Lesson 3: Program Flow

Lesson aim:

In this lesson we are going to learn how computers read the code that we write.

Why:

To write good programs we need to ensure that the computer understands our code the way we mean it to be understood. The order in which we write lines of code is important, much like in a novel, where the order of words on the page and pages in the book makes a big difference to the storyline!

The way computers 'read' code

Programs are lists of instructions to the computer.

Now we have written our first program we are going to talk about the order in which the computer runs our instructions.

In the example we've seen it's very simple: it reads **top-to-bottom**, just like we do.
Each line is a separate instruction to the computer, and each instruction is ended with a semicolon (";").

```
size(400,400);
ellipse(100,100,150,150);
```

1. First the computer will set the size of the display window (also called the canvas).
2. Then the computer will draw an ellipse.

We've made programs with one ellipse. Is that all we can do? NO!

We can **add as many instructions as we want**:

```
size(400,400);
ellipse(100,100,150,150);
ellipse(300,300,200,200);
```

1. Set window size
2. draw a circle - top left
3. draw a bigger circle, bottom right.

Try this out.

Task:

- Make a program that draws 8 different circles. Make some big, some small, some wide and some tall.
    - If you think you have drawn 8, but you can't see them all on-screen and you don't know why, move on to the next section.

Order of instructions

Processing executes instructions as it gets them. For example, instruction 1 happens before instruction 2. We call this **program flow.**

This matters when we try and draw **circles that overlap**. (Did you see this in the last task?) Enter this code in a new window:

```
size(400,400);
ellipse(200,200,250,250);
ellipse(200,200,300,300);
```

What do you see? Are there two circles? Why not? Try to figure out what has happened here before moving on.

We can only see the last, bigger circle when we run that code. Lets change the order of those two circles, what do we think will happen?

```
size(400,400);          //←instruction number 1 (setup window size)
ellipse(200,200,300,300); // ←instruction number 2 (draw a big circle, middle)
ellipse(200,200,250,250); // ←instruction number 3 (draw a smaller circle, middle)
```

Instead of drawing the big circle over the small circle we're now drawing the small circle over the big one - we can see both!

Task:

- Add 3 more circles to this program, each one smaller than the last, to make a target shape (seen below left).
- Extend this program to make two different targets, at different points on the screen. You should be aiming for something like the result seen below right.