What will we be learning in this lesson?

- We'll be introduced to `if()` statements and how to use them in our programs.
- See our second variable type, `boolean`.
- Build our first interactive program, where things change depending on the inputs of the user. Using four new variables, `mousePressed`, `mouseButton`, `keyPressed` and `key`.

Conditions:

In coding, we often want to test to see if certain conditions have been met. For example, in a computer game, the player's gun won't shoot unless the player presses the button that is associated with "fire." Most programs will have lots of different sections to their code, that can only be run by the computer **if** certain things happen.

The main way that programmers use to direct a computer to run a certain section of code is using `if()` statements.

NB: We recommend you open a new processing file and name it, LessonFourIfStatement.

`if()` statements:

`if()` statements are quite simple to understand. If the conditions inside the statement is true, the computer will run the code that is associated with that if statement. If the condition isn't true, it will simply skip the code associated with the if statement.

An important aspect of `if()` statements is that they do not use semicolons (**;**). Instead**, after an if statement you create a new code block using { and }, like** `setup()` **and** `draw()`. Also, you use two equal signs (**==**) inside an if statement, as this is the computers' equal sign, also known as **Logical Equal**.

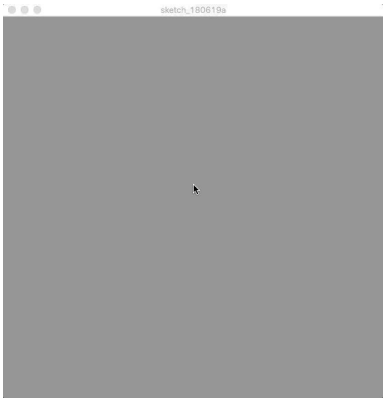
Now, let's use our first `if()` statement:

Task:

1. Write the following code and run it. When you click the mouse, a circle should appear on the screen:

```
void setup(){
 size(600, 600);
}

void draw(){
  background(150);
  if(mousePressed == true){
    ellipse(300, 300, 200, 200);
  }
}
```



2. Add a second `if()` statement to test **which** mouse button is being pressed. ***Hint**: The part inside the brackets will be `mouseButton == LEFT`.*

3. Edit the program so that if you press a key, you change the colour of the circle. ***Hint**: use `keyPressed` this time.*

What is happening here is that the computer is testing to see if the mouse has be pressed. If it has, it draws the circle. If it hasn't been clicked, no circle is drawn. The background covers any circle that have been drawn previously, so **the circle will only show when holding down the mouse.** The only aspect of the if statement that is completely unfamiliar is the part inside the brackets, `mousePressed == true`. This is because this part of the code uses a variable type that we haven't seen before. This variable type is known as a `boolean`.

Booleans:

A `boolean` is a type of variable that only has two possible values, true or false. We use them to say "has something happened?" or "do we want something to happen?". Creating a `boolean` is very similar to creating a `float`, the main difference being that now we are using `true` or `false` instead of a number. An example would be: `boolean isThisABoolean = true;`

We often use booleans and if statement together to create more complicated programs. Let's try it now:

NB: We recommend you open a new processing file and name it, LessonFourIfBooleans.

Copy this code and run it. Can you explain what will happen when you run it?

```
boolean moveBallLeft = false;
int circleHozPos = 0;

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

void draw(){
  fill(255);
  ellipse(circleHozPos, 200, 200, 200);
  if(circleHozPos >= 500){
    moveBallLeft = true;
  }
  if(moveBallLeft == true){
    circleHozPos = circleHozPos - 3;
  }
  if(moveBallLeft == false){
    circleHozPos = circleHozPos + 3;
  }
}
```

Task:

1. Change the program so only one ball can be seen at a time.

2. Make the ball bounce off the left hand side as well. You'll need to know at what point the ball hits the left hand side, create an `if()` and change the `boolean`.

3. Make the ball change direction when the **edges** of the ball hit the side of the screen, **not the middle.**

Bonus task:

Make a rectangle bounce off the top and bottom of the screen. It should have the same elements as the circle. Also, each time it bounces off the bottom, it's speed should go down by one. (You'll need a speed variable for this).

Testing the user input in more detail:

We've seen that we can use `mousePressed`, `mouseButton` and `keyPressed` to test the inputs of the users. `mousePressed` and `keyPressed` test to see if **any** mouse button / keys have been pressed, while `mouseButton` tests a **specific** mouse button. There is also something similar for `keyPressed`, known as `key`. To test if the user pressed the 'a' key, you would write this:

```
if (key == 'a') {
  // do something here
}
```

NB: Create a new file for this next task.

Task:

1. Use `fill()` to colour the circle. You should have 3 unique colours that can be showed, only if an unique key is pressed.

2. Be able to change the background to a random colour by pressing a specific key.

Questions for the end of the lesson?

- How many unique values can a `boolean` have?
- What symbols does the `if()` statement use to start a code block?
- Which variable to do use to test if 'R' has been pressed?
- What is the logical / computer symbol for equals?