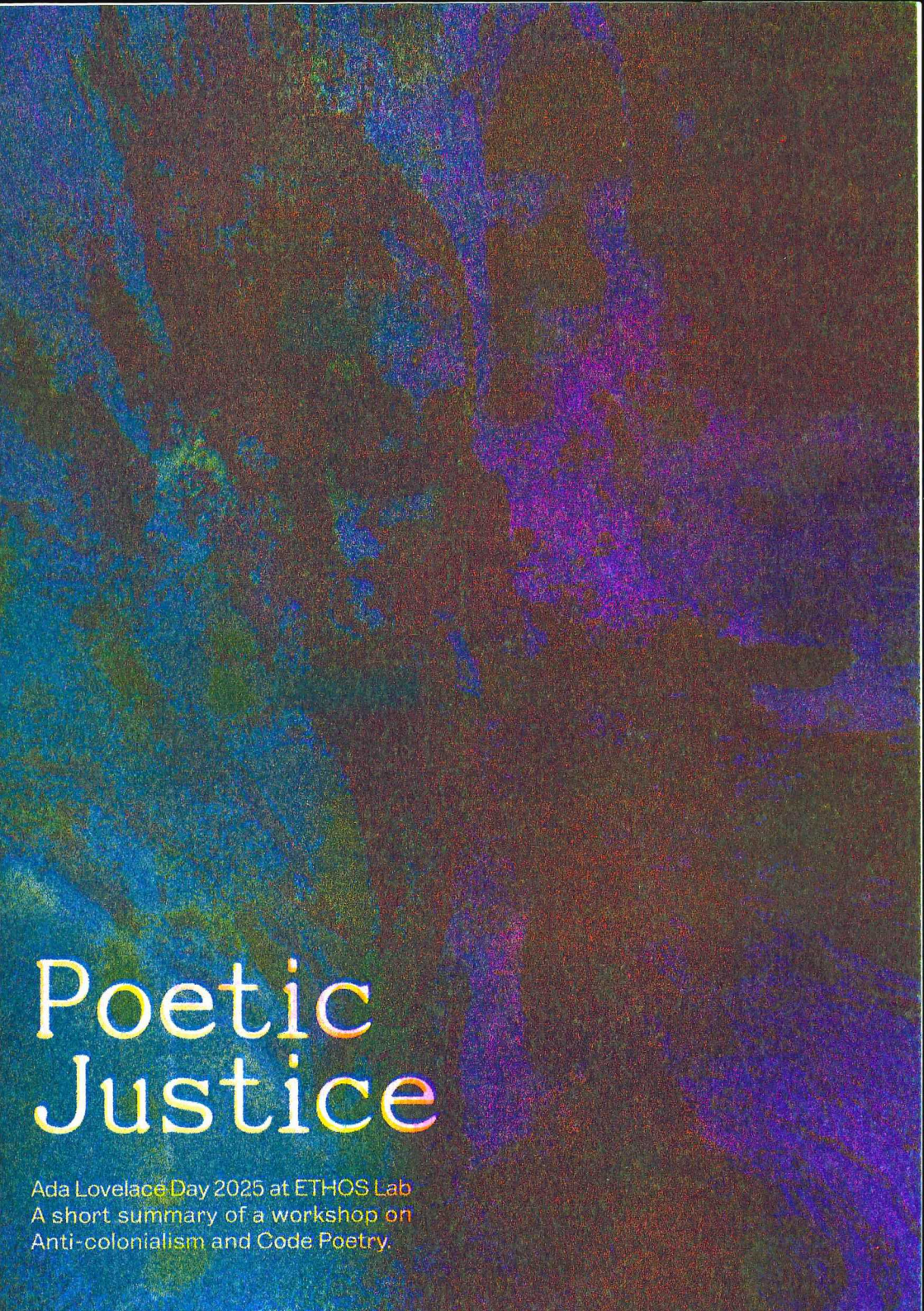
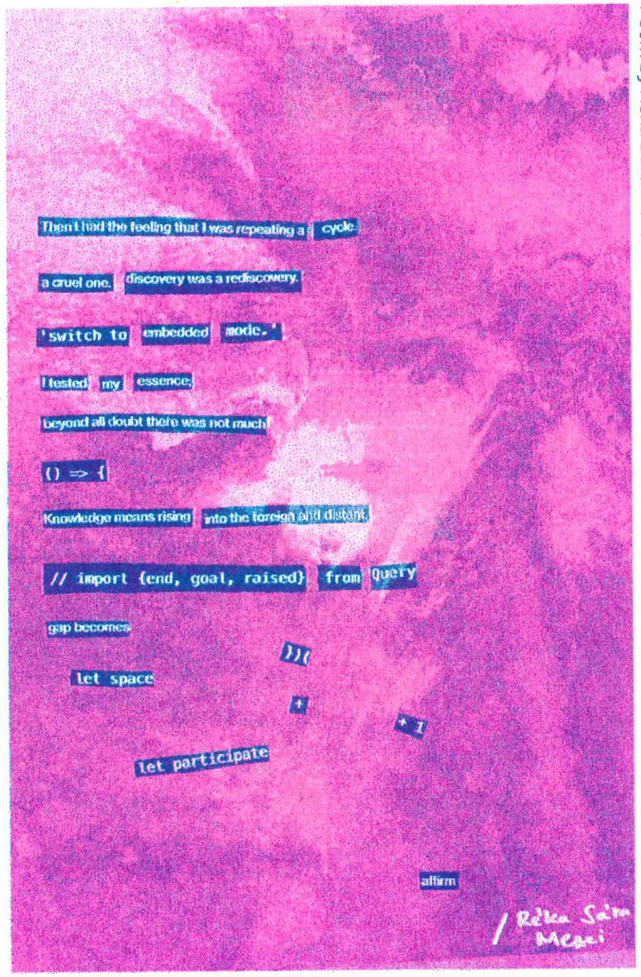
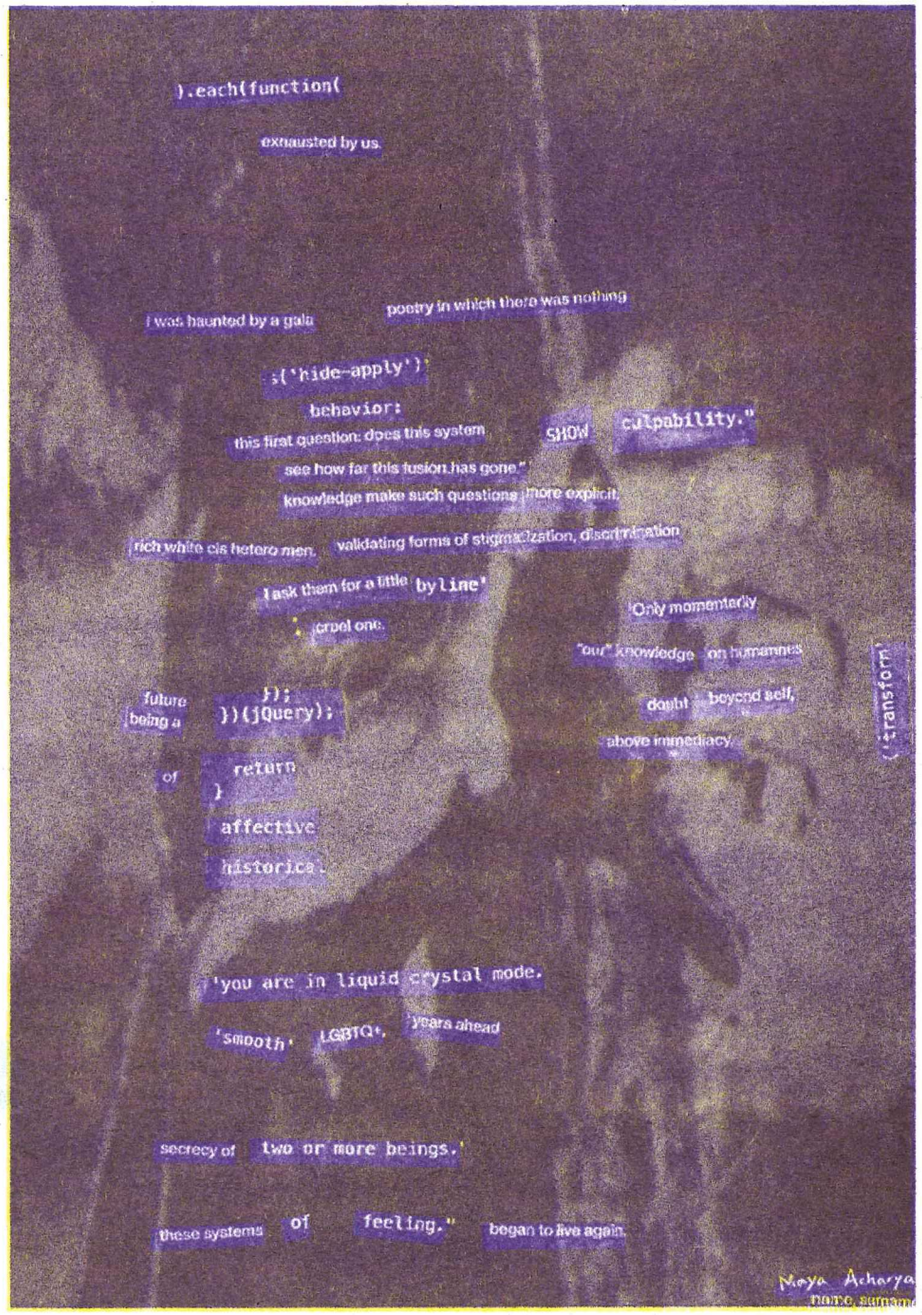


Poetic Justice

Ada Lovelace Day 2025 at ETHOS Lab
A short summary of a workshop on
Anti-colonialism and Code Poetry.



Poem by Réka Sára Mezei

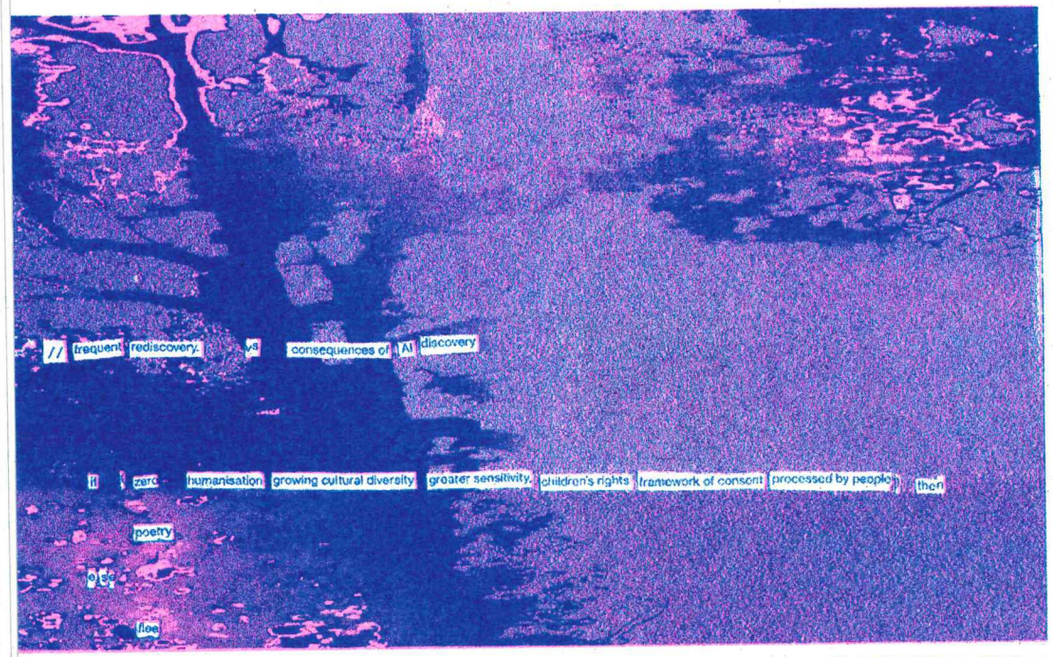


Poem by Maya Acharya

"If you can't give me poetry, can't you give me poetical science?"

Ada Lovelace, in a letter to her mother.

Contents

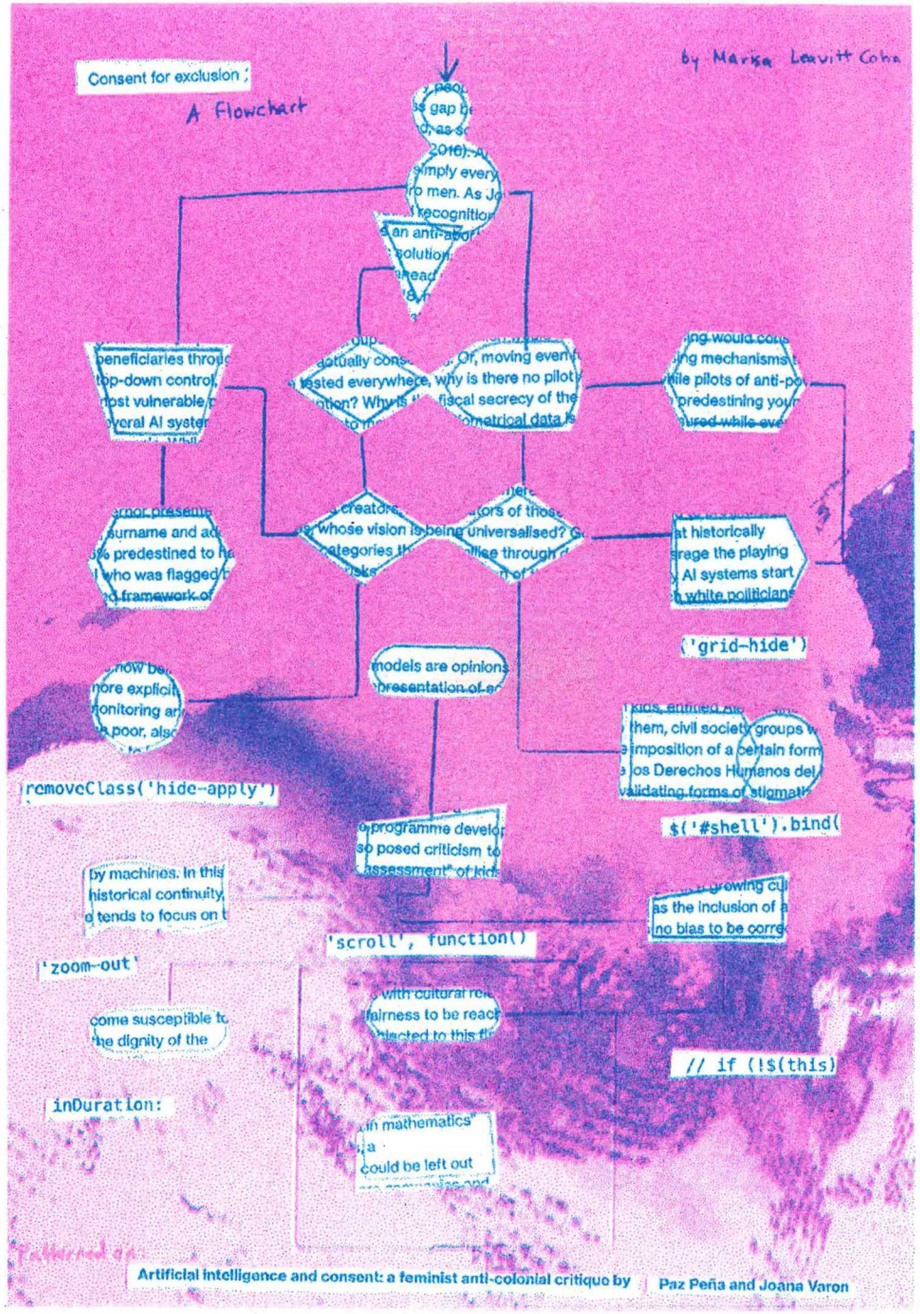


Poem by Hanna Stokes

Thanks to Nadia from Comfycakes who hooked up with delicious cupcakes. Thanks to all participants. Thanks to Xinyi at The Print house DK and of course ETHOS lab and staff.

All poems where created during the workshop by the participants from extracts of text from Fanon's Black skin, white masks, Said's Orientalism. and an article from Varon's & Peña's article "*Artificial intelligence and consent: A feminist anti-colonial critique*".

Afterwards the poems where edited with node-based, visual programming language (touch designer).



Poem by Marisa Levitt Cohn

Introduction

October 14 – Ada Lovelace Day – is celebrated worldwide. Its purpose is to encourage greater participation in the fields of science, technology, engineering, and mathematics (STEM) among women, but also other gender-nonconforming and marginalised groups within STEM domains. In 2025, I, Isabel Aguilar Johansson, a visual artist, was invited to ETHOS Lab to host a code poetry workshop.

ETHOS Lab is a place for critical methods and feminist computing at the IT University of Copenhagen. At ETHOS Lab, Ada Lovelace day has become a tradition of gathering, thinking, cutting, pasting, programming, deprogramming and, of course, eating cake.

In this short zine, I want to introduce you to a small part of the poetic world of code. You'll read about how engaging with code critically can help us challenge colonial biases in programming and, ultimately, in ourselves.

I'll also introduce a few terms used in academia (even though the thing I love about this art form is its inherent contradiction of resisting definition),

Who was Ada Lovelace?

Ada Lovelace was a British mathematician and writer. Described by scholars as the first person to write what we now call a

computer algorithm.
computer algorithm.
computer algorithm.
computer algorithm.
computer algorithm.

computer algorithm.

computer algorithm

Lovelace is often referred to as the first person to truly understand what a computer could become. In the early 1800s, when most saw machines as tools for numerical calculation, she argued that they could manipulate any symbolic structure like music, art, language, and abstract patterns of thought.

computer algorithm

computer algorithm

computer algorithm

Dismissed for a century

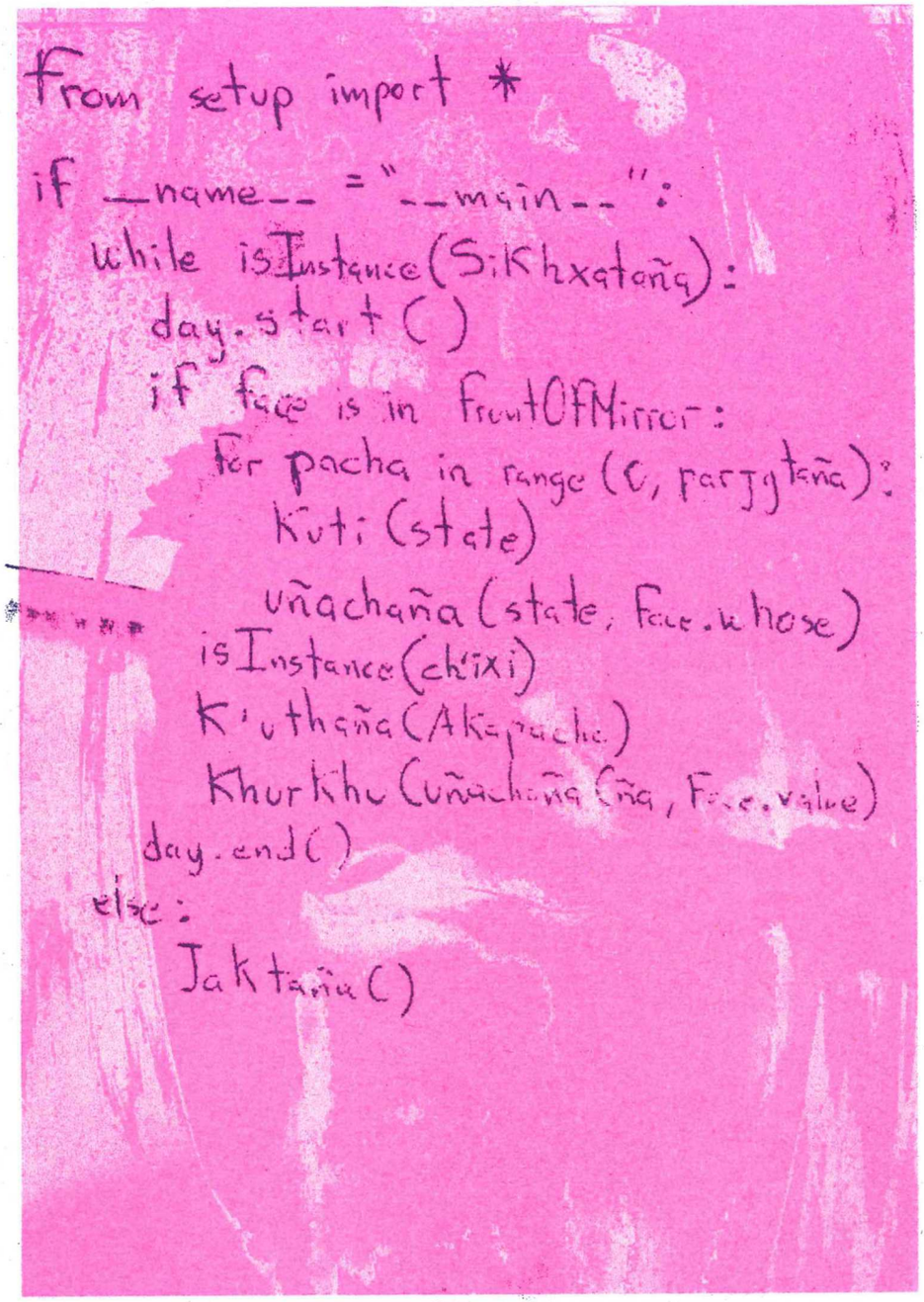
Many know of Lovelace's father, the poet Lord Byron. It was said that her mother was scared she might inherit his "dangerous imagination" and was determined to suppress any trace of this through strict mathematical education.

But Ada refused to be disciplined into a single category. She called herself a "poetical scientist" and in her journals algebra sits beside longing, differential calculus beside lines like "I want to put inimitable life into my work, a fire, a soul."

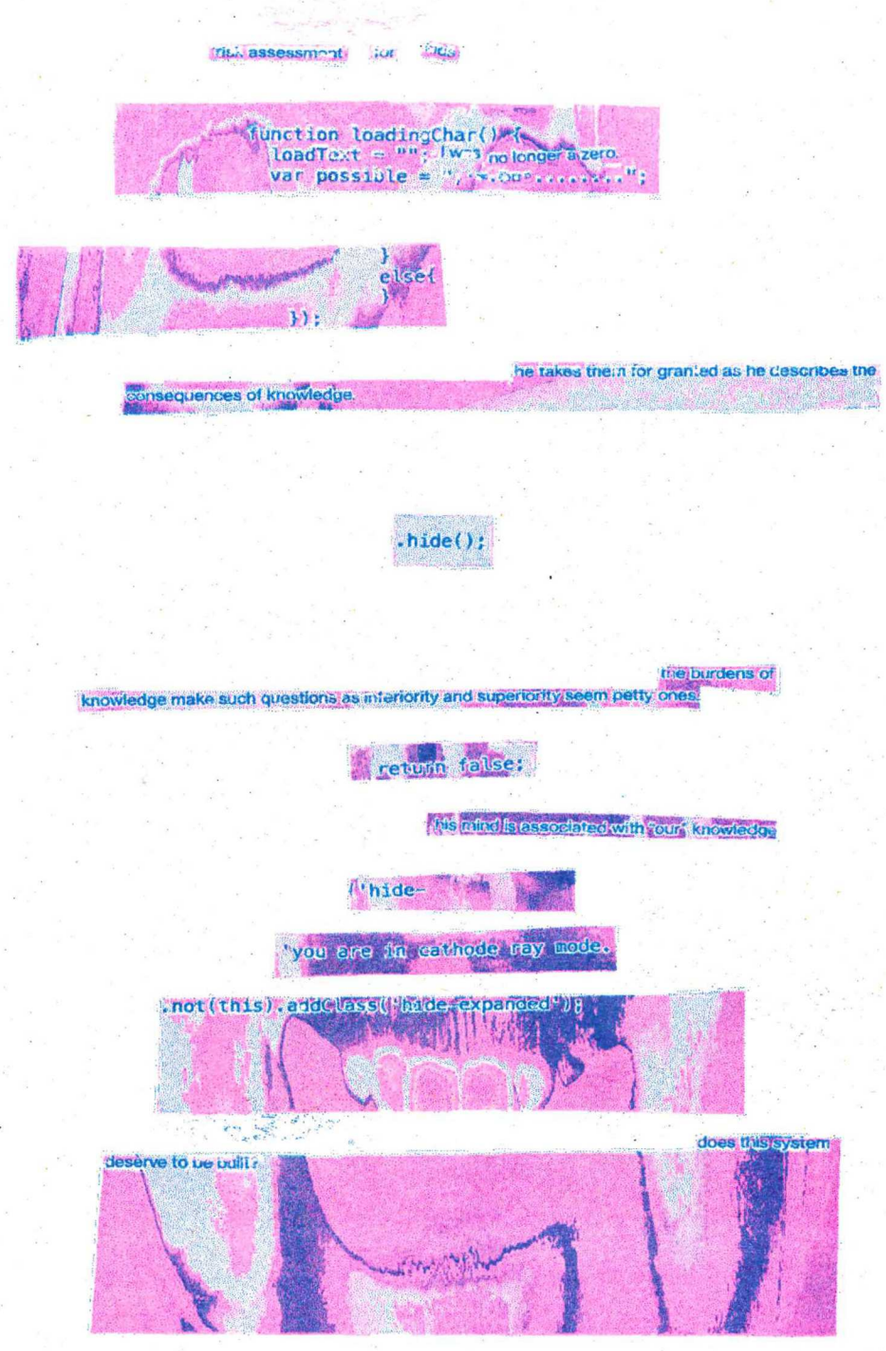
Her contributions to the field of programming were ignored for nearly a century. Historians credited Babbage, her mentor, for everything, and early accounts dismissed her as just eccentric, hysterical, or merely decorative.

Even in Alan Turing's foundational work on artificial intelligence, her name does not appear. It wasn't until feminist scholars in the 1970s returned to her notebooks that her influence was fully recognised.

A Poetical Scientist



Poem by Luis Alfredo Landa Mariaca



Poem by No longer A Zero

What is Code Poetry?

Poem by Undefined

Code poetry, E-poetry or digital poems (not to be confused with code work) is a subgenre of electronic literature. But answering what it really is, for me, a bit complex.

Creative coding, as an avant-garde art practice, often tries to dismantle the very frame it operates within: the syntax (the "grammar rules") of code, and also explores the computer's technical limitations. The practice exists both as an internet subculture, where communities organically form their own subcategories and labels, and within academia, where electronic literature is often studied in the context of critical code studies.

That makes code poetry difficult to pin down to one stable definition. There are ongoing debates among scholars about how to classify artworks within the sphere of electronic literature such as code poems. Even though, for me as an artist, those boundaries aren't central to my work, the debate still reveals something important: code as an artistic medium keeps producing new questions.

Eduardo Ledesma writes in "The Poetics and Politics of Computer Code in Latin America: Codework, Code Art, and Live Coding" (which I recommend, and will quote again later): "Latin American codework is hardwired into local material and physical realities." For me, this means that the constraints, the situated conditions of infrastructure, history, and place are not a limitation of the work, but can rather in itself be part of the critique. In this sense, codework and code art can function as a critical intervention into the supposedly "neutral" character of code, exposing what is often hidden beneath its operations, especially because of code's mainstream use in creating efficiency and profit.

And because electronic literature, creative coding, and net art are so tightly intertwined with the evolution of digital computers, the field keeps changing, constantly forced to rethink its own terms.

Still, I'll try to introduce a few examples. Why? It contradicts my position of not wanting to categorize, yet knowledge can create agency. Just keep in mind that this is a small slice of a constantly shifting sea..

These working definitions draw on N. Katherine Hayles' "Electronic Literature: What Is It?" in Doing Digital Humanities: Practice, Training, Research.

Workshop

Decoloniality and Code Poetry

Poem by Isabel Aguilar Johansson

In the autumn of 2025, ETHOS Lab hosted a workshop titled "Code Poetry Against Colonialism", facilitated by me: software developer and visual artist Isabel Aguilar Johansson. The workshop was inspired by Hanli Geyser's text "Decoloniality, Digital-coloniality and Computer Programming Education."

Geyers argues that programming education is structured around epistemologies of the Global North and that students are expected to adopt these worldviews in order to be considered competent. The notion of programming as a neutral language is tied to a positivist belief that the world can be fully defined, measured and verified through logic. Yet languages, whether poetic or computational, do not only describe reality, they create it. Every line codes decisions about what can appear, what must be excluded, what counts as valid and what becomes an error.

This has political consequences. Fanon reminds u,s in "Black Skin, White Masks" that dehumanization is not an accidental feature of colonialism but its structural condition. By defining "the human" against "the non-human," colonial thinking enables slavery, racial violence, the extraction of labor and the reduction of people to resources. This logic is not identical to programming, of course, but programming languages do operate through similar epistemic habits: categorisation, boundary-drawing, type enforcement, rejection of ambiguity. A semicolon can become a border. A variable can become a category. A function can collapse complexity into a single predetermined action.

The workshop attempted to make these structures visible. Participants were invited to cut and paste fragments from codebases together with excerpts from decolonial texts, assembling new compositions that were neither code nor poetry. As they rearranged symbols and sentences, the inherited logics of programming, its hierarchies began to surface. But the participant got to bend the syntax to meld with their own description of the world.

My approach as a facilitator was shaped in part by the situationist idea that playful creation can be a practice of freedom. The intention was not to teach participants to "fix" programming, but to challenge preconceived myths about code.

The practical method was simple: scissors, glue, paper, code, theory. Recontextualisation as a political tool, contrapuntal reading of code, translanguaging across epistemologies, and critical code literacy. By working through poetic form, we created a moment of disruption that exposed the colonial logics embedded in programming languages and opened up new ways of understanding code.

In the end, we created these code-poems. you've seen trough this zine,

rces

manities Initiative. (n.d.). Electronic literature: An introduction [Video].

1967). Black skin, white masks (C. L. Markmann, Trans.). Grove Press.

013, October 15). Ada Lovelace Day - tales of inspiring women. The Guardian.

(2024). Decoloniality, digital-coloniality and computer programming . ACM Transactions on Computing Education, 24(4), Article 54, 1–30. https:// .1145/3702332

K. (2007, January 2). Electronic literature: What is it? Electronic Literature ion.

K. (2008). Electronic literature: New horizons for the literary. University of ne Press.

E. (2015). The poetics and politics of computer code in Latin America: , code art, and live coding. Revista de Estudios Hispánicos, 49(1), 91–120. i.org/10.1353/rvs.2015.0016

n.d.). Who was Ada Lovelace? Ada Lovelace Day.

019, October 17). Ada Lovelace found poetry in computer algorithms. The kins News-Letter.

(1978). Orientalism. Pantheon Books.

Peña, P. (2021). Artificial intelligence and consent: A feminist anti-colonial nternet Policy Review, 10(4). https://doi.org/10.14763/2021.4.1602