

CODING AS A PLAYGROUND

Graphic Design ; Post-Digital Age ; Learn to Code vs. Code to Learn: Creative Coding Beyond the Economic Imperative
Essay by Silvio Lorusso Demian Conrad Onomatopoe, 2022

LEARN TO CODE

In 2021, the economic imperative to train and retrain has never been so strong. After the pandemic's dramatic impact on artists' economies, a skepticism about "creative" work is emerging, portraying it as unproductive daydreaming, and a wholly unessential industry. The emphasis is now on hard labor and effectiveness. The fundamental idea of Learn to Code is that the ability to program is a historical necessity for people working at a useless or obsolete job, and that these people must serve the economic imperatives of capitalism. This servitude is referred to as

digital Cramer highlighted has to do with the revival of old media. This might be a bit of a stretch, but what is more "old media", more 20th-century, than the idea of a workforce to be forged for the good of the nation? Of course, the Learn to Code narrative hints at the fact that jobs, skills and aspirations do not exist in a vacuum. However, due to a combination of disenchantment with programming and old-media labor rhetoric, coding emerges as a post-digital manifestation of capitalist realism, forcing graphic designers, journalists and coal miners alike to deal with their situations. All of them must go through mandatory updates, just like software. Is programming itself immune to this logic? Not really, it would seem, as the angelus novus of AI promises or threatens (depending on whom you ask) to automate the coder as well.

CODE TO

In a time when much is being said about the creativity of autonomous AI-powered machines, it is good to reconsider Licklider's notion of human-computer symbiosis. When you code, you instruct the computer to execute a more or less complex task, which is then immediately performed. You do not always know what to expect: the result might awe or disappoint you, allowing you to reorient or even redefine your initial goal. Part of the symbiosis is intrinsic to the language shared by user and machine—namely, code. Creativity unfolds through this micro-iterative learning process: it is neither in the mind nor in the machine, but rather in the continuous scripted dialogue between the subject and their extensions. The computer is just one of those extensions, but a particularly powerful one, since it is, to use Alan Kay's term, a metamedium, that is, a medium capable of simulating all others. As such the computer should be a thing that can be shaped and transformed. When it becomes less malleable, the computer is fixed within a stable media, which is perhaps more efficient, but also less surprising, less "creative." You do more but you learn less

and a capacity for listening; the output is fun and a sense of belonging. Coding can also be a bridge linking us to past users. We see this in Ted Davis' assignment to recreate pioneering computer works or with the Re-Programmed Art Project, where a series of contemporary designers reinterpreted "analog" works of the Italian collective Gruppo T, active in the 1960s.

CODING AS A CRAFT

While Learn to Code turns coding into a resumé-ready skill, Code to Learn is about coding as a craft. My understanding of craft is wide-ranging: "a good job well done," as Sennett defines it. A craft is a conveyer