Lesson 2: User Input

Mouse User Input

"User Input" lets us take instructions from the user - if you use the keyboard or mouse to do anything on a computer, that's user input! We're going to start with the mouse here.

Remember that the first two numbers in the ellipse function are the *position* of the centre of the ellipse (**distance from the left** and **distance from the top** of the screen). The numbers we put here tell the computer where to draw the ellipse.

We can also use *variables* to tell the computer where to draw the circle. These are like boxes with numbers in them - we tell the computer the name of the box and the computer uses whatever number is in the box to draw our circle.

Two good examples of variables are `mouseX` and `mouseY`. These are the horizontal (`mouseX`) and vertical (`mouseY`) co-ordinates of the mouse. Look at the code below and guess where the ellipse is going to appear when we run the code.

Type code on this side for "Task 2" below	**Code on this side is from Lesson 1 (for comparison)**

`void setup(){`
`  size(600,600);`
`}`
 	
`void draw(){`
`  ellipse(mouseX, mouseY, 100, 100);`
`}`
	`size(200,200);`
	
	`ellipse(100,100,150,150);`

Tasks (for each one, ask: what happens when I move the mouse right? What about when I move it down?)

1. Make a new program and save it: the file name **must** start with "**Lesson2**".
2. Copy the above code on the **left** exactly as it appears above.
3. Run the new program. What happens when you move the mouse around the window?
4. Replace `mouseX` with a number (say, 200). What happens now? Why is that?
5. Put `mouseX` back where it was, and now replace `mouseY` with a number. What happens? Is it different to what happened in (4)? Why?
6. For fun, switch `mouseX` and `mouseY` (so it's now "`ellipse(mouseY, mouseX, 100, 100);`"). What happens now? Why?
7. Now try having `mouseX` twice: eg "`ellipse(mouseX, mouseX, 100, 100);`". How does the circle move? Why? Try with two `mouseY`s.