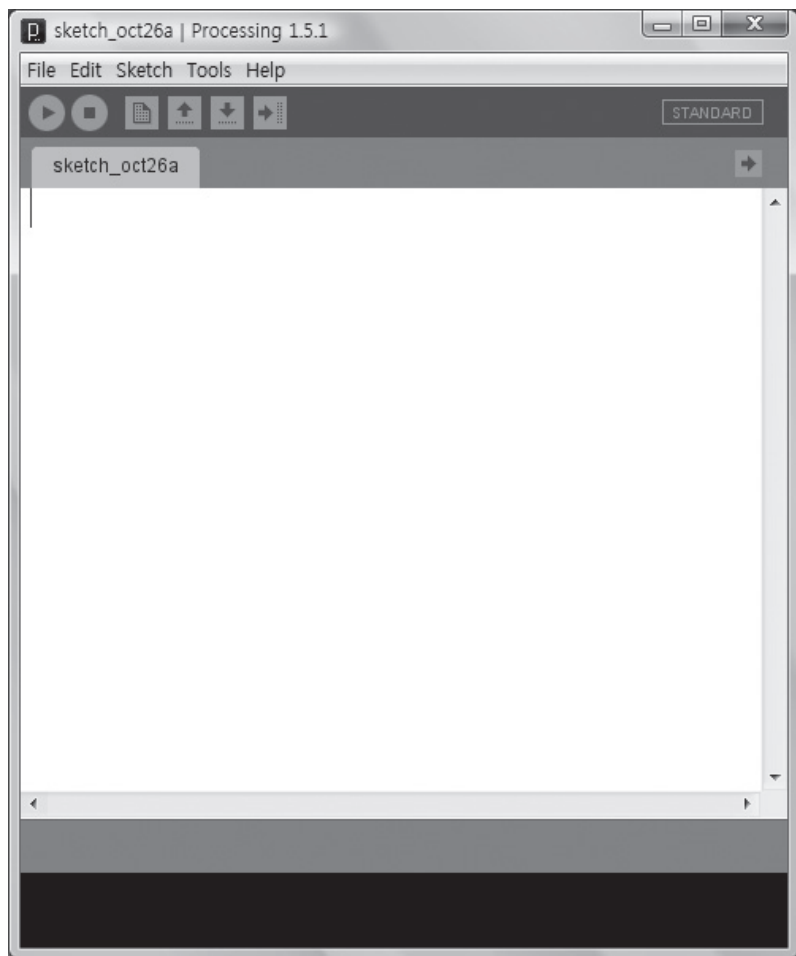
↓ Windows

↓ Mac OSX

↓ Windows [Without Java]\*

The [list of revisions](#) covers the differences between releases in detail. Please read the [changes](#) if you're new to the 1.0 series. Also check the [known problems](#) for this release.

# 실행화면

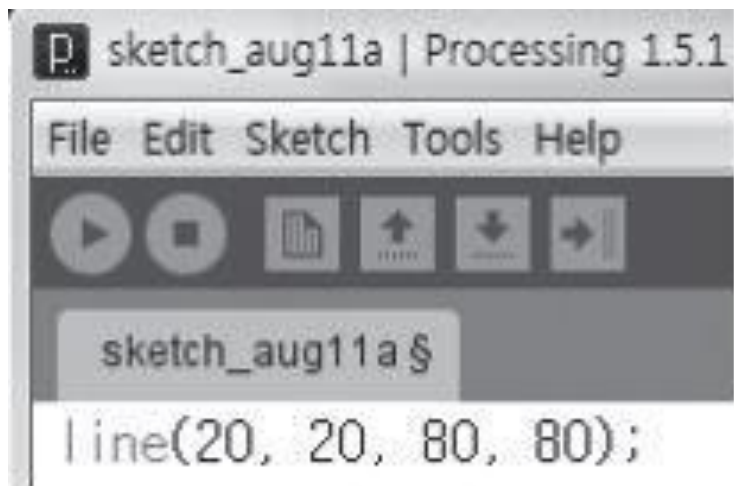


- 스케치
- 스케치북

STANDARD/ANDROID

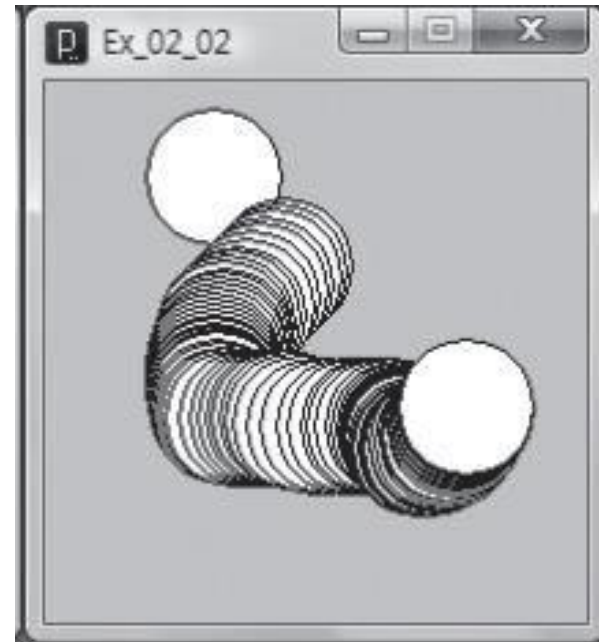
Application/Applet

# 선 긋기



# 마우스

```
void setup() {  
  size(200, 200);  
}  
void draw() {  
  if (mousePressed) {  
    ellipse(mouseX,  
            mouseY, 50, 50);  
  }  
}
```



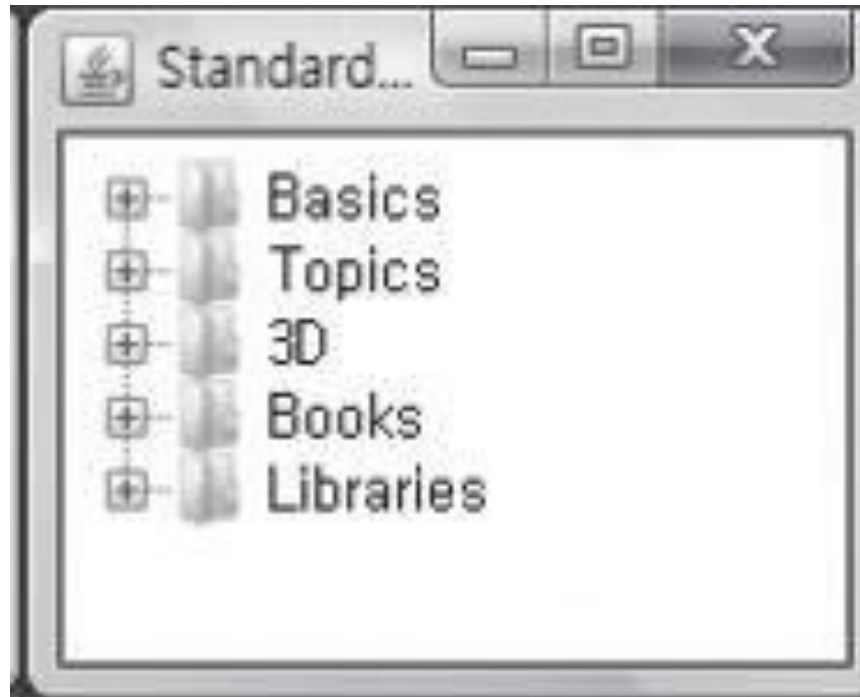
# 응용 프로그램

- Export Application)



# 프로세싱 예제

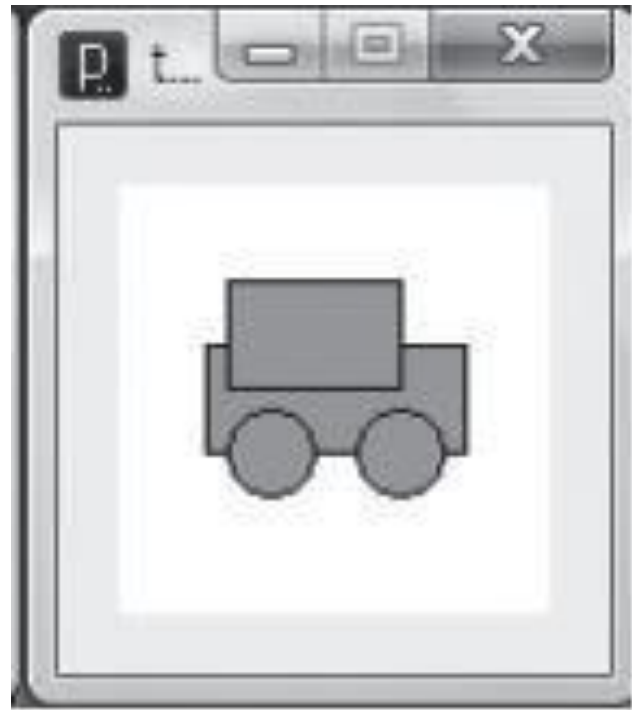
- File > Examples....



# 실습

- 자동차를 함께 디자인  
해 보요~

```
background(255);  
rectMode(CENTER);  
fill(160);  
rect();  
ellipse()
```



# 재미삼아 프로세싱 3장

도형 그리기



# 좌표

- **size**(500, 200);
- **point**(100, 50);
- **line**(100, 50, 100, 150);

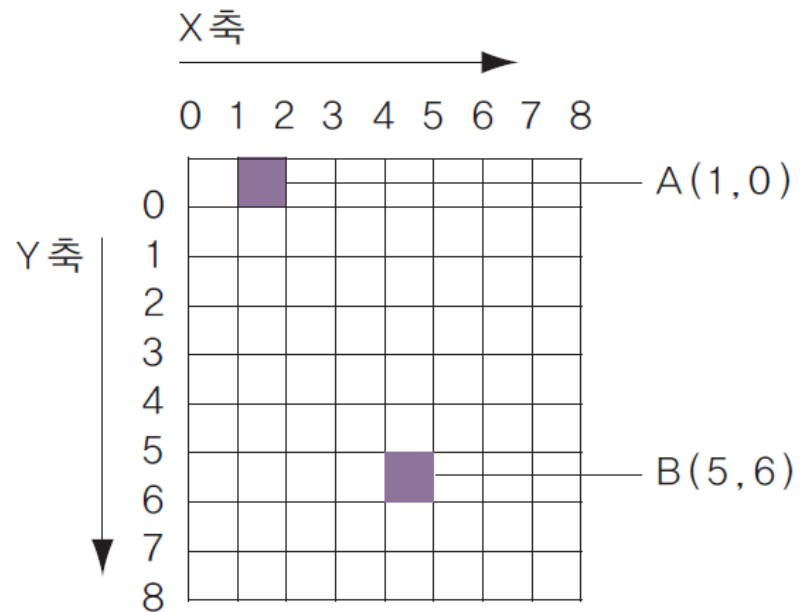
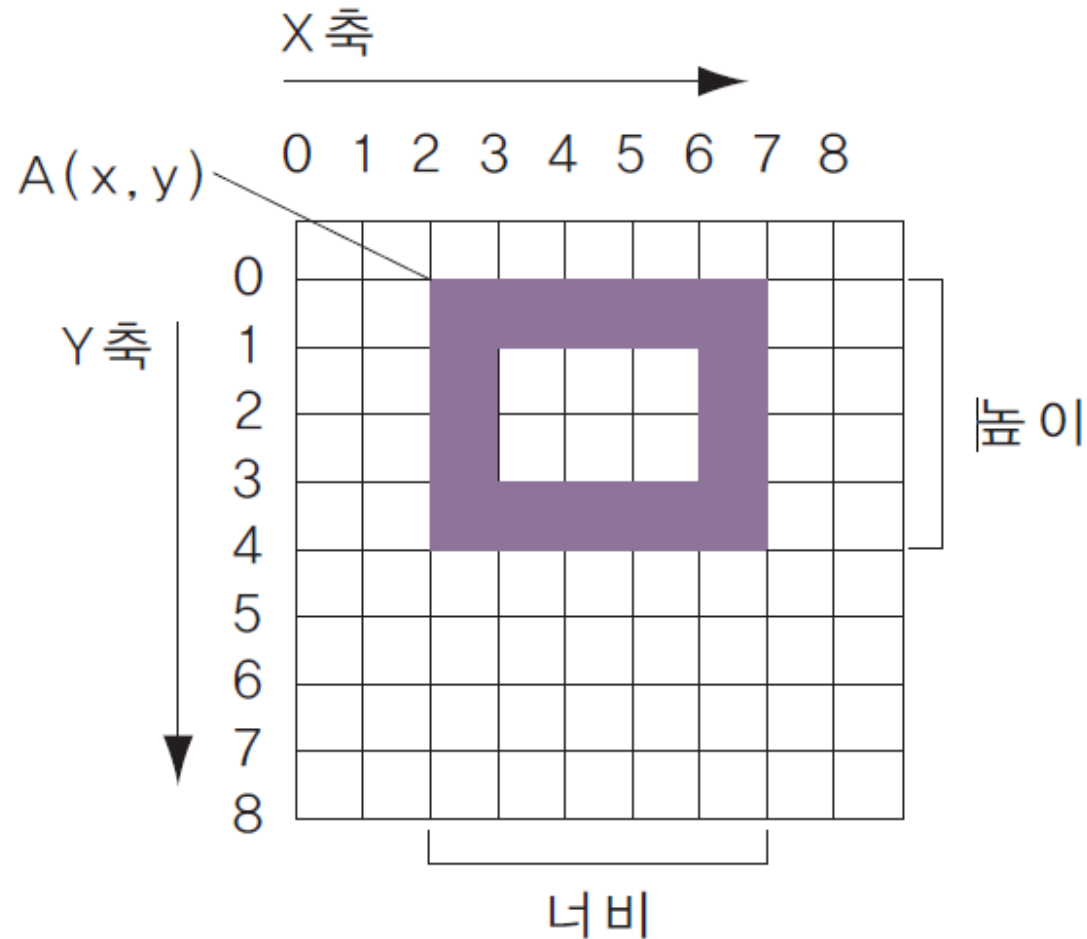


그림 3-1 좌표 이해하기

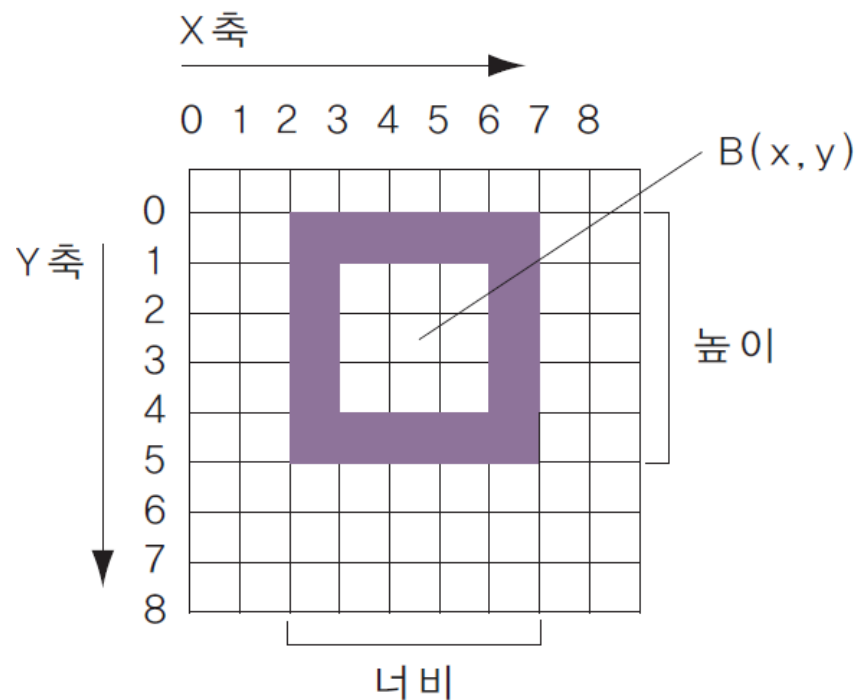
# 사각형 rect()

- **rect**(2,1,5,4



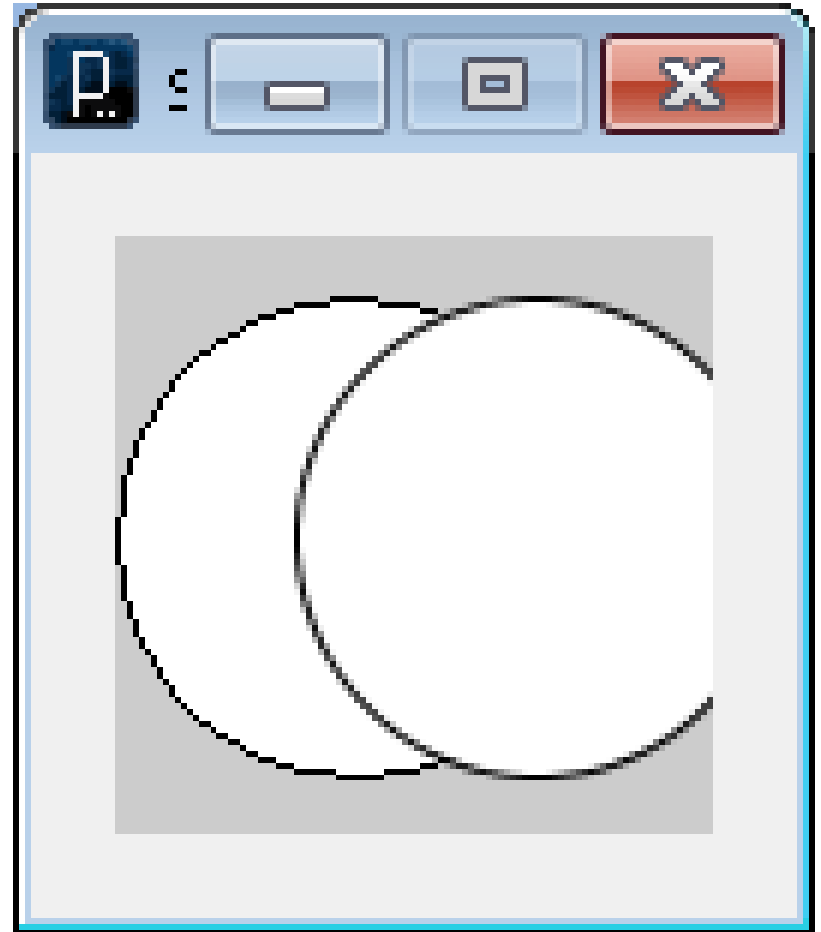
# rectMode(CENTER)

- **rectMode**(CENTER);
- rect(2,1,5,4);



# smooth()

- ellipse(40, 50, 80, 80);
- **smooth**();
- ellipse(70, 50, 80, 80);

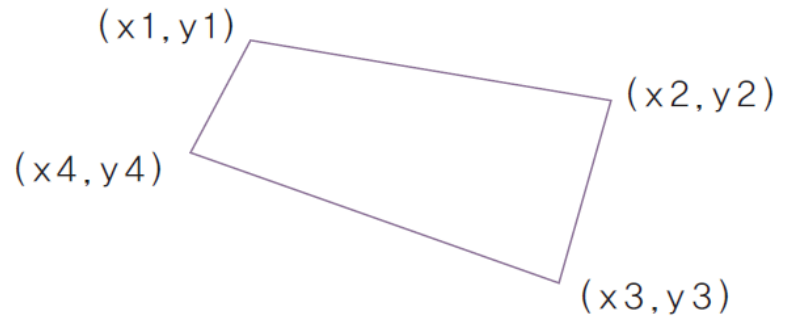


```
quad(x1, y1, x2, y2, x3, y3, x4, y4)
```

```
size(500, 200);
```

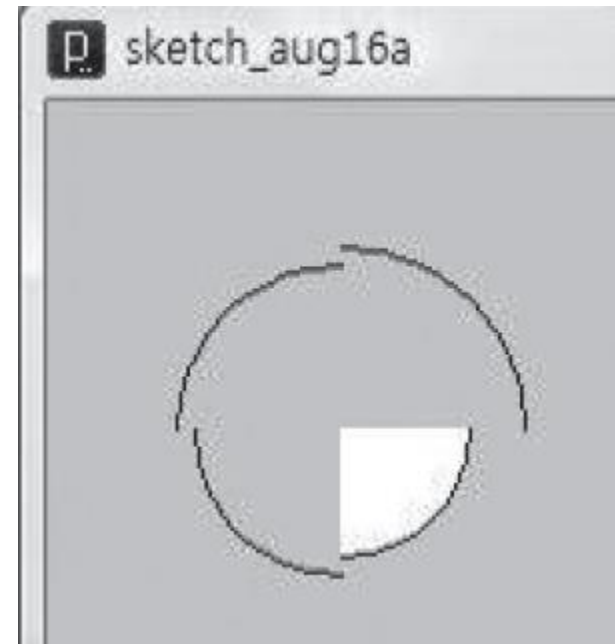
```
smooth();
```

```
quad(100, 50, 350, 75,  
      300, 170, 80, 120);
```



# `arc()`, `PI`, `TWO_PI`, `noFill()`

```
size(500, 200);  
arc(70, 90, 60, 60, 0, PI/2);  
noFill();  
arc(70, 90, 70, 70, PI/2, PI);  
arc(70, 90, 80, 80, PI,  
    TWO_PI-PI/2);  
arc(70, 90, 90, 90, TWO_PI-  
    PI/2, TWO_PI);
```



# 선의 폭 및 깎인 모양

// 선의 두께

**strokeWeight**(10);

// 모서리가 깎인 모양

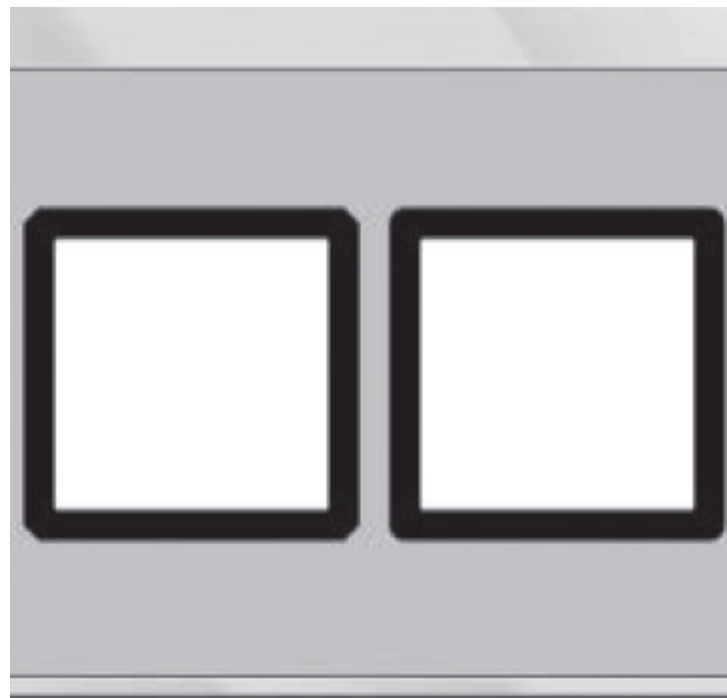
**strokeJoin**(**BEVEL**);

rect(170, 50, 100, 100);

// 모서리가 둥글게

strokeJoin(**ROUND**);

rect(290, 50, 100, 100);



# 선의 끝 모양

```
strokeWeight(10);  
// 선의 끝이 둥글게  
strokeCap(ROUND);  
line(20, 50, 120, 50);  
// 선의 끝이 일직선  
strokeCap(SQUARE);  
line(20, 100, 120, 100);  
// 선의 끝이 더 연장됨  
strokeCap(PROJECT);  
line(20, 150, 120, 150);
```





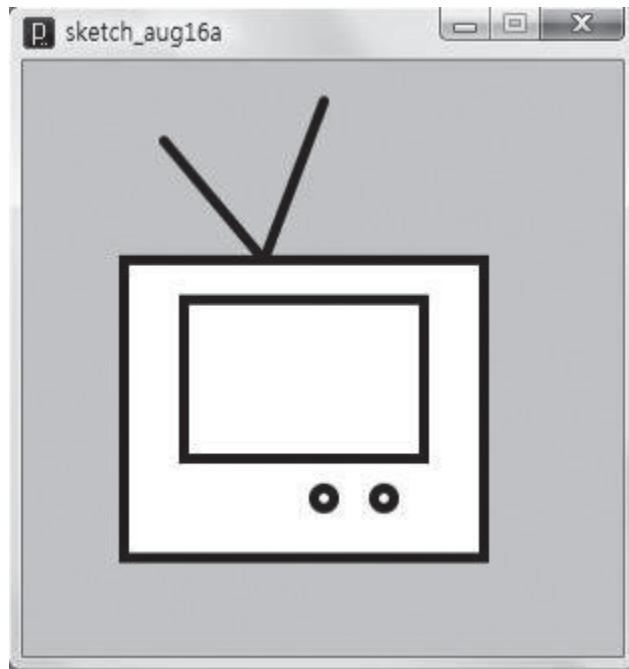
# 투명 도형 그리기

```
size(340, 120);  
fill(255, 0, 0, 255);  
rect(10, 10, 100, 100);  
fill(255, 0, 0, 128);  
rect(120, 10, 100, 100);  
fill(255, 0, 0, 0); //투명  
rect(230, 10, 100, 100);
```



# 실습

- TV를 디자인 해보자



# 재미삼아 프로세싱 4장

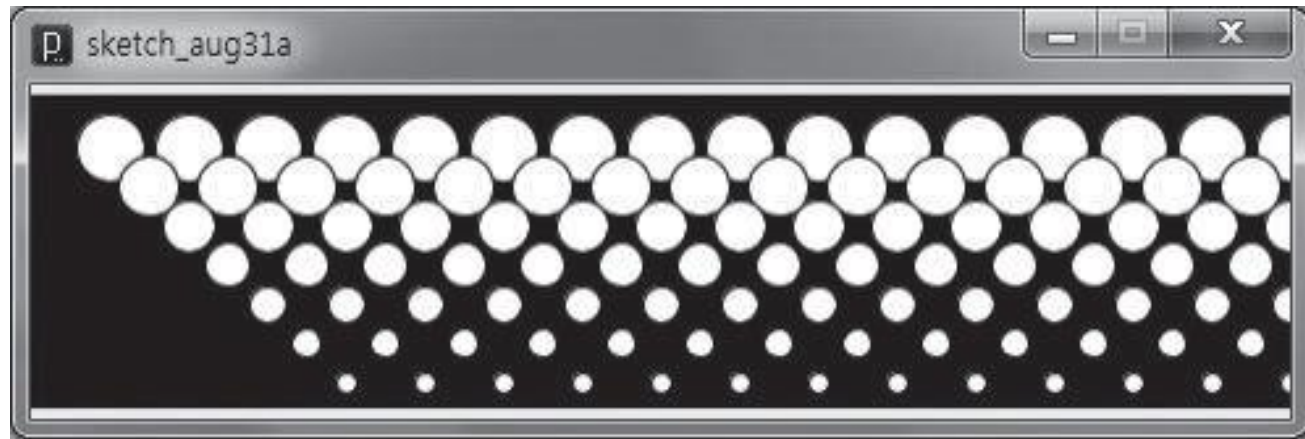
변수

# 시스템 변수

변수명	기능	변수명	기능
width	창의 너비	height	창의 높이
mouseX	마우스의 x 좌표값	mouseY	마우스의 y 좌표값
frameCount	프레임 처리 수	frameRate	초당 프레임의 수
keyPressed	키 누름 상태 확인	mousePressed	마우스 누름 상태 확인

# 이중 반복

```
for (int y = 20; y <= height; y += 15) {  
    for (int x = 10; x <= width; x += 30) {  
        ellipse(x+y, y, 30-y/5.0, 30-y/5.0);  
    }  
}
```



# 재미삼아 프로세싱 5장

키보드 입력

# 프로그램의 구조

```
void setup(){  
    // 1회 실행  
}  
void draw(){  
    // 여러 번 실행  
}
```

# 흔적 남기기

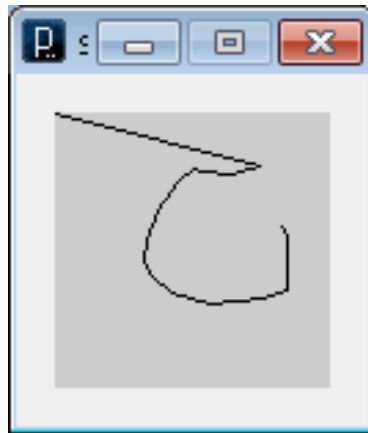
```
void setup() { fill(0, 102); smooth(); noStroke(); }  
void draw() { ellipse(mouseX, mouseY, 9, 9); }
```





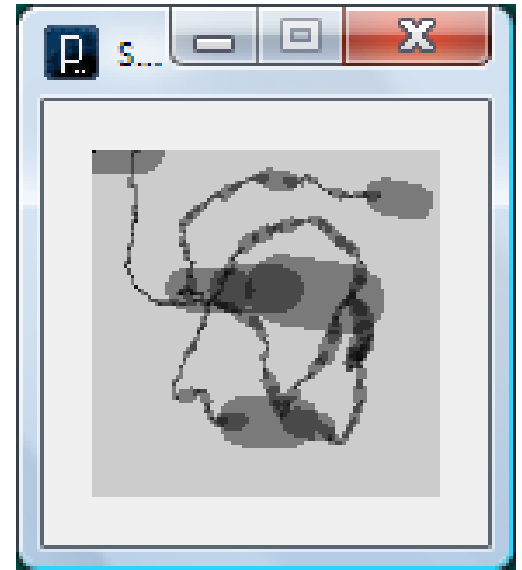
# 선 긋기

```
void draw() {  
    line(mouseX, mouseY, pmouseX, pmouseY);  
}
```



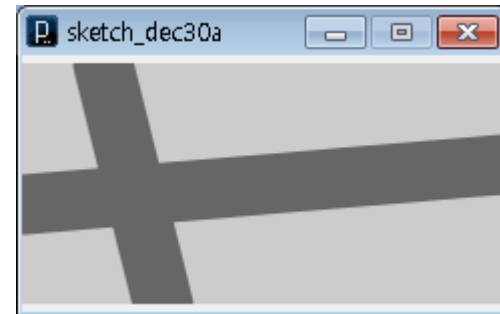
# 마우스 속도에 따른 선 굵기

```
void setup() { stroke(0, 102); }  
void draw() {  
  float weight =  
    dist(mouseX, mouseY, pmouseX, pmouseY);  
  strokeWeight(weight);  
  line(mouseX, mouseY, pmouseX, pmouseY);  
}
```



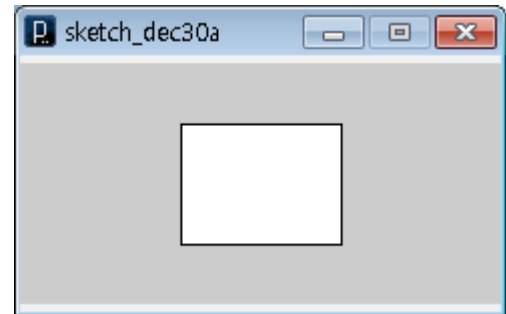
# 마우스 클릭

```
void setup() {size(240, 120); smooth(); strokeWeight(30);}
void draw() {
  background(204);
  stroke(102);
  line(40, 0, 70, height);
  if (mousePressed == true) {
    stroke(0);
  }
  line(0, 70, width, 50);
}
```



# 마우스의 위치 검사

```
int x=80; int y=30; int w=80; int h=60;
void setup() { size(240, 120);}
void draw() { background(204);
    if ((mouseX > x) && (mouseX < x+w) &&
        (mouseY > y) && (mouseY < y+h)) {
        fill(0);
    }
    else { fill(255); }
    rect(x, y, w, h);
}
```



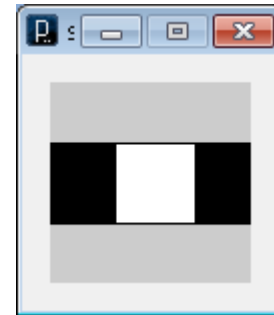
# 키보드 입력

```
void setup() {  
    size(120, 120);  
    textSize(64);  
    textAlign(CENTER);  
}  
  
void draw() {  
    background(0);  
    text(key, 60, 80);  
}
```



# 좌우 화살표 키

```
int x = 30;
void draw() {
    if (keyPressed && (key == CODED)) {
        if (keyCode == LEFT) {
            x--;
        } // the left arrow
        else if (keyCode == RIGHT) {
            x++;
        } // the right arrow
        rect(x, 30, 40, 40);
    }
}
```



# 재미삼아 프로세싱 6장

영상과 텍스트

# 지원하는 포맷

GIF, JPG, TGA, PNG

```
PImage img;  
loadImage("Test.jpg");
```

```
void draw(){  
    image(img, 0, 0);  
}
```



# 웹에서 영상 읽기

```
PImage img = loadImage("http://*.jpg");  
size(img.width, img.height);  
image(img, 0, 0);
```

# 디렉터리

- `img=loadImage("C:\\Processing\\picture.png");`
- `img=loadImage("input/picture.jpg");`

# 영상의 축소 확대

```
void setup(){  
    size(400,300);  
    img=loadImage("picture.jpg");  
}  
void draw(){  
    image(img, 0, 0);  
    image(img, 0, 0, 200, 150); //축소  
    image(img, 0, 0, 800, 600); //확대  
}
```

# 여러 장 영상 처리

- 영상배열
- `PImage images[] = new PImage[2];`

# 마우스로 영상크기 조절

```
PImage img;  
void setup(){  
  img = loadImage(http://\*.jpg);  
  size(img.width, img.height);  
}  
void draw() {  
  background(0);  
  image(img, 0, 0,  
    mouseX, mouseY);  
}
```

# 픽셀 처리 1/2

```
PImage img;  
void setup(){  
  img=loadImage("http://*.jpg");  
  size(img.width,img.height); // 크기  
}  
void draw(){  
  loadPixels(); // 픽셀 가져오기
```

# 픽셀 처리 2/2

```
for(int y=0; y<height; y++){
    for(int x=0; x<width; x++){
        int location=x+y*width;
        float r = red (255-img.pixels[location]);
        float g = green (255-img.pixels[location]);
        float b = blue(255-img.pixels[location]);
        pixels[location]=color(r,g,b);
    }
}
updatePixels();
}
```

# 텍스트 크기 및 정렬

```
void setup() {  
    textSize(64);  
    textAlign(CENTER);  
}  
  
void draw() {  
    text(key, 60, 80); // 입력하는 키보드 문자  
}
```



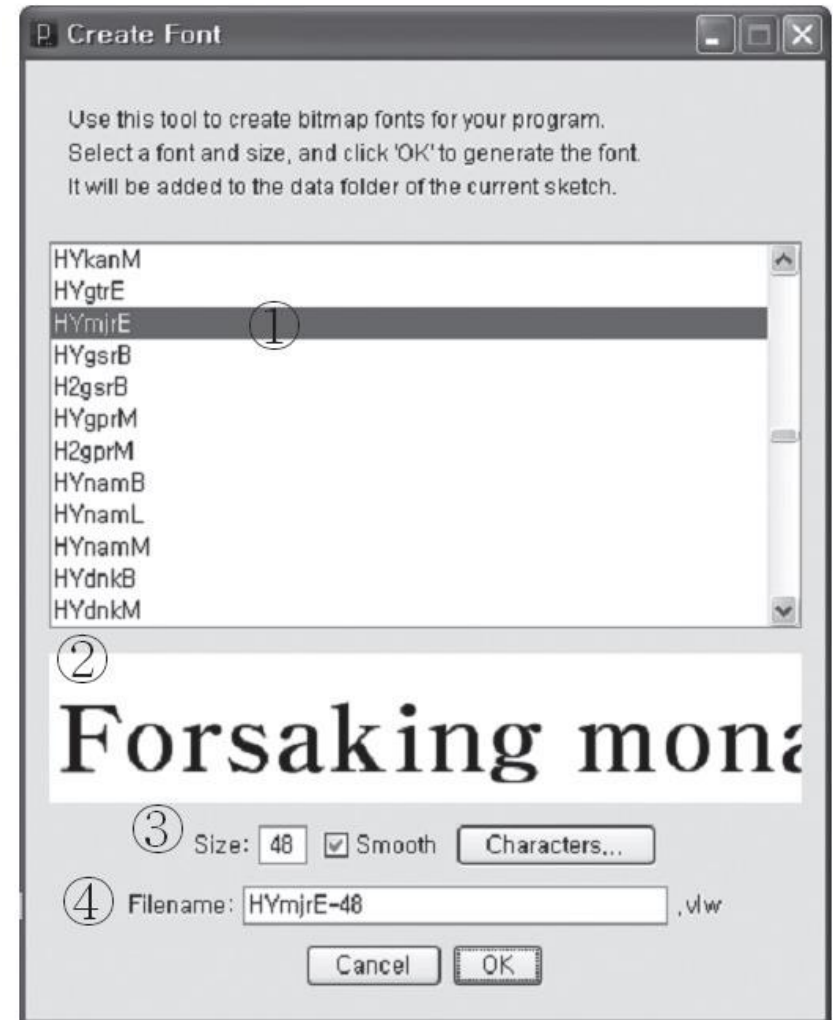
# 굴림 폰트 활용

```
PFont font;  
font = createFont("굴림", 48);  
textFont(font);  
fill(0);  
text("프로세싱", 0, 48);
```

# 폰트 생성 및 로드하기

Tools > Create font

```
PFont font;  
void setup(){  
  size(500,200);  
  font = loadFont(  
    "HYmjrE-48.vlw");  
}
```



# 문자출력

```
void draw() {  
    background(255);  
    textFont(font, 50);  
    fill(0);  
    textAlign(LEFT);  
    text("Fun life, 10, 70);  
}
```

# 사각형 속에 문자 출력

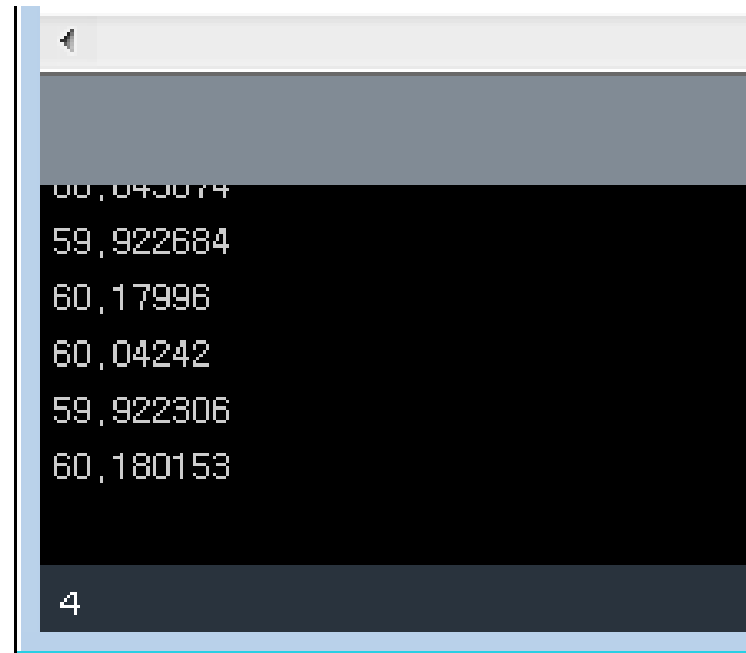
- `text("재미있는 프로세싱!!!!", 10, 50, 200, 200);`

# 재미삼아 프로세싱 7장

움직임

# 프레임 속도

```
void draw() {  
    println(frameRate);  
}
```



# frameRate 설정

```
void setup() {  
    frameRate(30);  
}  
void draw() {  
    println(frameRate);  
}
```

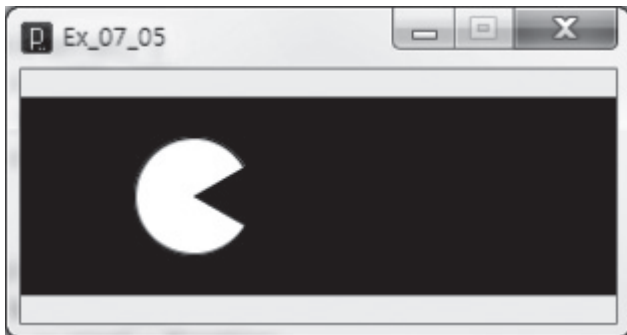
# 움직임

```
int x=0, speed=2;
void setup() {
  size(300, 100);
}
void draw() {
  background(0);
  x += speed;
  ellipse(x, 50, 50,50);
}
```



# 트위닝

```
if (direction == 1) { // 오른쪽 방향  
    arc(x, 50, radius, radius, 0.52, 5.76);  
} else { // 왼쪽 방향  
    arc(x, 50, radius, radius, 3.67, 8.9);  
}
```



# 원운동 시뮬레이션

```
float angle = 0.0; float offset = 60;  
float scalar = 30; float speed = 0.05;  
void setup() { size(120, 120); smooth(); }  
void draw() {  
    float x = offset + cos(angle) * scalar;  
    float y = offset + sin(angle) * scalar;  
    ellipse( x, y, 40, 40);  
    angle += speed;  
}
```

# 이동

```
void draw() {  
    translate(mouseX, mouseY);  
    ellipse(0, 0, 40, 40);  
}
```

# 회전

```
float angle = 0.0;  
void setup(){ smooth();}  
void draw() {  
    translate(mouseX, mouseY);  
    rotate(angle);  
    rect(-15, -15, 30, 30);  
    angle += 0.1;  
}
```

# 배율

```
float angle = 0.0;
void setup() { smooth();}
void draw() {
    translate(mouseX, mouseY);
    scale(sin(angle) + 2);
    ellipse(0, 0, 20, 20);
    angle += 0.1;
}
```

# 재미삼아 프로세싱 8장

함수

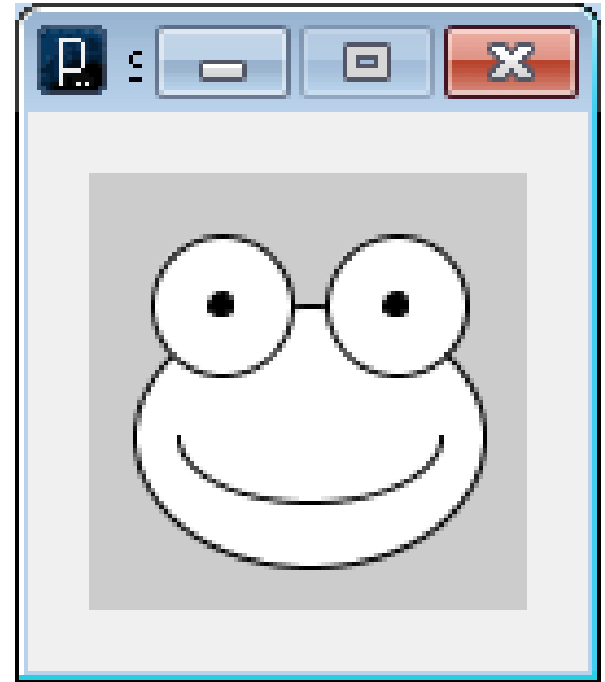
# 주사위 시뮬레이션

```
void setup() { // 주사위 3번 던지기
    dice();
    dice();
    dice();
}

void dice() {
    int d = int(random(1, 6));
    println("Rolling... " + d);
}
```

# 개구리 함수

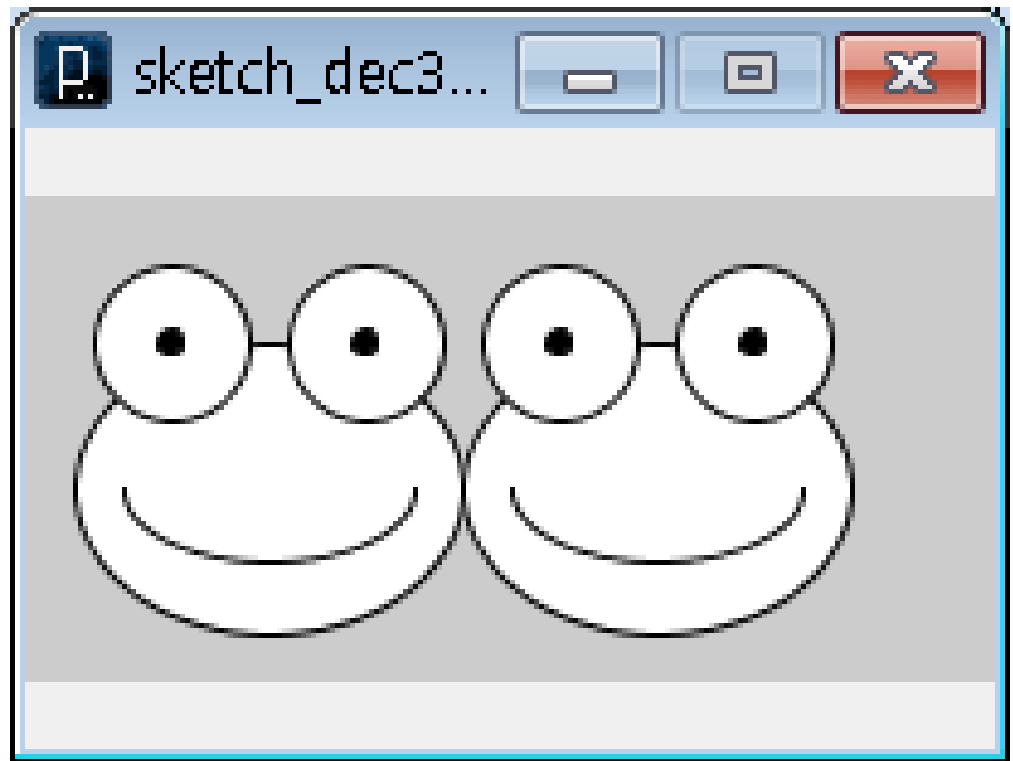
```
void frog(int x, int y){  
    pushMatrix(); // 추가된 부분  
    translate(x, y); // 위치의 이동  
    fill(255);  
    ellipse(50, 60, 80, 60); // face  
    arc(50, 60, 60, 30, 0, PI); // mouse  
    ellipse(50-20, 60-30, 32, 32); // left eye  
    ellipse(50+20, 60-30, 32, 32); // right eye  
    fill(0);  
    ellipse(50-20, 60-30, 5, 5); // left eye point  
    ellipse(50+20, 60-30, 5, 5); // right eye  
    popMatrix(); // 추가된 부분  
}
```





# 개구리 두 마리

```
void setup() {  
  size(200, 100);  
  smooth();  
  frog(0, 0);  
  // 위치 변경  
  frog(80, 0);  
}
```



# 개구리 여러 마리

```
for (int x = 0; x < width; x += 80) {  
    frog(x, 0);  
}
```

# 개구리 색상과 크기 변경 1/2

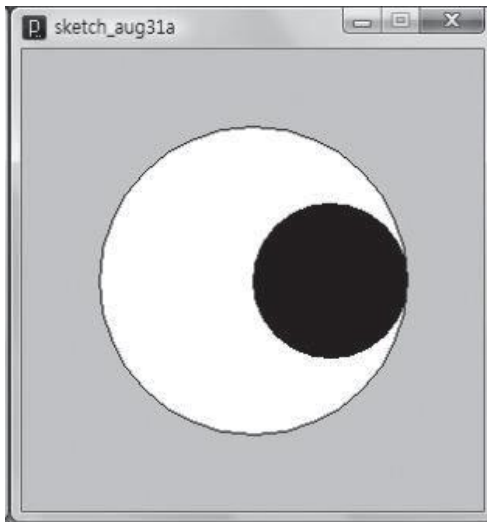
```
void setup() {  
    size(600, 100); smooth();  
    randomSeed(0);  
    for (int x = 0; x < width; x += 80) {  
        int g = int(random(0, 255)); // 색상 설정  
        float s = random(0.25, 1.0); // 크기 설정  
        frog(x, 0, g, s);  
    }  
}
```

## 개구리 색상과 크기 변경 2/2

```
void frog(int x, int y, int g, float s)
{
    pushMatrix();
    translate(x, y);
    fill(g); // 색상 바꾸기
    scale(s); // 크기 바꾸기
    ....
}
```

# 실습 확장

- 속이 빈 지름이 100인 원의 오른쪽에 속이 찬 지름이 50인 원을 스케치하라.
- 랜덤 컬러 개구리 만 들기



# 재미삼아 프로세싱 9장

객체

# 클래스 정의

```
class Ellip {                                // 클래스 이름 정의
    float xpos, speed;                       // 멤버 변수
    Ellip (float x, float s) {               // 생성자
        xpos = x;
        speed = s;
    }
    void move() {                            // 멤버 함수
        xpos += speed;
        if (xpos > width) { xpos = 0; }
        ellipse(0, xpos, height/3, 30, 30);
    }
}
```

# 공 1개 던지기

```
Obj Ellip;                                // Ellip 객체 선언
void setup(){
    size(200, 200); // 창의 크기
                    // 초기화하고 값을 할당
    Ellip = new Obj(20, 20);
}
void draw(){
    background(204);
    Ellip.move();                        // 원을 그려줌(메서드 실행)
}
```



# 재미삼아 프로세싱 10장

배열

# 바코드 생성

```
int [] code;
```

```
void setup(){
```

```
    code =new int[width];
```

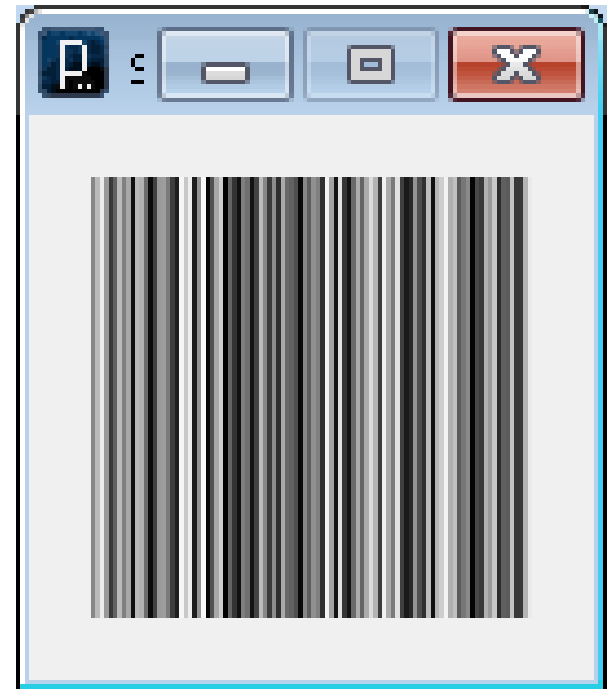
```
    for (int i=0;i<code.length;i++) {  
        code[i]=(int)random(0, 255);  
    }
```

```
}
```

```
void draw() {
```

```
    for (int i=0;i<code.length;i++) {  
        stroke(code[i]);  
        line(i, 0, i, height);  
    }
```

```
}
```



재미삼아 프로세싱 11장

# 객체의 배열

```
Obj [] cir = new Obj[70];
```

```
void setup() {  
    size(640, 480); smooth();  
    for (int i = 0; i < cir.length; i++){  
        int x = (int)random(width);  
        int y = (int)random(height);  
        int r = i + 2;  
        cir[i] = new Obj(x, y, r);  
    }  
}  
void draw() {  
    for (int i = 0; i < cir.length; i++) {  
        cir[i].display();  
    }  
}
```

```
class Obj {
```

```
    int x, y;
```

```
    int diameter;
```

```
    Obj(int tempX, int tempY, int  
        tempDiameter) {
```

```
        x = tempX; y = tempY;
```

```
        diameter = tempDiameter;
```

```
    }
```

```
    void display() {
```

```
        ellipse(x, y, diameter,  
                diameter);
```

```
    }
```

```
}
```

# 실행 결과

