Generative Design II

과제

계획

Drawing
Animation
Interaction
Variable
Conditional (if)
For
Images, Sound and Video
Functions / Class
p5js (web / mobile)
Arduino

Midterm : Idea presentation

Project Development (3 - 4 weeks)

Final Presentation

계획

Processing Basics

p5js Basics (web / mobile)

Project : Dynamic Typography
Project : Data Visualization
Project :
Project : Arduino?

Midterm : Idea presentation

Project Development (3 - 4 weeks)

Final Presentation

Processing Basics

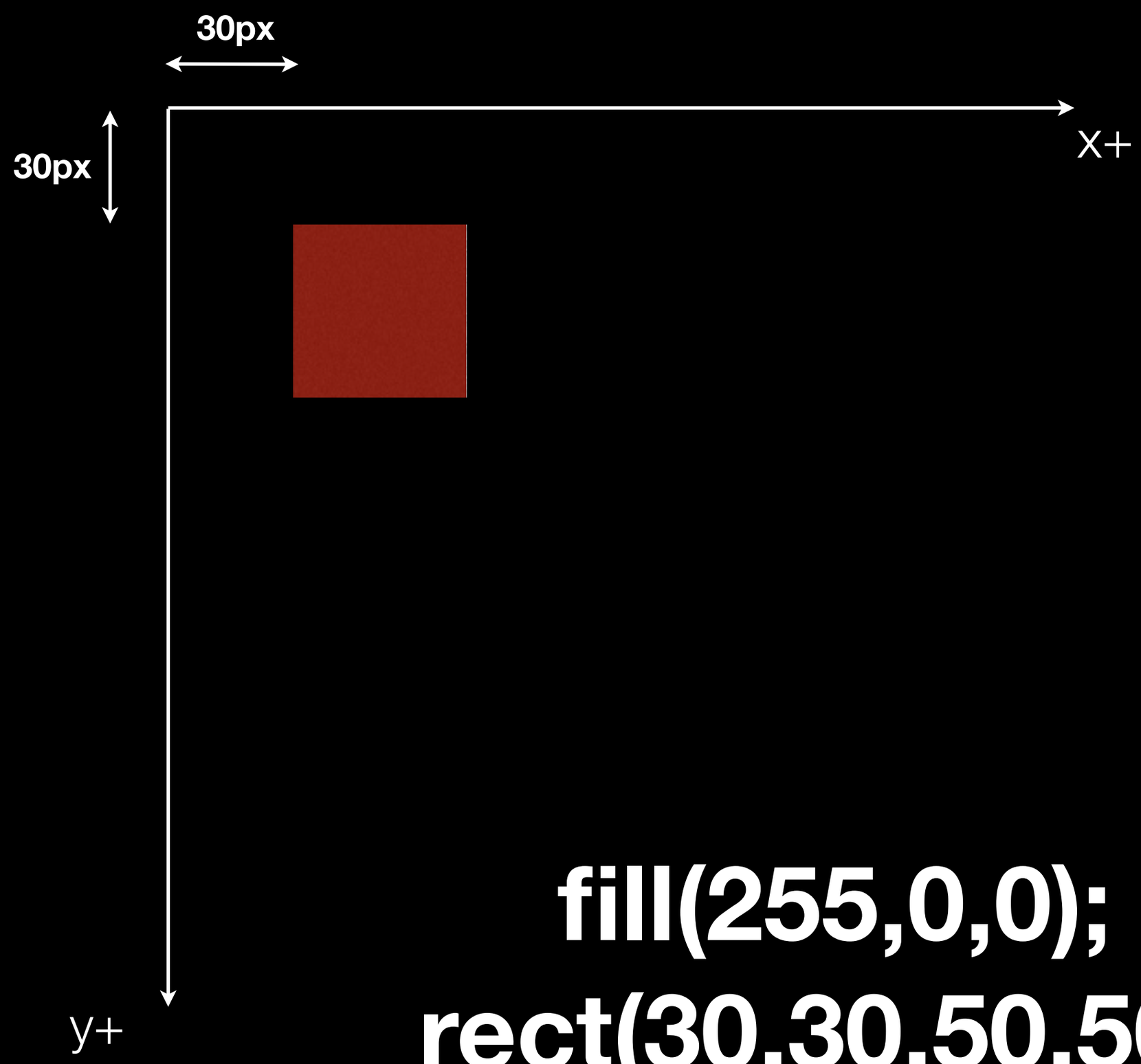
Transformation

Transformation

```
translate(x, y);
rotate( angle );
scale( x, y );
```

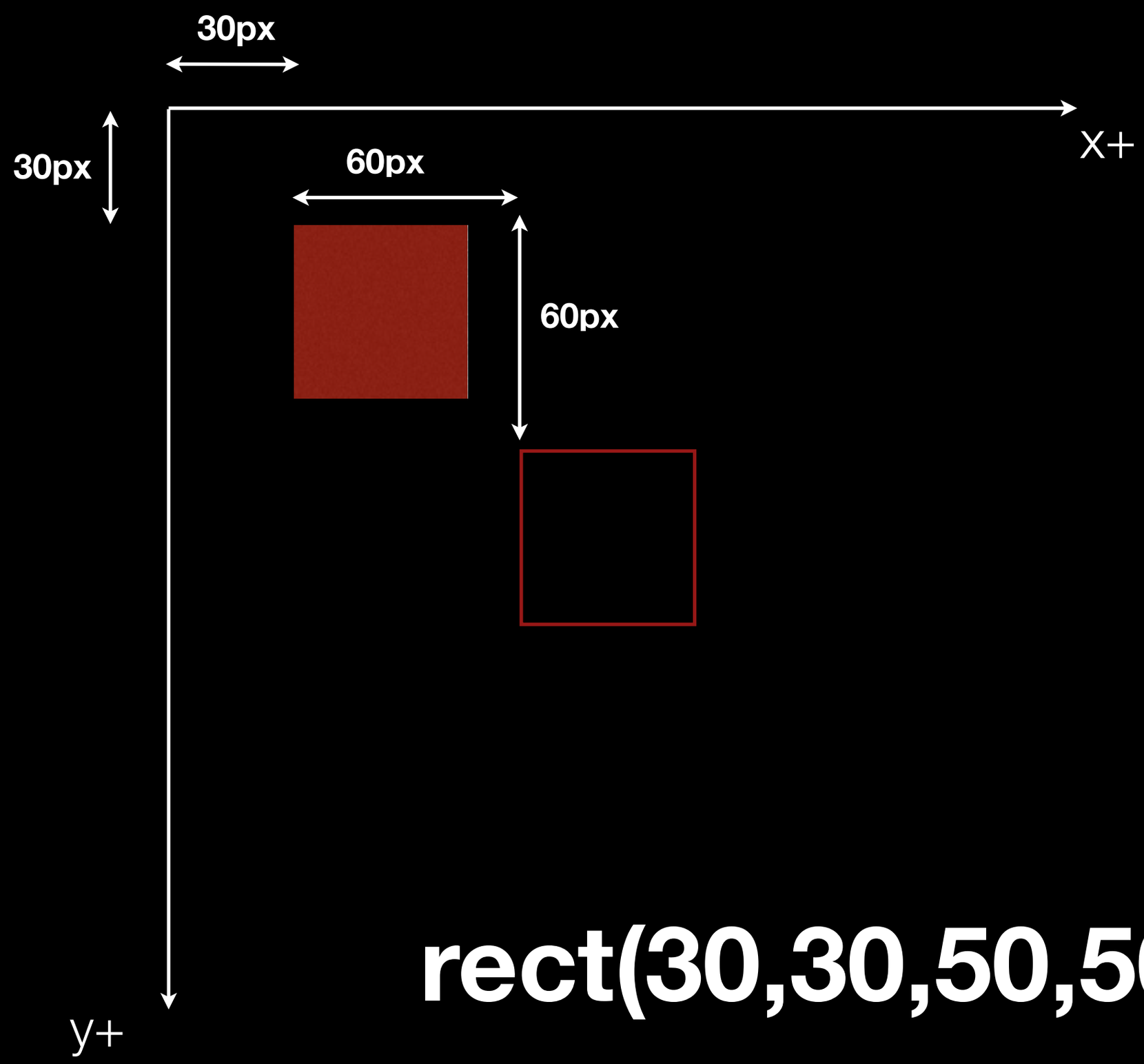
```
translate(x, y);
```

Translate



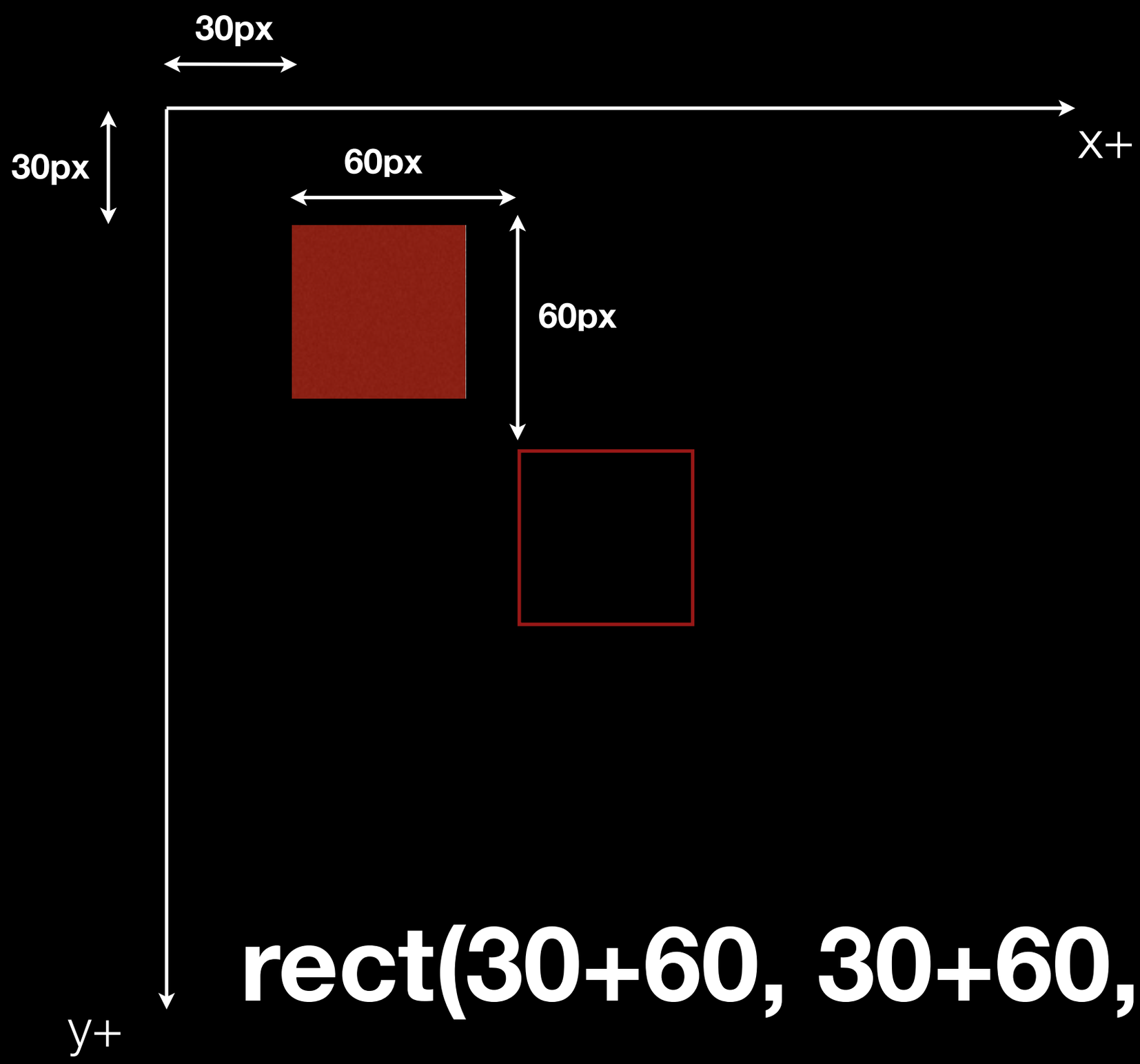

```
fill(255,0,0);
rect(30,30,50,50);
```

Translate



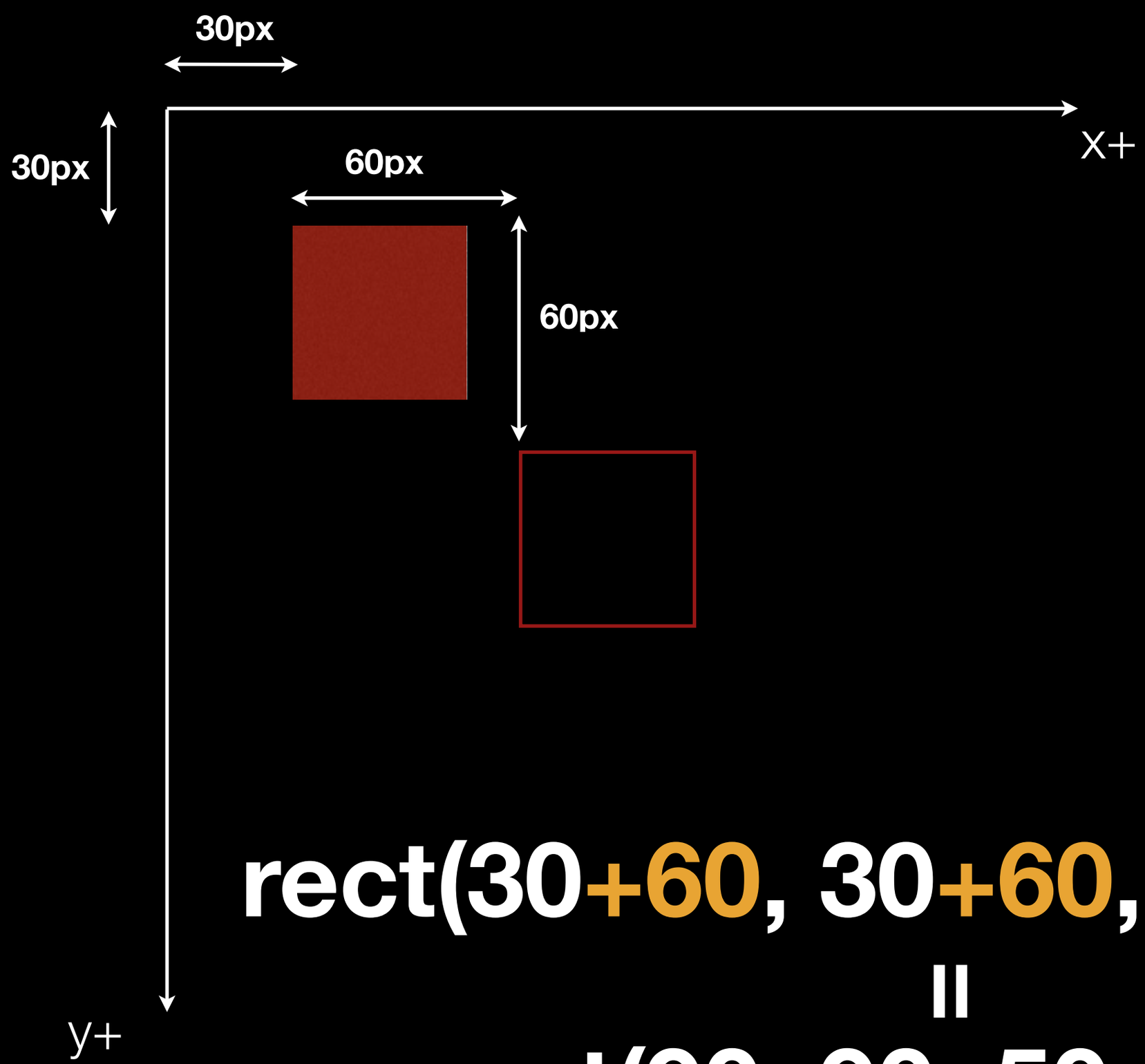
rect(30,30,50,50);

Translate



rect(30+60, 30+60, 50, 50);

Translate



rect(30+60, 30+60, 50, 50);

||

rect(90, 90, 50, 50);

Processing follows Order of Operations

Parentheses first
Multiplication and Division (left-to-right)
Addition and Subtraction (left-to-right)

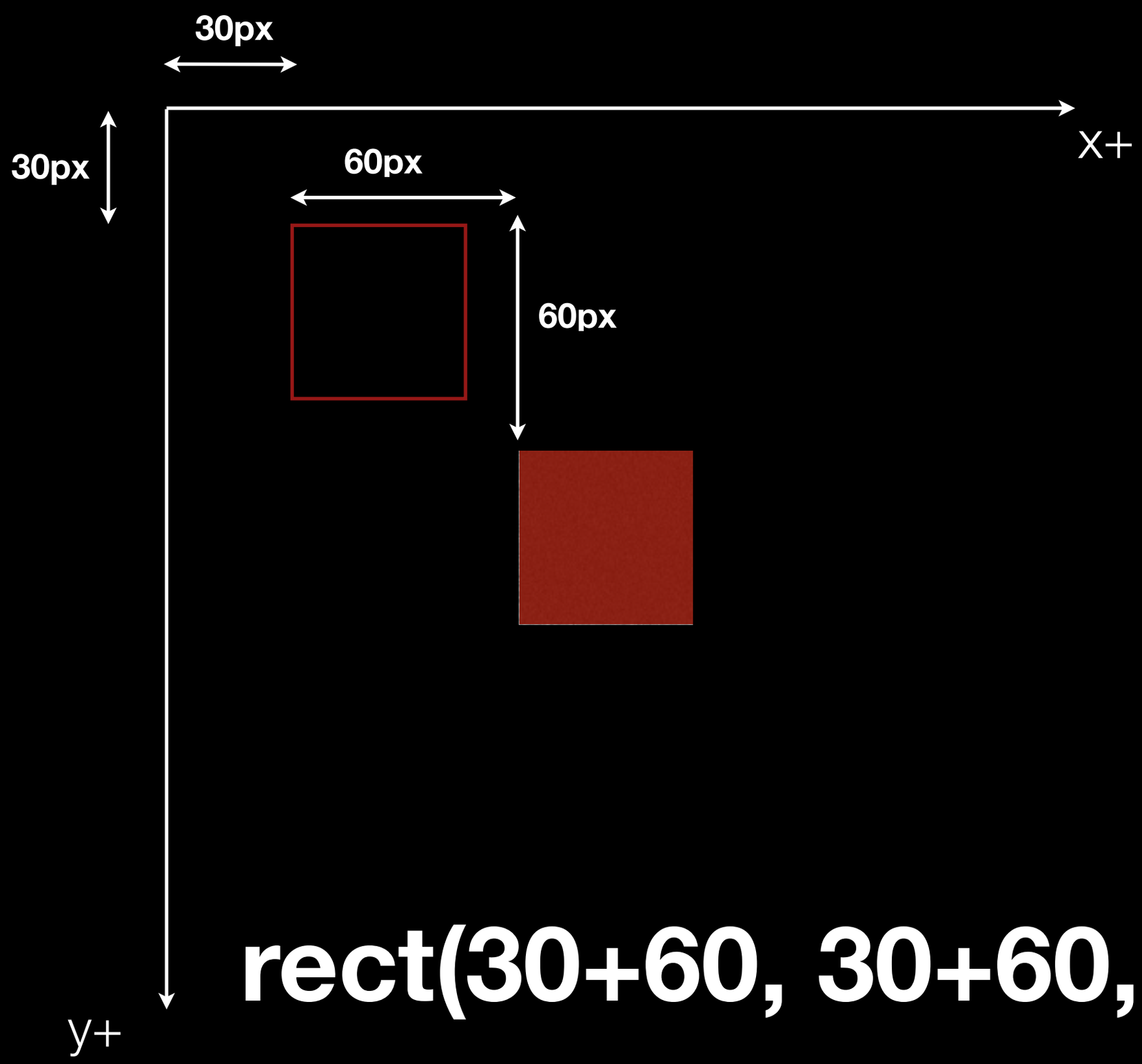
```
rect(30+60, 30-10, 50*2, 50/2);
rect(90, 20, 100, 25);
```

Processing follows Order of Operations

Parentheses first
Multiplication and Division (left-to-right)
Addition and Subtraction (left-to-right)

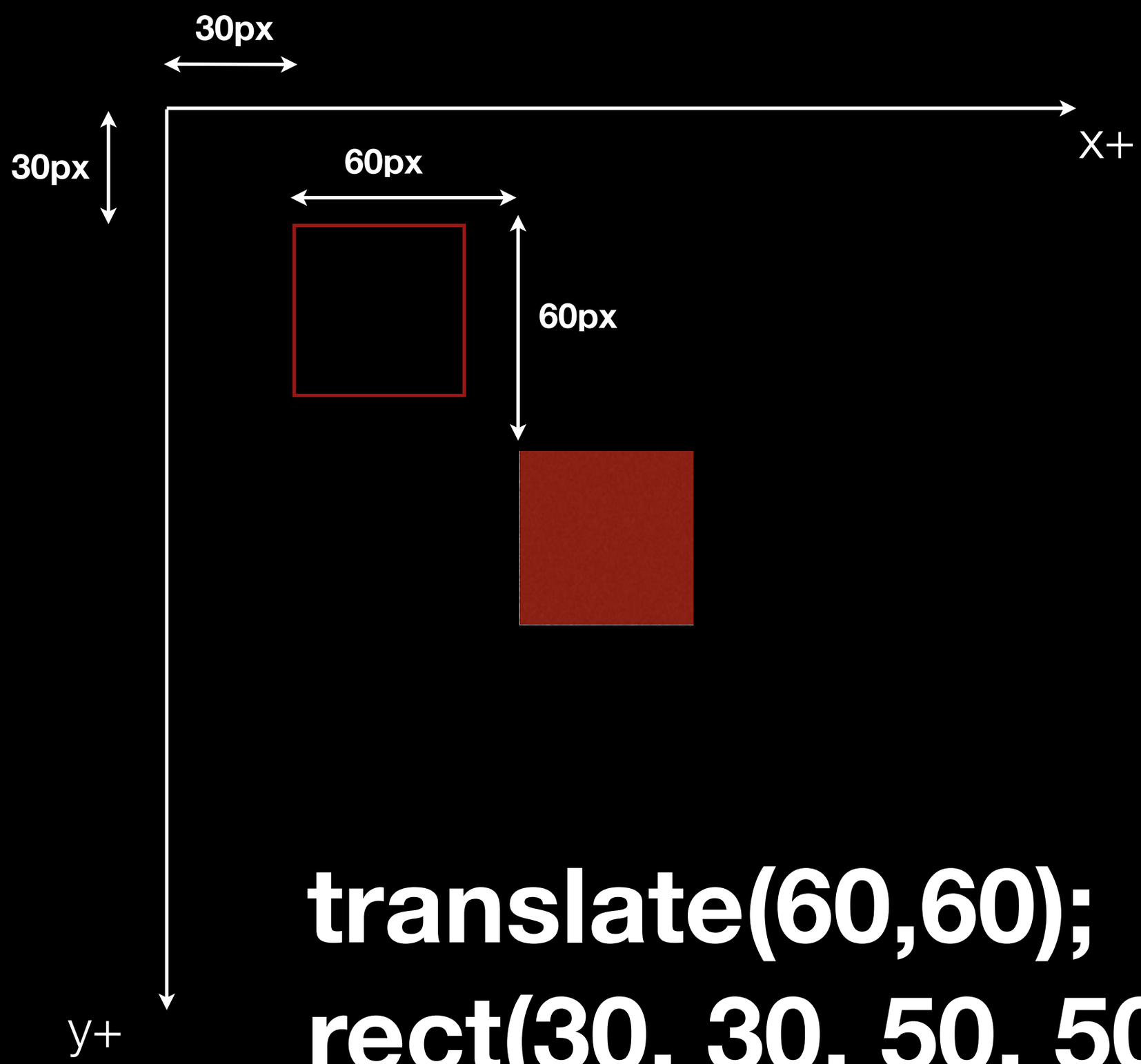
```
rect((30+60)*2, 30, 50, 50);

rect(180, 30, 50, 50);
```

Translate



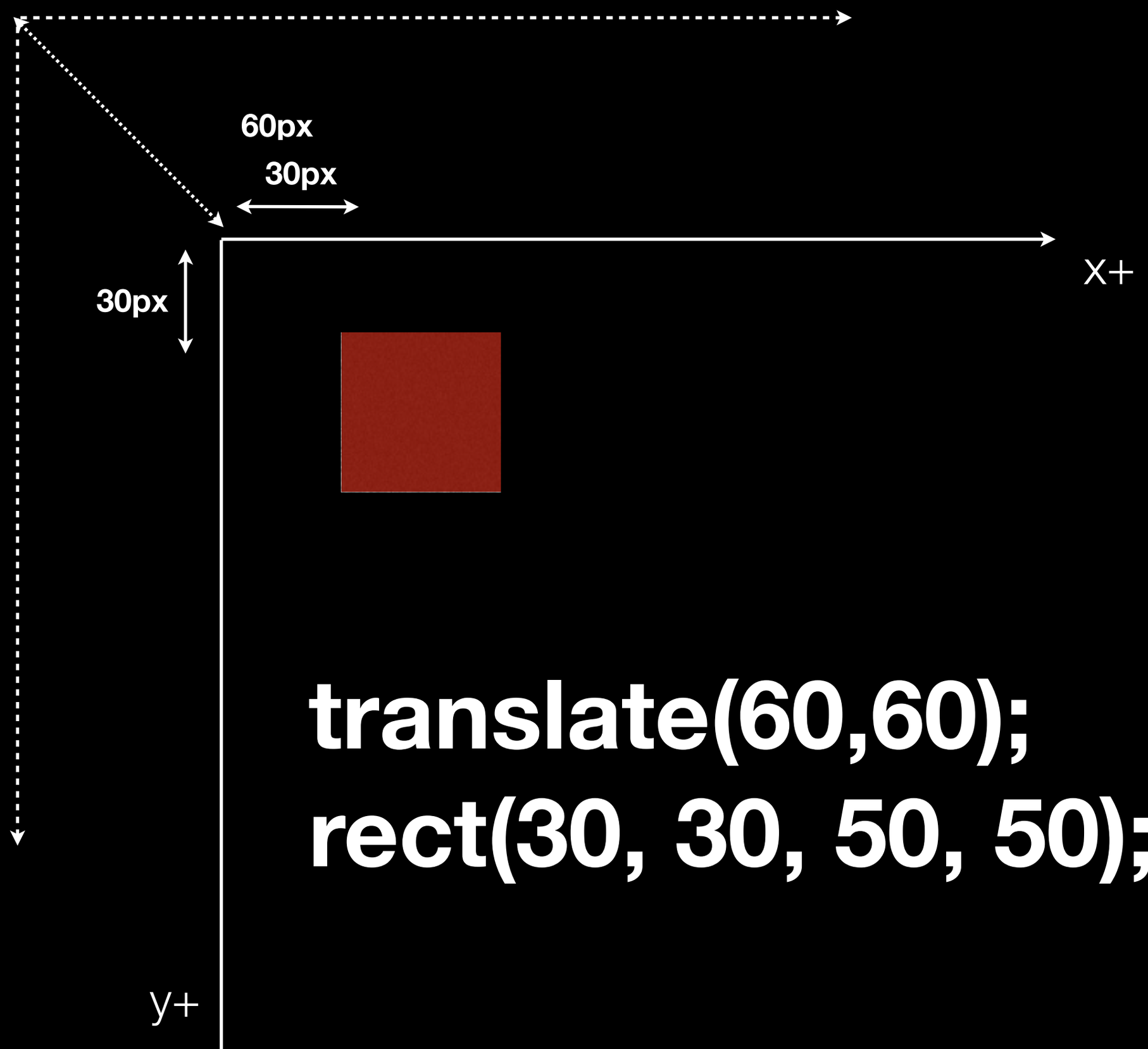
rect(30+60, 30+60, 50, 50);

Translate



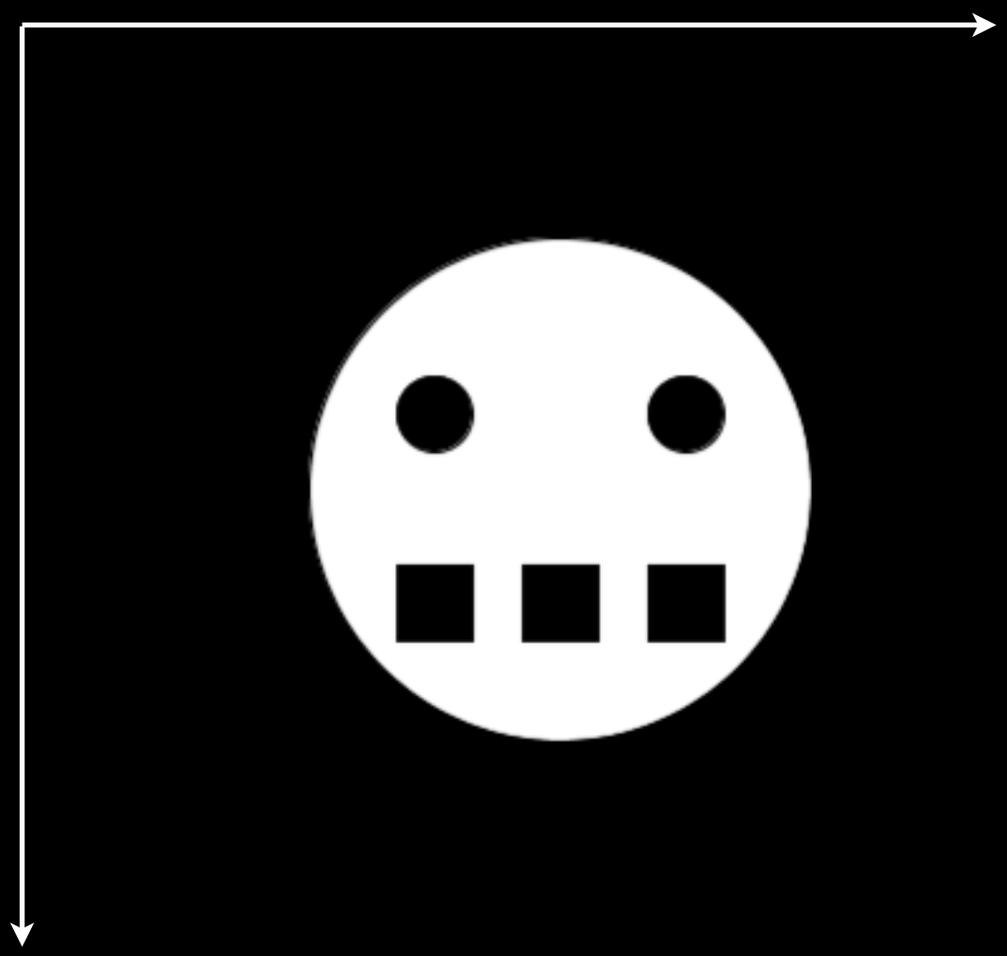

```
translate(60,60);
rect(30, 30, 50, 50);
```

Translate



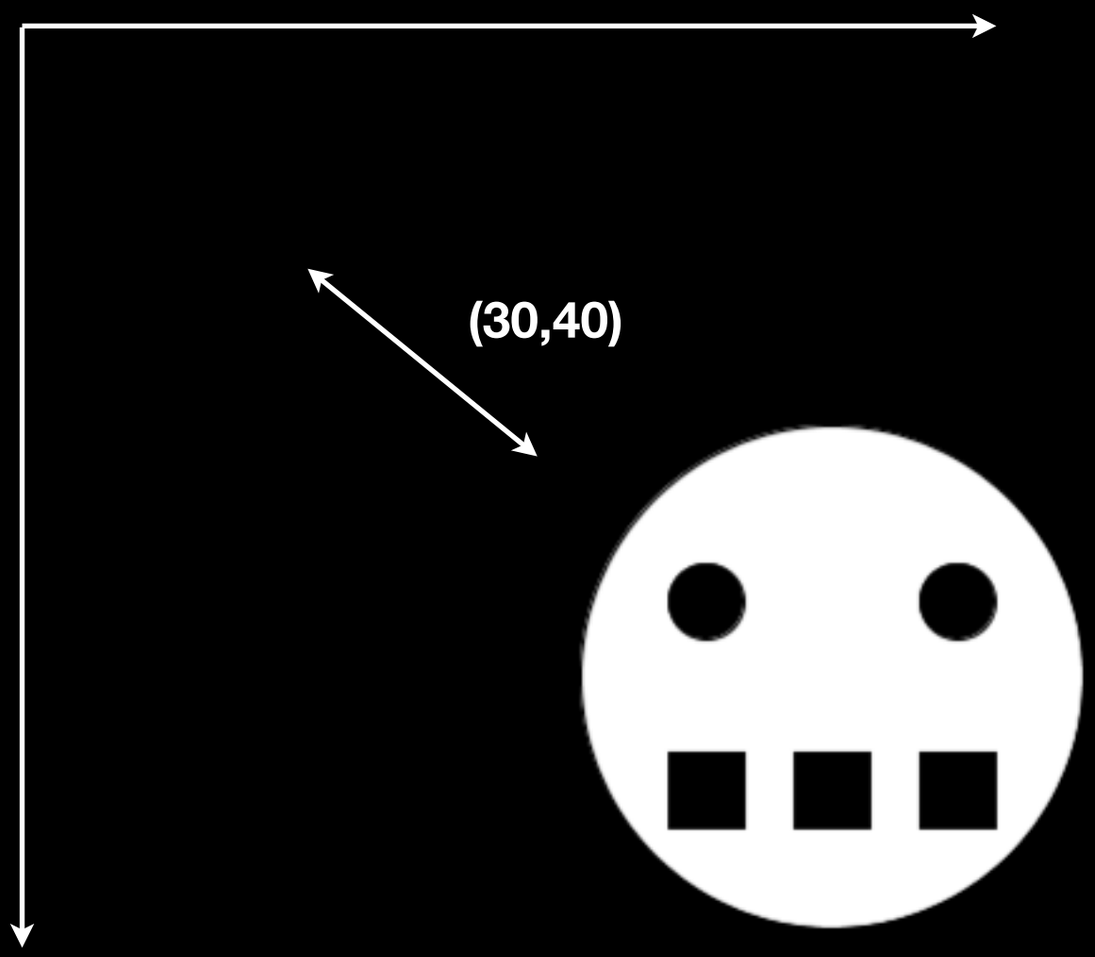

```
translate(60,60);
rect(30, 30, 50, 50);
```

Translate



```
fill(255,255,255);
ellipse(250,250,200,200);
fill(0,0,0);
ellipse(200,220,30,30);
ellipse(300,220,30,30);
rect(185,280,30,30);
rect(235,280,30,30);
rect(285,280,30,30);
```

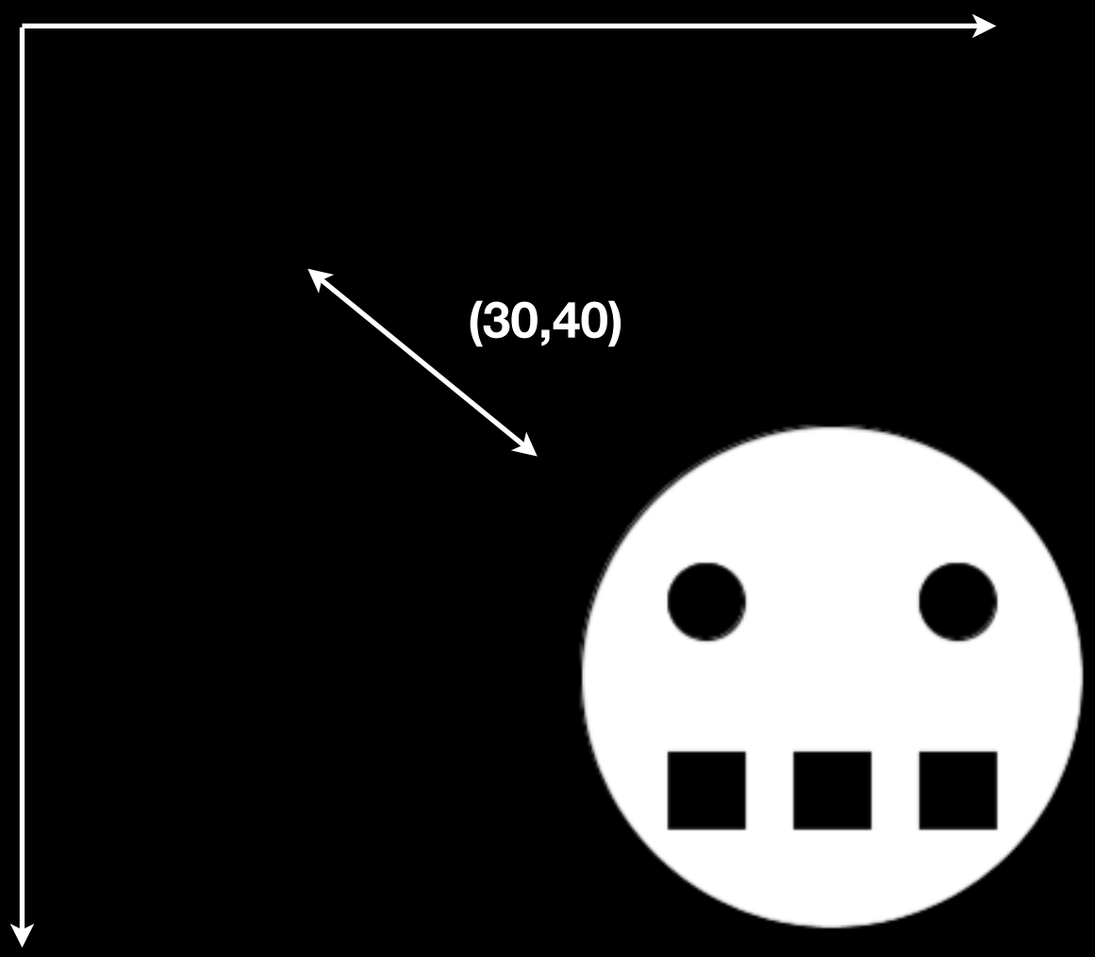
Translate




```
fill(255,255,255);
ellipse(250,250,200,200);
fill(0,0,0);
ellipse(200,220,30,30);
ellipse(300,220,30,30);
rect(185,280,30,30);
rect(235,280,30,30);
rect(285,280,30,30);
```

move 30px right, 40px up

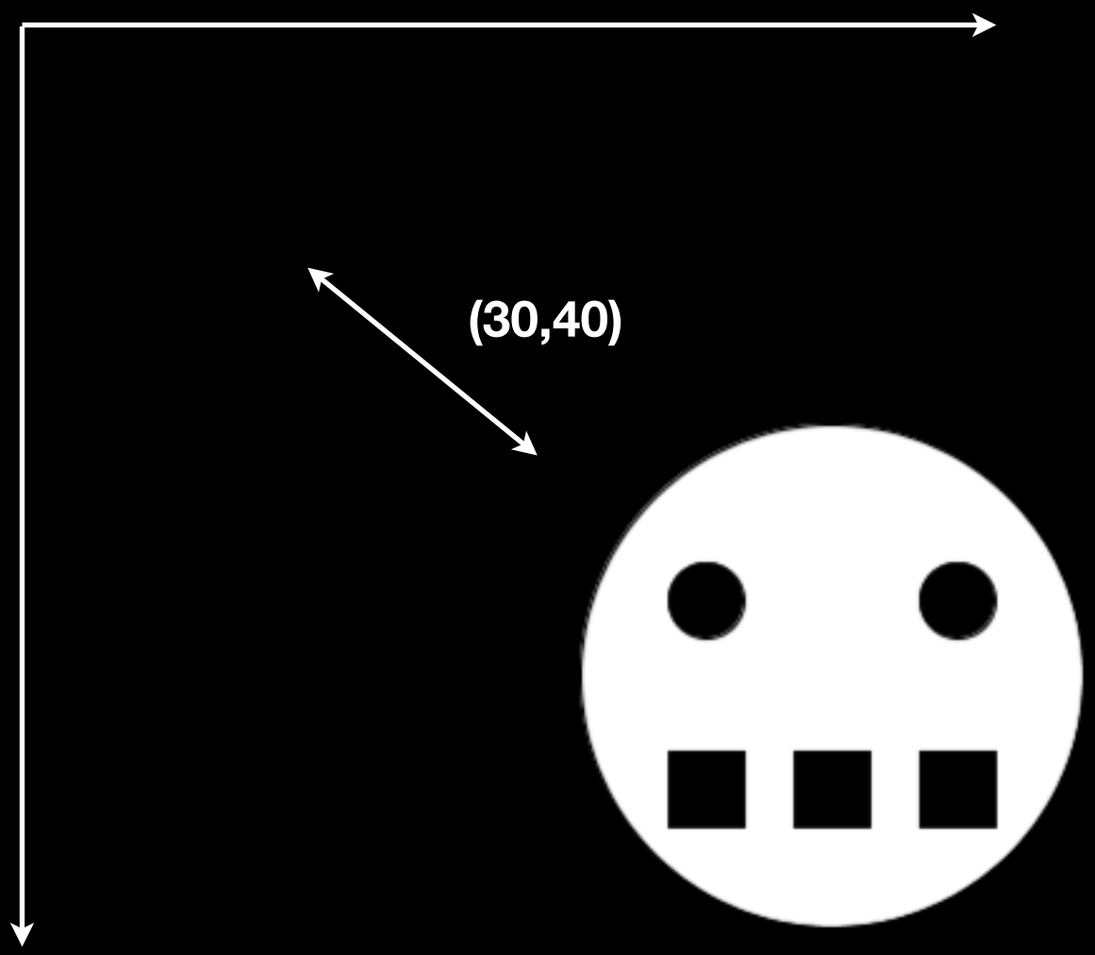
Translate




```
fill(255,255,255);
ellipse(250+30,250+40,200,200);
fill(0,0,0);
ellipse(200+30,220+40,30,30);
ellipse(300+30,220+40,30,30);
rect(185+30,280+40,30,30);
rect(235+30,280+40,30,30);
rect(285+30,280+40,30,30);
```

move 30px right, 40px up

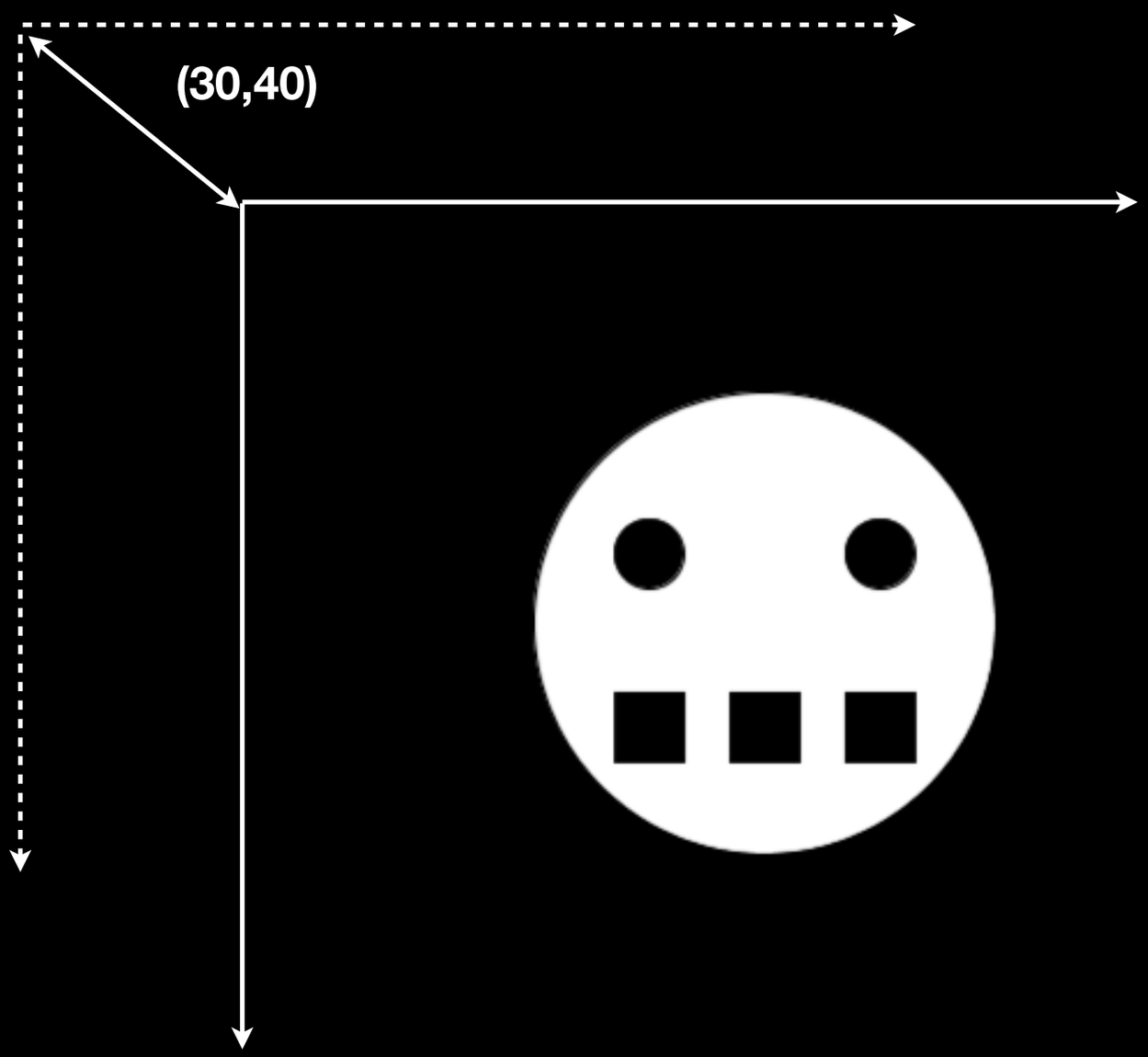
Translate




```
translate(30,40);
fill(255,255,255);
ellipse(250,250,200,200);
fill(0,0,0);
ellipse(200,220,30,30);
ellipse(300,220,30,30);
rect(185,280,30,30);
rect(235,280,30,30);
rect(285,280,30,30);
```

move 30px right, 40px up

Translate

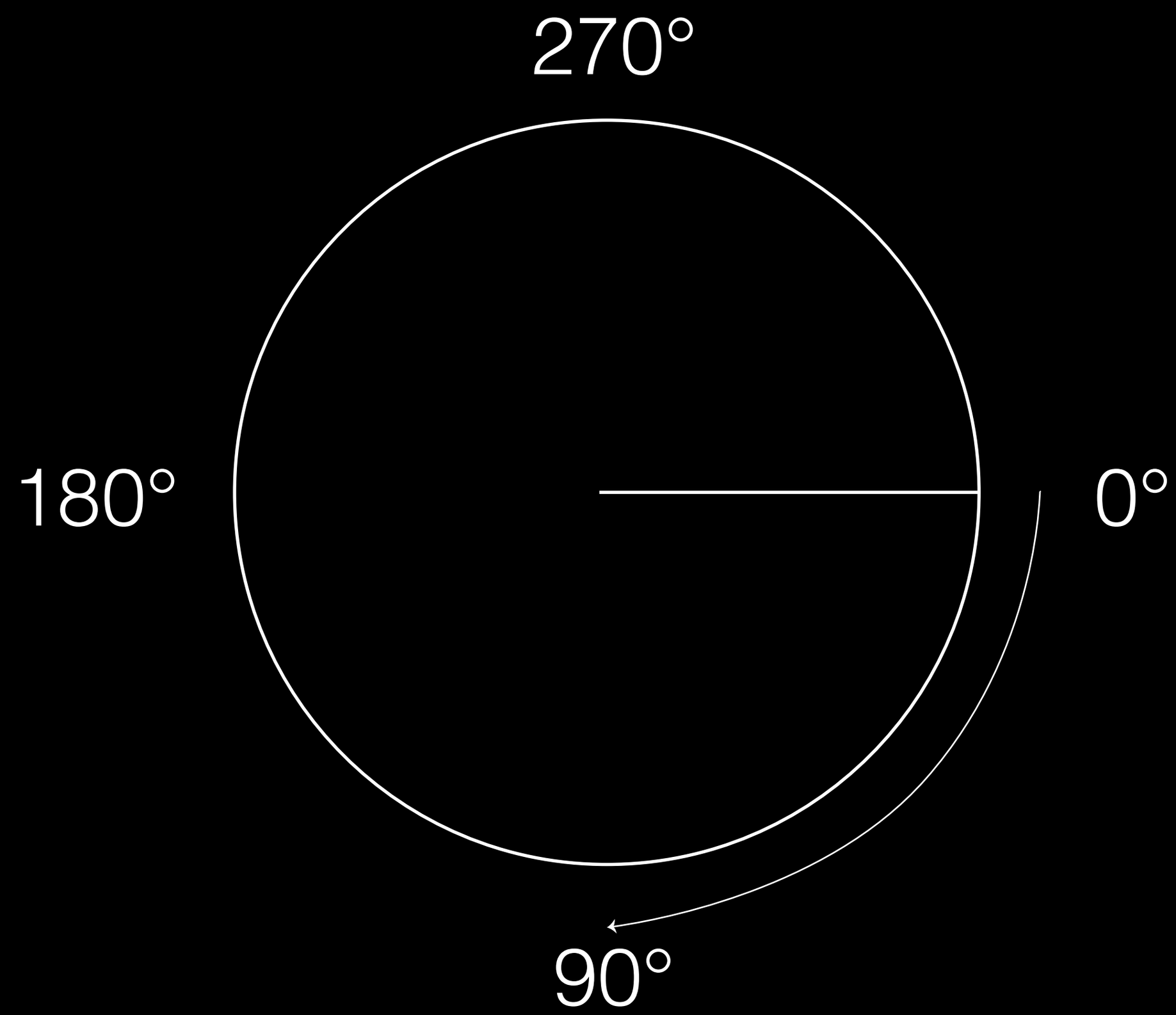



```
translate(30,40);
fill(255,255,255);
ellipse(250,250,200,200);
fill(0,0,0);
ellipse(200,220,30,30);
ellipse(300,220,30,30);
rect(185,280,30,30);
rect(235,280,30,30);
rect(285,280,30,30);
```

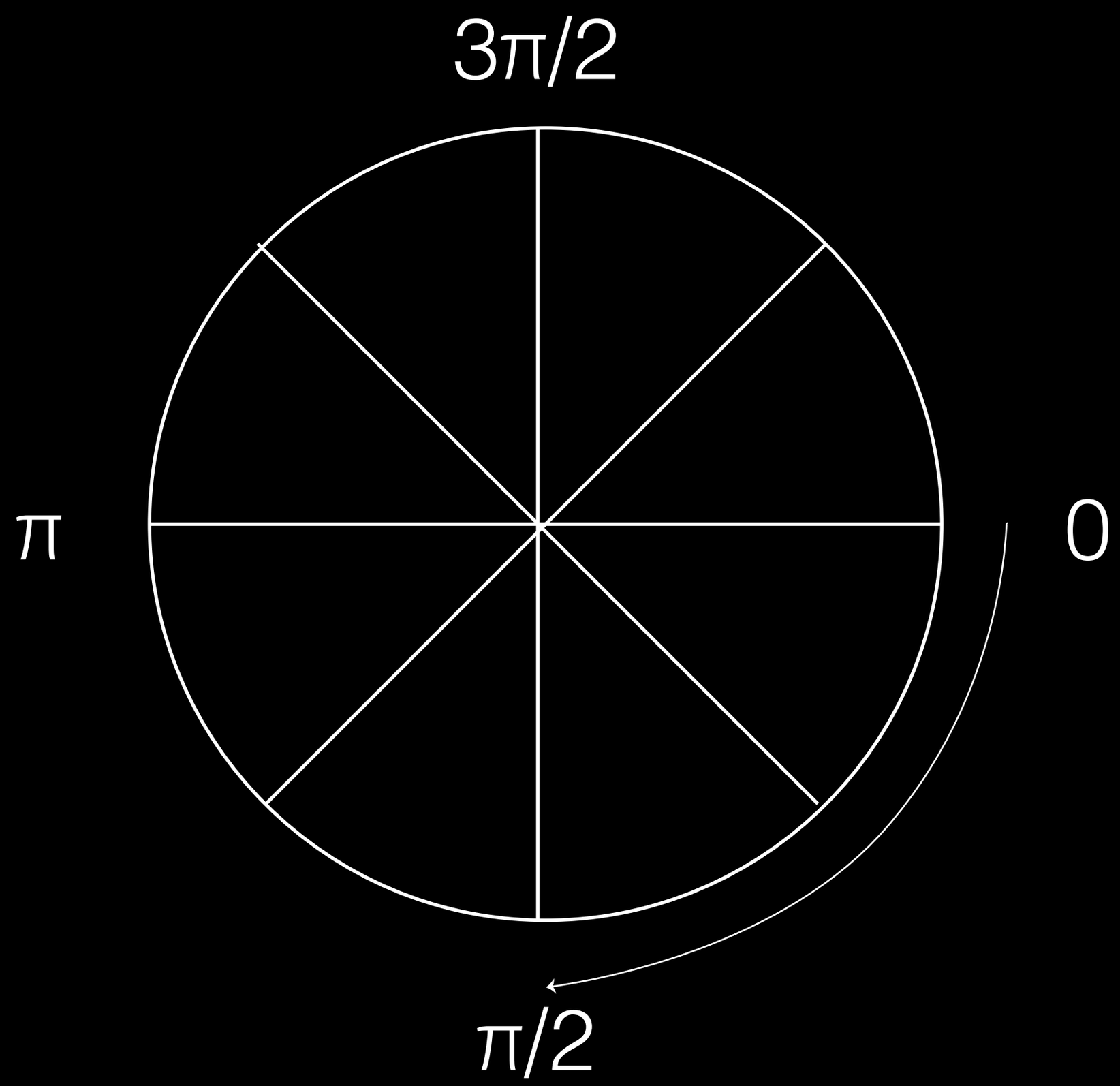
```
rotate(angle);
```

all the functions that have to do with rotation measure angles in radians rather than degrees

Degrees

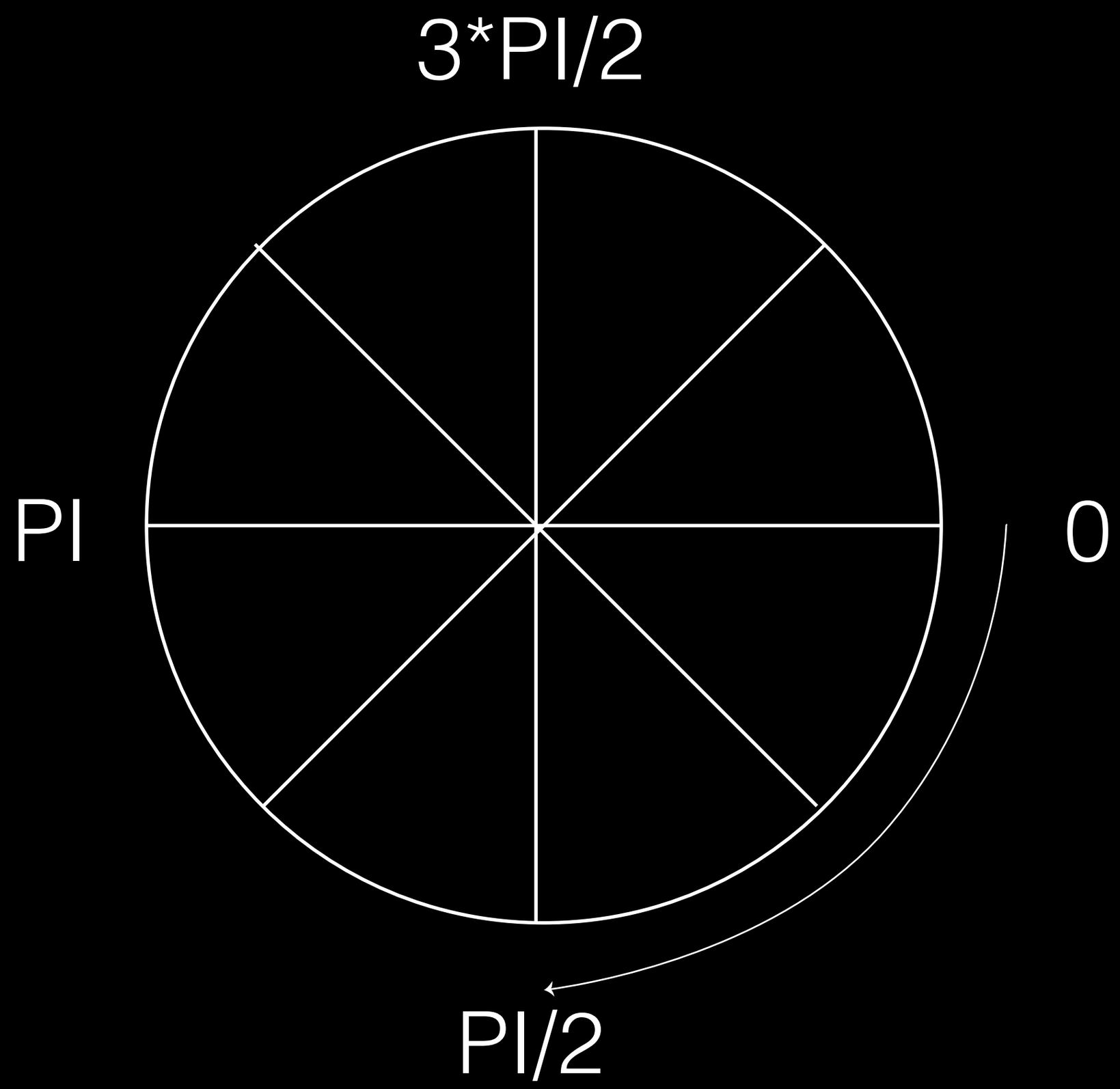


Radians



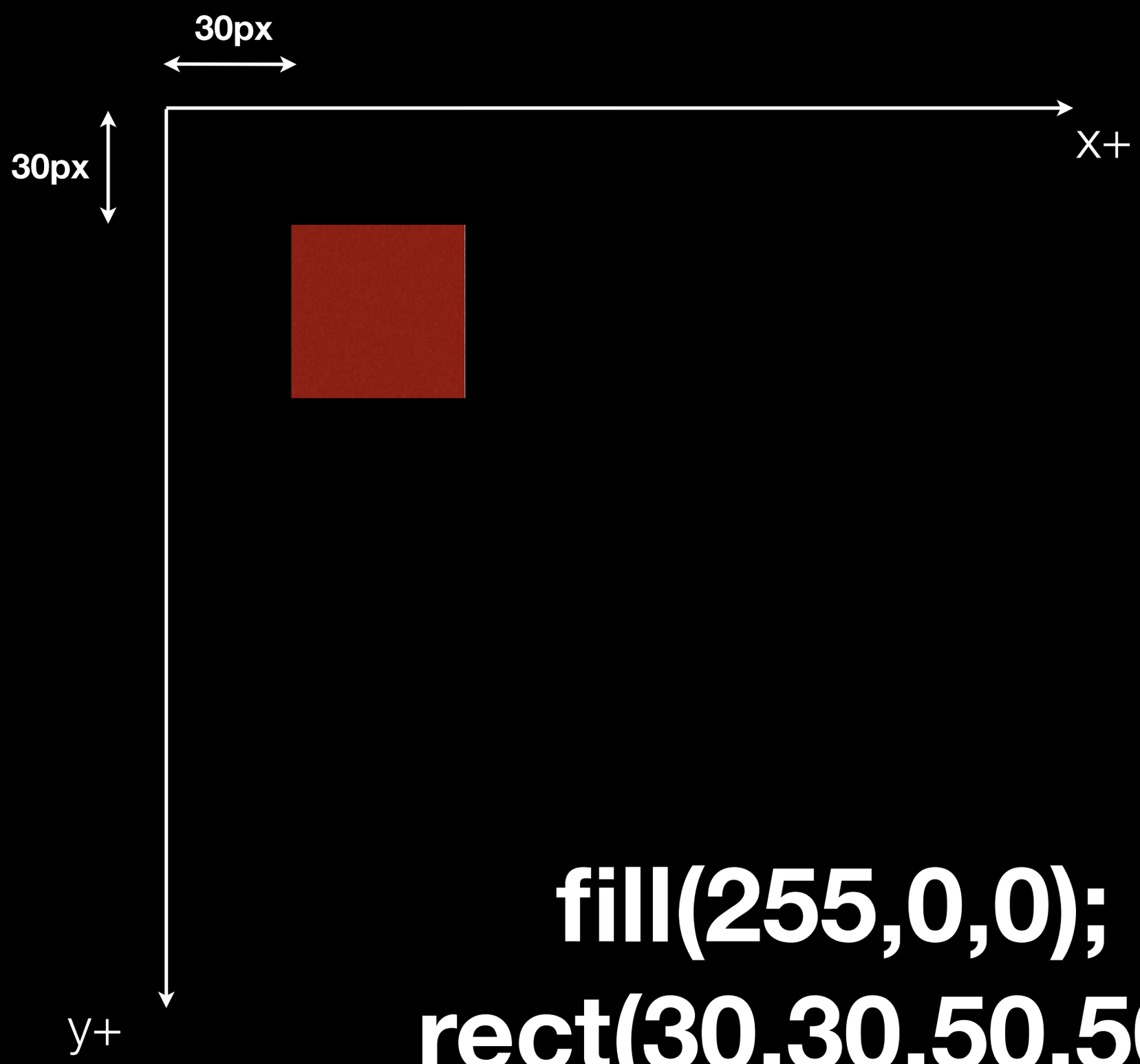
PI = π = 3.14159265358979328 46...

Radians



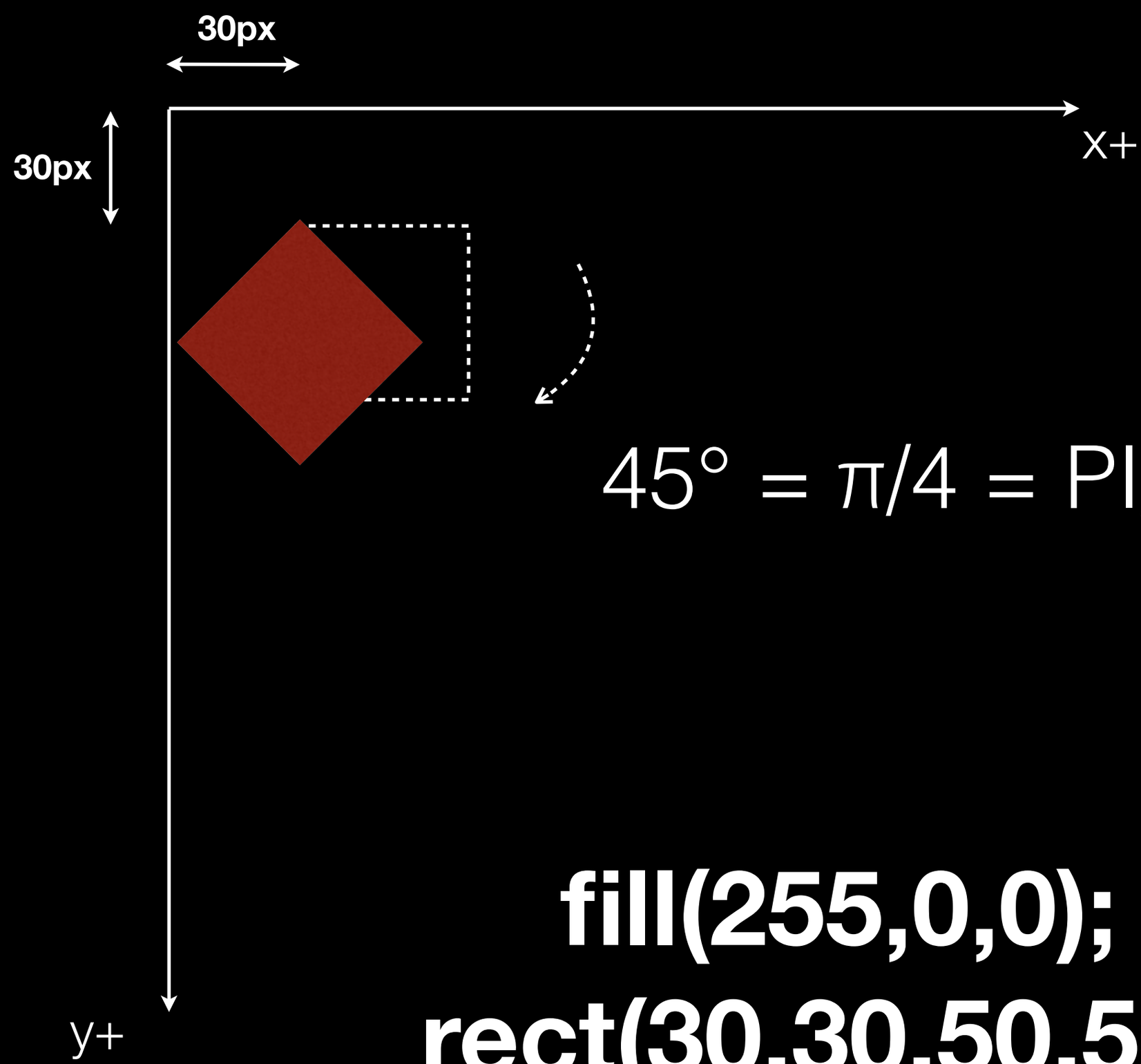
PI = π = 3.14159265358979328 46...

Rotate




```
fill(255,0,0);
rect(30,30,50,50);
```

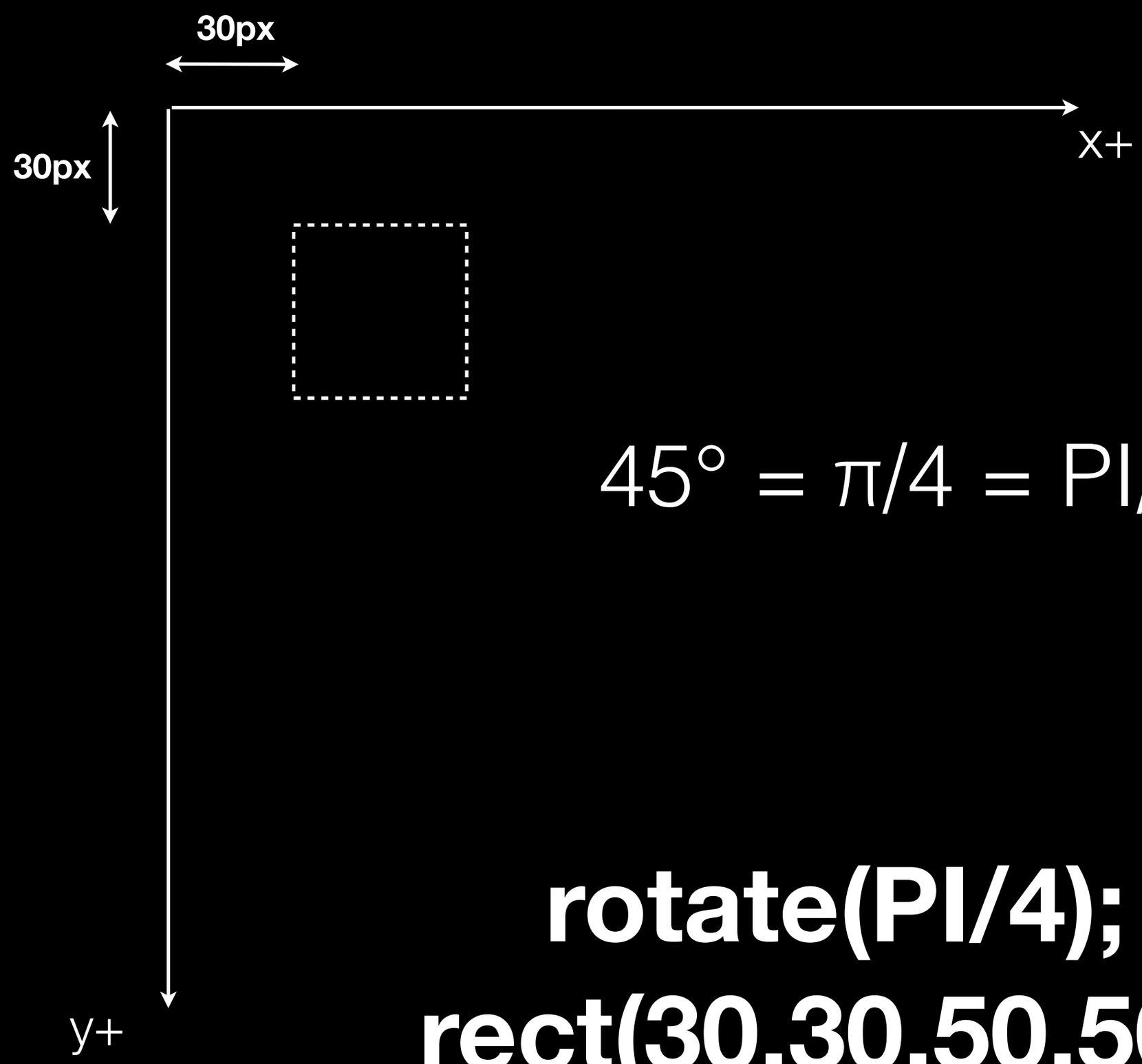
Rotate



45° = π/4 = PI/4

```
fill(255,0,0);
rect(30,30,50,50);
```

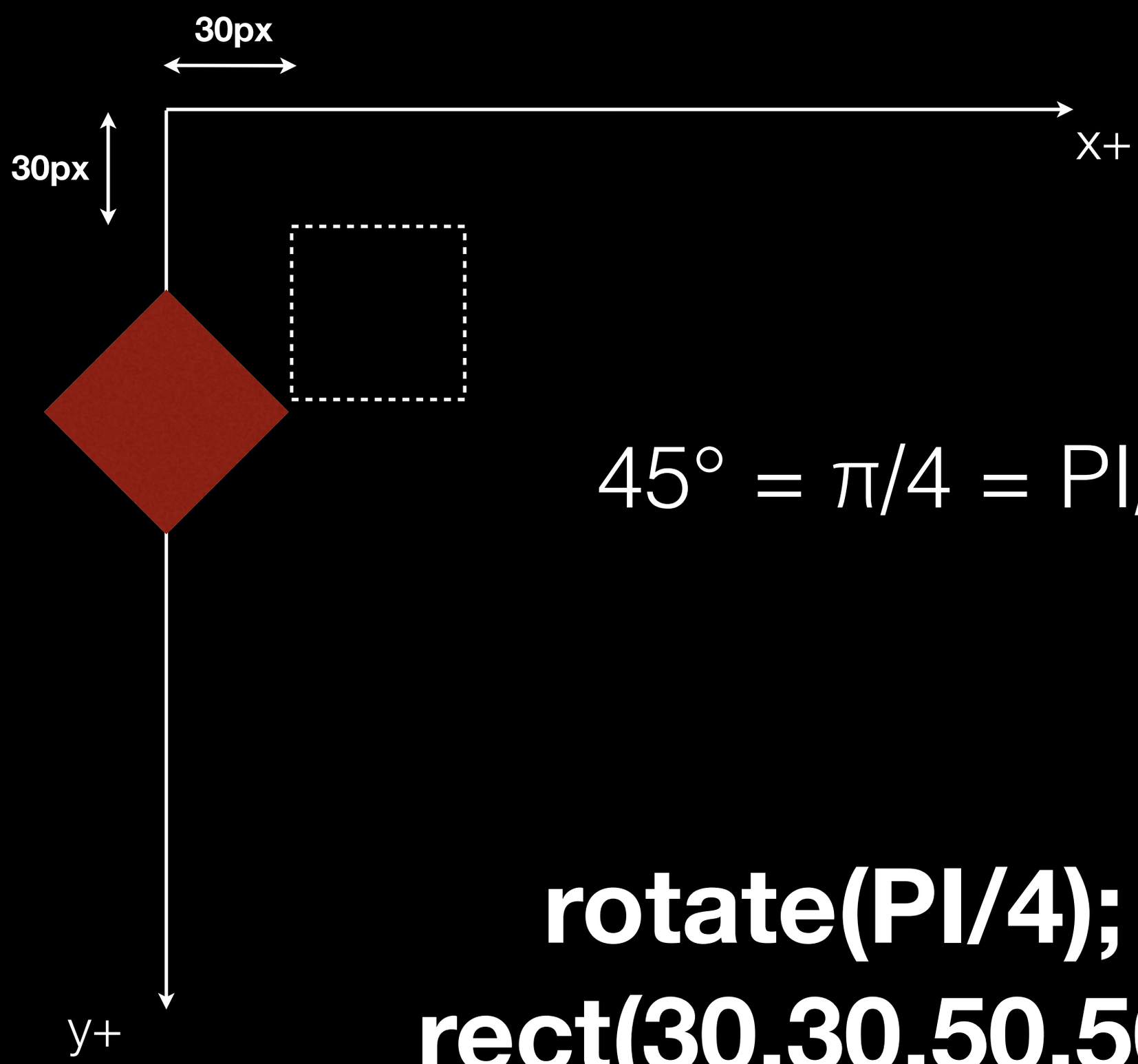
Rotate



45° = π/4 = PI/4

```
rotate(PI/4);
rect(30,30,50,50);
```

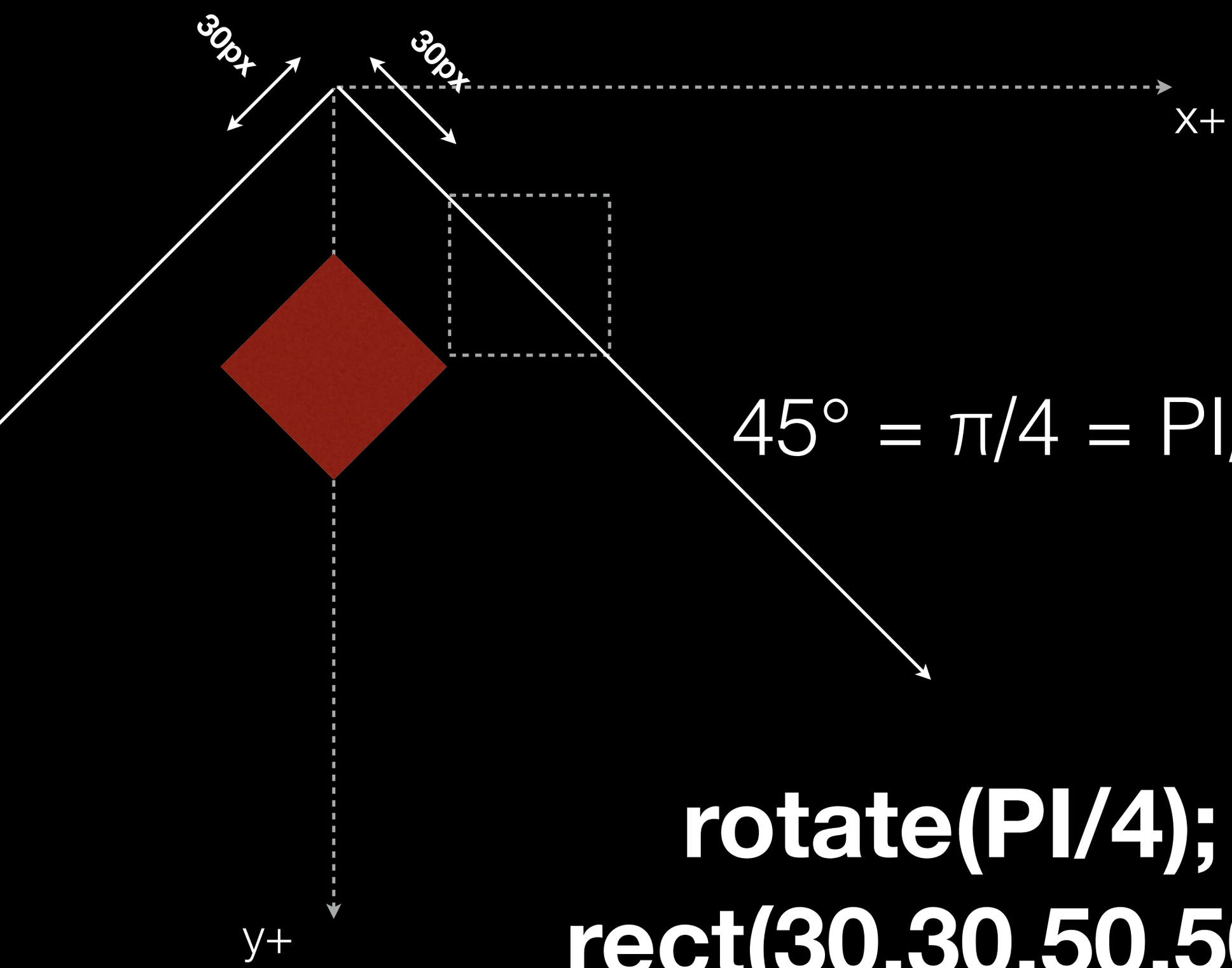
Rotate



$45° = \pi/4 = PI/4$

```
rotate(PI/4);
rect(30,30,50,50);
```

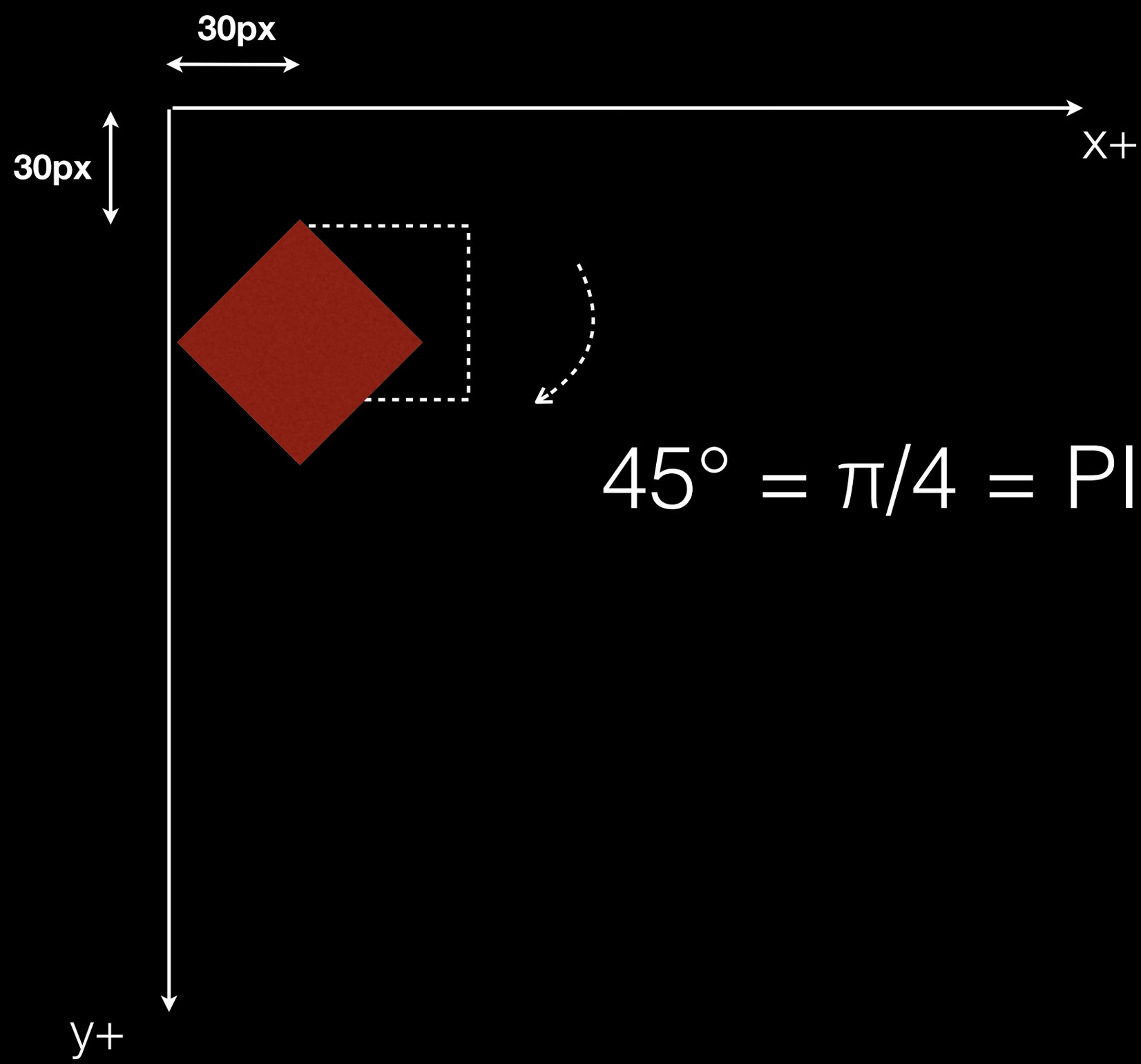
Rotate



45° = π/4 = PI/4

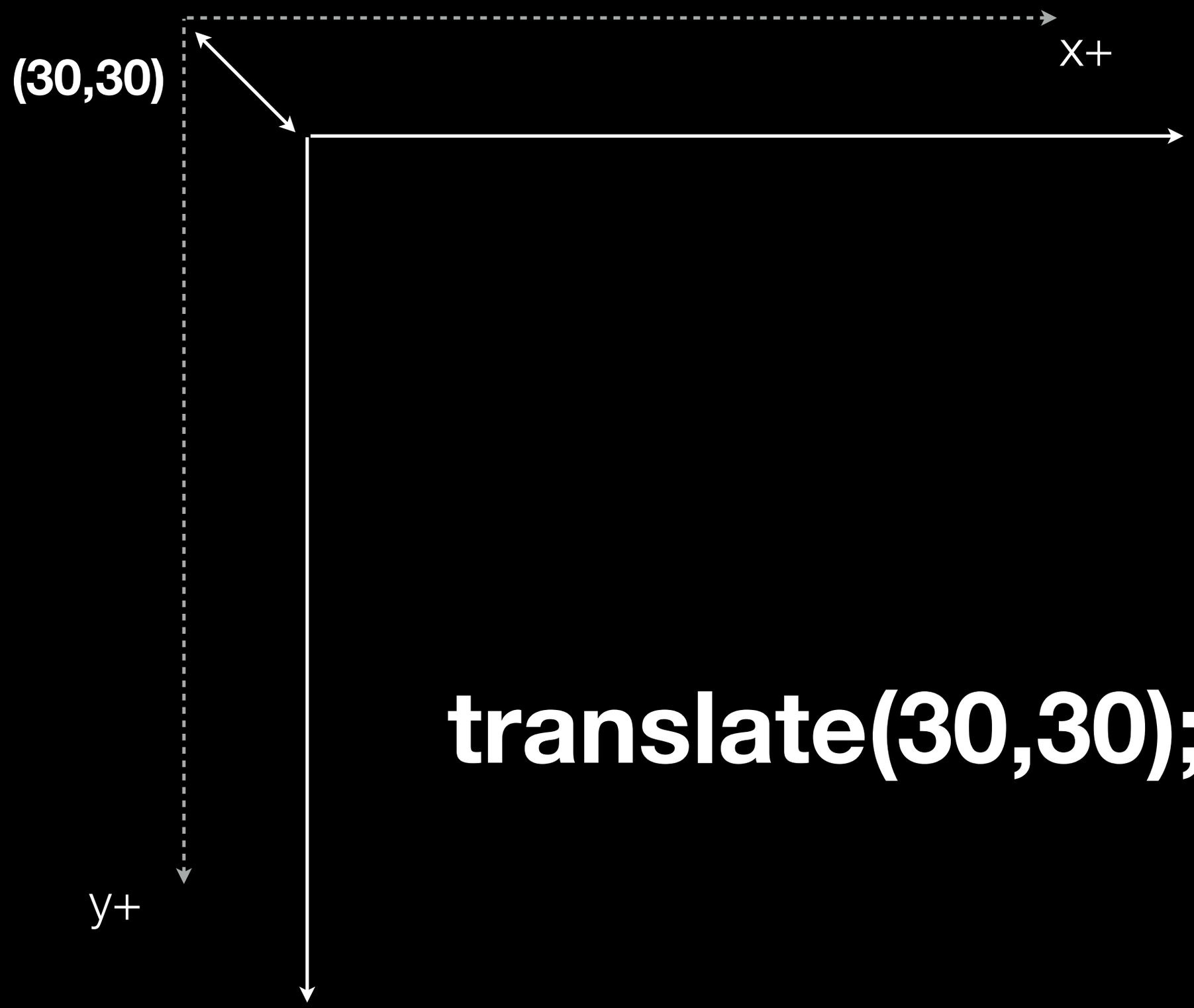
rotate(PI/4);
rect(30,30,50,50);

Rotate



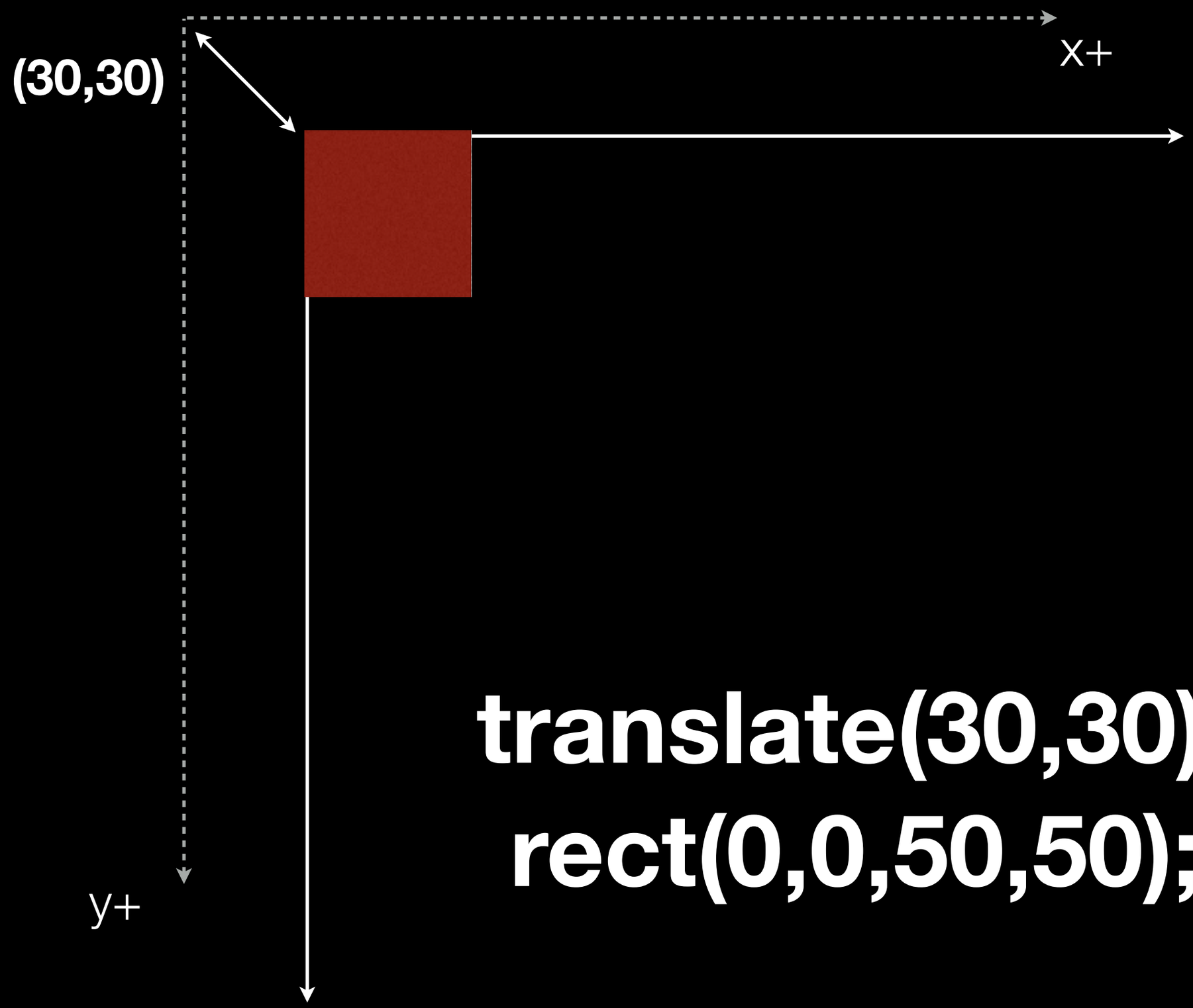
$45° = \pi/4 = PI/4$

Rotate



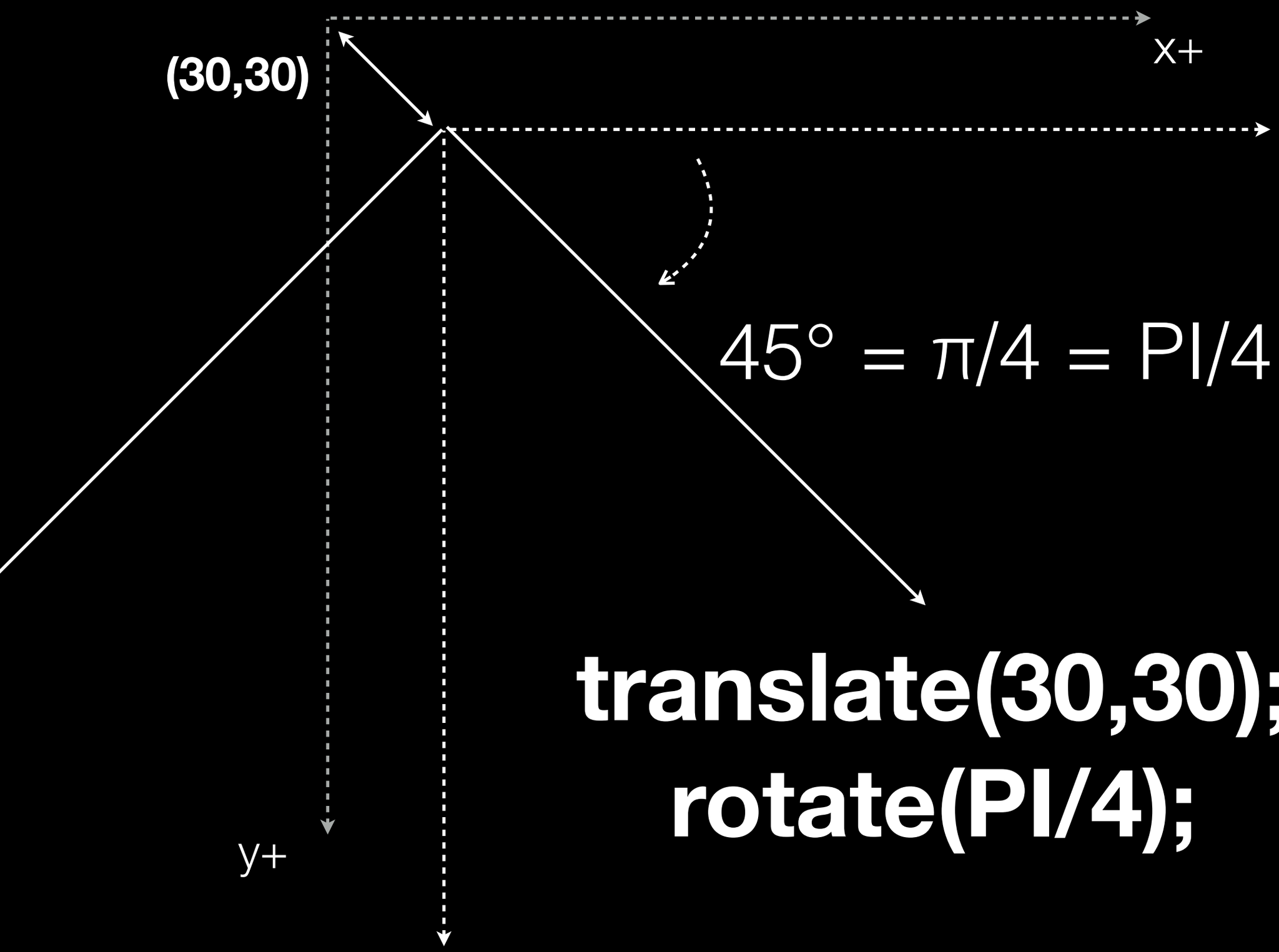
translate(30,30);

Rotate




```
translate(30,30);
rect(0,0,50,50);
```

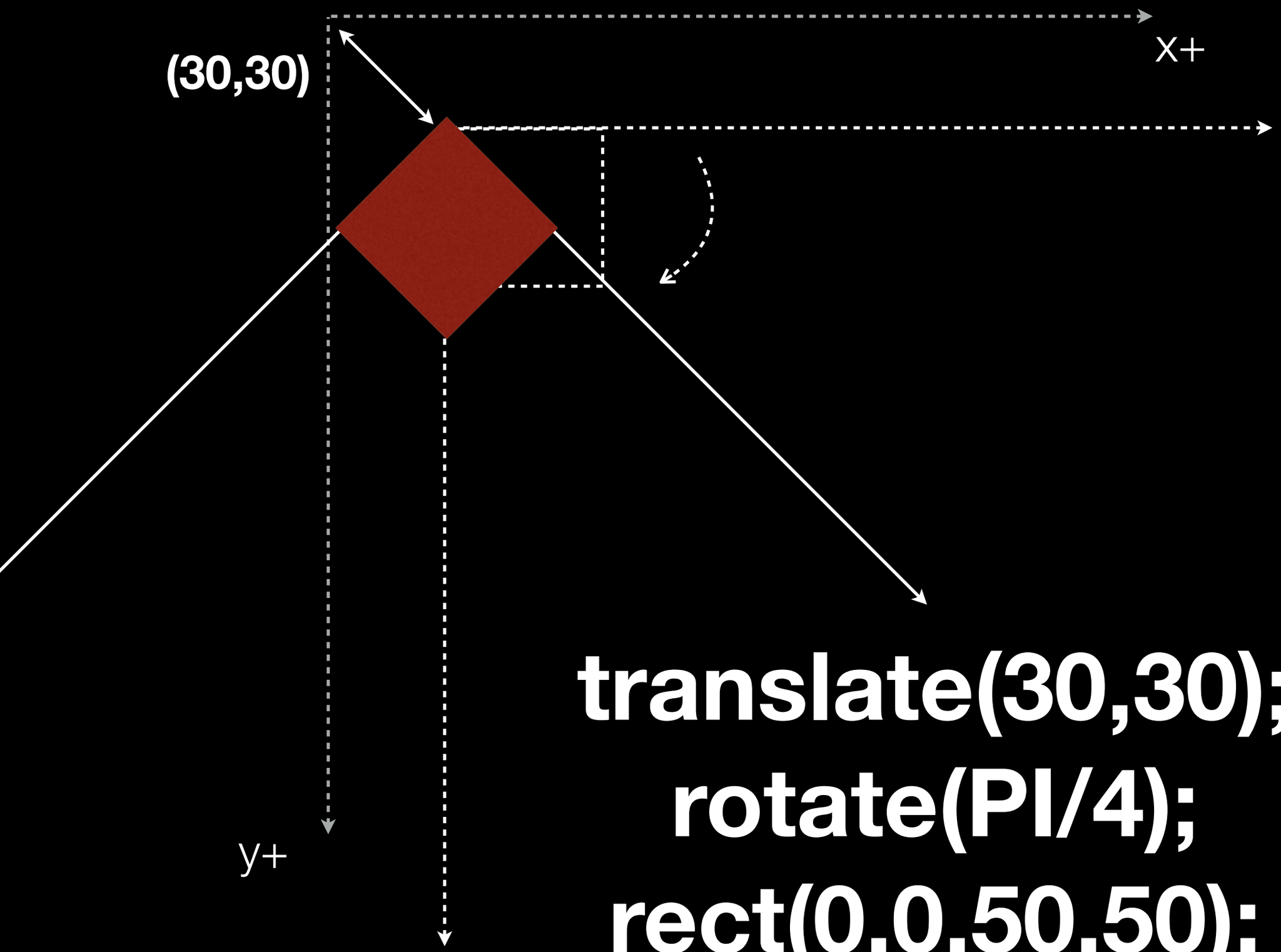
Rotate



45° = π/4 = PI/4

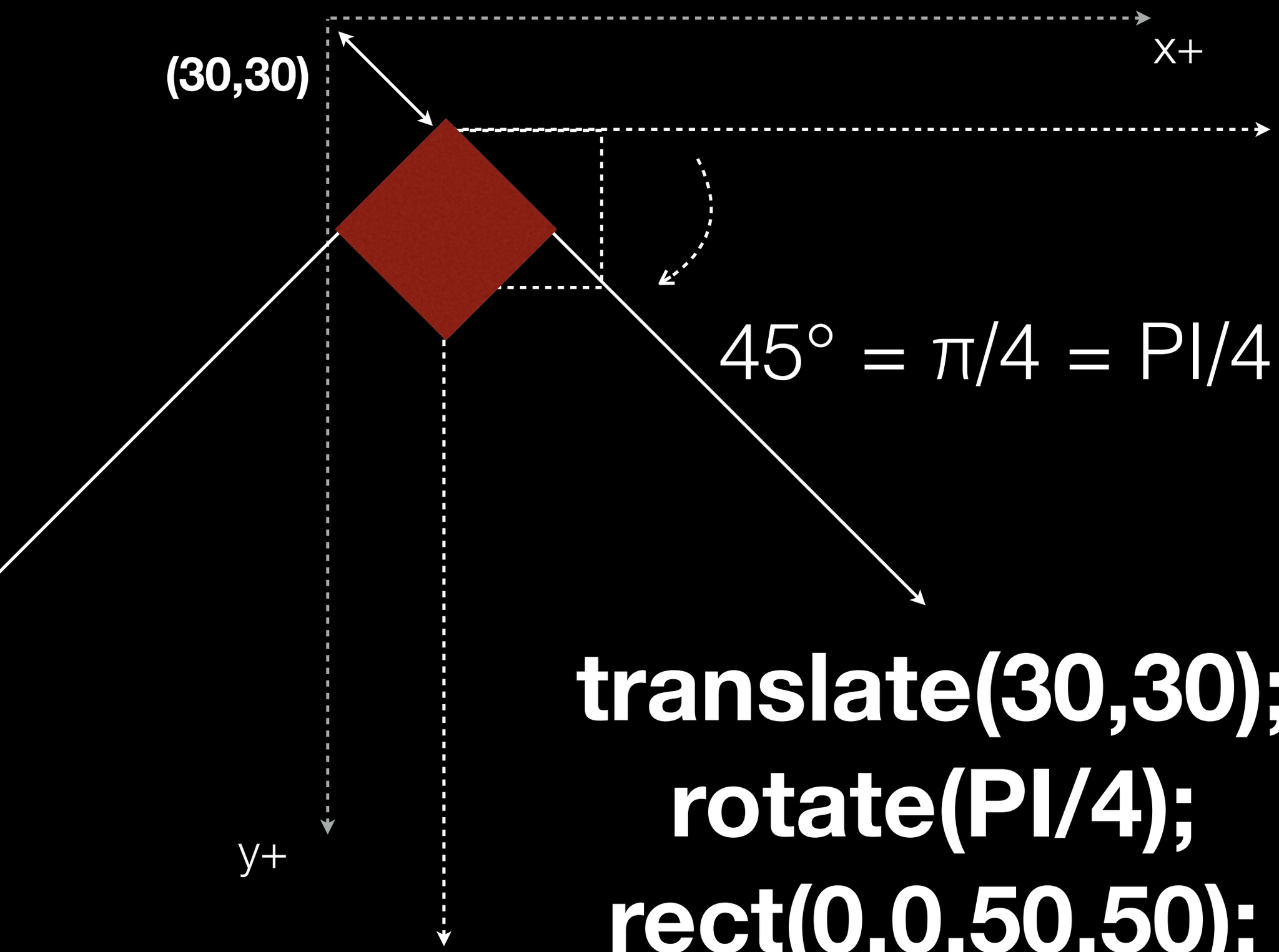
translate(30,30);
rotate(PI/4);

Rotate




```
translate(30,30);
rotate(PI/4);
rect(0,0,50,50);
```

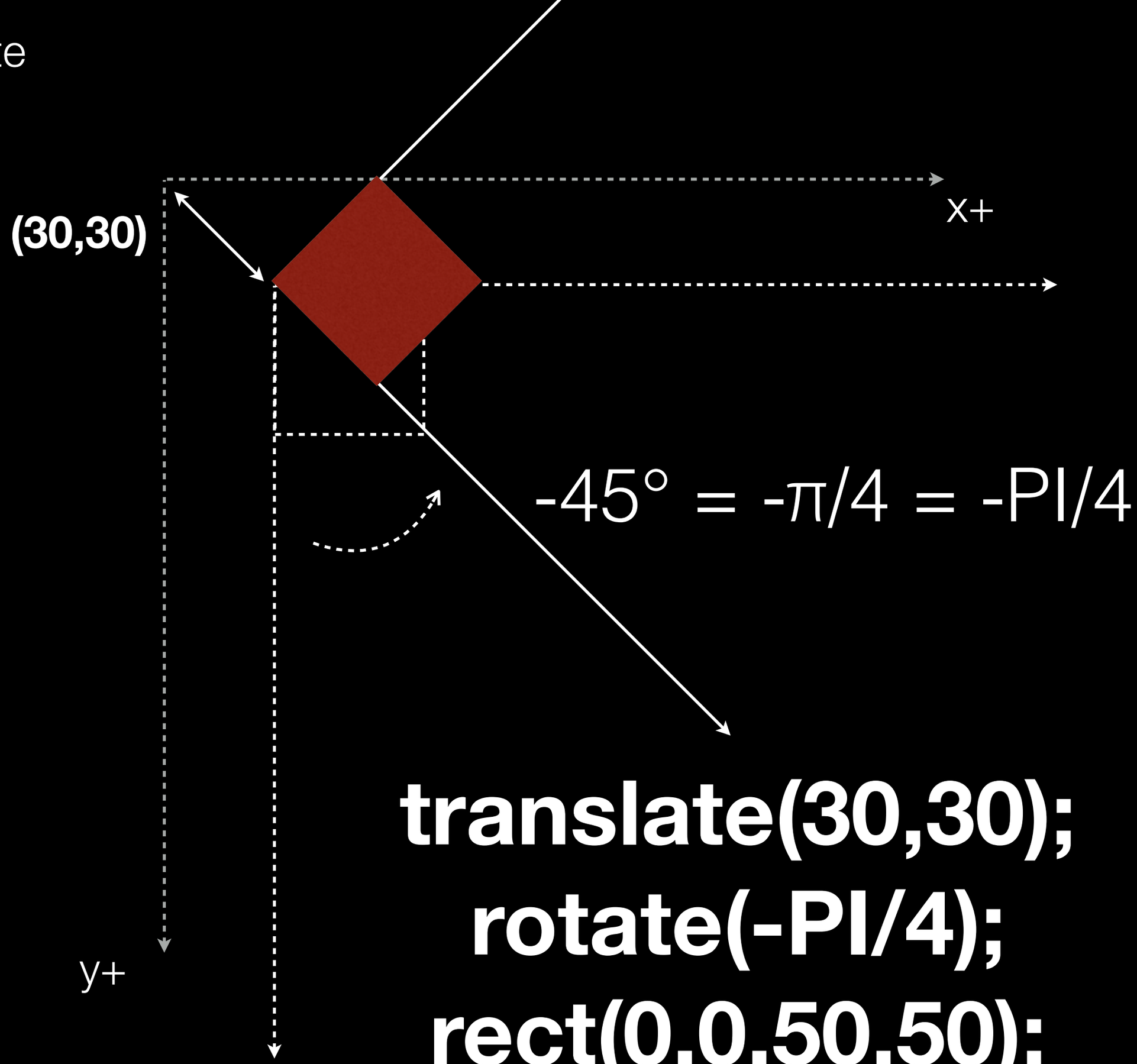
Rotate



45° = π/4 = PI/4

```
translate(30,30);
rotate(PI/4);
rect(0,0,50,50);
```

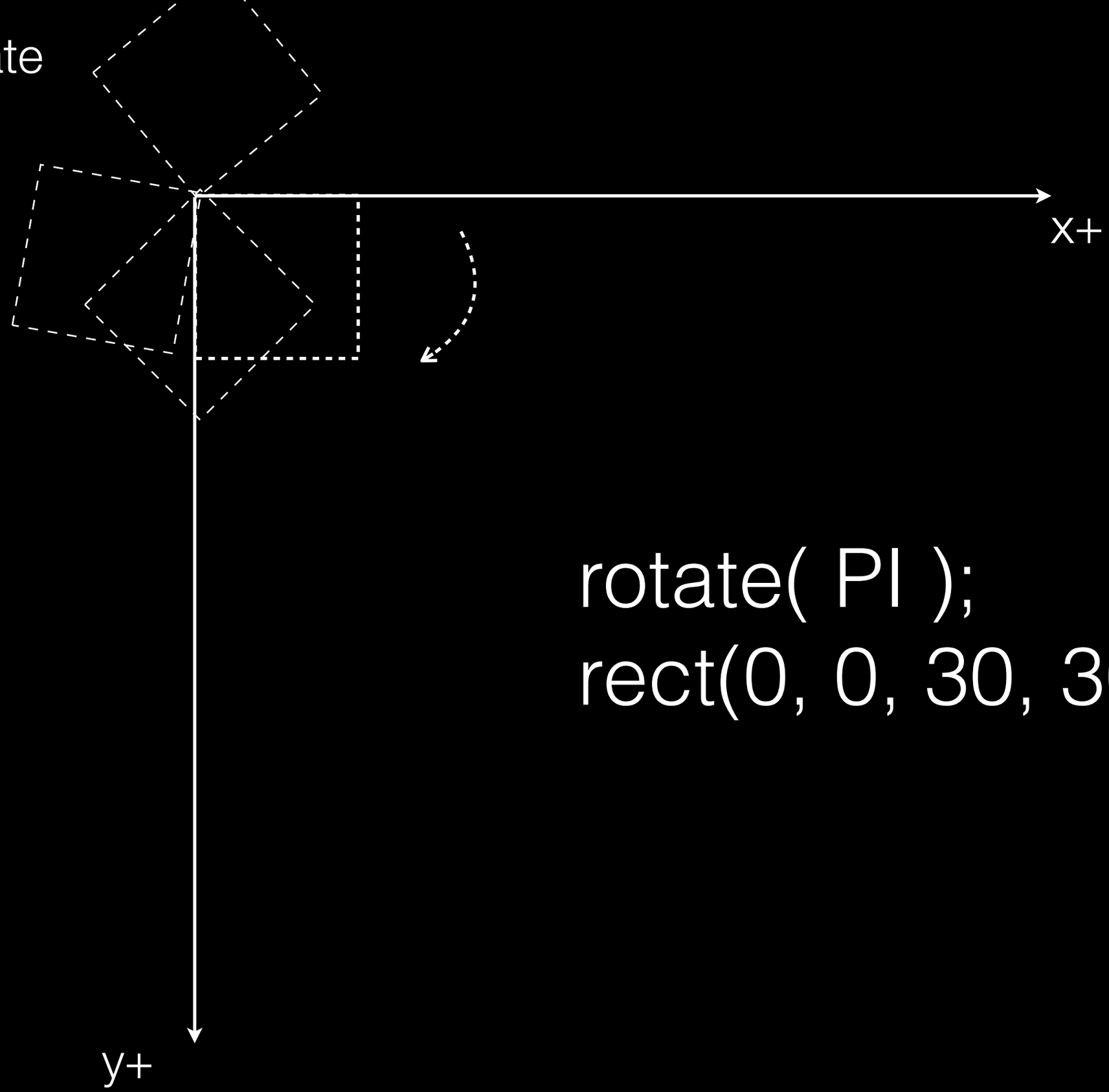
Rotate



-45° = -π/4 = -PI/4

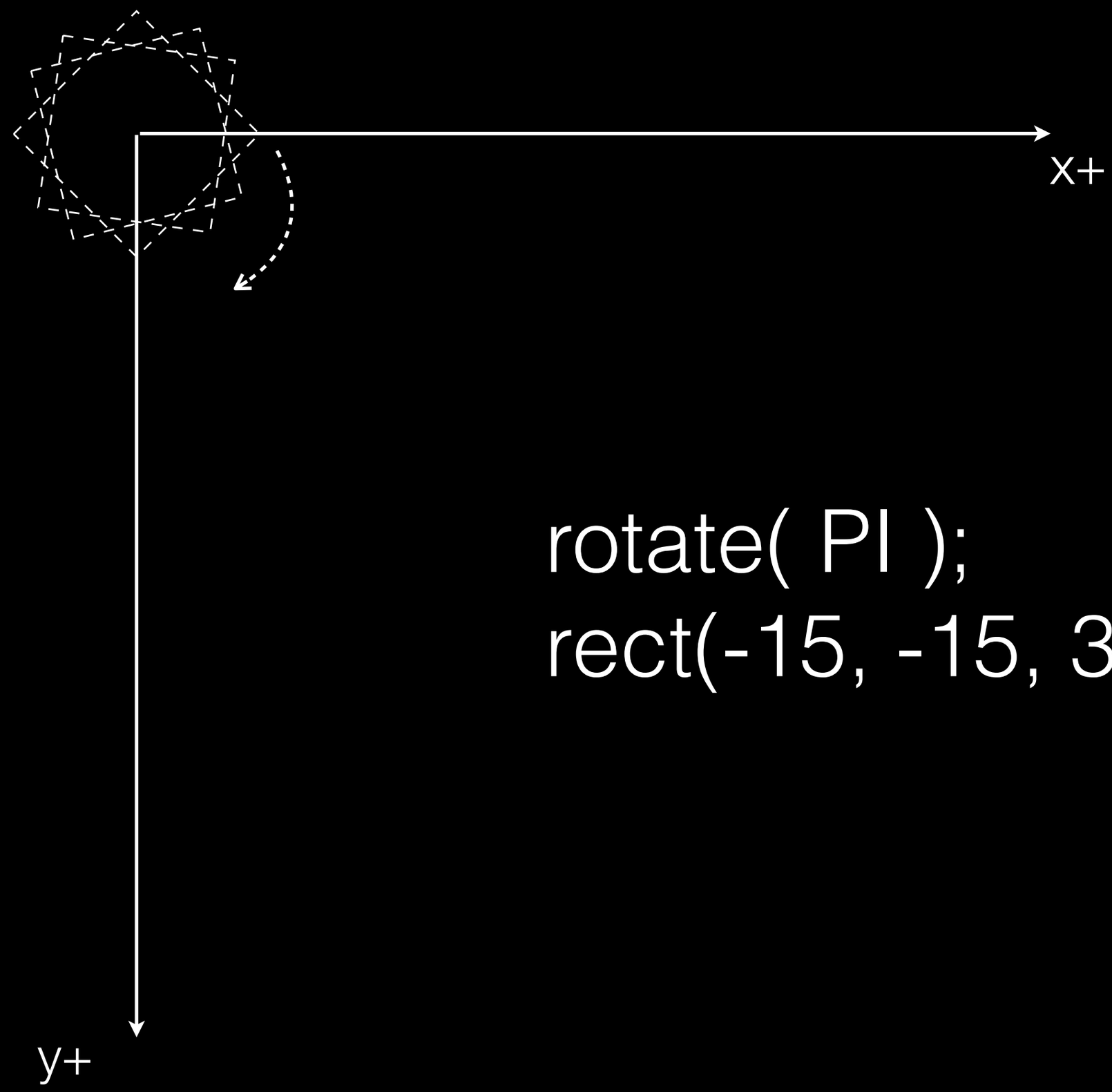
```
translate(30,30);
rotate(-PI/4);
rect(0,0,50,50);
```

Rotate




```
rotate( PI );
rect(0, 0, 30, 30);
```

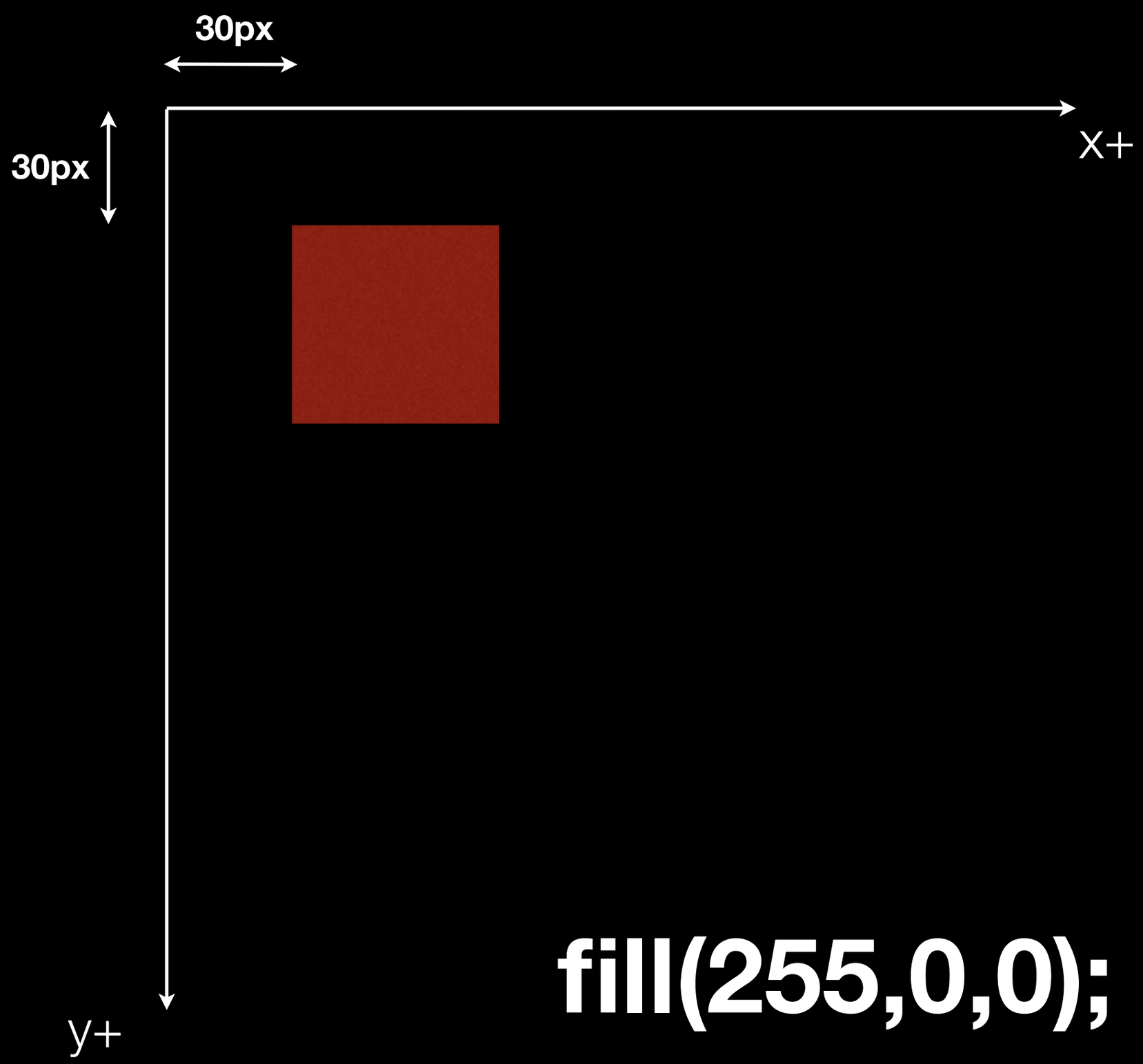
Rotate




```
rotate( PI );
rect(-15, -15, 30, 30);
```

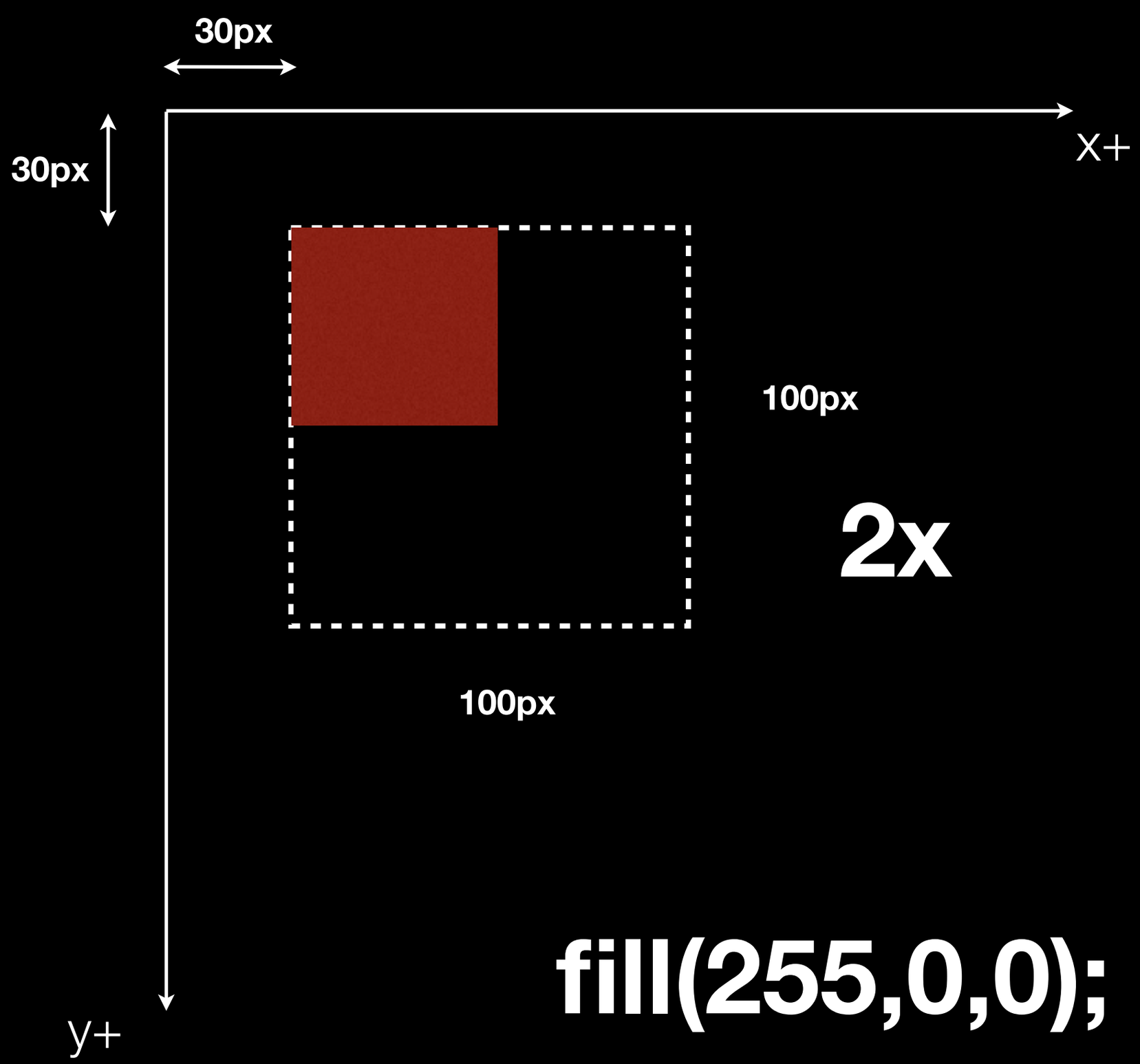
```
scale( s );
scale( x, y );
```

Scale




```
fill(255,0,0);
rect(30,30,50,50);
```

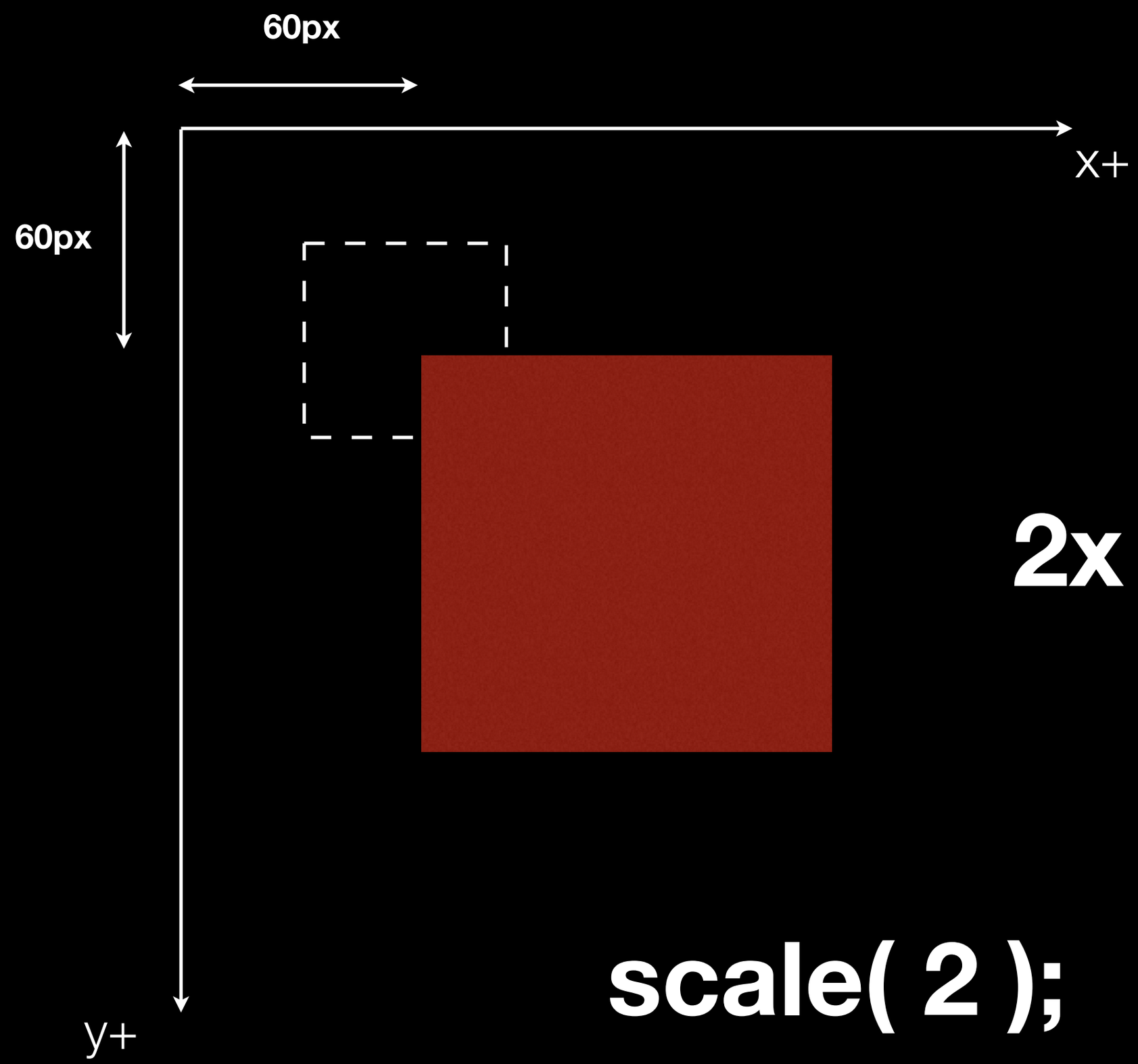
Scale



2x

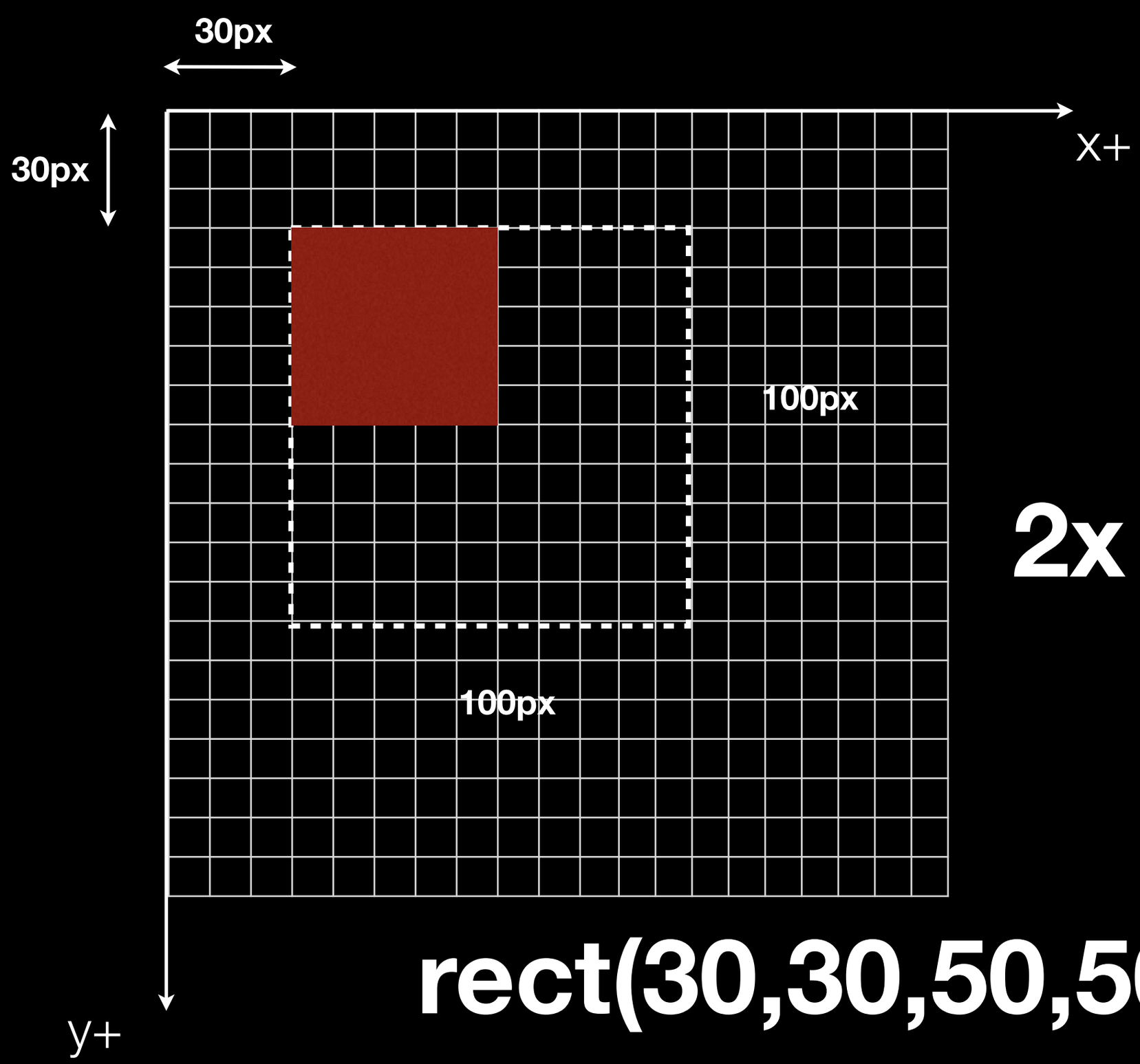
```
fill(255,0,0);
rect(30,30,50,50);
```

Scale




```
scale( 2 );
rect(30,30,50,50);
```

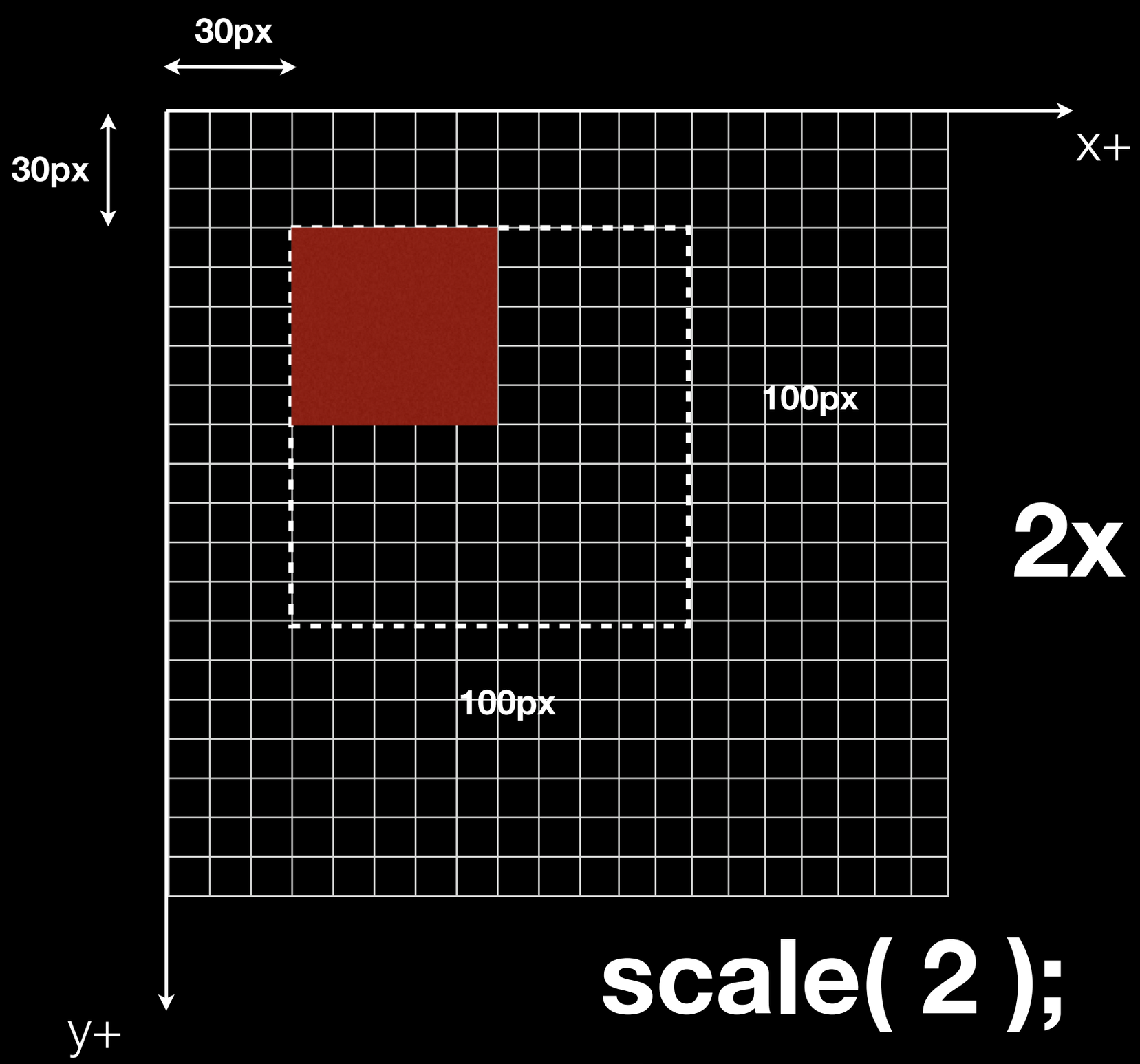
Scale



2x

rect(30,30,50,50);

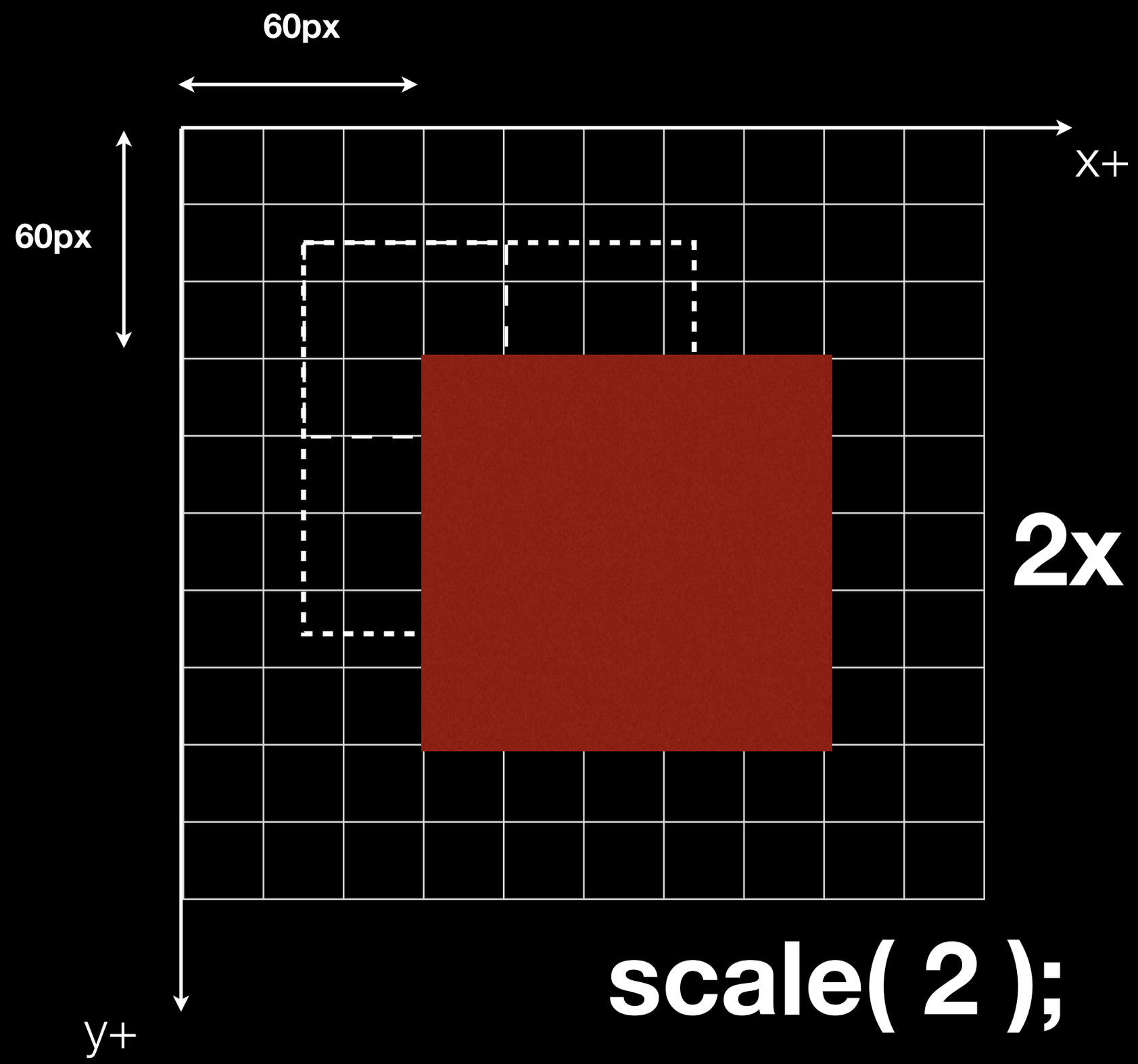
Scale



2x

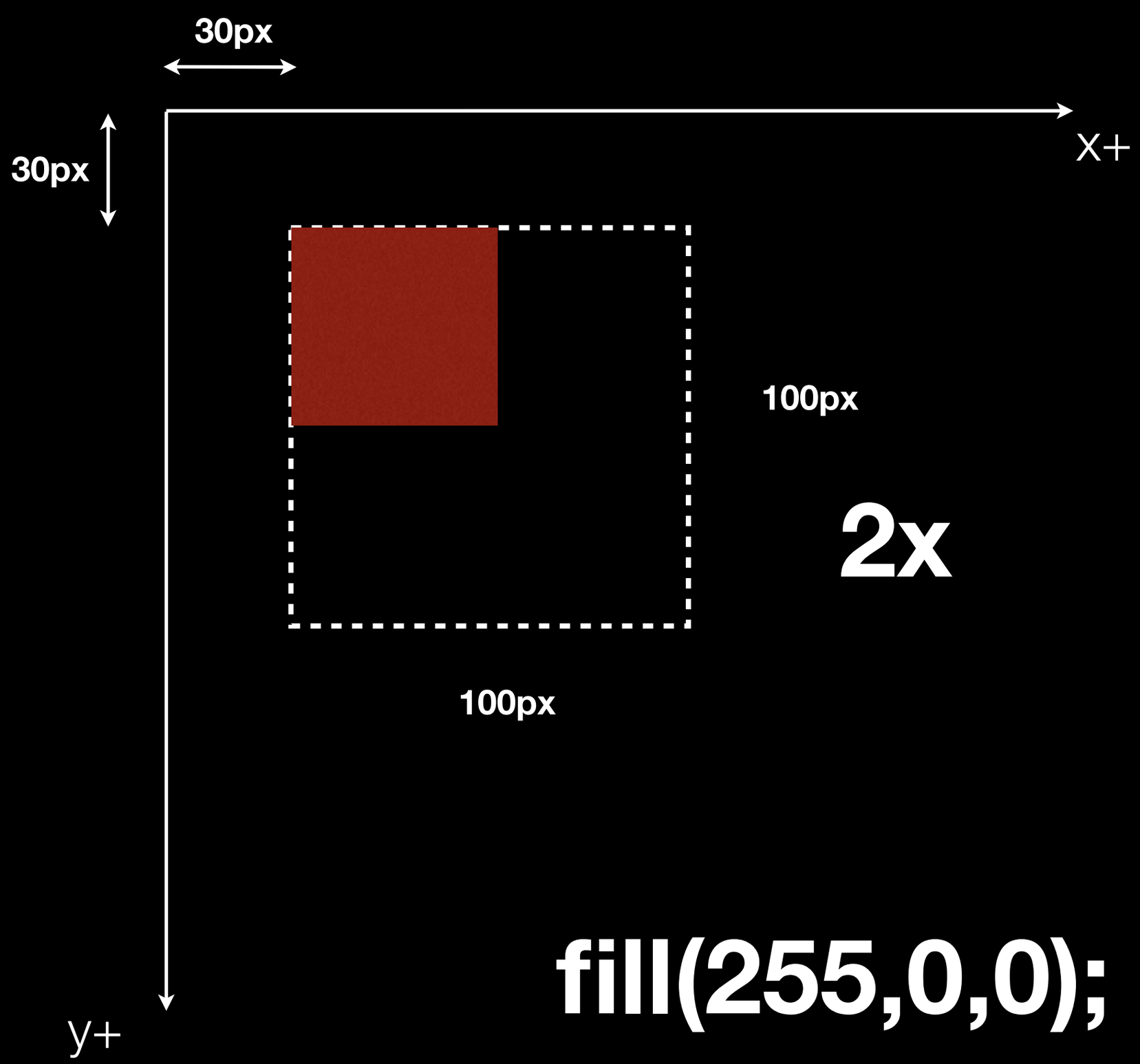
```
scale( 2 );
rect(30,30,50,50);
```

Scale




```
scale( 2 );
rect(30,30,50,50);
```

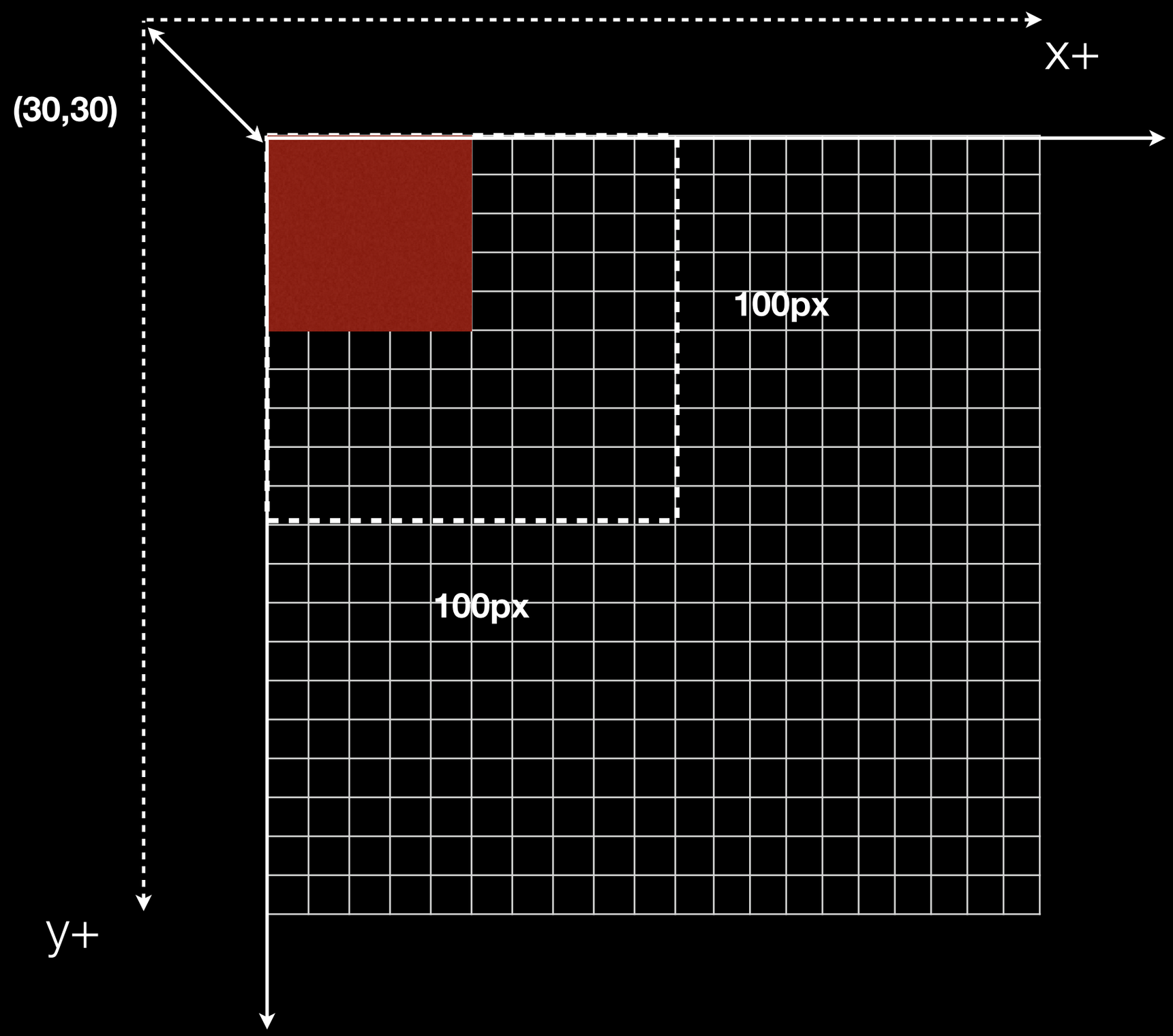
Scale



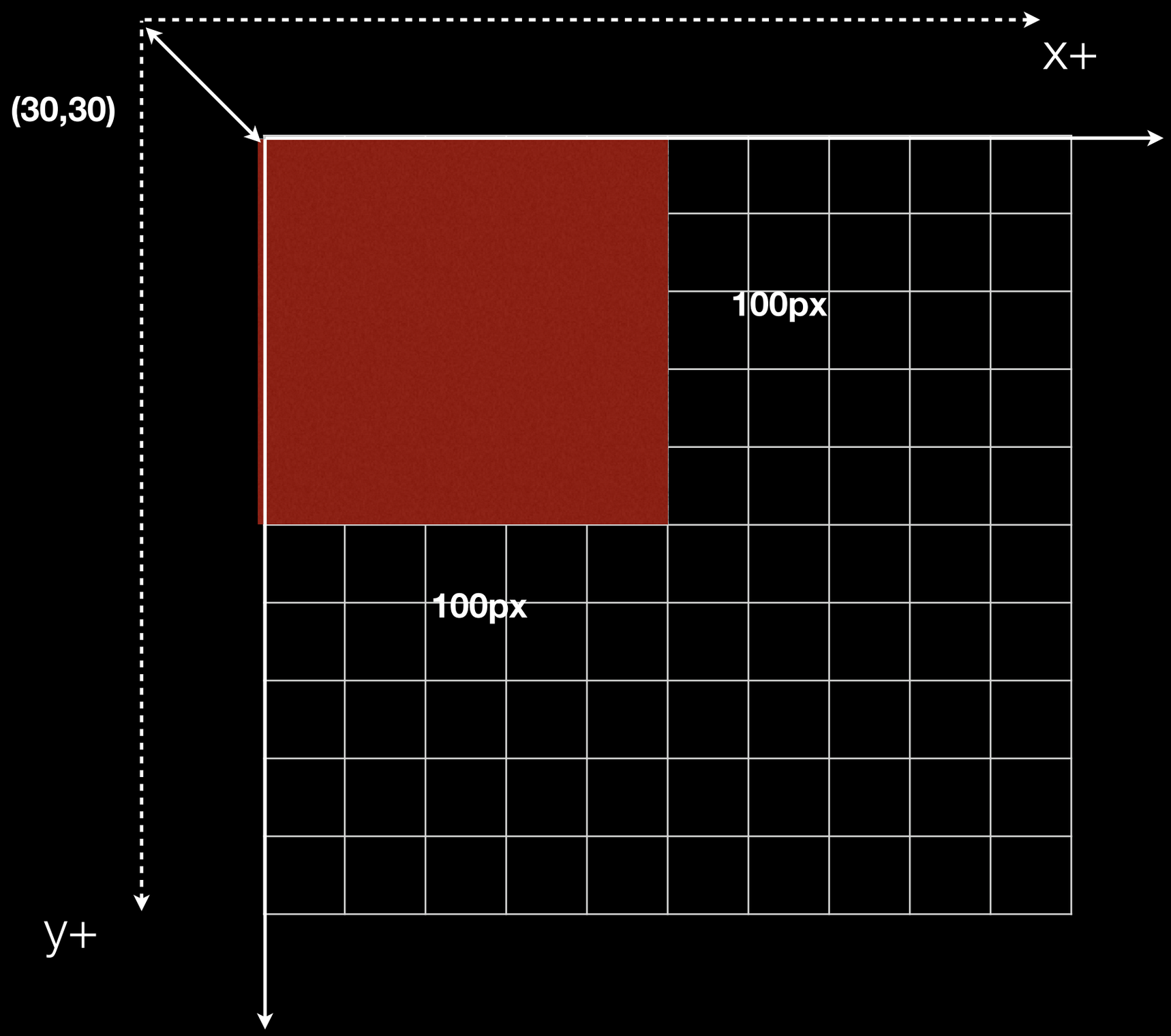
2x

```
fill(255,0,0);
rect(30,30,50,50);
```

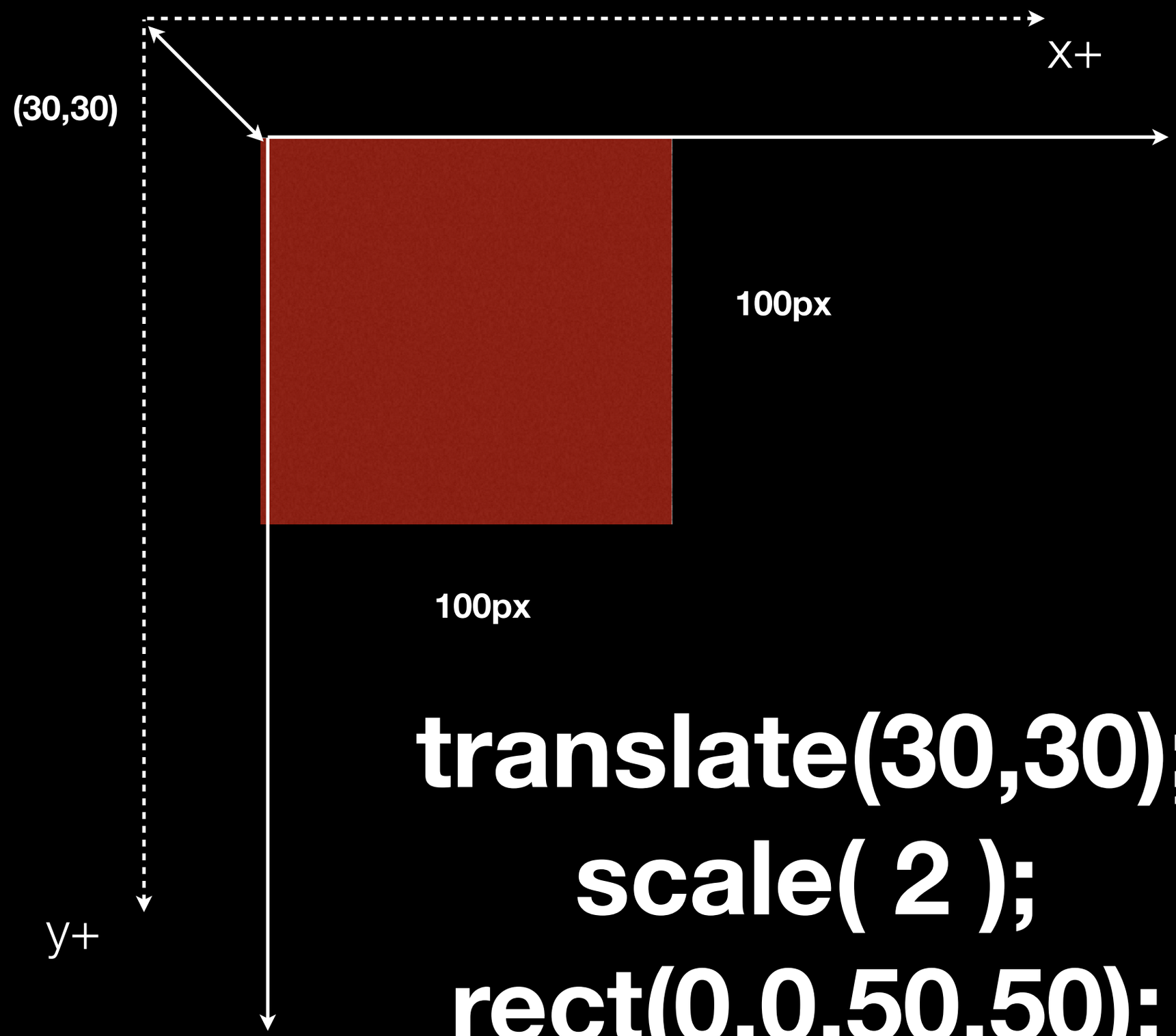
Scale



Scale



Scale

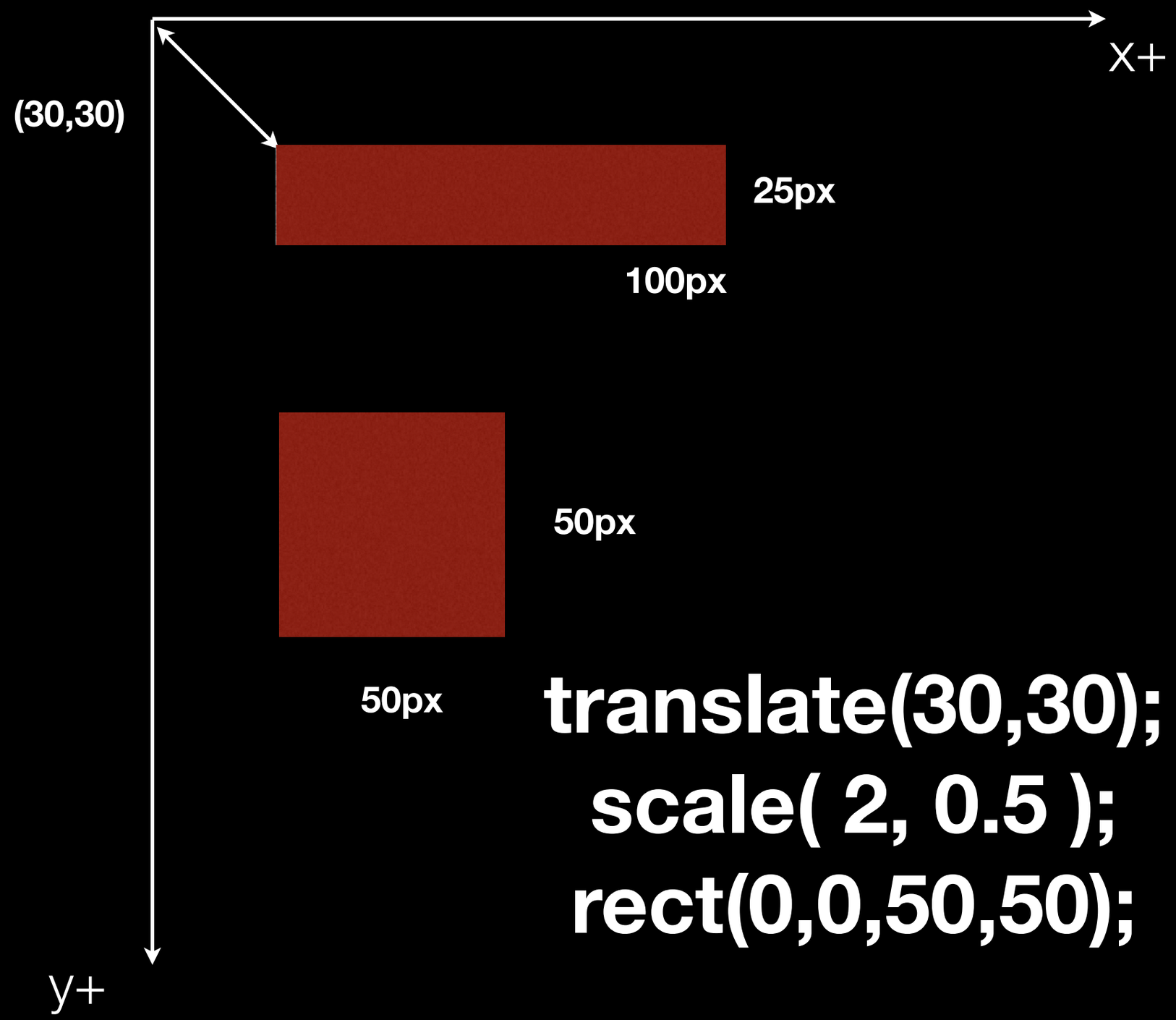



```
translate(30,30);
scale( 2 );
rect(0,0,50,50);
```

Scale

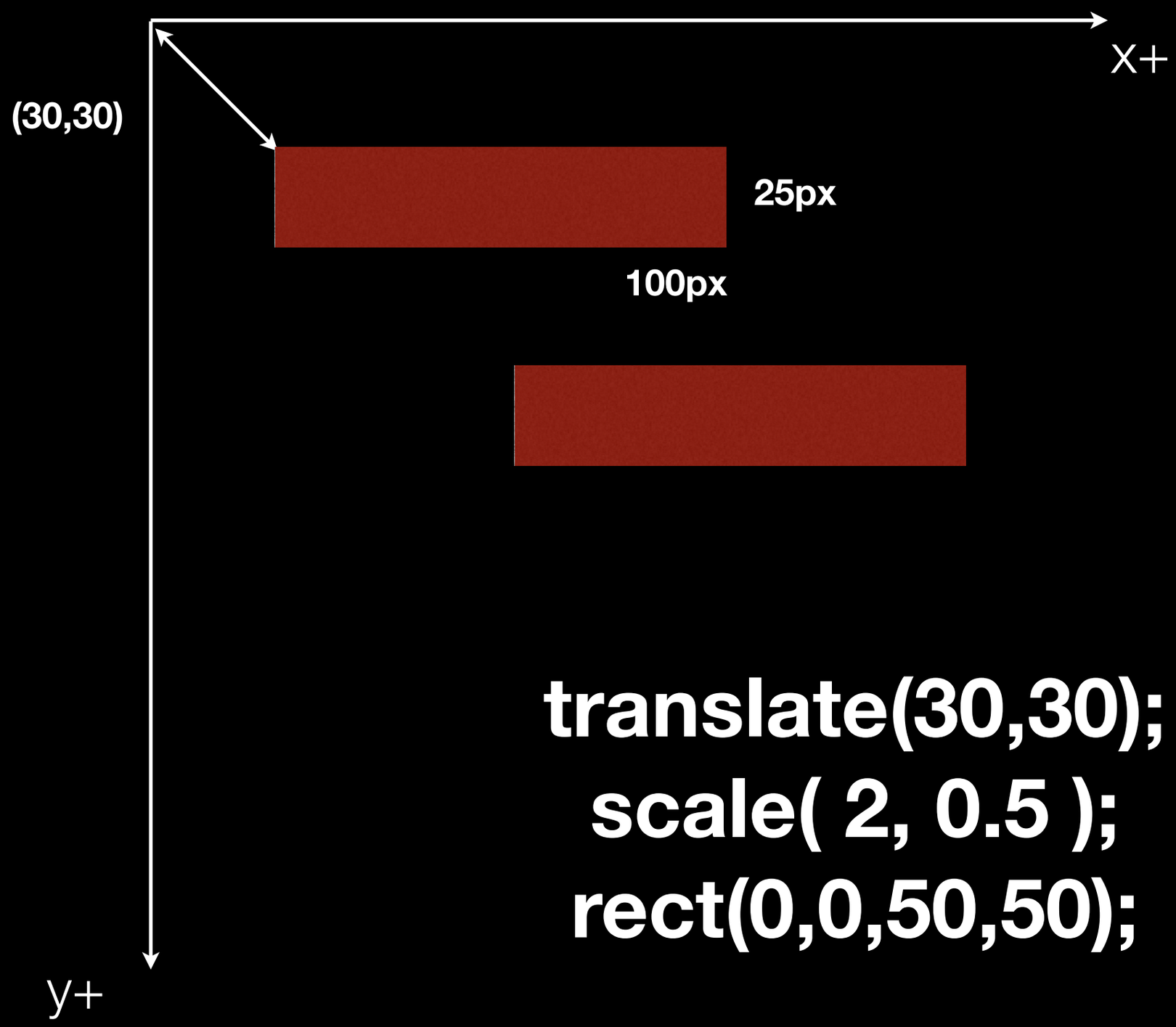

```
translate(30,30);
scale( 2, 0.5 );
rect(0,0,50,50);
```

Scale



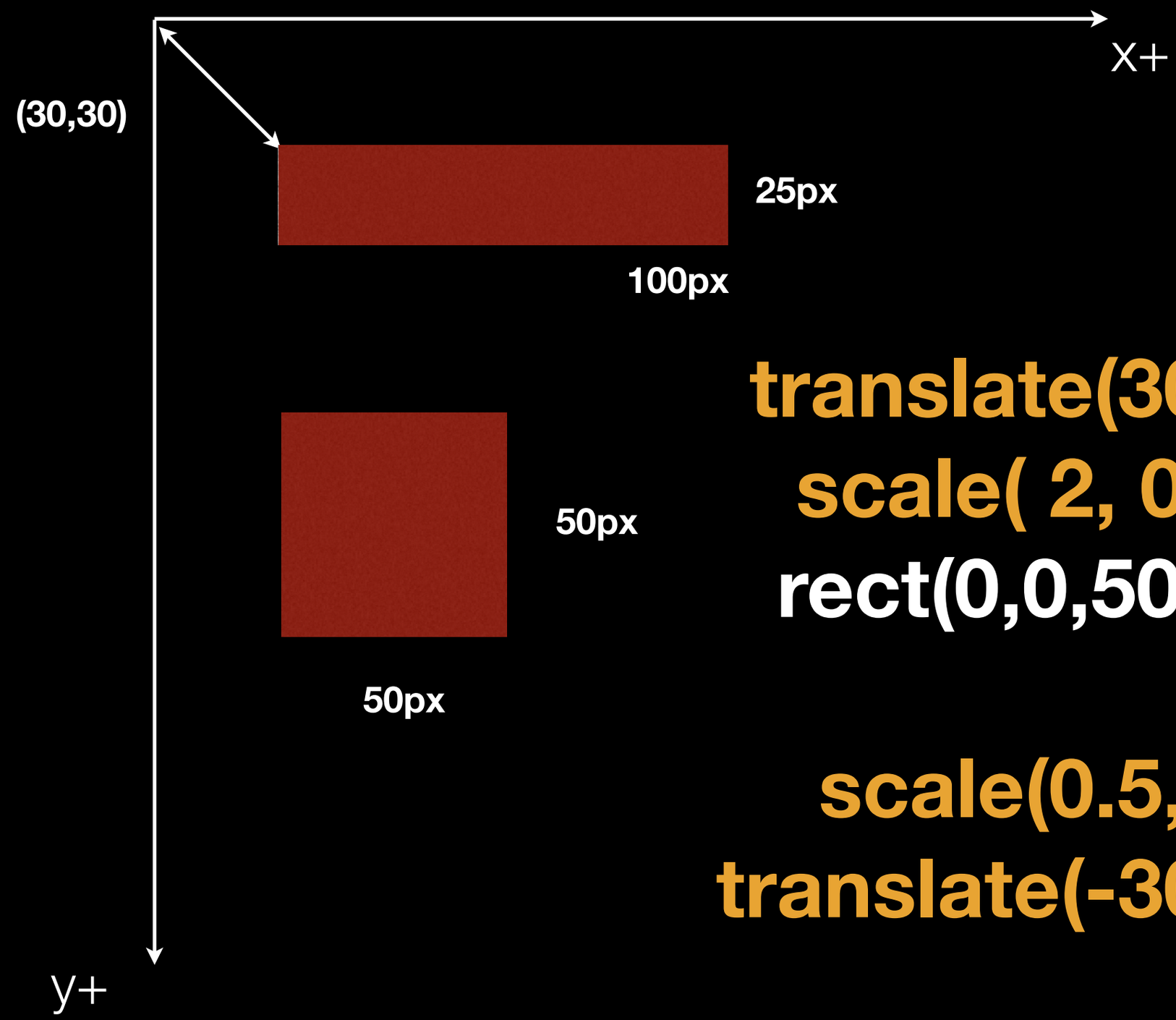

```
translate(30,30);
scale( 2, 0.5 );
rect(0,0,50,50);

rect(30,100,50,50);
```

Scale



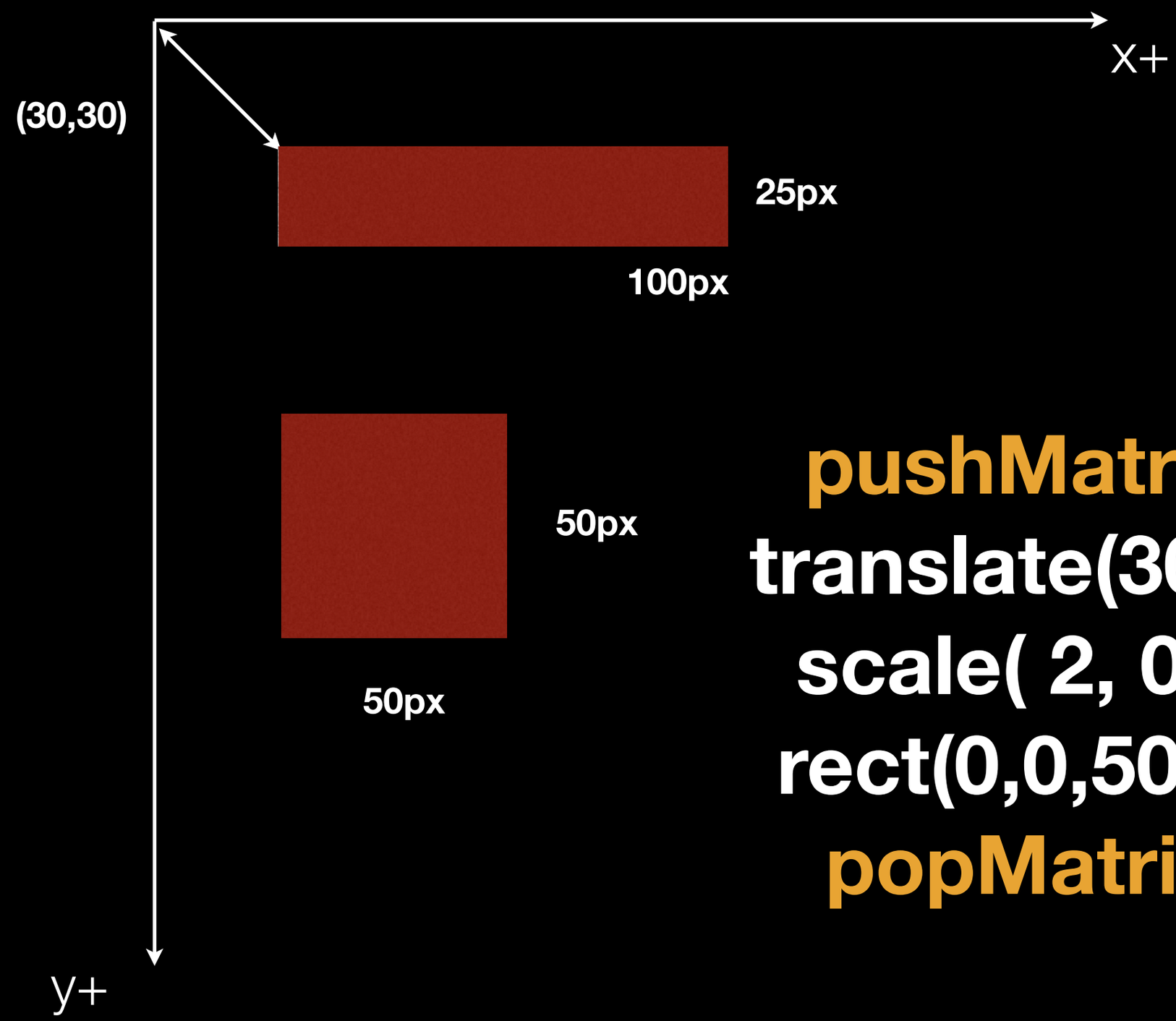

```
translate(30,30);
scale( 2, 0.5 );
rect(0,0,50,50);

rect(30,100,50,50);
```

Scale




```
translate(30,30);
scale( 2, 0.5 );
rect(0,0,50,50);

scale(0.5, 2);
translate(-30,-30);

rect(30,100,50,50);
```

Scale



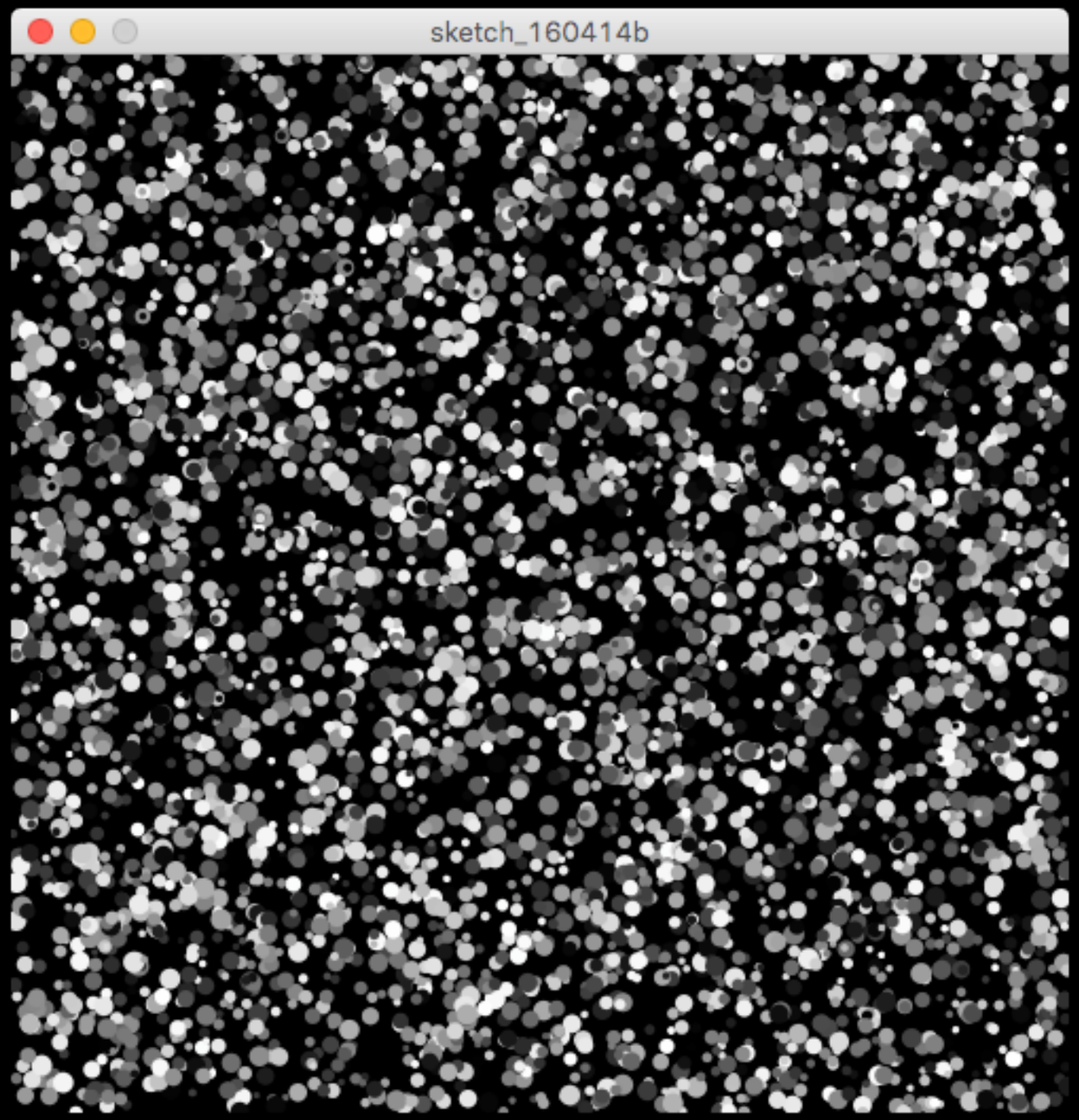

```
pushMatrix();
translate(30,30);
scale( 2, 0.5 );
rect(0,0,50,50);
popMatrix();

rect(30,100,50,50);
```

Scale

```
pushMatrix();
translate( … );
scale( … );
rotate( … );
rect(…);
ellipse(…);
popMatrix();

rect(30,100,50,50);
```

Order of transformation

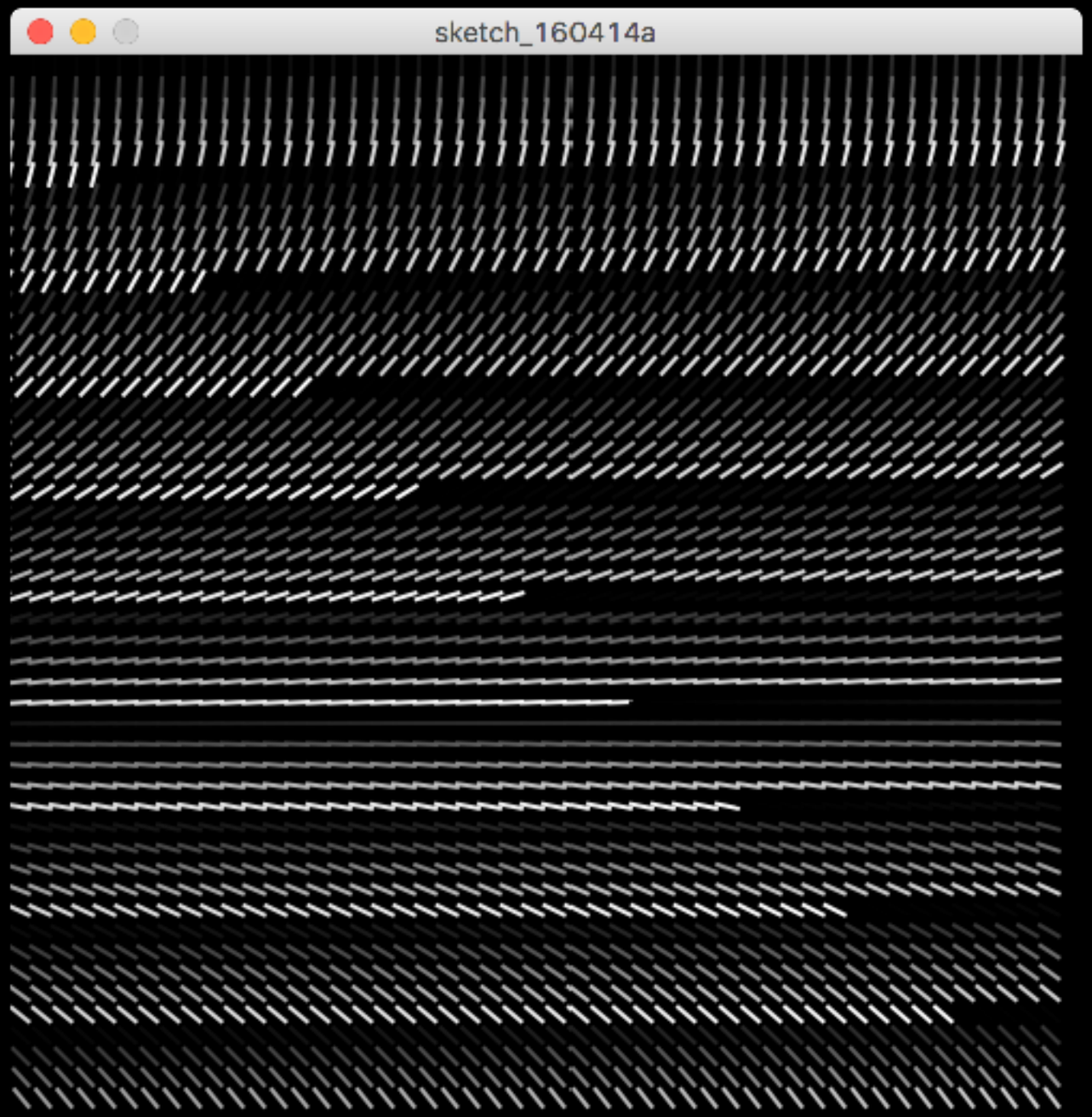
When you do multiple transformations, the order makes a difference. A rotation followed by a translate followed by a scale will not give the same results as a translate followed by a rotate by a scale.


sketch_160414b

For Loop

```
for( int i=0; i< 5000; i++ ){
  float r = random(3,10);
  fill(random(255));
  ellipse( random(width), random(height), r, r );
}
```

For Loop



For Loop

```
for( int i=0; i< 50; i++ ){
  for(int j=0; j< 50; j++ ){
    int idx = i + j*50;
    fill( idx%255 );
    pushMatrix();
    translate( i*10, j*10 );
    rotate( idx*0.001 );
    rect( 0, 0, 2, 12 );
    popMatrix();
  }
}
```

For Loop

```
for( int i = 0; i < 100; i++ ) {

}
```

For Loop

```
for( int i = 0; i < 100; i++ ){
    // code to run
    // do something 100 times

}
```

For Loop

```
for( int i = 0; i < 33; i++ ){
    // code to run
    // do something 33 times

}
```

For Loop

```
for( int i = 0; i < 100; i++ ){
    // code to run
   // do something repeatedly

}
```

STEP 1.

i 라는 이름의 integer type 변수를 만들고, 0값으로 셋팅.

For Loop

```
for( int i = 0; i < 100; i++ ){
    // code to run
    // do something repeatedly

}
```

STEP 2.

i 의 값이 100 미만인지 테스트.

아니면 바깥으로 탈출.

For Loop

```
for( int i = 0; i < 100; i++ ){
    // code to run
    // do something repeatedly

}
```

STEP 3.

중괄호 안의 코드를 한번씩 실행함.

For Loop

```
for( int i = 0; i < 100; i++ ){
    // code to run
    // do something repeatedly

}
```

STEP 4.

i 의 값에 1을 더하고 STEP 2 로 돌아감.

i++

i += 1

i = i + 1

For Loop

```
for( int i = 0; i < 10; i++ ){
    for( int j = 0; j < 10; j++ ){
        println (i , j)'
    }
}
```

과제 1: Interactive Self-portrait

마우스, 혹은 키보드에 반응하는 자화상 그리기

+ rotate() 혹은 scale() 한 번 이상 사용.
+ // 코드를 설명하는 코멘트 최대한 많이 달기. (최소 3회)

+ 1개 이상의 스크린샷 이미지
+ 스크린 동영상 캡쳐하여 유튜브 업로드 유튜브 embed
+ 간단한 설명 (왜 / 어떻게)
+ 코드 폴더 zip 하여 구글/네이버/드롭박스 등 클라우드 드라이브 링크
+ 코드 복사 붙여넣기

과제 2: Pattern Recognition

내 생활, 머리 속의 (의미 있는) 패턴을 찾아 코드로 그림그리기

+ for 구문을 이용하여 패턴을 그림.
+ // 코드를 설명하는 코멘트 최대한 많이 달기. (최소 3회)
+ noLoop() 사용해도 무방함.

+ 1개 이상의 스크린샷 이미지
+ 간단한 설명 (왜 / 어떻게)
+ 코드 폴더 zip 하여 구글/네이버/드롭박스 등 클라우드 드라이브 링크
+ 코드 복사 붙여넣기

과제

매주 월요일 밤 10시

기한 맞춰 제출

평가 항목
+ 과제별 요구사항
+ 아이디어, 디자인
+ 노력, 시간

과제

남의 코드 베끼지 말기. 가능한 직접 쓰기.

다른 사람/인터넷의 코드를 참조하는 경우,
+ 블로그와 코드 안에 출처 밝히고,
+ 이해해서 내 것으로 만든 경우에만 인정. (모르면 질문)

과제 검사 시, 질문할 수도.