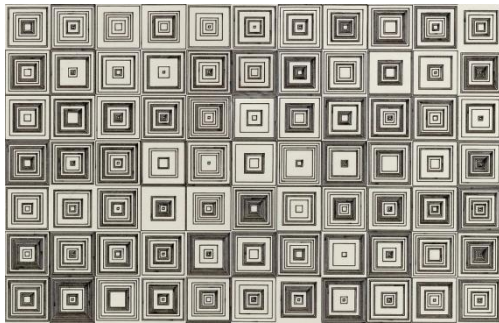


# Creative coding – pre-reading

## Generative Art



In traditional art, you draw every line. In generative art, you write a system or a set of rules (an algorithm) that creates the art for you. Imagine an artist like *Vera Molnár*. She used code to create thousands of variations of a single geometric pattern. You don't just create one piece of art; you create a "machine" that produces infinite art. This is how modern video games like *Minecraft* or *No Man's Sky* generate entire infinite worlds—no human could draw it all, so the code does it.

<https://www.youtube.com/watch?v=8tNESHtkr0>

## Fashion & wearables



Coding is moving from the screen to the body. In **digital fashion**, designers now create 3D "skins" for avatars or AR filters. This reduces waste because no physical fabric is used during the design phase. With **generative patterns**, brands like *Iris van Herpen* use algorithms to model how water flows or birds fly, then 3D print those shapes into dresses that would be impossible to sew by hand.

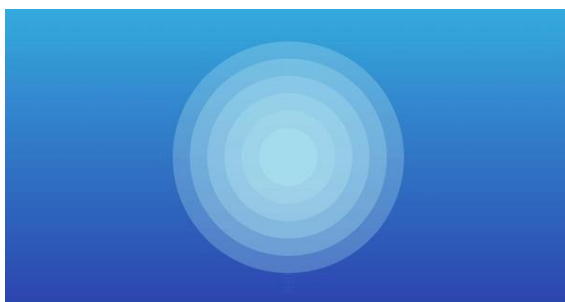


Algorithms can also be used to 3D print intricate jewellery or create AR (augmented reality) "try-on" filters for brands like Sephora or Gucci. The code has to track your face/body in real-time (input) and map a digital 3D object onto it (output).

<https://youtu.be/Oyc77c6hKx0>

## Tech for Good: Eco-visualisation

Creative coders take complex data, such as live air pollution levels in London, and turn it into something beautiful. Like a digital flower on your phone that glows bright green when the air is clean but wilts and turns grey when pollution rises. This uses conditionals (*if pollution > x, then change colour*). Case study - <https://www.youtube.com/watch?v=bmjuDK1jSQE>



## Tech for Good: Calm Tech

In a world of stressful notifications, we can code for "wellbeing." For example: Generative "breathing" circles that pulse at a specific frequency to help users regulate heart rates, or apps that generate unique, soothing ambient music that never repeats.

<https://www.youtube.com/watch?v=5-ttqEsf518>

## Interactive installations: The "Lived" experience



Code that reacts to you. For example: The *Rain Room* (Random International). It's a room where it's always raining, but because of sensors and code, the rain stops wherever a person is standing. You walk through a rainstorm without getting wet.

<https://youtu.be/FslABAyj2OA>

## Creative coding cheat sheet – Javascript

The Structure (The "Skeleton") - Every sketch needs these two blocks to live.

| Code                                  | What it does                             | Real-world Analogy                     |
|---------------------------------------|------------------------------------------|----------------------------------------|
| <code>function setup() { ... }</code> | Runs <b>once</b> at the very start.      | Prepping your canvas and paints.       |
| <code>function draw() { ... }</code>  | Runs <b>forever</b> (60 times a second). | The "flipbook" that creates animation. |

### Key areas for the canvas

| Code                                 | What it does                                                                                                         |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| <code>createCanvas(600, 400);</code> | Set the canvas area in pixels                                                                                        |
| <code>background(255)</code>         | Background colour of the canvas from 0-255. 0 = black and 255 = white. In between is a gradient from black to white. |

Shapes & Drawing (The "Brushes") Computers draw from the top-left corner (0,0).

| Code                                       | What it does                   |
|--------------------------------------------|--------------------------------|
| <code>ellipse(x, y, width, height);</code> | Circle                         |
| <code>rect(x, y, width, height);</code>    | Square                         |
| <code>line(x1, y1, x2, y2);</code>         | Line                           |
| <code>fill(red, green, blue);</code>       | Colour - Numbers from 0 to 255 |
| <code>stroke(red, green, blue);</code>     | Outline                        |
| <code>noStroke();</code>                   | No outline                     |

### Motion & Interaction (The "Magic")

Use these words inside your draw() loop to make the art move or react.

| Code                          | What it does                                                        |
|-------------------------------|---------------------------------------------------------------------|
| <code>mouseX, mouseY</code>   | Follows your cursor position.                                       |
| <code>pmouseX, pmouseY</code> | Where the mouse was <i>one frame ago</i> (great for drawing lines). |
| <code>random(min, max)</code> | Picks a random number. Use for "glitch" or "sparkle" colours.       |
| <code>frameCount</code>       | A number that keeps getting bigger. Use it for constant movement.   |

### Programming Concepts (The "Logic")

Variables (Storing Information) - Use these to "remember" numbers like speed, size, or position.

| Code                          | What it does                             |
|-------------------------------|------------------------------------------|
| <code>let xPos = 0;</code>    | Create it at the very top of your code   |
| <code>xPos = xPos + 1;</code> | Put this inside draw to make things move |

## Conditionals (Making Decisions)

| Code                                                                                                                                          | What it does                                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <pre>if (mouseIsPressed) {<br/>  fill(255, 0, 0); // Turn RED if clicking<br/>} else {<br/>  fill(0, 0, 255); // Stay BLUE if not<br/>}</pre> | The "If-Then" logic that makes art interactive.                                             |
| <pre>for (let i = 0; i &lt; 10; i++) {<br/>  ellipse(i * 40, 200, 20, 20);<br/>}</pre>                                                        | Instead of drawing 10 circles, tell the computer to repeat.<br>// Draws a row of 10 circles |

If your code isn't working, check these top 3 errors:

1. **The case sensitive trap:** background works, but BackGround will break.
2. **The missing punctuation:** Did you forget a semicolon ; or a closing bracket }?
3. **The "invisible" Sshape:** Is your fill the same colour as your background? Is your shape drawing off the edge of the screen?