

2025 Spring semester

Interaction Lab (1)

week 4

Keunjung Bae

변수(Variables)

❖ 변수와 변수명

메모리에 임의의 데이터를 저장할 수 있는 공간을 '변수', 그 공간의 주소를 '변수명' 이라 한다.

❖ 자료형(Data Type)

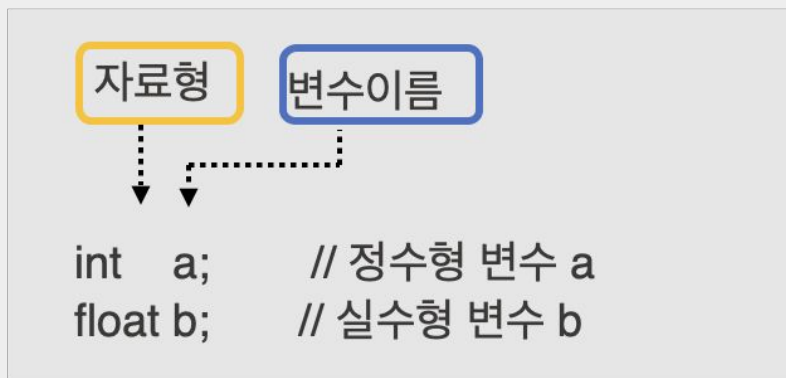
데이터의 크기는 '자료형'에 의해 결정된다. 자료형이란 데이터의 형식을 말하며, 정수형(integer), 실수형(float) , 문자형(character) 등이 있다. 이 데이터의 형식에 따라 필요한 공간의 크기가 달라진다.



❖ 자료형의 종류와 크기

자료형		크기	데이터
정수형	int	4byte	...-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5...
실수형	float	4byte	...-0.2, -0.1, -0.02, -0.01, 0.0, 0.01, 0.02, 0.1, 0.2...
문자형	char	1byte	A~Z, a~z...
논리형	boolean	1byte	true(참) or false(거짓)

❖ 변수의 선언



저장할 데이터 타입을 정하고, 주소를 부여하는 것을 변수를 ‘선언한다 (declare)’라고 한다. 변수의 선언은 먼저 자료형을 명시하고, 그 뒤에 변수 이름을 써 준다.

❖ 변수의 초기화

자료형 변수이름;
변수이름 = 초기값;

```
int a;      // 정수형 변수 a선언  
a = 0;     // 변수 a에 0을 저장
```

```
float b;    // 실수형 변수 b선언  
b = 1.2;   // 변수 b에 1.2 저장
```

변수이름

초기값

변수를 선언한 후에는 변수를 초기화 해준다. 초기화란 선언한 변수에 처음 값을 저장해주는 것을 말한다. 값을 저장할 때에는 연산자 '='를 사용한다.

❖ 변수명 만들기

변수 이름은 사용자가 필요에 따라 직접 만들 수 있다. 주로 소문자를 사용하며, 이미 내장된 변수나 예약어(keyword)는 사용할 수 없다. 예약어란 이미 특정한 용도로 사용이 예약된 이름을 말한다.

예) int, float, char, boolean, for, if,

변수 이름을 설정할 때에는 너무 길지 않도록 하며, 변수 이름으로 저장될 데이터의 종류를 예측할 수 있도록 짓는 것도 좋은 방법이다.

```
예) int dist;      // 거리(distance)에 대한 값을 저장하고 읽기 위한 변수  
    float ang;     // 각도(angle)에 대한 값을 저장하고 읽기 위한 변수  
    float xpos;    // 마우스의 x좌표를 저장하고 읽기 위한 변수
```

Scaling *an input value for visual effect*

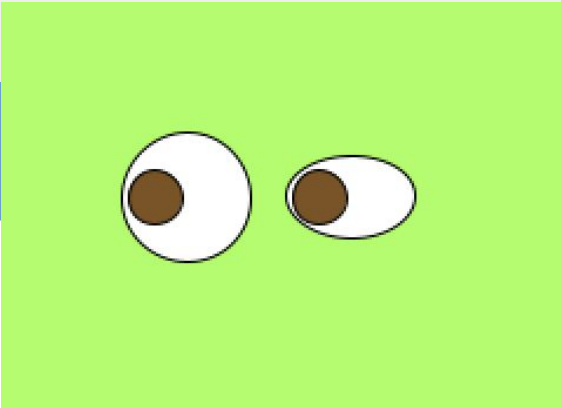
▶

■

전체 화면

sketch 250325b

```
1 void setup() {  
2   size(400, 400);  
3 }  
4  
5 void draw() {  
6   background(182, 252, 112);  
7   // White eye  
8   fill(255);  
9   // ellipse(217, 110, 55, 55 - 6 * mouseX / 100.0);  
10  ellipse(217, 110, 55, 55 );  
11  
12  ellipse(287, 110, 55, 55 - 6 * mouseX / 100.0);  
13  
14  // Brown eye  
15  fill(120, 84, 40);  
16  ellipse(217 - 2 * mouseX / 50.0, 110, 23, 23);  
17  ellipse(287 - 2 * mouseX / 50.0, 110, 23, 23);  
18 }  
19  
20  
21  
22  
23  
24
```





정지



Java ▾

sketch 250321a ▾

```
1 int a = 2+4;
2 println(a);
3
4 int b = 2-4;
5 println(b);
6
7 int c = 2*4;
8 println(c);
9
10 int d = 2/4;
11 println(d);
12
13 float e = 2/4;
14 println(e);
15
16 float f = 2.0/4.0;
17 println(f);
18
```

```
6
-2
8
0
0.0
0.5
```

Console

Errors



Java ▾

sketch 250321a ▾

```
1 float xpos = 0;
2 float xspeed = 2;
3
4 void setup() {
5   size(400, 400);
6   smooth();
7 }
8
9 void draw() {
10  background(255);
11
12  //xpos = xpos + xspeed;
13  xpos += xspeed;
14  ellipse(xpos, 200, 20, 20);
15 }
```

자동 서식 적용 완료





Java ▾

sketch 250321a ▾

```
1 float a = 2.0;
2 println(a);
3
4 //a += 4.0;
5 //println(a);
6
7 //a *= 4.0;
8 //println(a);
9
10 //a /= 4.0;
11 //println(a);
12
13 a = a- 4.0;|
14 a-=4.0;
15 println(a);
16
17 a = a+1;
18 a += 1;
19 a++;
20
21 a--;
```



Java ▼

sketch 250321a ▼

```
1 int a = 8/3;  
2 println(a);  
3  
4 int b = 7 % 2;  
5 println(b);  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```





Java ▾

sketch 250321a ▾

```
1 int a =3;
2 int b =4;
3
4 //boolean c = (a==b);
5 //println(c);
6
7 //boolean c = (a < b);
8 //println(c);
9
10 boolean c = (a ≠ b);
11 println(c);
12
13
14
15
16
17
18
19
20
```

```
sketch 250321a
1 int a = -3;
2 int b = abs(a);
3 println(b);
4
5 float a = 3.2;
6 float b = ceil(a);
7 println(b);
8
9 float a = 3.2;
10 float b = floor(a);
11 println(b);
12
13 float a = 3.2;
14 float b = round(a);
15 println(b);
16
17 float a = dist(0,0,10,20);
18 println(a);
19
20 float c = sqrt(2);
21 println(c);
22
23
24
```

(0,0)  (5,4)

dist(x1, y1, x2, y2);



Java ▼

sketch 250321a ▼

```
1 float a = 5.5;
2 float b = constrain(a, 2, 4);
3 println(b);
4
5 float score = 80; // 0~100점 중 하나
6 float rating = map(score, 0, 100, 0, 5);
7 println(rating);
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```



Java ▾

sketch 250321a ▾

```
1 void setup() {  
2   size(500, 500);  
3 }  
4  
5 void draw() {  
6   float xpos = random(100, 400);  
7   float ypos = random(100, 400);  
8   //float size = random(5,10);  
9   float size = map(mouseX, 0, width, 5, 50);  
10  
11   ellipse(xpos, ypos, size, size);  
12   //ellipse(constrain(xpos, 180, 320), ypos, size, size);  
13 }
```



연습문제 : random() 함수이용해서 실행창에

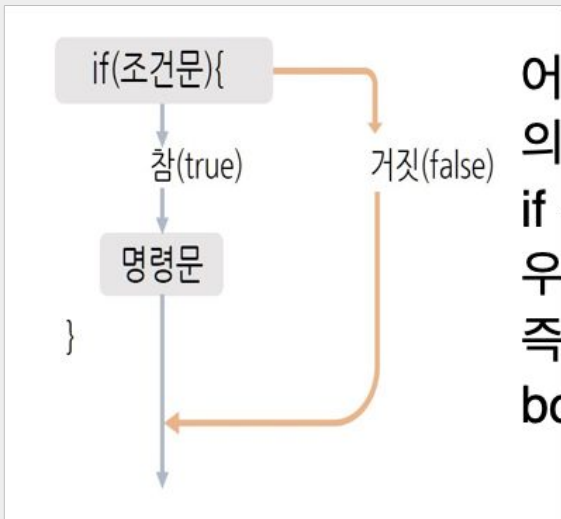
무작위의 위치, 무작위 크기, 무작위 색으로 생기는 원을 계속 그려보세요.

화면에 원이 계속 쌓이도록 그려보세요.

조건문

if () { }

❖ If구문의 기본구조



어떠한 조건을 만들기 위해서는 ‘만약 ~라면’의 뜻을 갖는 ‘if’를 사용한다.

if 구문은 if 다음 괄호 안의 조건을 만족할 경우, 아래의 명령문이 실행된다.

즉 조건에 맞는지 맞지 않는지를 판단하는 boolean 논리를 따른다.

if () { }

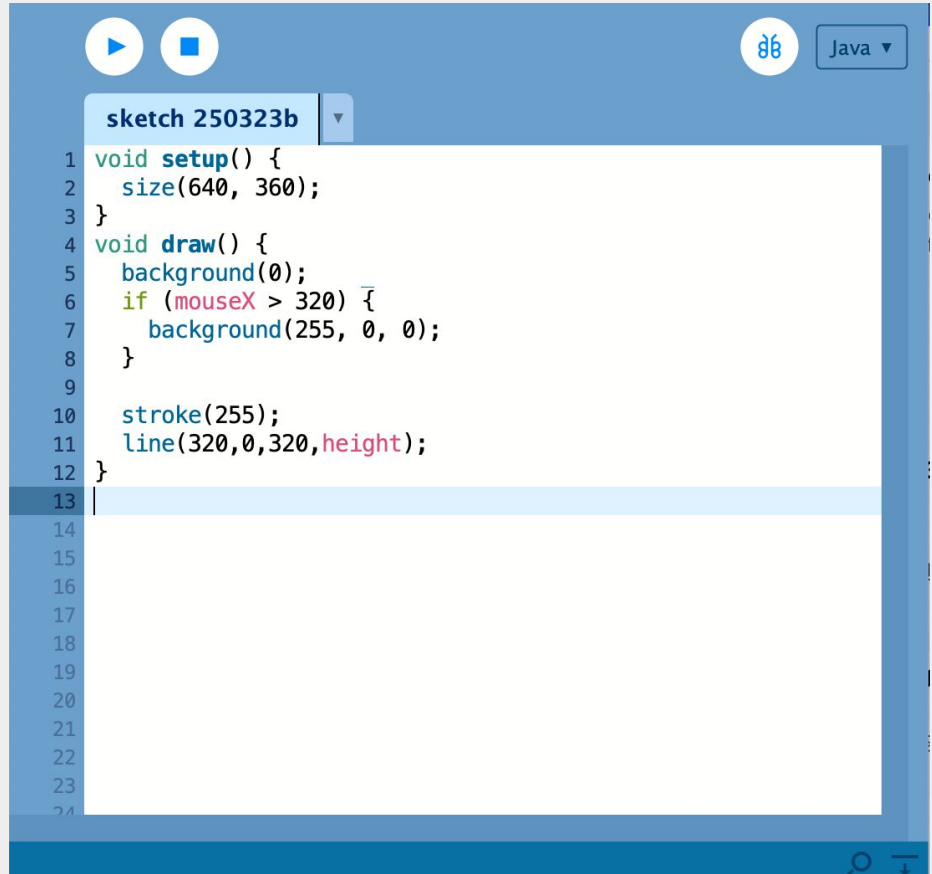
비교 연산자(Comparing Operator)

비교 연산자는 두 개의 값을 비교하여 참 또는 거짓과 같은 논리(bool)형으로 결과값을 반환하는 연산자이다. 주로 if 구문에서 조건에 대한 참, 거짓을 판단할 때 사용된다.

연산자	의미	예제	풀이
>	크다	if(x>10)	x가 10보다 크다면
<	작다	if(x<10)	x가 10보다 작다면
>=	크거나 같다	if(x>=10)	x가 10보다 크거나 같다면
<=	작거나 같다	if(x<=10)	x가 10보다 작거나 같다면
==	같다	if(x==10)	x가 10이라면
!=	같지 않다	if(x!=10)	x가 10이 아니라면

(320,,0)

```
if( mouseX > 320 ) {  
  
}
```



```
sketch 250323b  
1 void setup() {  
2   size(640, 360);  
3 }  
4 void draw() {  
5   background(0);  
6   if (mouseX > 320) {  
7     background(255, 0, 0);  
8   }  
9  
10  stroke(255);  
11  line(320,0,320,height);  
12 }  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24
```

연습문제 :

마우스를 자유자재로 이동할때 원이 붓처럼 나오는 스케치 그리는데
마우스를클릭할때 배경이 초기화되게 해보세요.

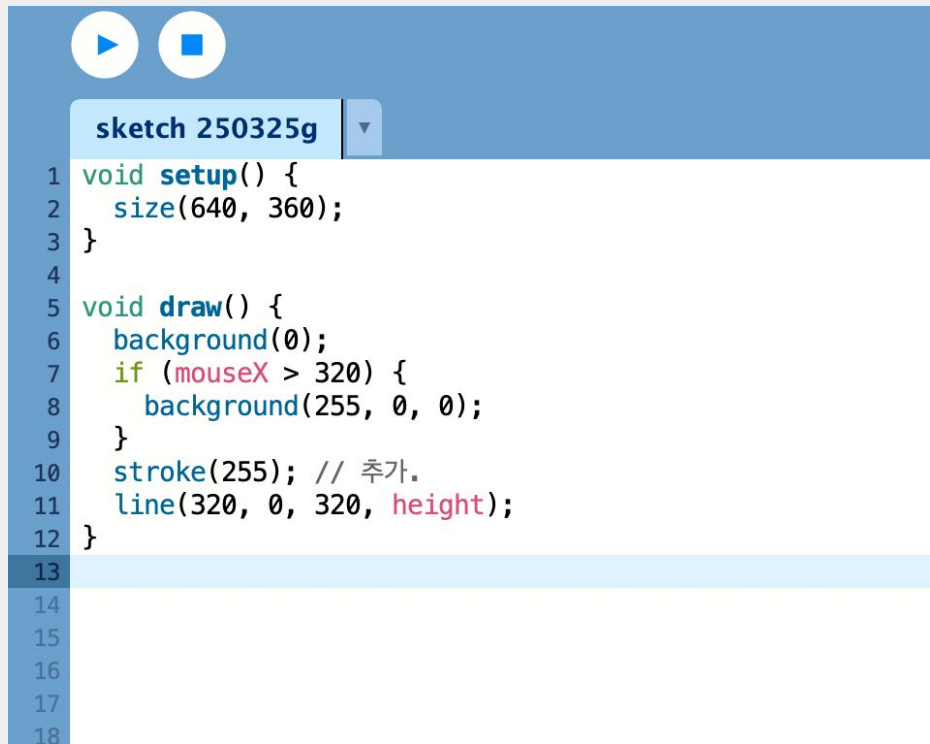
아래 조건을 만족하는 Processing 코드를 작성하면 됩니다.

1. 화면 크기는 640x360으로 설정한다.
2. 마우스 위치에 따라 원의 크기를 다르게 한다: (if 조건문쓰기)
 - 마우스가 화면 **왼쪽 절반**에 있을 때는 작은 원 (크기 30)
 - **오른쪽 절반**에 있을 때는 큰 원 (크기 80)
3. 원의 색은 매번 **무작위 색상**으로 설정한다.
4. 마우스를 자유자재로 움직이며 그림을 그리듯 원이 계속 나타나야 한다.(붓처럼)
5. 마우스를 클릭하면 **배경이 초기화**되어 화면이 지워진다.

```
if( ) {
```

```
}else{
```

```
}
```



```
sketch 250325g
1 void setup() {
2   size(640, 360);
3 }
4
5 void draw() {
6   background(0);
7   if (mouseX > 320) {
8     background(255, 0, 0);
9   }
10  stroke(255); // 추가.
11  line(320, 0, 320, height);
12 }
13
14
15
16
17
18
```

if /if

(각자 따로 검사)

- 모든 조건을 전부 검사함
- 조건이 여러 개라면, 참인 건 전부 실행됨

```
void setup(){
  size(400,400);
}
void draw() {
  if (mouseX > 100) {
    println("100 보다 큼");
  }

  if (mouseX > 200) {
    println("200 보다 큼");
  }
}
```

if /else

((둘 중 하나만 실행))

- 두 경우 중 하나만 처리할 때

```
void draw() {
  if (mouseX > 100) {
    println("100 보다 큼");
  } else {
    println("100 이하임");
  }
}
```

if /else if /else

(순차 조건 검사)

- 위에서부터 차례로 검사하고,
- 처음 맞는 조건만 실행한 후 나머지는 건너뛴

```
void draw() {
  if (mouseX > 200) {
    println("200보다 큼");
  } else if (mouseX > 100) {
    println("100보다 큼");
  } else {
    println("100 이하임");
  }
}
```



Java ▾

sketch 250323e ▾

```
1 void setup() {  
2   size(640, 360);  
3 }  
4  
5 void draw() {  
6   //background(0);  
7   if (mouseX > 320) {  
8     background(255, 0, 0);  
9   }else{  
10    background(0);|  
11  }  
12  stroke(255); // 추가.  
13  line(320, 0, 320, height);  
14 }  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24
```





Java ▾

sketch 250323e ▾

```
1 void setup() {  
2   size(640, 360);  
3  
4 }  
5  
6 void draw() {  
7   background(0);  
8   fill(175);  
9   rectMode(CENTER);  
10  
11   if (mouseX > 320) {  
12     ellipse(320,180,100,100);  
13   }  
14   else{  
15     rect(320,180,100,100);  
16   }  
17  
18   stroke(255);  
19   line(320, 0, 320, height);  
20 }  
21  
22  
23  
24
```



Java ▾

sketch 250323e ▾

```
1 void setup() {
2   size(600, 360);
3 }
4
5 void draw() {
6   background(0);
7   fill(175);
8   rectMode(CENTER);
9
10  if ( mouseX > 400) {
11    ellipse(300, 180, 100, 100);
12  } else if (mouseX > 200){
13    rect(300, 180, 100, 100);
14  } else{
15    line(250,130,350,220);
16  }
17
18  // if (mouseX < 200) {
19  //   line(250,130,350,220);
20  //} else if (mouseX < 400) {
21  //   rect(300, 180, 100, 100);
22  //} else {
23  //   ellipse(300, 180, 100, 100);
24  //}
25
26  stroke(255);
27  line(200, 0, 200, height);
28  line(400, 0, 400, height);
29 }
30
31
```

논리 연산자(Logical Operator)

논리 연산자는 두 개 이상의 비교식을 하나의 식으로 결합하여 참 또는 거짓의 결과값을 반환하는 연산자이다. 논리곱(AND), 논리합(OR), 논리 부정(NOT) 등이 있다.

연산자	의미	예제	풀이
&&	그리고(AND)	if(x>10 && x<50)	만약 x가 10보다 크고 50보다 작다면(두 조건이 모두 만족할 경우)
	또는 (OR)	if(x>10 x<50)	만약 x가 10보다 크거나 또는 50보다 작다면(두 조건 중 하나라도 만족할 경우)
!	~가 아닌(NOT)	if(x!=100)	만약 x가 100이 아니라면



Java ▾

sketch 250323g ▾

```
1
2 float x = 280;
3 float y = 100;
4 float size = 100;
5
6 void setup() {
7   size(640, 320);
8 }
9
10 void draw() {
11   background(0);
12   //fill(175);
13
14   if (mouseX > x && mouseX < x+size && mouseY > y && mouseY < y+size ) {
15     fill(255);
16   } else {
17     fill(175);
18   }
19
20   square(x, y, size);
21 }
22
23
24
```





Java ▾

sketch 250324a ▾

```
1 float x = 0;
2
3 void setup() {
4   size(640, 360);
5 }
6 void draw() {
7   background(0);
8   noStroke();
9   fill(255);
10  ellipse(x, height/2, 50, 50);
11
12  x = x + 7;
13
14  if ( x >= 640) {
15    x = 0;
16  }
17
18 }
19
20
21
22
23
24
25
26
```





Java ▾

sketch 250323h ▾

```
1 float x = 0;
2
3 void setup() {
4   size(640, 360);
5 }
6
7
8 void draw() {
9   background(0);
10  noStroke();
11  fill(255);
12  ellipse(x, height/2, 50, 50);
13
14  if ( x >= 640 ) {
15    x = x - 7;
16  } else {
17    x = x + 7;
18  }
19  println(x);
20 }
21
22
23
24
25
26
```





Java ▾

sketch 250323h ▾

```
1 float x = 0;
2 float speed = 7;
3
4 void setup() {
5   size(640, 360);
6 }
7
8 void draw() {
9   background(0);
10  noStroke();
11  fill(255);
12  ellipse(x, height/2, 50, 50);
13
14  x = x + speed;
15
16  if ( x >= 640) {
17    speed = -7;
18  }
19  if (x <= 0) {
20    speed = 7;
21  }
22  println(x);
23 }
24
25 // 7, 14, 644, 637, 630
26
27
```





Java ▾

sketch 250323h ▾

```
1 float x = 0;
2 float speed = 7;
3
4 void setup() {
5   size(640, 360);
6 }
7
8 void draw() {
9   background(0);
10  noStroke();
11  fill(255);
12  ellipse(x, height/2, 50, 50);
13
14  x = x + speed;
15
16  if ( x >= 640) {
17    // speed = -7;
18    speed = speed * -1;
19  }
20
21  if (x <= 0) { // 추가.
22    // speed = 7;|
23    speed = speed * -1;
24  }
25
26  println(x);
27 }
28 // 7, 14, 644, 637, 630
29
30
```



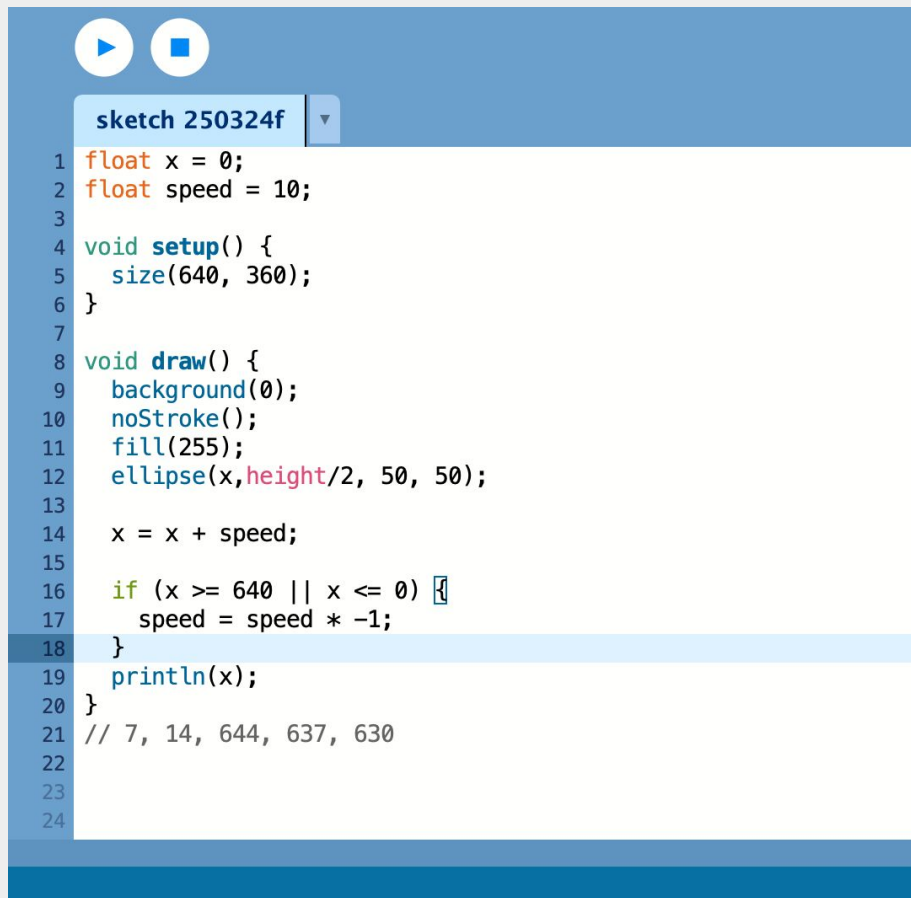


sketch 250323h

```
3
4 void setup() {
5   size(640, 360);
6 }
7
8 void draw() {
9   background(0);
10  noStroke();
11  fill(255);
12  ellipse(x, height/2, 50, 50);
13
14  x = x + speed;
15
16
17  if(x >= 640 || x <= 0) {
18    speed = speed * -1;
19  }
20
21  println(x);
22 }
23 // 7, 14, 644, 637, 630
24
25
26
27
28
29
30
31
32
```

연습문제 :

다음의 코드에서 원에 yspeed를 적용하여
원의 상하움직임의 방향전환을 추가해보세요.
(기존 좌우방향전환에 더하여 결국은
상하좌우로 공이 다 방향전환하게!)



```
sketch 250324f
1 float x = 0;
2 float speed = 10;
3
4 void setup() {
5   size(640, 360);
6 }
7
8 void draw() {
9   background(0);
10  noStroke();
11  fill(255);
12  ellipse(x,height/2, 50, 50);
13
14  x = x + speed;
15
16  if (x >= 640 || x <= 0) {
17    speed = speed * -1;
18  }
19  println(x);
20 }
21 // 7, 14, 644, 637, 630
22
23
24
```

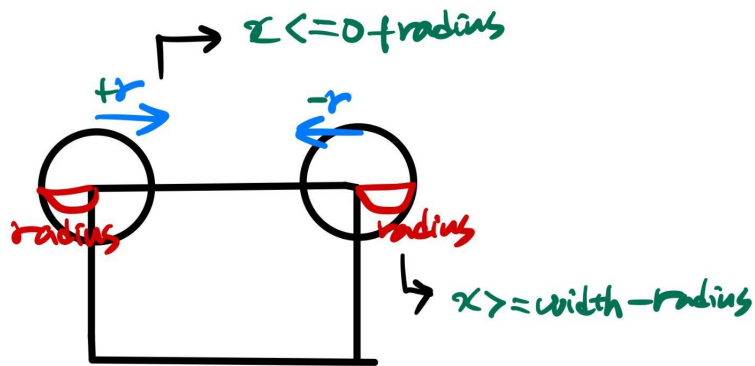
연습문제 :

옆의 그림을 참조하여

앞서 만든 코드에서 원이 완벽하게 가장자리에서 튕기게 해보세요.

(맨 윗줄에 반지름에 대한 변수이름 radius를 만들어서 해보세요.)

```
if(x >= width || x <= 0) {  
    xspeed = xspeed * -1;  
}
```



if($x \geq width$ || $x \leq 0$)

연습문제 :

이전 코드에서 (공 튀기는 예제) 원이 벽에 부딪힐때(좌우벽, 상하벽) 원의 색이 random() 함수를 써서 랜덤하게 바꿔보세요.



Java ▾

sketch 250324c ▾

```
1 float x = 26;
2 float y = 26;
3 float xspeed = 7;
4 float yspeed = 7;
5 float radius = 25;
6
7 void setup() {
8   size(640, 360);
9 }
10
11 void draw() {
12   background(0);
13   noStroke();
14   fill(255);
15   ellipse(x, y, 50, 50); // 움직이는 원
16
17   if (!mousePressed) {
18     x = x + xspeed;
19     y = y + yspeed;
20   }
21   if (x >= width - radius || x <= radius) {
22     xspeed = xspeed * -1;
23   }
24
25   if (y >= height - radius || y <= radius) {
26     yspeed = yspeed * -1;
27   }
28 }
29
30
31
```





Java ▾

sketch 250324c ▾

```
1 float x = 26;
2 float y = 26;
3 float xspeed = 7;
4 float yspeed = 7;
5 float radius = 25;
6 boolean going = false; //추가.
7
8 void setup() {
9   size(640, 360);
10 }
11
12 void draw() {
13   background(0);
14   noStroke();
15   fill(255);
16   ellipse(x, y, 50, 50); // 움직이는 원
17
18   if (going) {
19     x = x + xspeed;
20     y = y + yspeed;
21   }
22   if (x >= width - radius || x <= radius) {
23     xspeed = xspeed * -1;
24   }
25
26   if (y >= height - radius || y <= radius) {
27     yspeed = yspeed * -1;
28   }
29 }
30
31 void mousePressed() {
32   //if (going) {
33   //  going = false;
34   //} else {
35   //  going = true;
36   //}
37
38   going = !going;
39 }
40
```



