

Creative coding – Challenge

Warm up

Partner A: You are describing. Think of a simple character using shapes like a triangle for the body, add 3 round eyes etc You need to describe the drawing in simple steps to be followed by your partner. They need to be clear and precise.

Partner B: You are drawing. By following **only** the instructions you receive. You cannot think and add in detail, follow exactly what you have been told to draw.

Describing steps	Following steps

Challenge 1 - Animate Simple Shapes

The program needs to create a circle that moves from the left to the right.

- Inside the `draw()` function:

Set the background	Use <code>background(220)</code> Setting the background to grey. 0 is black moving through to 255 for white.
Create a circle	Add <code>ellipse(x, 200, 50, 50);</code> X is the position stored in the variable, 200 is the height. 50 x 50 is the size of the circle.
Set the movement	Add <code>x = x + 2;</code> Every frame, this will move the circle 2 pixels to the right

Extension: Use the `fill()` function to randomly create a sparkle effect in the draw function.

Challenge 2: Make it interactive

Create a program to create a brush that follows the mouse but only paints when the user clicks. The click will generate a circle to represent a star.

Activity steps:

Set the Canvas:	Inside <code>setup()</code> Use <code>createCanvas(windowWidth, windowHeight);</code> Add a <code>background(0);</code> here so the trail stays on the screen.
Create the draw function	Add <code>draw() {</code>
Add the condition	Inside the draw function add <code>if (mouseIsPressed) {</code> Remember the indentation
Follow the Cursor:	Add <code>fill(random(150, 255), 50, random(200, 255), 150);</code> This changes the colour gradient and transparency each time it is drawn. Add <code>noStroke();</code> Add <code>let circleSize = random(5, 25);</code> This is used to change the size of the circle each time it is drawn Add <code>ellipse(mouseX, mouseY, circleSize, circleSize);</code> This will use the mouse position and the circle size to define the position and size of the circle drawn
Close brackets	<code>}</code> <code>}</code> If you have opened, remember to close brackets
Clear the canvas	Add a function to clear the screen when a key is pressed, <code>function keyPressed() {</code> <code>background(0);</code> <code>}</code>

Extension: Can you change the size range for the circles drawn?

Challenge 3: The Debugging Task

1. What will happen when this program code is run?

```
function setup() {  
  createCanvas(400, 400)  
}  
  
function draw() {  
  background(0);  
  fill(255, 0 0);  
  ellipse(200, 200, size, size);  
}
```

2. Run the program, identify the errors and fix them.

Extension: The circle was supposed to be yellow. Explain why this has not been highlighted as an error when the program is run. Change the code to output a yellow circle.

Challenge 4: The client brief

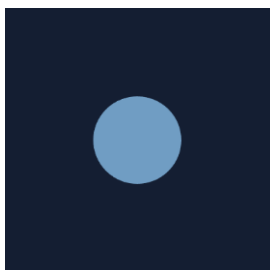
You have been commissioned to create a calming digital art piece for a "Wellbeing App." The client wants a flowing, river-like pattern that subtly changes colour as it moves, like light reflecting off water. They also want it to respond to the mouse position, creating a ripple effect.

Activity steps:

Set up	Create your <code>setup()</code> and <code>draw()</code> functions. Inside <code>setup()</code> use <code>createCanvas(600, 400);</code> and <code>background(20, 20, 80);</code> (a deep blue). Use <code>noStroke();</code> Remember to close the bracket on the function }
Make it fade	Inside <code>draw()</code> Add <code>background(20, 20, 80, 5);</code> The 5 makes it very transparent, creating the fade.
Create many waves with a for loop	Inside <code>draw()</code> Add <code>for (let i = 0; i < 50; i++) {</code> This means whatever is inside the curly brackets will run 50 times!
Position the waves	Inside the for loop, you need two new variables for position: 1. <code>let xWave = (frameCount * 2 + i * 10 + mouseX) % width;</code> This line makes the waves move, spread out, and ripple with the mouse. The % width makes the wave wrap back to the start of the screen. 2. <code>let yWave = height / 2 + sin(xWave * 0.05 + frameCount * 0.1) * (i * 2);</code> This creates the up-and-down wavy motion, spreading out from the centre.
Colour the waves	Still inside the for loop, set a colour. Try <code>fill(50, 150 + i * 2, 255, 150);</code> This makes the waves gradually brighter.
Draw the waves	Still inside the for loop, Use <code>ellipse(xWave, yWave, i * 0.5 + 5, i * 0.5 + 5);</code> The <code>i * 0.5 + 5</code> makes the waves slightly different sizes.
Close brackets	<code>}</code> <code>}</code> If you have opened, remember to close brackets

Extension – The breathing circle

- Create a canvas 400 x 400.
- On the canvas create a breathing circle.
- Use the code you have used for background, fill, ellipse.



Add into the draw function this line of code to create the breathing illusion:

```
let s = sin(frameCount * 0.05) * 50 + 100;
```

What can you add or edit to create your own version?

Reflection

Use the space below to reflect on what you have learnt from these activities.

- What surprised you?
- What effect are you happiest with?
- What would you do next if you had more time?