Projection 1

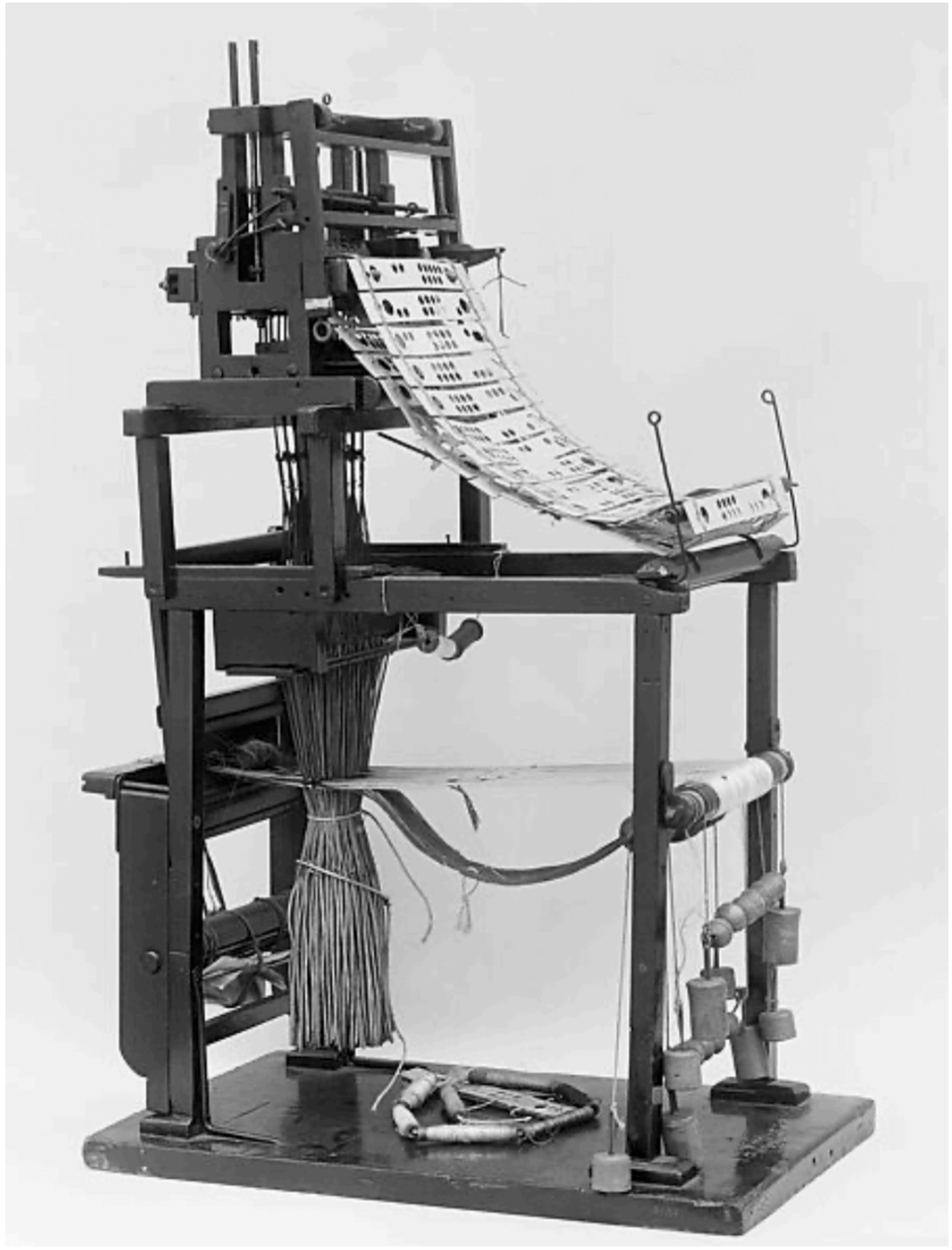
THE BACK REVEALS THE SYSTEM

TEXTILES AND COMPUTATION

My project explores the relationship between textiles and computers.

Historically, there is a strong link between the computer and textiles. The development of the computer can be traced back to textile logic, most clearly expressed through the Jacquard loom using binary punched cards to automatically control woven patterns.

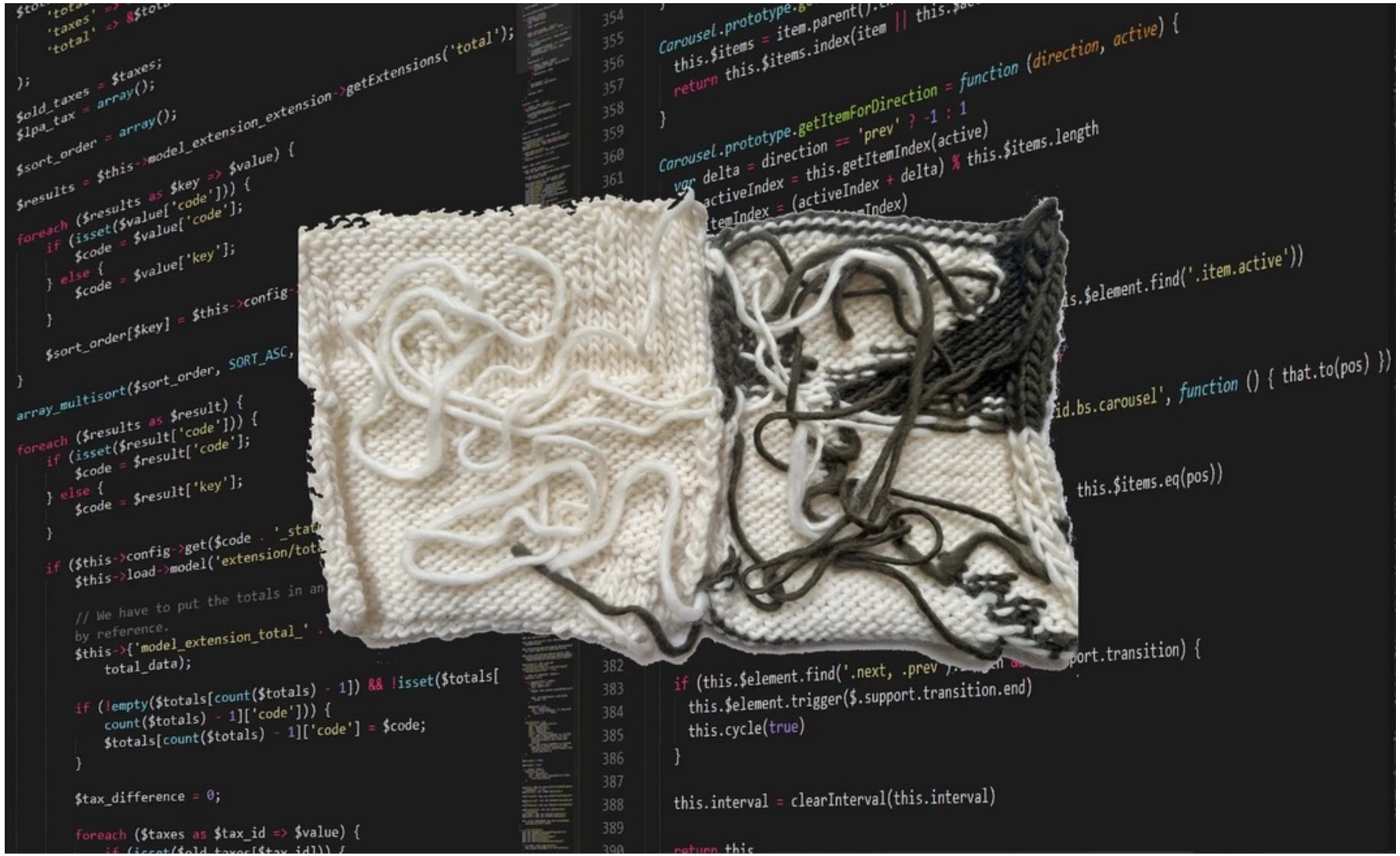
(More on this in projection 2)



THE REVERSE SIDE:
REVEALING THE SYSTEM

In both knitting and programming, attention is usually placed on the front and the final outcome. We are intended to see only the surface, while the knots, errors, and processes remain hidden on the back.

This project shifts the focus to the reverse side - the backend - to reveal the system and highlight the similarities between textile processes and coding.



HIDDEN BACKEND SYSTEMS

Coding languages operate as backend systems, running behind the scenes as hidden structures that determine how a website behaves.

Python control words act as instructions that guide and control the program's logic and actions.

`if` → do something *if* a condition is true

`else` → otherwise do this

`elif` → else-if (another condition)

`for` → repeat something a certain number of times

`while` → repeat while something is true

`break` → stop a loop

`continue` → skip to next loop step

CODING LANGUAGE VS KNITTING LANGUAGE

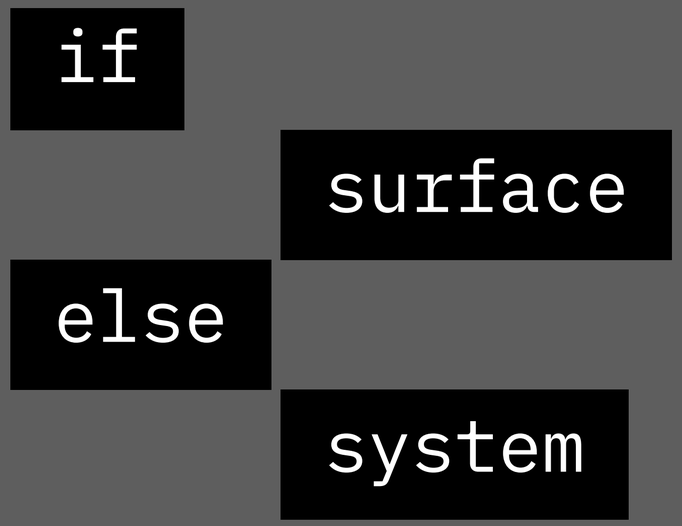
Comparing knitting instructions with Python functions reveals a shared logic. By focusing on the backend of each system, similarities emerge in their languages of instruction, structure, and process. Both rely on loops, repetition, abbreviations, and mathematical logic to produce an outcome.

```
(A) = colour A

(B) = colour B

K = knit
P = purl

** repeat from * to *

Cast on 67 stitches (A)

Row 1: P (A)

Row 2 (K): 3 (A), 3 (B), 5 (A), 2 (B), 3 (A), 2 (B), 1 (A),
4 (B), 9 (A), 4 (b), 9 (A), 8 (B), 6 (A), 6 (B), 2 (A).

Row 3 (P): 2 (A), 8 (B), 4 (A), 8 (B), 7 (A), 8 (B), 5 (A),
9 (B), 3 (A), 2 (B), 4 (A), 3 (B), 4 (A)

Row 4 (K): 5 (A), 3 (B), 3 (A), 2 (B), repeat ** twice * 3
(A), 3 (B), 4 (A), 3 (B), 9 (A), 2 (B), 7 (A), 2 (B), 4 (A),
2 (B), 2 (A)
```

```
# Definitions
A = "A"   # Colour A
B = "B"   # Colour B

K = "knit"
P = "purl"

# Cast on
pattern = {
    "cast_on": {
        "stitches": 67,
        "colour": A
    },

    "rows": {

        1: {
            "stitch": P,
            "sequence": [
                (67, A)
            ]
        },

        2: {
            "stitch": K,
            "sequence": [
                (3, A), (3, B), (5, A), (2, B), (3, A), (2, B), (1, A),
                (4, B), (9, A), (4, B), (9, A), (8, B), (6, A), (6, B), (2,
            ]
        },

        3: {
            "stitch": P,
            "sequence": [
                (2, A), (8, B), (4, A), (8, B), (7, A), (8, B),
                (5, A), (9, B), (3, A), (2, B), (4, A), (3, B), (4, A)
            ]
        },

        4: {
            "stitch": K,
            "sequence": [
                (5, A), (3, B), (3, A), (2, B),

                # repeat section
                {"repeat": 2,
                 "pattern": [
                     (3, A), (3, B), (4, A), (3, B)
                 ]
                },

                (9, A), (2, B), (7, A), (2, B), (4, A), (2, B), (2, A)
            ]
        }
    }
}
```

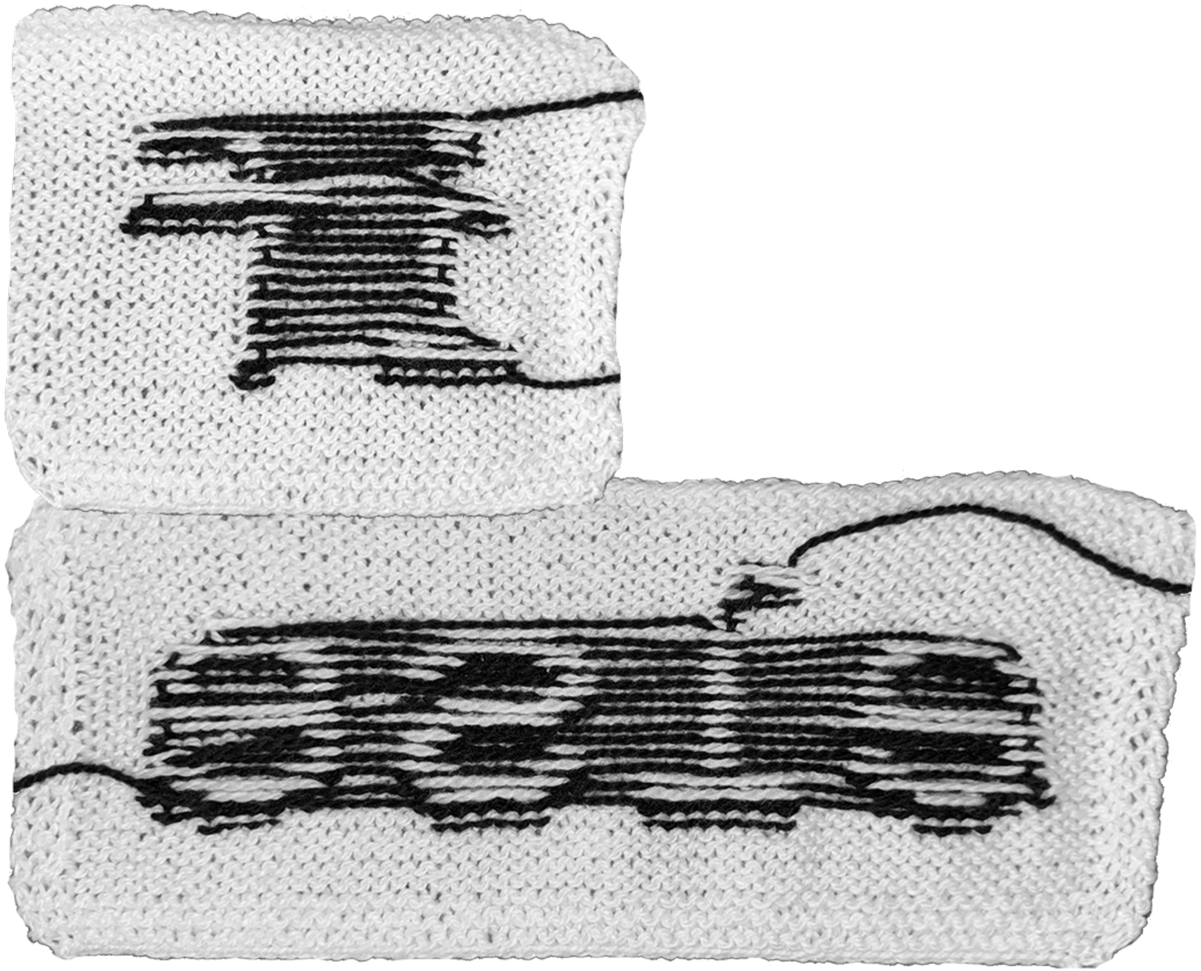
ENQUIRY

How can shifting attention from the front end to the backend reveal the usually hidden systems and the shared logic that shapes both textiles and coding?


if
surface
else
system

MATERIALISING THE INVISIBLE CODE

I took something purely digital and usually invisible, Python control words, and made it tangible by hand-knitting them into a physical form.



IF / ELSE:
FRONT AND BACK LOGIC

The knitted “if” is seen the front and the “else” from the back, reinforcing the idea that the front surface represents the outcome of a system, while the back reveals it.

The reverse side exposes the underlying threads, showing how the two colours are constructed and knitted to produce the visual outcome on the front.