Chapter 5

Conditionals

Making Decisions

- If you wish to defrost, press the defrost button; otherwise press the full power button.
- Let the dough rise in a warm place until it has doubled in size.
- If the ball reaches the side of the display change it's direction.

Boolean Expressions

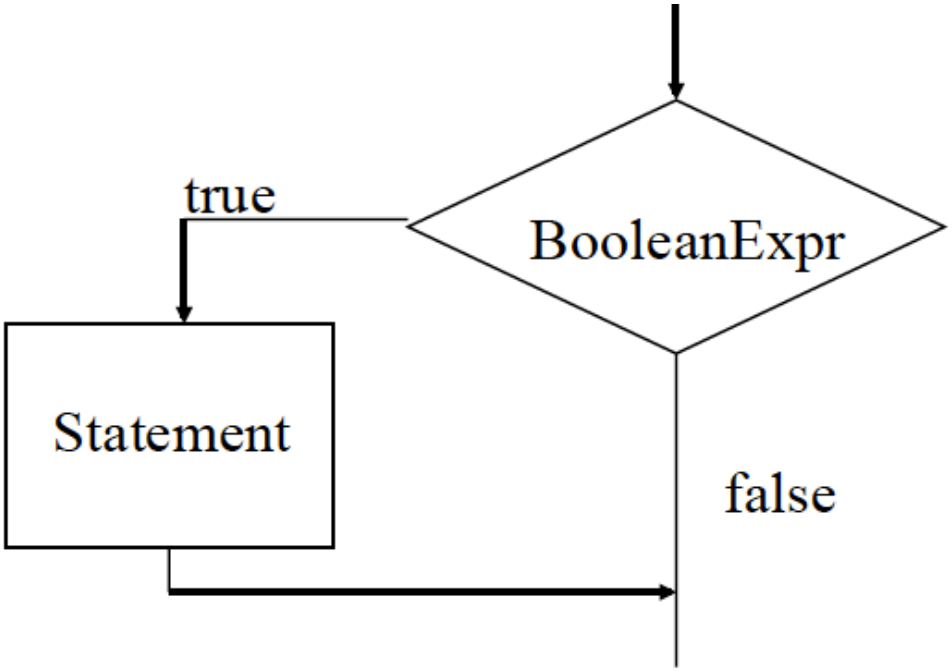
- Any expression that evaluates to true or false.
- Relational operators, `<, <=, >, >=.`
- Equality operators, `==, !=.`
- For example:

```
int i = 3, j = 4;
5 < 6
i == j
(j + 2) <= 6
```

The `if` statement

if (*BooleanExpression*)

Statement



```
int count = 0;

void setup() {
   frameRate(2);
}

void draw() {
   background(120);
   int x = 20, y = 30, size = 40;
   if (count % 2 == 0) {
      ellipseMode(CORNER);
      fill(255, 0, 0);
      rect(x, y, size, size);
      fill(0, 255, 0);
      ellipse(x, y, size, size);
   }
   count = count + 1;
}
```

```
int circleX, circleY, dia = 40;

void setup() {
   circleX = width/2;
   circleY = height/2;
}

void draw() {
   fill(0,255,0);
   if (dist(mouseX, mouseY, circleX, circleY) <= dia/2)
      fill(255,0,0);
   ellipse(circleX, circleY, dia, dia);
}
```

```
int ballX, ballDia = 50;
void setup() {
    size(400,400);
    ballX = -ballDia/2;
}
void draw() {
    background(120);
    if (ballX > width+ballDia/2)
        ballX = -ballDia/2;
    ellipse(ballX, height/2, ballDia, ballDia);
    ballX = ballX + 1;
}
```

A. Ball moves across jumping back to left edge as soon as it touches the right edge.

B. Ball moves across moving off the right edge then moves back in from the left edge.

C. Ball moves across until half way off the right edge (showing just a half circle) then reappears as a half circle on the left edge.

D. Ball moves across then disappears and doesn't come back.

E. Ball moves left to right then right to left after reaching the right edge, and repeats.

Expressions and Statements

- Expression statements are formed by adding a semicolon to the end of certain types of expressions.
- An assignment expression is an expression involving an assignment.

```
area = width * height;
```

- A method call expression has no assignment.

```
rect(…);
```

Non-statements

- Not all expressions can be turned into statements. The following are syntax errors.
  - x+y;
  - width > 20;
- The above do not make sense as statements. They don't DO anything. Statements must DO something.
  - assign a new value to a variable
  - cause some output to occur (println(), rect())
  - change some internal "state" (background(255), noStroke())

Blocks

Several statements can be grouped into a block using { }.

```
{
   int x = 20, y = 30, size = 40;
   ellipseMode(CORNER);
   fill(255, 0, 0);
   rect(x, y, size, size);
   fill(0, 255, 0);
   ellipse(x, y, size, size);
}
// x, y, and size above cannot be used
// here
```

```
int circleX, circleY, dia = 40;
void setup() {
   circleX = width/2;
   circleY = height/2;
}
void draw() {
   background(255);
   fill(0,255,0);
   if (dist(mouseX, mouseY, circleX, circleY) <= dia/2) {
      background(0);
      fill(255,0,0);
   }
   ellipse(circleX, circleY, dia, dia);
}
```

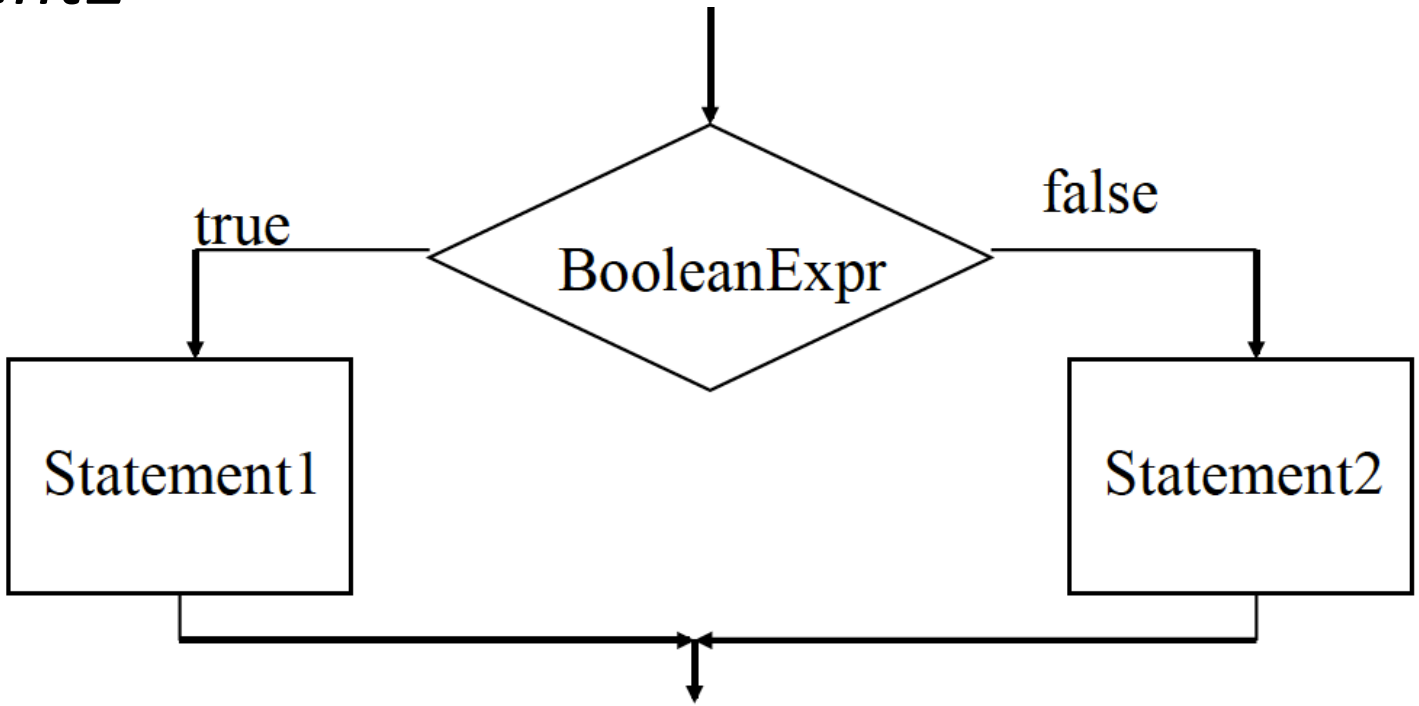
The `if-else` Statement

`if` (*BooleanExpression*)

Statement1

`else`

Statement2



```
//draw replaces what was written in slide 5
void draw() {
   background(120);
   int x = 20, y = 30, size = 40;
   ellipseMode(CORNER);
   if (count % 2 == 0) {
      fill(255, 0, 0);
      rect(x, y, size, size);
      fill(0, 255, 0);
      ellipse(x, y, size, size);
   }
   else {
      fill(0, 255, 0);
      rect(x, y, size, size);
      fill(255, 0, 0);
      ellipse(x, y, size, size);
   }
   count = count + 1;
}
```

Nested if-else Statements

```
if ( BooleanExpression )
   if( BooleanExpression )
      Statement
   else
      Statement
else
   Statement2
```

Nested if-else Statements

```
if ( BooleanExpression )
   Statement1
else
   if ( BooleanExpression )
      Statement
   else
      Statement
```

```
int ballX, ballDia = 50, speed = 1;
void setup() {
    size(400,400);
    ballX = ballDia/2;
}
void draw() {
    background(120);
    if (ballX > width - ballDia/2) {
        speed = -1;
    }
    else _____________________________ {
        speed = 1;
    }
    ellipse(ballX, height/2, ballDia, ballDia);
    ballX = ballX + speed;
}
```

A. Leave it blanks (as is).
B. if (ballX > ballDia/2)
C. if (ballX < ballDia/2)
D. (ballX > ballDia/2)
E. (ballX < ballDia/2)

What goes in the blank so that the ball moves back and forth, reversing direction whenever it reaches the edge?

Semicolons and the if statement

```
if (carX < 0)
   carX = width;
   // draw car
...
```

```
if (carX < 0) {
   carX = width;
   carY = carY + 10;// move down each time
                    // it wraps around
}
```

Semicolons and Common Error

```
void draw() {
   if (carX < 0);
      carX = width;
   // draw car
   ...
}
```

Logical Operators

- Operators that take boolean values as operands.
- x && y - true if x AND y are both true
- x || y - true if either x OR y are true, or both
- !x - true if x is false - read NOT x

Logical Operator Truth Tables

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

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

```
void setup() {
   size(200, 200);
   rectMode(CORNERS);
}
int boxLeft = 50, boxRight = 150,
boxTop = 50, boxBottom = 150;
void draw() {
   if (mouseX > boxLeft && mouseX < boxRight &&
       mouseY > boxTop && mouseY < boxBottom ) {
      fill(255,0,0);
   }
   else {
      fill(0,255,0);
   }
   rect(boxLeft, boxTop, boxRight, boxBottom);
}
```

```
int circleX, circleY, dia = 40;
void setup() {
   circleX = width/2;
   circleY = height/2;
}
void draw() {
   if (_____________________________)
      fill(255,0,0);
   else
      fill(0,255,0);
   ellipse(circleX, circleY, dia, dia);
}
```

Which expression completes this program so that it shows a red circle when the mouse is inside of the circle and a green circle when the mouse is outside of the circle?

A. `dist(mouseX, mouseY, circleX, circleY) <= dia/2`

B. `dist(mouseX, mouseY, circleX, circleY) >= dia/2`

C. `dist(mouseX, mouseY, circleX, circleY) < dia`

D. `dist(mouseX, mouseY, circleX, circleY) > dia`

E. `abs(mouseX-circleX) < dia/2 && abs(mouseY-circleY) < dia/2`

Bouncing Ball

- $pos_{t+1} = pos_t + velocity$
- $velocity_{t+1} = velocity_t + acceleration$
- gravity provides constant acceleration downward

```
float velocity = 2;
float yPos = 0;
int ballRadius = 10;
void draw() {
    background(255);

    // if hit the ground reverse the velocity
    if (yPos > height-ballRadius) {
        velocity = -velocity;
    }

    // adjust position based on velocity
    yPos = yPos + velocity;

    // draw the ball
    ellipse(width/2, yPos, ballRadius*2, ballRadius*2);
}
```

```
float velocity = 2, yPos = 0;
int ballRadius = 10;
float gravity = 0.1;
void draw() {
    background(255);

    // if hit the ground reverse the velocity
    if (yPos > height-ballRadius) {
        velocity = -velocity;
    }

    // adjust position based on velocity
    yPos = yPos + velocity;

    // adjust the velocity - increasing due to gravity
    velocity = velocity + gravity;

    // draw the ball
    ellipse(width/2, yPos, ballRadius*2, ballRadius*2);
}
```

```
void draw() {
   background(255);

   // if hit the ground reverse the velocity
   if (yPos > height-ballRadius) {
      velocity = -velocity;
   }

   // adjust position based on velocity
   yPos = yPos + velocity;

   // adjust the velocity - increasing due to gravity
   velocity = velocity + gravity;

   // add some drag
   velocity = velocity*0.99;

   // draw the ball
   ellipse(width/2, yPos, ballRadius*2, ballRadius*2);
}
```

Recap

- boolean valued expressions using relational operators: <, >, <=, >=, ==, !=
- boolean operators &&, ||, !
- `if` (*booleanExpression*) {

    sequence of statements

  }

  else {

    sequence of statements

  }

Write a program that shows a circle that drops down from the top center of the display until it reaches the exact center and then just stays there.

```
int ballDia = 50, ballY;
void setup() {
   size(400,400);
}
```

On the next slide are three possible draw() methods to go with the above (it wouldn't all fit on one slide). You will be asked which of the draw methods completes the program so that it shows a ball (circle) dropping down from the center of the top until it reaches the center. When it reaches the center it stops and just stays there.

```
void draw() { // CHOICE A
    background(120);
    if (ballY < height/2)
        ballY = ballY + 1;
    ellipse(width/2, ballY, ballDia, ballDia);
}
```

Choice D is All 3 work
Choice E is none work

```
void draw() { // CHOICE B
    background(120);
    if (ballY >= height/2)
        ballY = ballY - 1;
    ballY = ballY + 1;
    ellipse(width/2, ballY, ballDia, ballDia);
}
```

```
void draw() { // CHOICE C
    background(120);
    ballY = ballY + (1-(ballY/(height/2)));
    ellipse(width/2, ballY, ballDia, ballDia);
}
```

```
// a ball bouncing around in a box
float xVel, yVel;
float xPos, yPos;
int ballRadius = 20;

void setup() {
    size(200,200);
    fill(0); // ball will be black
    // starts in the center
    xPos = width/2;
    yPos = height/2;
    // assign small random velocities
    xVel = random(1,3);
    yVel = random(1,3);
}
```

```
void draw() {
   background(255);
   // if hit the top or bottom need to reverse
   if (yPos > height-ballRadius || yPos < ballRadius) {
      yVel = -yVel;
   }

   // check if hit the left or right edge
   if (xPos > width-ballRadius || xPos < ballRadius) {
      xVel = -xVel;
   }

   // adjust position based on velocity
   xPos = xPos + xVel;
   yPos = yPos + yVel;
   // draw the ball
   ellipse(xPos, yPos, ballRadius*2, ballRadius*2);
}
```

Make the necessary additions to the ball in a box program so that if the user clicks the mouse on the ball it changes to a new random color.

```
int ballX, ballDia = 50, ballSpeed = 1;
void setup() {
    size(400,400);
    ballX = ballDia/2;
}
void draw() {
    background(120);
    if (ballX > width - ballDia/2 || ballX < ballDia/2)
        ballSpeed = -ballSpeed;
    ellipse(ballX, height/2, ballDia, ballDia);
    ballX = ballX + ballSpeed;
}
```

A. Ball moves across jumping back to left edge as soon as it touches the right edge.

B. Ball moves across moving off the right edge then moves back in from the left edge.

C. Ball moves across until half way off the right edge (showing just a half circle) then reappears as a half circle on the left edge.

D. Ball moves across then disappears and doesn't come back.

E. Ball moves left to right then right to left after reaching the right edge, and repeats.