

# unlearning of a web code

Master Thesis von Fernanda Braun Santos  
Matrikel-Nummer: 2465357  
Hamburg, April 2024

Klasse Digitale Grafik  
HFBK Hochschule für bildende Künste Hamburg

Betreut durch Prof. Dr. Hanne Loreck  
(Studienschwerpunkt Theorie und Geschichte)  
Prof. Konrad Renner und Prof. Christoph Knoth  
(Studienschwerpunkt Grafik / Fotografie / Typografie)

# Inhalt

1. Einleitung
2. Coding oder Programming oder Software-Engineering
3. Wie Computer Sprache geschrieben wurde — Eine kurze Coding Geschichte
4. Ein Netz wird um die Erde gespannt — Das Internet
5. Unlearning des kolonialisierten Codes
6. Die Null — Der Neuanfang und das Nichts
7. Befehl und Performance — Menschliche und unmenschliche Sprache
8. Was willst du später werden? — Coder oder User sein
9. Als Frauen noch Computer waren und heutiger Tech-Sexismus
10. Technologie gab's lange vor Geburt Christi
11. Rassistische Codes und koloniale Plattformen
12. Trial and Error — Wo Dekolonialisierung anfängt und aufhört
13. World white Web
14. Academic Coding a.k.a. Inaccessible Contemporary Sites
15. Hacking, Decolonizing, Unlearning
16. Nachwort

# Anmerkung zur Sprache

In dieser Arbeit wird gegendert. Wenn nicht, ist ausdrücklich die männliche Form gemeint.

Die englische Form eines Begriffes wird ausschließlich verwendet, wenn es kein äquivalentes deutsches Wort gibt.

Vor / nach unserer Zeitrechnung, kurz: v. u. Z. / n. u. Z, wird hier verwendet, um den christlichen Fokus im Text zu verringern. Solange es keine dekoloniale Lösung gibt, bleibt es eine symbolische Entscheidung, da die westliche und christliche Perspektive trotzdem beibehalten wird.

Schwarze / Schwarze Personen ist hier eine Selbstbezeichnung und beschreibt von Rassismus betroffene Personen. Schwarz wird in dieser Arbeit systematisch großgeschrieben, denn es handelt sich nicht um eine Eigenschaft, sondern um Konstruktionen in unserer Gesellschaft. Schwarz soll als Abgrenzung zu vom Begriff ‚nicht-weiß‘ verstanden werden. ‚Colorism‘<sup>01</sup> und ‚white Privilege‘<sup>02</sup> werden vollkommen im Begriff mit einbezogen und anerkannt, auch wenn diese nicht weiter ausgeführt werden.

01 Colorism: Vorurteilsbehaftete oder bevorzugte Behandlung von Menschen gleicher Race allein aufgrund ihrer Hautfarbe. Colorism wird gefördert durch ein europäisch/westlich geprägtes Schönheitsideal.

02 White Privilege hier: Schwarze von der Gesellschaft weiß wahrgenommene Personen, die von weißen Privilegien profitieren.

# 1. Einleitung

*„Wenn wir mit der heutigen Technologie Lichtgeschwindigkeit erreichen wollen, werden dann alle Länder der Welt in der Lage sein, die gleiche Geschwindigkeit der Informationsprozesse zu erreichen? Werden alle Länder in der Lage sein, an der Erfassung des lokalen Wissens in der Welt teilzunehmen? Wenn alles so schnell geht: Werden wir in der Lage sein, das westliche historische epistemologische Modell zu revidieren und neu zu hinterfragen? [...]“<sup>03</sup> [Abb.1]*

Wer codet ist automatisch Gestalter:in der digitalen Welt. Doch wer codet? Und für wen? Alles, was wir im Digitalen finden können, wurde mal in Form von Codezeilen gestaltet; Auch wenn wir ein Baukastensystem oder Verbindungen in Form von Nodes<sup>04</sup> verwenden, werden automatisch im Hintergrund Codezeilen geschrieben. Dazu gehören der lokale Blog der HFBK oder die Website des Hamsters deines Nachbarn, aber auch Giganten wie Google, Airbnb und Uber. Obwohl zwischen diesen lokalen und globalen Beispielen über 2 Milliarden Codezeilen und 2 Milliarden Nutzer:innen Unterschied liegt, folgen sie der gleichen Logik.

In dieser Arbeit wird die Verbindung zwischen Coding und Design analysiert und in den Kontext von Dekolonialisierungs- und Unlearning<sup>05</sup>-Theorien eingebettet. Sexismus, Rassismus, Klassismus und Kapitalismus werden hierfür zusammenhängend betrachtet. Für eine weitreichende kritische Auseinandersetzung mit Code und Design werden die Geschichte des Codings und dessen Auswirkungen, Herkunft bestimmter Glyphen, Sprache und Befehle, Technologie Definitionen und Ästhetik Theorien im Zusammenhang mit digitalem Grafikdesign einbezogen. Ähnlich wie beim Schreiben von Code mit mehreren verschiedenen Programmen und Programmiersprachen, die miteinander verknüpft werden, weil sie sich als nützlich oder sogar notwendig für den Inhalt und Kontext erweisen, werden in diesem Beitrag diverse Disziplinen und Perspektiven zusammengeführt, um einen umfassenden Überblick über die Kolonialisierung von Code zu bieten.

03 Wildi Merino, Ingrid: Arquitectura de las Transferencias. Arte, Política y Tecnología. 2012–2014. [http://www.ingridwildimerino.net/images/recherche/ADT\\_version\\_ESP.pdf](http://www.ingridwildimerino.net/images/recherche/ADT_version_ESP.pdf). Übersetzung: Fernanda Braun Santos.

04 Node-based-programming: Programmieren indem Code-Blocks oder Code-Schnipsel miteinander verbunden werden. Jeder Block steht für ein oder mehrere Befehle.

05 Unlearning: Konzept aus dekolonialer Forschung und Kunst. Es bedeutet Fähigkeiten, Meinungen und Kenntnisse zu überprüfen und andere anzuerkennen, hegemoniales Wissen und eingeübte Praktiken korrigieren, Wissen und Nichtwissen bewusst wahrnehmen.

Als Chilenin bleibe ich immer wieder an der Definition der eigenen Identität hängen. Chile — die Oasis Lateinamerikas — wo Politik, Leben und Archive geprägt sind von Neoliberalismus und hohem Wirtschaftswachstum, doch auch vom Erbe der Kolonialisierung und der Pinochet-Militärdiktatur. Trotz des ständigen Protests und Erinnerung, Aufarbeitung und Sichtbarkeit bleibt für mich die Frage offen, wo wir uns in den Debatten über Rassismus und Kolonialisierung einordnen. Und während ich diesen Fragen nachgehe und im Lateinamerikanischem Kontext einordne, beobachte ich die Geburt einer globalen digitalen Identität aus unendlichem Code. Penetrant, kontinuierlich und präzise wird in diesen Zeilen alles bis ins kleinste Detail definiert und implementiert. Ohne das meiste wirklich lesen und verstehen zu können, sind wir mit allen Ländern der Welt mittendrin — Viele im Globalen Süden haben jedoch kaum den eigenen Teil dazu geschrieben.

Mein Interesse für das Coden wurde durch Unlearning und Decolonizing Theorien erweckt. Seit Beginn meines Bachelors interessiere ich mich für Regeln und Systeme im Design: Wer ist Designer:in? Wer kann designen? Und wer designt nicht? Dabei wurde immer wieder klar, dass die Codes in unserer Designsprache tief verankert und mit kapitalistischen, rassistischen und sexistischen Programmen verflochten sind.

Je mehr ich mich der Praxis des digitalen Grafikdesigns widme, desto mehr verstehe ich, dass die diskriminierenden, kolonialen Strukturen des analogen Grafikdesigns hier teilweise verschärft stattfinden. Trends und Macht können viral werden, doch nicht alle haben Zugang zu den Materialien, die selten zu finden und Sprachen, die schwer zu verstehen sind.

## 2. Coding oder Programming oder Software-Engineering

,Coding‘ wird auf deutsch mit dem Wort ,Programmierung‘ übersetzt. Die Begriffe bedeuten allerdings nicht das selbe. Wer programmiert, setzt sich nicht nur mit der Entwicklung von Computerprogrammen mithilfe von Programmiersprachen auseinander, sondern auch mit Dokumentation, Übersetzung, der Behebung von Fehlern, Test, Analyse, Gestaltung und weiteren Prozessen, die zum Endprodukt beitragen. Programming beinhaltet also vielschichtige Aufgaben.

Coding ist ein kleiner Teil des Aufgabenbereichs der Programmierung. Coding umfasst das Zeichen und Zeilen schreiben, in einer

bestimmten Programmiersprache oder Syntax folgend, die von Maschinen wie zum Beispiel Computern (das kann ein Desktop-Computer, ein Smartphone oder Tablet sein) gelesen werden können, um diese anschließend zu performen. Code ist Text und Text ist Code, denn für beide werden Zeichen und Symbole verwendet, um etwas anderes zu repräsentieren. Coding ist nach dieser Betrachtungsweise also abstrakter und dadurch allgemein umfassender als Programming. Die Grundlage fast aller digitalen Informationen ist bis heute Binärkode, auch Maschinencode genannt. Binärkode basiert auf 0 oder 1, wahr oder falsch, Sein oder Nichtsein.

Coding ist auch das Implementieren von Algorithmen, das Kombinieren von Programmiersprachen, das Verstehen des Ziel Subjekts, Zusammenhänge von Unterprogrammen oder Hilfsmodulen der Programme herstellen und die Fähigkeit logisch denken und schreiben zu können, im Sinne von Schlussfolgerungen ziehen und Zusammenhänge definieren.<sup>06</sup>

*„Programming today carries her legacy. First and foremost, code is a language, with syntax and grammar and vocabulary, plus some logic. We can invent code much like a sentence, choosing the words that fit best. Code is text both in the sense that it is composed by letters and that it can be written and read.“<sup>07</sup>*

Trotz der Bemühungen, in der Coding-Geschichte eine allgemeine Definition oder gar Sprache finden zu wollen, wurde sich nie offiziell auf eine geeinigt. Die Vielschichtigkeit, Parallelität und rasante Entwicklung von Coding, Programming und Software-Engineering machte dies unmöglich.

### 3. Wie Computer Sprache geschrieben wurde – Eine kurze Coding Geschichte

Eines der ersten Beispiele für Programmierung sind Uhrwerke seit 200 – 70 vor unserer Zeitrechnung oder mechanische Musikinstrumente ab dem 9. Jahrhundert. Ada Lovelace, eine britische Mathematikerin, gilt allgemein als erste Computer Programmiererin. Sie erschuf 1840 ein Programm zur Berechnung von Zahlen und hatte darüber hinaus die Idee, Symbole wie Buchstaben oder Musiknoten anhand von Zahlenkombinationen auszudrücken. Dies war die erste Idee einer Applikation über mathematische

<sup>06</sup> Computer programming: [https://en.wikipedia.org/wiki/Computer\\_programming](https://en.wikipedia.org/wiki/Computer_programming) (Stand: 01.03.2024)

<sup>07</sup> Cairns, Anna: Code, Text, Type. c0da, 2023.  
<https://c0da.org/contributions/code-text-type> (Stand: 01.03.2024)

Funktionen hinaus,<sup>08</sup> blieb allerdings noch in der Theorie ihrer Zeit und wurde nicht in die Praxis umgesetzt.

Die ersten Codes wurden speziell für Endgeräte erstellt. Rund 100 Jahre nach Ada Lovelace gab es die ersten Beispiele mathematischer Codes, die an Computern angewendet werden konnten, die aus heutiger Perspektive eine extrem limitierte Rechen- und Speicherkapazität hatten. Die ersten Programmiersprachen, die einfache Befehle eng an die Mechanik der Maschine gebunden waren, mussten händisch angepasst werden.<sup>09</sup>

Alan Turing gilt als der wichtigste Theoretiker der Grundlagen von Informations-, Computertechnologie und Algorithmen. Er erfand 1935 / 37 die nach ihm benannte Turingmaschine, die mathematische Systeme mithilfe eines Algorithmus berechenbar macht. Dieses mathematische System folgte strikten Regeln (Algorithmus), aus denen mathematische Funktionen entstanden, die Zeichen manipulierte und bearbeitete.

Hierbei wurde bewiesen, dass — wenn das richtige Regel-Konstrukt oder der richtige Algorithmus verwendet wird — jede mathematische Rechnung berechnet werden kann.<sup>10</sup> An dieser Maschine wurden die ersten Programmcodes von verschiedenen Mathematikern erfunden oder angewendet. Zum Beispiel entwickelte der Us-amerikanischen Alonzo Church ein mathematisches System mit den ersten Variablen<sup>11</sup> und abstrakten, zusammengefassten Funktionen.

Es folgten mehrere Theoretiker:innen, Mathematiker:innen und Informatiker:innen, die zur Entstehung einer Computer Maschine beitrugen und diese für verschiedene Zwecke und Tätigkeiten neu und weiterentwickelten, bis es 1941 den ersten elektronischen Computer gab. Einer dieser Computer war der Electronic Numerical Integrator and Computer (ENIAC). Dieser war zunächst streng geheim und wurde erst 1946 der Öffentlichkeit bekannt gegeben. Damit einhergehend wurden die ersten vollzeit Computer Jobs von Rechnerinnen besetzt, die noch ohne Programmiersprache arbeiteten. Die Programmierung des ENIAC erfolgte durch die Verbindung einzelner Bestandteile mit Kabel und Drehschaltern.<sup>12</sup> [Abb.2]

Bevor es überhaupt die Idee der analytischen Maschine gab, mussten Menschen für ungefähr 200 Jahre die Aufgabe der Rechnung und

08 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018. S.9.

09 ebd.

10 [https://de.wikipedia.org/wiki/Alan\\_Turing](https://de.wikipedia.org/wiki/Alan_Turing) / <https://de.wikipedia.org/wiki/Turingmaschine> (Stand: 01.03.2024)

11 Variablen: Abstrakter Platzhalter für einen bestimmten oder unbestimmten Wert.

12 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018. S.16.

Verarbeitung von Informationen erledigen. Mitte des 20. Jahrhunderts haben nahezu ausschließlich Frauen ‚computing‘ oder ‚computations‘ ausgeübt<sup>13</sup>. Dafür sprach die Annahme, dass Frauen geduldiger und genauer wären, dies galt damals als typische Frauenarbeit. Sie wurden zu den ersten Operator:innen und bildeten dadurch ein Netzwerk, das ungefähr so funktionierte wie die erste zusammenhängende Computer-Maschine, lange bevor es überhaupt die Idee einer solchen gab — Frauen waren Computer.<sup>14</sup>

Die erste *High-Level Language* wurde 1952 von der Us-Amerikanerin Grace Hopper entwickelt. Zuvor wurde ausschließlich mit Machine Language gearbeitet — eine *Low-Level Language*. *Low-Level Language* ist die Bezeichnung für Anweisungen an eine bestimmte Hardware, die oft nach einem binären System geschrieben wurde. Hier wurde schon mit Abkürzungen und Spezifizierung für Operationen und Befehle gearbeitet, was dazu führte, dass je nach Maschine verschiedene Systeme und Sprachen angewandt werden mussten.<sup>15</sup>

Hopper erkannte, dass es viel zugänglicher und verständlicher ist, mit Englisch einen Befehl zu schreiben, als mit den bisherigen eher abstrakten Programmiersprachen. Der Computer verstand nun die englische Sprache und nicht andersrum. Von Hoppers Vorgängern und Kollegen wurde dies für unmöglich gehalten und stieß auf Unverständnis. Dennoch entstand eine heute immer noch verwendete Processing Language und somit auch die *High-Level Languages*.<sup>16</sup> Es folgten mehrere Sprachen, die die Codes durch Abstraktion auf Englisch verständlicher machten. In den späten 60er-Jahren waren Computer in der Lage Programme direkt als Text eingetippt zu bekommen statt analog auf Papier. Die neuen Möglichkeiten von Texteditoren machten Veränderungen und Verbesserungen am Code schneller und flexibler.<sup>17</sup> Hopper ist eine von vielen Personen, die zwischen 1950 und 1970 eine Vielzahl an relevanten *High-Level Languages* entwickelten, die kein Hardware-Wissen beanspruchen, sondern unabhängig von der Maschine mithilfe dieser Sprachen Befehle ausführen können.<sup>18</sup>

Neu zu dieser Zeit war, dass Effizienz und Überschaubarkeit des Codes durch Logik und Struktur erzeugt wurden. Anstatt wie bisher Befehle im Code zu schreiben und immer wieder zwischen ihnen

13 Computing, computations: Zielorientierte Tätigkeiten für oder am Computer.

14 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018. S.7.

15 [https://en.wikipedia.org/wiki/Computer\\_programming#History](https://en.wikipedia.org/wiki/Computer_programming#History) (Stand: 01.03.2024)

16 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018. S.12–15.

17 [https://en.wikipedia.org/wiki/Computer\\_programming#History](https://en.wikipedia.org/wiki/Computer_programming#History) (Stand: 01.03.2024)

18 [https://en.wikipedia.org/wiki/History\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/History_of_programming_languages) (Stand: 01.03.2024)

zu springen, wurden Sequenzen und Selektion durch Verzweigung und Wiederholungen eingesetzt.

Wie ein Code gelesen wird, also in mehreren Schritten (langsamer, mit Unterordnungen) oder in einem einzigen Durchlauf (sehr schnell und effizient, mit einmal Definierungen), kann auch seither bestimmt werden. Dadurch entstehen neue Möglichkeiten der Geschwindigkeit und neue Herausforderungen in der Strukturierung von Code und der Kapazität des Computers.<sup>19</sup>

Im Jahr 1968 wurde bei einer Konferenz die ‚software-crisis‘ thematisiert, die aus der Motivation organisiert wurde, mit der rasanten technologischen Entwicklung mithalten zu wollen. Es wurde ‚programming‘ in ‚software engineering‘ umbenannt, was unmittelbar das Feld und dessen Akteur:innen beeinflusste. Die wachsende Professionalisierung der Branche exkludierte zuvor praktizierende Programmier:innen wie Mütter und diejenigen, die das Programmieren selbst beziehungsweise inoffiziell außerhalb akademischer und militärischer Einrichtungen lernten. Computer, die bisher Frauen waren, wurden spätestens ab diesem Zeitpunkt von männlichen Personen größtenteils ersetzt.<sup>20</sup> Das Potenzial von Programmieren und Software Engineering wurde erkannt, wodurch die Industrie nach und nach wuchs. Die Code-Geschichte und die Coding-Möglichkeiten werden in diesem Zeitraum immer unübersichtlicher und vielschichtiger.

Es gab um 1980 über 450 etablierte und weniger etablierte Programmiersprachen und noch keine Standardisierung. Das US-amerikanische Militär machte sich daher zum Ziel, eine geeignete einheitliche Sprache für eine Standardisierung zu finden und erschuf eine neue objekt-orientierte Sprache, die durch viele Modifizierungen und Namensänderungen, verschiedene Personen und Institutionen im Jahr 1996 schließlich von Java verdrängt worden ist.<sup>21</sup>

Die vorausgehende kurze historische Einordnung konzentriert sich auf Aspekte, die den jetzigen technologischen Stand stark beeinflusst haben. Die Entwicklung von Programmiersprachen und Coding ist weitaus komplexer, simultaner und unabgeschlossener als hier dargestellt, abgelaufen.

19 [https://en.wikipedia.org/wiki/History\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/History_of_programming_languages) (Stand: 01.03.2024)

20 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018. S.29.

21 [https://de.wikipedia.org/wiki/Geschichte\\_der\\_Programmiersprachen](https://de.wikipedia.org/wiki/Geschichte_der_Programmiersprachen) (Stand: 01.03.2024)

# 4. Ein Netz wird um die Erde gespannt – Das Internet

Schon in den 60er Jahren gab es eine Art vorläufiges Internet: das Arpanet. Es wurde an Universitäten und Forschungseinrichtungen hauptsächlich als ein Netzwerk für Kommunikation mit dem Ziel, Rechenkapazitäten zu maximieren, genutzt. Die elektronische Nachricht, kurz: E-Mail, war unter anderem eine der wichtigsten Errungenschaften. In den 80er Jahren entstanden anschließend die Grundlagen des Internets, die eine weltweite Verbindung möglich machten. Die digitale Kommunikation vernetzte damals verschiedene Universitäten innerhalb der USA und schon bald auch über ihre Grenzen hinaus. Zehn Jahre später wurde entschieden, das Internet auch außerhalb von Bildungsstätten und somit auch für kommerzielle Zwecke zugänglich zu machen.<sup>22</sup> Das Internet ist ein weltweiter Verbund von Rechnernetzwerken und einzelnen autonomen Systemen, die es jedem Computer möglich machen, sich miteinander zu verbinden.

Das World Wide Web besteht aus elektronischen Hypertext<sup>23</sup>-Dokumenten, (Webseiten) die über das Internet zugänglich sind und mittels HTML geschrieben werden. 1993 entstanden die ersten Webbrowser für das World Wide Web. Diese haben die Darstellung von Inhalten online ermöglicht, wodurch die Menge an Content rasant wuchs. Es bot eine völlig neue Grundlage für Programmiersprachen. Hierbei spielten Java (1996) und vor allem JavaScript (1995) ebenfalls eine große Rolle und waren sehr präsent. HTML (Hypertext Markup Language, 1992) erlaubt durch Formatierung und Gliederung von Text und weiteren Daten die Strukturierung des WwW's.<sup>24</sup> Es war nun möglich eine lineare Form des Lesens aufzubrechen und mit einer Netzstruktur durch zum Beispiel Hyperlinks dynamisch zu strukturieren.<sup>25</sup> Die Gestaltung erfolgte mit der Sprache Css (Cascading Style Sheets, 1993).

Die Internetvernetzung und die Browser-Softwares boten die Möglichkeit, „Open Source“ zu arbeiten, indem Code offengelegt, inspiziert und mit anderen geteilt werden konnte. Dadurch wurden online Communities gebildet, die Code miteinander entwickelten und sich gegenseitig an diesen bedienen konnten. Die ersten Wellen an unabhängigen Projekten wurden zum Beispiel von Künstler:innen, Coder:innen und Autor:innen

22 <https://de.wikipedia.org/wiki/Internet#Geschichte> (Stand: 01.03.2024).

23 Hypertext: Über links verbundenes dynamisches Netz aus Text-, Bild- und Dateneinheiten.

24 [https://de.wikipedia.org/wiki/Geschichte\\_der\\_Programmiersprachen](https://de.wikipedia.org/wiki/Geschichte_der_Programmiersprachen) (Stand: 01.03.2024)

gecodet, doch schnell wurde das Web ein kommerzielles Medium. Je größer und komplexer Web-Projekte wurden, desto verschachtelter und unzugänglicher wurde der Code dahinter. Dadurch konnten Firmen immer stärker ihren Fortschritt Gatekeepen<sup>26</sup>.

Das Www, das Internet und der Webbrowser sind unterschiedliche Elemente. Das Www ist ein signifikanter Bestandteil des Internets, ebenso wie E-Mails und verschiedene Netzwerkprotokolle. Das Www wird und wurde über Browser-Software zugänglich gemacht. Zum Beispiel werden E-Mails über das Internet versendet und können auch über einen Webbrowser abgerufen werden, sind aber nicht unbedingt Teil des Www und unabhängig vom Webbrowser. Der Webbrowser dient in diesem Fall lediglich zur Darstellung, wofür auch Software verwendet werden kann, die nicht spezifisch für das Www entwickelt wurde.

Die damals neu entstandenen Plattformen entwickelten sich zu einem geschlossenen dezentralen System, das vermehrt ausschließlich über Webbrowser zugänglich war. Sie sind nicht mehr lokale Anwendungen auf einem Endgerät, sondern bedienen sich verschiedener Netz-Systeme des Internets. Code konnte zum Beispiel mit verschiedenen Bibliotheken vernetzt sein und sich auf andere Code-Schnipsel im Internet beruhen. Dadurch wurde es teilweise schwieriger zu verstehen, woher und wohin Daten kursieren und wo diese gelagert werden. Code wurde weniger ‚Open Source‘. Durch Social-Media-Plattformen wurde die Interaktivität und Verbindung zwischen Nutzer:innen konstant wichtiger und es waren zunehmend Inhalte wie Fotos, Videos und Links im Umlauf. Doch auch Endgeräte waren untereinander besser vernetzt, damit Inhalte und Nutzer:innen Daten verbreiten konnten.<sup>27</sup> Seit einigen Jahren geht der Trend von Browser-Anwendungen wieder zurück. Diese werden durch Applikationen, kurz: Apps, lokaler. Dadurch müssen Informationen und Daten nicht mehr den Browser-Anbieter durchlaufen, der diese für sich nutzen könnte, sondern können direkt von der lokalen App an den App-Betreiber weitergeleitet werden. Auf den neuen Businessmodellen und Marketingstrategien der 2000er, um Geld aus dem Internet zu machen oder Geld ins Internet zu bringen, basieren die heute einflussreichsten und mächtigsten Unternehmen weltweit.<sup>28</sup>

Verglichen mit den frühen Möglichkeiten von HTML, Css und JavaScript können Coder:innen durch vereinfachte Computerprogramme und

26 Gatekeeping: Zugang zu Informationen, Ressourcen oder sozialen Gruppen kontrollieren oder sogar verwehren.

27 <https://de.wikipedia.org/wiki/Internet#Geschichte> (Stand: 01.03.2024)

28 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018. S.67.

Baukastensysteme heute effizient und mit niedrigem Kostenaufwand im WWW und außerhalb digital gestalten. Dokumentation, Tutorials und Bibliotheken helfen dabei.<sup>29</sup> Die bis heute rasant wachsenden Social-Media-Plattformen machen diesen Austausch umso einfacher.

## 5. Unlearning des kolonialisierten Codes

Um zu einzuschätzen, ob eine Dekolonialisierung von Coding notwendig ist, sollte zunächst die Definition von Kolonialismus und Kolonialisierung näher betrachtet werden. Diese lautet: Die Eroberung nicht-europäischer Länder mit dem vorrangigen Ziel der wirtschaftlichen Ausbeutung. Bei der Kolonialisierung wurde missioniert, doch der Fokus lag auf dem Zuwachs von Reichtum der jeweiligen Kolonialmacht, der Ausbeutung, Expansion und Macht-Beibehaltung. Die Beziehung zwischen Kolonialisten und Kolonialisierten ist extrem ungleich. Kolonialisierung bedeutet die Enteignung von Land, Arbeit, Material und Wissen als auch den Völkermord an indigenen Völkern und deren Versklavung. Trotz einer massiven ‚Dekolonisation‘<sup>30</sup> nach 1945 bestehen bis heute kulturelle und wirtschaftliche Abhängigkeitsstrukturen.<sup>31</sup>

Kolonialismus dient der Herrschaft über Menschen und deren Land, Region oder Gebiet. Die bis heute andauernde Moderne, die durch den Kolonialismus herbeigeführt wurde, ist geprägt durch koloniale Arten des Wissens und der Existenz, unter anderem durch Kategorisierung und Klassifizierung, die sich in Technologien spiegeln und weiter verfestigen. Die gesellschaftlichen Folgen der Kolonialisierung sind bis heute unter anderem stark ausgeprägte Rassismen, Sexismen und Klassismen in der Gesellschaft als auch der tief verankerte Kapitalismus weltweit. Es könnte allgemein gesagt werden, dass Technologien insofern kolonial sind, als dass sie aus der Moderne entstanden sind<sup>32</sup>.

*„Unlearning is not forgetting, it is neither deletion, cancellation nor burning off. It is writing bolder and writing anew. It is commenting and questioning, it is giving new footnotes to old and other narratives. It is wiping off the*

29 [https://de.wikipedia.org/wiki/Geschichte\\_der\\_Programmiersprachen](https://de.wikipedia.org/wiki/Geschichte_der_Programmiersprachen) (Stand: 01.03.2024)

30 ‚Dekolonisation‘: Repräsentiert hier lediglich eine Dimension von Dekolonialisierung. Diese bezieht sich auf die Erlangung der Unabhängigkeit von kolonisierten Ländern durch den Verlust von Besitz oder den Rückzug der Kolonialmächte, sei es auf friedliche Weise oder mit Krieg.

31 <https://www.bpb.de/kurz-knapp/lexika/politiklexikon/17718/colonialismus/> (Stand: 01.03.2024)

32 Mustafa Ali, Syed: A brief introduction to decolonial computing. Research Gate, 2016.  
[https://www.researchgate.net/publication/303980379\\_A\\_brief\\_introduction\\_to\\_decolonial\\_computing](https://www.researchgate.net/publication/303980379_A_brief_introduction_to_decolonial_computing) (Stand: 01.03.2024)

*dust, clearing the grass, and cracking off the plaster that lays above the erased, unlearning is flipping the coin and awakening the ghosts, unlearning is looking in the mirror and seeing the world.“<sup>33</sup>*

In den kommenden Kapiteln werden die einzelnen Punkte der andauernden Kolonialisierung im Code unter dem Gesichtspunkt von Unlearning untersucht, kommentiert, infrage gestellt, neu gedeutet, demaskiert, verbunden, subjektiviert und kontextualisiert.

## 6. Die Null — Der Neuanfang und das Nichts

Wenn wir die Bestandteile im Code etwas besser verstehen wollen, sollten wir bei der 0 anfangen. Das allgemeine Verständnis einer Null, wie wir sie heute kennen, als Richtung, Trennzeichen, Platzhalter, Symbol, leere Menge und die Abwesenheit einer Quantität, begann mit der 0 aus Indien und der Maya. Die 0 der Maya (32 v.u.Z.) fand chronologisch und geografisch isoliert statt, während die indische 0 (ca. 7 Jh. n.u.Z.) heute universal und als Erfindung der Zahl gelten könnte. In Babylon, China, Kambodscha und Ägypten und weiteren Orten der Welt gab es ebenfalls bereits die 0 mit einzelnen Attributen und Eigenschaften, jedoch nicht allen oben genannten. Eine 0 in diesem Sinne gab es in Europa erst Jahrhunderte nach unserer Zeitrechnung.<sup>34</sup>

Bei Computerprogrammen hat die 0 eine enorme Bedeutung, da diese aus riesigen aneinander gereihten Kombinationen von Nullen und Einsen bestehen. Bei manchen Programmiersprachen kann eine 0 jedenfalls auch eine wichtige Bedeutung haben, also nicht möglich oder nicht vorhanden sein, im Sinne von ‚wird ignoriert‘. Im Binärcode, bestehend aus 0 und 1, gibt es nur wahr / falsch, ja / nein, ein / aus, männlich / weiblich — gibt es eines, wird das andere ausgeschlossen.

Wenn wir davon ausgehen, dass „*Weibliches etwa als das Geschlecht, das nicht eins ist, das sich selbst berührt*“<sup>35</sup> entspricht die 0 dem weiblichen Geschlecht, dem verbindenden Element, das Raum und Möglichkeit lässt, ohne zu sein oder eben vieles zu sein. Sie ist nicht an eine Zahl geknüpft. Sie erhält ihre Bedeutung und definiert sich im Zusammenhang mit weiteren

<sup>33</sup> <https://twitter.com/SavvyArtSpace/status/718837212838764545> (Stand: 01.03.2024)

<sup>34</sup> Gheverghese Joseph, George: A Brief History of Zero. Research Gate, 2008.

[https://www.researchgate.net/publication/228783983\\_A\\_Brief\\_History\\_of\\_Zero](https://www.researchgate.net/publication/228783983_A_Brief_History_of_Zero) (Stand: 01.03.2024)

Zahlen, Zeilen und Symbolen um sie herum. Die Position, die sie vertritt, ist relativ — und doch ist sie unverzichtbar.<sup>36</sup>

Die 0 hatte lange Zeit, bevor sie die wichtigste Hälfte eines modernen Computers wurde, auch eine spirituelle Bedeutung. Sie war die Leere, das Erneuern, das Nichtsein und Teil einer Prophetie.<sup>37</sup> Dieser vielschichtige Gehalt der Zahl und ihre diversen Möglichkeiten sind heute im Globalen Norden durch ihren Zweck als Abstraktion und Verschlüsselung eher unbedeutend.

Die Glyphen der 0 folgen fast in jeder Sprache der gleichen Logik, ein rundes, berührendes, geschlossenes Symbol: Punkt, Kreis, Oval. [Abb.3] Selbst bei den Mayas war die 0 ein rundes Schildkrötenpanzer. [Abb.4] Die 1 steht, neigt und biegt sich und ein Ende bleibt dabei offen.

## 7. Befehl und Performance — Menschliche und unmenschliche Sprache

Bei der Grundbeschaffenheit des Computers — mit der Anwendung der binären Codes 0 und 1 — einigen wir uns bereits auf arabische Ziffern, die ebenfalls im lateinischen, kyrillischen und griechischen Alphabet vorkommen. Davon ausgehend basieren die meisten Sprachen, auf denen wir uns mit dem Computer verständigen, auf Englisch: HTML, Css, Java, JavaScript, php, C, C++, Python und weitere. An dieser Stelle ist es wichtig zu erwähnen, dass es viele Programmiersprachen gibt, die sich gegenseitig ergänzen und je nach Projekt von einer oder einem Dutzend Programmiersprachen Gebrauch machen.

Es gibt Beispiele nicht-englischer Programmiersprachen, die nicht-lateinische oder neue Zeichen, Zahlen, Glyphen und Code Richtungen ermöglichen. Eine davon ist die Programmiersprache APL. Sie war eine nicht-englische, auf vollkommen neuen Zeichen und Glyphen basierende Programmiersprache, die von einem kanadischen Mathematiker und Informatiker entwickelt wurde.<sup>38</sup> Es gab und gibt weitere Programmiersprachen, die auf ähnliche Art dem Englischen entbunden sind, doch diese ist wohl die bekannteste und die Basis für darauf folgende Beispiele mit ähnlicher Logik. Das Besondere an dieser Sprache ist, dass sie in den

36 Schlüters, Eske: Ich – ein Transformer? oder: How to code a feminist transformer?. c0da, 2022. <https://c0da.org/contributions/ich-ein-transformer-oder-how-to-code-a-feminist-transformer> (Stand: 01.03.2024)

37 Gheverghese Joseph, George: A Brief History of Zero. Research Gate, 2008. [https://www.researchgate.net/publication/228783983\\_A\\_Brief\\_History\\_of\\_Zero](https://www.researchgate.net/publication/228783983_A_Brief_History_of_Zero) (Stand: 01.03.2024)

60ern und 70ern weltweit verwendet wurde. Für viele Programmierer:innen war es viel einfacher, diese Sprache zu lernen, als zunächst englisch und anschließend die Programmiersprache. [Abb.5]

Citrine, eine Programmiersprache, die in jede geschriebene Sprache übersetzt werden kann und wurde 2014, mit der Motivation Coder:innen mit diverser linguistischer Herkunft mit einer Programmiersprache auszustatten, entwickelt. Sie ist kostenlos, ‚Open Source‘ und hat das Ziel, lesbärer und minimalistischer zu sein.

Quorum ist ein Beispiel von vielen Initiativen, die barrierefreies Programmieren möglich machen. Die Syntax von Code.org, eine Sprache, die überwiegend für Schüler:innen und Gleichberechtigung im Coding entwickelt wurde, wurde hier so adaptiert, dass sie mit Screenreader, Vergrößerungswerkzeugen und vor allem in Tutorials und Foren zugänglicher ist als andere Programmiersprachen.<sup>39</sup> <sup>40</sup> Da beide Beispiele hauptsächlich in der Bildung eingesetzt werden, sind sie breit verbreitet und zum Beispiel durch Lehrpläne grundlegend etabliert. Code.org sowie andere in Deutschland reichlich unterrichtete Programmiersprachen für Schüler:innen sind heute von Google, Amazon oder Microsoft entweder entwickelt, vertrieben oder nutzen umfassend die Cloud Angebote dieser Unternehmen.

Viele Programmiersprachen sind in andere Sprachen übersetzt worden. Zum Beispiel ist Zhpy eine Variation von Python (englisch basierend), die auf chinesisch geschrieben und codiert werden kann. Üblicherweise sind das Grundgerüst und die Basis als auch das Backend<sup>41</sup> auf englisch und werden sozusagen im Zwischenschritt übersetzt.

Die Programmiersprache Qalb / بَلْقَ , die 2012 von dem Künstler und Computerwissenschaftler Ramsey Nasser entwickelt wurde, ist eine Antwort auf ausschließlich auf englisch existierende Programmiersprachen. In بَلْقَ kann man komplett auf arabischer Sprache Funktionen anwenden und schreiben. [Abb.6]

*„All modern programming tools are based on the ASCII character set, which encodes Latin Characters and was originally based on the English Language. As a result, programming has become tied to a single written culture. It carries with it a cultural bias that favors those who grew up reading and writing in that*

39 <https://www.washington.edu/accesscomputing/sites/default/files/doit-sync/files/QuorumCSP.pdf> (Stand: 01.03.2024)

40 Krishika, K.L.: 6 Popular Non-English Programming Languages. aim (Analytics India Magazine), 2023. <https://analyticsindiamag.com/6-popular-non-english-programming-languages/> (Stand: 01.03.2024)

41 Backend: Infrastruktur hinter einer Software, die für Benutzer:innen nicht sichtbar ist. Die für Benutzer:innen sichtbare Seite ist das Frontend.

cultural. بـلـق explores and challenges that by presenting a language that deviates almost entirely from ASCII.“<sup>42</sup>

Wer Programmiersprachen verstehen kann, hängt stark von Dokumentationen, Tutorials und Informationen ab, die zur Verfügung gestellt werden. Deswegen sind online oder analoge Communities von besonderer Relevanz. Diese kommunizieren meistens auf Englisch.

In einem Interview sagte Yehwan Song, eine koreanische, digitale Künstlerin, die viel mit Websites arbeitet, dass Entwickler ihren Code mit Absicht maskieren und undurchsichtig machen und dadurch unnötig verkomplizieren. Sie sagt, Sprache mache den Zugang schwieriger, wenn bedacht wird, dass alle Domains und Cookies-Fenster auf englisch geschrieben sind. Dies seien Fakten, die von nicht englisch sprachigen Personen einfach akzeptiert werden müssen, um das Internet zu nutzen und dort arbeiten zu können.<sup>43</sup>

Englisch ist die dominierende Sprache von Coding und ohne diese zu beherrschen, gibt es kaum Zugang zu globalen online Netzwerken, Infrastrukturen und Bildungssystemen. Beispielsweise sprechen in Indien 12 % der Bevölkerung die englische Sprache, wodurch die Barriere immerhin deutlich niedriger als zum Beispiel in ganz Lateinamerika ist, wo viel weniger Menschen die englische Sprache beherrschen. In Ländern, die zum Beispiel das kyrillische, hebräische oder chinesische Alphabet verwenden, gibt es kaum eine andere Möglichkeit, als die Grundlagen des lateinischen Alphabets und englisch zu lernen und darüber hinaus eine Hardware zu anzuschaffen, auf der das lateinische Alphabet eintippbar ist. Obwohl mittlerweile weltweit auf fast allen Tastaturen das lateinische Alphabet ebenfalls oder ausschließlich abgedruckt ist, muss die Führung der Finger auf der Tastatur ebenfalls neu gedacht, gelernt und automatisiert werden.

Programmiersprache ist die Sprache, die uns Menschen mit der Maschine verbindet. Diese wird durch Digitalisierungsprozesse immer allgegenwärtiger und zu einem relevanten Teil unserer Kommunikation. Vor allem ange-sichts der Zukunft von Machine Learning ist es wichtig, dass wir mit der Maschine gut und fehlerfrei kommunizieren können, was wir wollen und wo unsere Grenzen liegen. Die Maschine interagiert millionenfach am Tag

42 Nasser, Ramsey: بـلـق. 2013.  
<https://nas.sr/%D9%82%D9%84%D8%A8/> (Stand: 01.03.2024)

43 Song, Yehwan: A conversation with Yehwan Song. trcc, 2022.  
<https://timrodenbroeker.de/yehwan-song/> (Stand: 01.03.2024)

auf dem ganzen Globus mit der menschlichen Sprache<sup>44</sup>. Wenn wir von der Maschine immer besser verstanden werden, sollten wir die Möglichkeit haben, der Maschine in Form von Code mitzuteilen, was sie daraus machen soll — das heißt, sie steuern können. Ein bemerkenswerter Unterschied zwischen Code und menschlicher Sprache ist, dass Code den einzigen Zweck hat, einen Befehl auszuführen. In der menschlichen Sprache ist ein Befehl oder eine Deklarierung vom Kontext und der Kraft dieser Aussage auf komplexe Weise abhängig. In beiden kann die Ausführung negiert werden, doch der Computer würde das im Gegensatz zum Menschen nur tun, wenn er die Aussage nicht versteht oder diese fehlerhaft ist<sup>45</sup>. Ansonsten ist er völlig bereitwillig zur Ausführung. „*Code has become arguably as important as natural language because it causes things to happen, which requires that it be executed as commands the machine can run.*“<sup>46</sup> Menschliche Sprache bringt viel „Noise“<sup>47</sup> mit sich und viele unwichtige, unpräzise Informationsstränge. Hingegen ist Code sehr genau und toleriert keine Fehler. Insofern sind wir Menschen vielverständnisvoller, flexibler und begreifen eigentlich die Maschine viel besser, als sie uns.

Wie weiter oben genannt, gibt es Programmiersprachen, die in Leichter Sprache sind. Sie helfen Menschen dabei, Code zu verstehen, die nicht englisch sprechen, Schwierigkeiten beim Lernen, Lesen und Schreiben haben, junge oder alte Menschen. Bei diesen Sprachen ist nicht nur der Code verständlich und zugänglich, sondern auch das Ergebnis und das Ziel, zum Beispiel einer Website, wird klar formuliert. Bei den meisten Initiativen der Entwicklung von Programmiersprache in Leichter Sprache geht es nicht nur um das Programmieren, sondern auch darum, Informationen zu teilen. Es ist wichtig, Fragen zu klären wie: Was ist Coden und warum ist es wichtig zu verstehen, was mit Code erreicht werden kann? Wie können wir unseren Teil dazu beitragen?

„*Information is power. It's a thing many of us understand about the world in many aspects of life—particularly with regard to politics or education, but with regard to disability it is a real tool of empowerment. Because it lets us determine for ourselves whether we can engage, to anticipate what our experience will be like and what it might cost us, and to determine whether we want to invest that time and energy.*“<sup>48</sup>

44 Hayles, N. Katherine: *My Mother Was a Computer. Digital Subjects and Literary Texts*. University of Chicago Press, 2005. S. 39.

45 ebd. S. 50.

46 ebd. S. 49.

47 Noise hier: Irrelevante, unerklärte oder unerwartete Informationen, z. B. Geräusche, die nicht nützlich oder störend sind und ignoriert werden können.

48 Cunningham, Claire: *Equations of Care & Responsibility*. DANCEOLITICS. Uferstudios GmbH, 2021. S. 74.  
<https://online.fliphtml5.com/zgewh/lssv/#p=1> (Stand: 01.03.2024)

Bei Leichter Sprache gibt es klare Regeln. Eine davon ist, dass oben genannter ‚Noise‘ reduziert wird, indem nur relevante Informationen in wenigen Wörtern, kurzen Sätzen und einzelnen Absätzen verpackt werden<sup>49</sup> — ähnlich wie bei vielen Programmiersprachen. Dennoch sind die Bedeutungen, Symbole und Zahlen zu komplex und verschachtelt und können von betroffenen Personen meist nicht verstanden werden<sup>50</sup>. Personen, die eher Schwierigkeiten mit emotionalen und sozialen Informationen haben, könnte das Programmieren insgesamt verständlicher vorkommen und einfacher fallen.

„esolangs“ ist eine Programmiersprachen-Sammlung einer Internet Community, die seit 1993 besteht und Programmiersprachen von der Praktikabilität lösen wollen und eher esoterische Ziele verfolgen. Der Begriff besteht aus „esoteric“ und „language“. Verschiedene Projekte werden auf einer Seite gesammelt die die Programmierer:innen auf- und herausfordern über Coden nachzudenken. Daraus entstehen künstlerische, spekulative und konzeptuelle Arbeiten die als Gegenbewegung zum kommerziellen Coden stattfinden.<sup>51</sup> Dadurch dass assoziativ, emotional und gegen eine klare Logik gearbeitet wird, sind die meisten hier hochgeladenen Sprachen nicht für außenstehende verständlich und gehen eher auf ein sehr persönliches Bedürfnis des:r entsprechenden Künstler:in ein, ohne ansprechend für Leser:innen zu sein. [Abb.7]

## 8. Was willst du später werden? — Coder oder User sein

Ein häufiges Argument, wenn es um die Demokratisierung durch das Internet geht, ist, dass weltweit sehr viele Menschen Zugang zu einem elektronischen Gerät haben und somit auch die Möglichkeit zu coden. Es gibt Kampagnen und Projekte, die versuchen, durch die Verteilung von Laptops, Tablets und Desktops an Orten, die keine ausreichenden finanziellen Ressourcen haben, zugänglicher zu machen. Dies geschah zum Beispiel beim Projekt OLPC (One Laptop Per Child) an Schulen.

Dieses non-profit Projekt wurde 2005 ins Leben gerufen und 2014 beendet. Das Ziel war durch die Bereitstellung und Finanzierung von technischen Geräten die Bildung von Kindern von ungefähr 9 bis 12 Jahren weltweit

49 Leichtfuß, Anne: Leichte Sprache. (Vortrag, Januar 2023). Klasse Digitale Grafik, Field trip: Simple Language.

50 ebd.

51 [https://esolangs.org/wiki/Main\\_Page](https://esolangs.org/wiki/Main_Page) (Stand: 01.03.2024)

zu fördern.<sup>[Abb.8]</sup> Das Team und Sponsoren nahmen sich zum Ziel, durch ihre ausschließlich dafür designten Geräte Besitz, Ungleichheit und Ungerechtigkeit auf globaler Ebene zu überwinden.<sup>52</sup> Auf dem ersten Blick war dies ein Schritt in Richtung Demokratisierung und Chancengleichheit. Doch wer ein solches Endgerät besitzt, ist eher Konsument:in als Produzent:in oder (Mit)Gestalter:in. Es wurden bigTech und vor allem Us-Produkte zwangsläufig an Orte eingeführt und implementiert, an denen es noch keine gab. Eine schlussfolgernde Frage wäre in diesem Fall: Können die Menschen eine eigene Form der Digitalisierung wählen und sich gegen die Adaptierung der gelieferten Softwareprodukte entscheiden?

Weltweit werden Plattformen von Google, Microsoft, Amazon und so weiter zum Coden genutzt. Diese bieten die Möglichkeit eines Netzwerks, Infrastruktur, Clouds und meist reibungslos funktionierende Systeme – aber auch Kontrolle, Abhängigkeit und Dominanz.

OLPC sollte 2008 ein neues Betriebssystem von Microsoft bekommen, welches viele Gegenstimmen – auch innerhalb des Teams – erhielt. Die hauptsächliche Kritik war, dass der Fokus von Laptop Lernen zu Laptop ‚Penetration‘<sup>53</sup> überging.<sup>54</sup>

Es ist fragwürdig, dass es immer wieder Projekte gibt, die ihren Fokus auf den Globalen Süden richten und die Bedürfnisse der Betroffenen in der Planung und Design nicht berücksichtigen, sondern gar als Experiment ihrer eigenen Technologien nutzen, um dann nach dem Scheitern wieder zu verschwinden. Es ist herablassend und rassistisch, für Personen das Beste zu wissen und zu wollen, anstatt sie teilhaben und entscheiden lassen, insbesondere wenn es um durch und durch koloniale Strukturen von Plattformen geht.

*„However, OLPC is an exemplar of the tension between global capitalism and philanthropy in the developing world, and technological hopes, dreams and desires. It also illustrates that the developing world no longer has to blindly accept American technological efforts.“<sup>55</sup>*

Bei einer Umfrage aus dem Jahr 2023 von ungefähr 90.000 befragten Coder:innen geben 41% an, einen Bachelor-Abschluss, 10% einen Schulabschluss und 13% einen anderen schulischen Abschluss abgeschlossen zu haben. 23% absolvierten ein Masterstudium und 4,5% haben einen

52 Fouché, Rayvon: *Race After the Internet From Black Inventors to One Laptop Per Child*. Routledge, 2012. S.61–83.

53 Penetration hier: Vermarktung und Marktdurchdringung.

54 Fouché, Rayvon: *From Black Inventors to One Laptop Per Child. Race After the Internet*. Routledge, 2012. S.77.

55 ebd.

Doktortitel. Trotzdem geben über 80 % der Befragten an, Coding mit Online Ressourcen gelernt zu haben, also mit Videos, auf Foren und in Dokumentationen.<sup>56</sup> Wer lernen will zu coden, kann es ohne Probleme mithilfe von YouTube Videos oder Coding Foren tun. Und trotzdem scheint es eher wenig Softwareentwickler:innen zu geben, die keinen Abschluss gemacht haben.

Das könnte unter anderem daran liegen, dass es essentiell ist, sich mit Anderen auszutauschen, um an aktuelle oder notwendige Informationen zu gelangen. Dafür ist ein Netzwerk an Programmierer:innen sehr relevant und kann im akademischen und schulischen Umfeld gut etabliert werden. Die Sozialisierung der einzelnen Personen ist ebenfalls wichtig. Personen, denen nicht früh genug die Rolle als Programmierer:in zugetraut wurde, werden höchstwahrscheinlich nicht im Berufsalter anfangen zu coden. Das hängt unmittelbar mit dem Bild des Programmierers und Nerds zusammen, der eher weiß, cis-männlich und objektiv ist. Die finanziellen Ressourcen und lokalen Infrastrukturen müssen ebenfalls einigermaßen stabil sein, um sich zuverlässig mit dem Internet verbinden zu können. Darüber hinaus ist eine funktionsfähige, aktuelle Hardware erforderlich, die das Coden und Eintippen von Zeichen überhaupt möglich macht. Beides kann gegebenenfalls auch eine Schule, Akademie oder Universität bieten.

Die explizit zugängliche und kostenlose Bildung in Informatik an Schulen ist extrem wichtig, damit Frauen, Schwarze und Personen mit knappen finanziellen Ressourcen überhaupt erst die Möglichkeit haben, problematische Vorstellungen darüber, wer programmieren kann, zu ‚unlearnen‘. Wie eben beschrieben, ist es nicht zwangsläufig relevant, dass Personen an Universitäten studieren oder an teuren Weiter- oder Umschulungskursen teilnehmen, sondern dass sie eine Community haben, in der Hardware, Software, Wissen und Informationen geteilt werden können.

## 9. Als Frauen noch Computer waren und heutiger Tech-Sexismus

Durch strukturelle Diskriminierung wurden Programmierer:innen und Coder:innen aus dem Berufsfeld gedrängt – trotz der damaligen vielzahl im Berufsfeld und hohem Grad an Tolerierung. Frauen, queere Personen und einzelne Schwarze Personen wurden als Objekte des Austauschs von

Informationen und Daten toleriert und akzeptiert, wenn sie relevant für das Erstellen von Code oder Rechnen waren. Es resultiert ein Gender Pay Gap in der Tech-Industrie in den USA im Jahr 2022 von 20 %. Schwarze, Native American und Hispanische/Latinx<sup>57</sup> Frauen verdienen 20 %–25 % weniger als weiße Frauen.<sup>58 59</sup>

Aus mehreren Berichten großer Tech- und Computing Unternehmen weltweit lässt sich zusammenfassen, dass ungefähr 15 %–30 % Frauen in den jeweiligen Unternehmen Programmierer:innen sind oder anderen Tech-Tätigkeiten nachgehen.<sup>60 61 62</sup> Ungefähr 50 % der Frauen sind weiß, 20 %–50 % sind asiatisch — in mehreren Unternehmen ist mindestens die Hälfte davon indisches, also 10 %–25 % — nur 4 %–9 % sind Hispanisch/Latinx, 2 %–9 % Schwarz und weniger als 0,7 % sind Natives. In Indien sind die Hälfte der Programmierer:innen Frauen. Sie geben an, dass die Bildung von Programmierung in Indien kaum geschlechtsabhängig ist.<sup>63 64 65 66 67 68</sup>

In den USA gibt es die meisten Programmierer:innen weltweit und davon sind nur 15 % weiblich. Chile hat in dieser Statistik die kleinste Programmiererinnen Rate mit 3 %.<sup>69</sup> Hispanische/Latinx Frauen verdienen mit Abstand am wenigsten in diesem Bereich.<sup>70</sup>

Aktuelle Forschungen und genaue Untersuchungen relevanter feministischer Stimmen haben gezeigt, dass Frauen eine essenzielle Rolle in der Geschichte des Programmierens gespielt haben — auch wenn sich diese nachträglich änderte. Vor allem beim genauen Betrachten der Geschichte des Zweiten Weltkrieges lassen sich nachträglich Belege für die relevante weiblich besetzte Anzahl an Programmiererinnen finden.<sup>71</sup>

57 Hispanisch: Personen aus einem spanischsprachigen Land, insbesondere aus Lateinamerika stammend oder deren Eltern und Großeltern aus einem diesen Ländern stammten.

Latinx: Selbstbezeichnung von Menschen lateinamerikanischer Herkunft. Inklusive und geschlechtergerechte Alternative für Latino/Latina.

58 <https://www.codecademy.com/resources/blog/gender-pay-gap-in-tech/> (Stand: 01.03.2024)

59 <https://www.census.gov/content/dam/Census/library/publications/2020/demo/p60-270.pdf#page=18> (Stand: 01.03.2024)

60 <https://www.statista.com/chart/4467/female-employees-at-tech-companies/> (Stand: 01.03.2024)

61 <https://punkt.de/de/blog/2018/frauen-in-der-softwareentwicklung.html> (Stand: 01.03.2024)

62 <https://de.statista.com/infografik/13283/frauen-in-der-tech-branche/> (Stand: 01.03.2024)

63 <https://code.likeagirl.io/which-countries-have-the-most-skilled-female-developers-b532fc53c7c8> (Stand: 01.03.2024)

64 <https://www.pewresearch.org/science/2021/04/01/stem-jobs-see-uneven-progress-in-increasing-gender-racial-and-ethnic-diversity/> (Stand: 01.03.2024)

65 <https://about.fb.com/news/2015/06/driving-diversity-at-facebook/> (Stand: 01.03.2024)

66 [https://static.googleusercontent.com/media/about.google/en//belonging/diversity-annual-report/2022/static/pdfs/google\\_2022\\_diversity\\_annual\\_report.pdf?cachebust=1093852](https://static.googleusercontent.com/media/about.google/en//belonging/diversity-annual-report/2022/static/pdfs/google_2022_diversity_annual_report.pdf?cachebust=1093852) (Stand: 01.03.2024)

67 <https://blogs.microsoft.com/blog/2020/10/21/microsofts-2020-diversity-inclusion-report-a-commitment-to-accelerate-progress-amidst-global-change/> (Stand: 01.03.2024)

68 <https://de.statista.com/infografik/13283/frauen-in-der-tech-branche/> (Stand: 01.03.2024)

69 <https://code.likeagirl.io/which-countries-have-the-most-skilled-female-developers-b532fc53c7c8> (Stand: 01.03.2024)

70 <https://www.pewresearch.org/science/2021/04/01/stem-jobs-see-uneven-progress-in-increasing-gender-racial-and-ethnic-diversity/> (Stand: 01.03.2024)

71 Wajcman, Judy: Technofeminism. Polity Press. 2004. S.13.

Es sind ausnahmsweise nicht einzelne Akteur:innen zu benennen, sondern Sie sind Teil des Systems und der Geschichte, die ohne Sie wahrscheinlich nie geschrieben worden sei.

Judy Wajcman schreibt unter anderem in ihrem Buch „Technofeminism“ darüber, wie Diskriminierung von Frauen zum Ausschluss der Teilnahme an wissenschaftlichen und technischen Berufen führte. Im Globalen Norden wurde dies durch strukturellen Sexismus an Bildungsstätten durchgeführt. An Schulen und Universitäten wurden Frauen und Männern, mit der damaligen wissenschaftlichen Erklärung, Frauen seien dafür weniger geeignet, andere Themen beigebracht, und sie wurden demnach gezielt anders gebildet. Die zweite Feminismus Welle thematisierte den Mangel an Frauen in Wissenschaft und Technologie und forderte Zugänglichkeit. Die Problematik bestand immer wieder in den gängigen Ideologien des Zeitalters, dass Frauen nicht in der Lage seien, die Symbolik, wie wir sie in jeder Sprache der Technologie vorfinden, zu verstehen und wenn sie diese doch erlernen wollten, müssten sie ihre Femininität aufgeben.

Es spielte ebenfalls eine große Rolle, dass Arbeitsstellen für Frauen geschaffen wurden, bei denen das Erziehen von Kindern kompatibel mit der Karriere war. Erst als die Computerindustrie sich zu einem immer wichtigeren und größeren Teil der Gesellschaft entwickelte, mit dem immer mehr Profit gemacht werden konnte, veränderte sich leicht der Prozentsatz an Frauen.

72 % Frauen geben bei einer Umfrage an, ‚Bro Culture‘ sei allgegenwärtig in ihrem Tech Beruf.<sup>72 73</sup> Es gibt keine strenge definition von ‚Bro Culture‘, doch allgemein kann sie als weiß, männlich, sportaffin, zur Mittel- bis Oberschicht angehörend, konkurrenzgetrieben und exklusiv — Frauen und Schwarze seien aber an sich willkommen, behaupten sie — beschrieben werden.<sup>74</sup> ‚Bro Culture‘ ist eher ein Phänomen der 2000er und anders als der Computer-Nerd handelt es sich eher um ein Bild des modernen Mannes, der durchaus stylish und sportlich sein kann. Die Rolle des Computer-Nerds kursiert seit den 80ern und wird ebenfalls einer männlichen Figur zugeschrieben. In Filmen und Erzählungen scheint es fast so, als würden bestimmte Personen als Nerds geboren werden — als wäre Nerd-Sein eine Art Bestimmung. Der Nerd ist extrem objektiv und einzigartig. Er kann Mathematik- und Informatik-Probleme aufgrund seiner extremen Begabung lösen.<sup>75</sup> Das

72 <https://www.trustradius.com/buyer-blog/women-in-tech-report> (Stand: 01.03.2024)

73 <https://zeitfuerx.de/arbeit/wie-die-bro-kultur-einer-diversen-arbeitswelt-schadet/> (Stand: 01.03.2024)

74 [https://en.wikipedia.org/wiki/Bro\\_culture](https://en.wikipedia.org/wiki/Bro_culture) (Stand: 01.03.2024)

75 Schelander, Esther: Wie Programmieren männlich wurde. Deutschlandfunk Kultur, 2023.

<https://www.deutschlandfunkkultur.de/frauen-informatik-programmieren-geschichte-100.html> (Stand: 01.03.2024)

sind nur zwei Beispiele der medial präsenten Archetypen von Programmierern in den letzten Jahrzehnten.

## 10. Technologie gab's lange vor Geburt Christi

Vielleicht gab es außerhalb der europäischen modernen Gesellschaft die ersten Coderinnen. Das indigene Volk der Maya lebte von ca. 3000 vor unserer Zeitrechnung bis 900 nach unserer Zeitrechnung. Es ist bekannt, dass die Mayas der Mathematik und Zahlensystemen eine große Bedeutung gaben und die Praktizierenden zu den wichtigsten Personen in deren Gesellschaft galten. Anhand zweier untersuchten Vasen der Maya lässt sich entnehmen, dass es unter dem kleinen elitären und prophetischen Kreis der Mathematiker:innen zwei Frauen gab.<sup>76</sup> Es ist belegt, dass Maya Frauen bei spirituellen und wirtschaftlichen Tätigkeiten eine große Rolle spielten. Wenn die als Frauen erkannten Personen tatsächlich Mathematikerinnen waren, wären es wahrscheinlich die ersten weltweit.<sup>77</sup>

Der Quipu oder Khipu bezeichnet eine Zählmethode, die im Inkareich (7. Jh. n.u.Z. bis zur Mitte des 20. Jh.) ihren Ursprung hat. Mithilfe dieser Technik konnte auf vierstichtiger Ebene Informationen in Form von Zahlen bis in den vierstelligen Bereich erfasst werden. Dies ermöglichte die effiziente Verwaltung von Lagerbeständen, die Weitergabe von Produkten und die Durchführung von Zählungen. Durch die Verwendung verschiedener Tierhaare, Färbungen und Knoten Richtungen wurden äußerst komplexe Systeme geschaffen, die wir heute als frühe Technologie verstehen können.<sup>78</sup> [Abb.9]

Wajcman argumentiert, dass das Konzept der Technologie historisch in verschiedenen Epochen und Kulturen eine andere Bezeichnung hatte und sogar etwas anderes bedeutete als das, was wir heute unter Technologie verstehen. Aus heutiger Perspektive beschreibt der Begriff das männlich Technologische und die Tätigkeit von Frauen wurde außerhalb dieses Begriffs verschoben.<sup>79</sup> Technologie entspricht heute nahezu ausschließlich dem Bild der technischen Produkte des weißen, mittelständigen Mannes im Silicon Valley und nicht den gewebten Stoffen der Frau in den Anden

76 Gheverghese Joseph, George: A Brief History of Zero. Research Gate, 2008.  
[https://www.researchgate.net/publication/228783983\\_A\\_Brief\\_History\\_of\\_Zero](https://www.researchgate.net/publication/228783983_A_Brief_History_of_Zero) (Stand: 01.03.2024)

77 Wegen Kolonialgeschichte nicht mehr nachvollziehbare Daten.

78 Museo Chileno de Arte Precolombino: EXPOSICION. Quipu: Contar anudando en el Imperio Inka.  
[https://precolombino.cl/mods/expo\\_temporales/detalle/43/html/005a.htm](https://precolombino.cl/mods/expo_temporales/detalle/43/html/005a.htm)

Lateinamerikas. Ein Beispiel von marginalisierter Technologie ist *Untitled* von Marilou Schultz [Abb.10] aus der Navajo-Kultur. Ihre Arbeit macht die enge Verbindung von Computerchips und gewebten Stoffen sichtbar, die nicht nur in der Geschichte von Navajo-Frauen in Intel-Fabriken geprägt wurde, sondern auch in ihrer technologischen Praxis des Webens<sup>80</sup>. Diese hat eine Ähnlichkeit zur Arbeitsweise beim „Apollo Guidance Computer“ von 1966, der eine ‚core rope memory‘ besaß, die von Frauen mit Nähkenntnissen aus Textilindustrien hergestellt wurde<sup>81</sup>. [Abb.11]

In manchen traditionellen Praktiken war die Idee der Programmierung ein fester Bestandteil lange vor der Erfindung des Computers. Zum Beispiel Stricken, Weben, Kochen, Musik und Poesie folgen einer Art System und sind oft nicht nur eine trockene ‚wenn-dann‘ oder ‚wiederhole-bis‘ Funktion, sondern haben auch eine spirituelle Bedeutung. Die Definition von Coding und Technologie scheint demgegenüber stark verankert mit Kolonialen Ideen der Moderne wie Effizienz und Objektivität. Tools werden oft spezifisch für große Unternehmen für viel Geld und mit großem Aufwand entwickelt statt für ein soziales Netz und nicht-kommerzielle Zwecke. Das, was wir heute unter Coding verstehen, ist weit weg vom Alltag oder gemeinschaftlicher, familiärer Arbeit, sondern eher verschlüsselt, kryptisch und verfremdet.

## 11. Rassistische Codes und koloniale Plattformen

Eine Plattform zu besitzen bedeutet, eine Software zu besitzen, also Codezeilen. Google besitzt 2 Billionen Codezeilen, die einen digitalen Ort zum Interagieren von Privatpersonen oder Unternehmen schafft. Daten werden erhoben und kontrolliert. Das heißt konkret: Codezeilen werden in großem Umfang geschrieben, um einen Mechanismus der Extraktion und Erhebung von Daten zu etablieren.<sup>82</sup>

Was einst während der Kolonialisierung vor 1945 durch die Entnahme wertvoller Ressourcen und den Aufbau von Infrastruktur mit guter Absicht geschah, wiederholt sich heute mit dem Einsatz großer digitaler Infrastruktur, die den Menschen im Alltag helfen soll. Kolonialisierung ist nicht

80 Hopkins, Candice: Marilou Schultz. Documenta 14, 2017.

<https://www.documenta14.de/en/artists/22610/marilou-schultz> (Stand: 01.03.2024)

81 Ken Shirriff's blog: Software woven into wire: Core rope and the Apollo Guidance Computer.

<https://www.righto.com/2019/07/software-woven-into-wire-core-rope-and.html> (Stand: 01.03.2024)

82 Kwet, Michael: Digital colonialism. US empire and the new imperialism in the Global South. Institute of Race Relations, 2019.

nur die aggressive Oppression, Raub und Dominanz, sondern auch die Legitimierung dieser eingesetzten Strukturen durch Ideologien und Versprechen, die Widerstand unbedeutend machen.<sup>83</sup>

Weltweit werden lokale Unternehmen durch globale ersetzt. Vor allem im Globalen Süden hat es fatale Folgen, denn die Abhängigkeit von digitalen Plattformen ist unendlich groß, nachdem die lokalen Strukturen und Unternehmen verdrängt werden. Die üblichen Beispiele kennen wir alle: Lokaler Transport=Uber, Grab; Soziale Netzwerke=Facebook, Instagram, X; Werbung=Google, Facebook; Film, Kino, Video=Netflix, YouTube, Amazon, Disney; Geräte und deren Systeme=Apple (iOS), Microsoft Windows, Android.

Nie war es einfacher und schneller, etwas zu verkaufen oder zu konsumieren. Digitale Plattformen erreichen die größte Form von Konzentration und Monopol. Die zugrunde liegenden digitalen Technologien und der damit verbundene Code können nun als ein neuer exponentieller Faktor betrachtet werden, der zur Spaltung der Gesellschaft beiträgt. In Klassengesellschaften können sie zudem den gegenwärtigen Zustand unterstützen und bereits etablierte Strukturen der Ungleichheit weiter festigen.

Ein wesentlicher Widerspruch dieser Unternehmen wie zum Beispiel ‚Meta‘ (Instagram, Facebook, X) oder ‚OpenAi‘ (ChatGPT, Dall-E) ist, dass sie vorgeben, sozial gesellschaftlich geformt und deswegen natürlicherweise unpolitisch seien — ein Spiegel der Gesellschaft zu sein. Sie behaupten, der freie Markt würde sie formen und sie würden lediglich den Regeln des Marktes folgen, obwohl die tatsächliche Monopolisierung dem Konzept des freien Marktes entgegen läuft. Die eigenen Methoden der Manager:innen und Autor:innen des Codes sind allerdings gesamtpolitisch geprägt und übertragen ihre Werte zwangsläufig auf ihre Arbeit an den Softwares. Obwohl sie teilweise vorgeben demokratisch zu sein, sind sie gewollt oder ungewollt immer definiert oder gar manipuliert durch die Expertise der Personen, die Code unzugänglich und undurchsichtig machen. Wenn wir eine kleine, selbst gemachte Website besuchen, würden wir oft mit den einfachsten Grundlagen von HTML verstehen, wie die Seite aufgebaut ist. Ganz anders sieht es zum Beispiel bei der Website von ‚Airbnb‘ aus, bei der überhaupt nicht mehr ersichtlich ist, woher welche Information kommt und welche Attribute den einzelnen Elementen gegeben werden. Was sonst als bloße objektive und neutrale Technologie dargestellt und betrachtet wird, kann deshalb unter anderem durch feministische und dekoloniale Theorien subjektiviert und anschließend global kontextualisiert werden.

Diese Prozesse der Objektivierung spitzen sich bei Künstlicher Intelligenz zu. Diese besteht aus Algorithmen, also automatisierten Anweisungen in Form von Code und Daten. Algorithmen lernen gesellschaftliche und soziale Unterschiede wie wir Menschen sie kennen und reproduzieren diese. Sie werden so programmiert, dass sie der Maschine unendlich viele Daten beibringen, um Systeme zu erkennen: Gesichter, Satzbildung, Schönheitsideale, Straftaten und so weiter. Die Daten werden meist allerdings nicht strukturiert, geordnet und sauber eingegeben, sondern werden fast ungefiltert verarbeitet. Die Algorithmen sind hier alles andere als Objektiv und steril. Wenn wir als Gesellschaft bestimmte Menschen strukturell aufgrund bestimmter Merkmale und Lebensweisen ausschließen, tut es ein Algorithmus ebenfalls, da es von Menschen erhobenen Daten lernt. Folgen dessen könnten sein, dass Algorithmen anhand der Daten schon einen festgefahrenen Blick und eine damit einhergehende Einstellung entwickeln, die es unmöglich macht das Tatsächliche zu sehen, sondern nur das, was diese bereits wissen.

Die Vermessung und Archivierung von menschlichen Körpern wurde zu Zeiten der Kolonialisierung vor 1945 vor allem genutzt, um Narrative und Bilder rassifizierter Körper, die Ausgrenzung dieser Körper und die Herrschaftslegitimation weißer Körper möglich zu machen.

Moderner Kolonialismus bedeutet auch die Ausbeutung von Menschen auf Kosten ihrer mentalen Gesundheit für den Wohlstand des Globalen Nordens. Das Arbeitsumfeld und das private Leben der meist weißen, hetero cis-männlich Programmierer:innen von Plattformen ist eher wohlhabend und luxuriös. Moderator:innen von Inhalten dieser Plattformen arbeiten meist im Globalen Süden für ein vielfach geringeres Gehalt von 2 USD pro Stunde, sehen den verstörendsten, gewalttätigsten Content an und regulieren diese nach fremd erstellten Regeln. Die meisten Moderator:innen leben und arbeiten unter schlechten Bedingungen und leiden unter extremen negativen psychischen Folgen, wie Depression, posttraumatische Belastungsstörung und Medikamentenmissbrauch. Sie müssen zusätzlich moralische und soziale Regeln des Westens beziehungsweise Globalen Norden annehmen und ihre eigenen ignorieren.<sup>84</sup> Die heute verbesserte weltweite Vernetzung durch Internetzugang weltweit bedeutet ebenfalls, dass mehr Arbeitsplätze geschaffen werden können, die zu mehr Ausbeutung und unfairen Arbeitsbedingungen im Globalen Süden führen. Die meisten Firmen hinter globalen Plattformen nutzen ihre Technologien, um wirtschaftliche

84 Zuazo, Natalia: #153 ChatGPT, la Carrera de la inteligencia artificial y la desigualdad. El Hilo de Radio Ambulante Estudios, 07.04.2023.

Vorherrschaft zu erlangen, die in lokale Strukturen und sogar in die Regierungen des Globalen Südens eingreifen können.

Wenn wir wieder auf Coding blicken, finden wir ähnliche Strukturen in der manuellen Arbeit, also im tatsächlichen Entwickeln und Coden. Diese Aufgaben werden aus Deutschland für ungefähr 75 %– 40 % weniger Gehalt an Unternehmen in Osteuropa, Indien oder Vietnam outsourced.<sup>85</sup> Während im Globalen Norden Firmen designen, planen und entwerfen, werden Einzelpersonen oder Unternehmen, die Mitarbeiter zur Verfügung stellen, für das günstige und schnelle Schreiben und Testen von Code hinzugeholt. Da die Arbeit nicht unbedingt persönliche off- oder online Meetings erfordert und über das Hin und Her schieben von Designdateien oder Code erfolgen kann, können die Arbeitsbedingungen häufig unbekannt sein.

Kleinere Web Projekte, die selbst programmiert werden können, werden wahrscheinlich sehr selten oder gar nicht outsourced.

## 12. Trial and Error — Wo Dekolonialisierung anfängt und aufhört

„Postkoloniales Computing“ ist eine Antwort auf das Thema dekoloniales Programmieren. Dabei wird der Begriff kritisiert, da „post“ impliziert, dass die Kolonialisierung abgeschlossen wäre und es eine darauf folgende Zeit gäbe. Die Kolonialisierung wurde jedoch durch die anhaltenden Machtstrukturen und Abhängigkeiten vom Globalen Nordens nie beendet. Demnach wäre der Begriff Postkolonial nicht zutreffend. Wichtige Dekoloniale Stimmen wie Edward Said, Gayatri Spivak und Homi Bhabha beziehen den Begriff „Postkolonial“ auf eine Zeit der Kämpfe gegen die Kolonialisierung oder der Erkämpfung der Unabhängigkeit. Der Kampf um Erbe, Geschichten und kulturellen Formen wird weitergeführt und das bedeutet, dass es eine kritische Auseinandersetzung mit dem Erbe und der Hinterlassenschaft aus der Perspektive der damals kolonisierten heute geben muss.<sup>86</sup>

An sich möchte „Postkoloniales Computing“ genau das tun und kulturelle und politische Fragen in den Bereichen Informatik, Designmethoden, Informations- und Kommunikationstechnologien stellen. Doch es gibt mehrere nennenswerte Kritikpunkte. Anfangs wurde „Postkoloniales Computing“ wegen eines gewissen Stils, Depolitisierung und der Neigung zu Kultur

85 <https://www.objectbay.com/blog/outsourcing-softwareentwicklung> (Stand: 01.03.2024)

86 Mustafa Ali, Syed: A brief introduction to decolonial computing. Research Gate, 2016.

als nicht hinreichend objektiv kritisiert worden. Heute wird die individualistische, subjektive Wahrnehmung, die es nicht schafft, die Komplexität und Zusammenhänge der Globalisierung und Kapitalismus zu verstehen, kritisiert. Die Zusammenhänge von Rassismus, Kapitalismus, Sexismus, Klassismus und Kolonialisierung sind essentiell und können deshalb nicht von Ländern und Institutionen auf Individuen delegiert werden. Das vielleicht problematischste an ‚Postkoloniales Computing‘ ist, dass sich Postkolonialität auf Ideen von Foucault, Lacan und Derrida fokussiert, doch die Auseinandersetzung mit Theorien aus dem Globalen Süden und die Verwendung ihrer Begriffe vernachlässigt. Dies macht das Konzept des postkolonialen Computings anfällig für den Vorwurf, in einer Schleife gefangen zu sein, die intern in Europa verblieb. Letztendlich sei die postkoloniale Theorie eine eurozentrische Kritik am Eurozentrismus selbst, wobei Befürworter des ‚Postkolonialen Computings‘ zumindest teilweise diese Problematik eingestehen.<sup>87</sup> Wenn wir den Kerngedanken aus Spivaks Essay „Can the Subaltern Speak?“ einbeziehen, würden zum Beispiel Schwarze Frauen, die programmieren, also das ‚Postkoloniales Computing‘ ausüben, keine Subalternen mehr darstellen, weil sie bereits den Zugang zu Werkzeugen und Infrastruktur haben.<sup>88</sup>

Trotzdem kann das aktive Einbeziehen von marginalisierten Personen einen erheblichen Teil zur Dekolonialisierung im Code beitragen. Es gibt jedoch dekoloniale Stimmen, die zurecht eher vorsichtig mit einer Quote umgehen würden, denn zum Einen ist es unmöglich durch das Einbeziehen von Einzelpersonen, die einer marginalisierten Gruppe angehören, tief verwurzelte strukturelle Systeme zu lösen. Zum Anderen sind die Einzelpersonen eventuell aus vielen möglichen Gründen nicht in der Lage, systematische Unterdrückung zu erkennen oder zu handhaben, was für sie selbst und andere marginalisierte Personen eine Gefahr darstellen kann. Die Hauptfrage in Unternehmen und Institutionen ist fälschlicherweise: Warum treten marginalisierte Personen nicht in die Informatik ein? Statt: Was sollte getan werden, damit insbesondere marginalisierte Personen in die Informatik einsteigen und sich eingebunden und gefördert fühlen?<sup>89</sup>

Für eine wirklich fundamentale Änderung im Feld des Codings ist eine Umstrukturierung erforderlich, die nicht nur punktuell marginalisierte Personen einbezieht, sondern die Bereitschaft nach außen trägt, grundsätzlich Arbeitsabläufe und Strukturen mit diesen Personen neu zu verhandeln.

87 Mustafa Ali, Syed: A brief introduction to decolonial computing. Research Gate, 2016.

88 Spivak, Gayatri Chakravorty: Can the Subaltern Speak? [https://abahlali.org/files/Can\\_the\\_subaltern\\_speak.pdf](https://abahlali.org/files/Can_the_subaltern_speak.pdf)

89 Birhane, Abeba; Guest, Olivia: Towards decolonising computational sciences. Women, Gender & Research, 2021. S.6. [https://pure.mpg.de/rest/items/item\\_3287104\\_1/component/file\\_3287105/content](https://pure.mpg.de/rest/items/item_3287104_1/component/file_3287105/content) (Stand: 01.03.2024)

# 13. World white Web

Die Dekolonialisierung des Designs besteht darin, eine politische Praxis zu etablieren und Klassifizierungen wie gut oder schlecht, richtig oder falsch, billig oder hochwertig aufzulösen. Beim Coden eines Designs sind durch etablierte Programme und deren Sprachen Regeln und Bedingungen von Hard- und Software bereits weitgehend vordefiniert. Äquivalent dazu wären bei der analogen Gestaltung eines Buches die Beschaffenheit des Materials des Endproduktes sowie die Tools zum Entwerfen, wie zum Beispiel die Ausstattung der Bleisatzwerkstatt oder die Werkzeuge bei InDesign. Im Falle einer Website wird Code geschrieben, damit zum Beispiel ein Browser ihn lesen, interpretieren und ausführen kann. Der Browser ist hier also der Hauptrezipient, der nur eine ganz bestimmte Art der Beschreibung versteht und dadurch nicht alle Vorstellungen und Wünsche umsetzen kann. Tools wie Templates oder Nodes<sup>90</sup> können dabei helfen, dem gewünschten Design und Aufbau näher zu kommen, doch das kann eventuell eine Einschränkung in den Designentscheidungen bedeuten. Je mehr selbst gecodet wird, desto experimenteller oder präziser kann gestaltet werden. Dies kann sich teilweise deutlich auf die Benutzeroberfläche einer Website auswirken.

Es ist jedoch unmöglich, über Design in diesem Kontext zu sprechen, ohne ein wenig in die Geschichte der Ästhetik einzutauchen. Das europäische Subjekt hat zu Zeiten der Kolonialisierung und Sklaverei die Sprache der Universalität und Objektivität erschaffen, die vollkommen sein sollte. Damit einhergehend fand die Institutionalisierung der Ästhetik statt, die sich als Instrument des Adels und Bürgertums entwickelte, um sich von unteren Schichten und vor allem von den Kolonialisierten abzugrenzen. Nur bestimmte Menschen waren in der Lage, intellektuelle Bestimmungen zu (re-)produzieren, denn die Anderen mussten die Rolle einnehmen, diese zu empfangen und wahrzunehmen. So betrachtet ist die Ästhetik aus dem Bedürfnis geboren, Unterschiede und Ausgrenzung zu erzeugen. Der weiße Universalismus hatte durch die Einführung der Objektivität klare Grenzen und war sehr spezifisch, wodurch es genau das Gegenteil versuchte, aufrecht zu erhalten, nämlich eine dauerhafte selbstgerechte Art, subjektive Meinungen als eine Objektive auszugeben.<sup>91</sup> Diese Überlegenheit des Globalen Nordens durch Epistemologien von Ästhetik ist einer der Grundbausteine des heutigen Kapitalismus und hält bis heute an.

<sup>90</sup> Node-based-programming: Programmieren indem Code-Blocks oder Code-Schnipsel miteinander verbunden werden. Jeder Block steht für ein oder mehrere Befehle.

<sup>91</sup> Gikandi, Simon: *Slavery and the Culture of Taste*. Princeton University Press (Reprint Edition), 2014. S. 42.

*„By the same token, the colonial experience had been consigned to the domain of political economy by the beginning of the 19th century. The separation between aesthetics and political economy has been firmly established. My assumption here is simply that any attempt to separate aesthetic questions from ideological ones was doomed to fail because the two categories were products of the same moment, indeed operated within the same economy of discourse. It is not by accident that the foundational moment of modern colonialism — the 18th century — is also the age of the aesthetic.“<sup>92</sup>*

Es gibt heute eine stark verankerte standardisierte Ästhetik der Benutzeroberfläche im WWW, die vom Globalen Norden heraus in fast alle Ecken der Welt verbreitet wurde. Als Beispiel nenne ich hier das Template „Hales“<sup>[Abb.12]</sup> bei Squarespace, einem der beliebtesten Anbieter für All-Inclusive-Websites weltweit, die Design-Templates, Hosting<sup>93</sup>, Domains und so weiter im Paket anbieten. „Hales“ ist eine sehr übersichtliche und aufgeräumte Website. Helle Farben, große Weißräume, seriflose Schrift und dogmatische Designentscheidungen machen sie aus. Websites wie diese wurden durch unschlagbare Preise und einfaches, funktionierendes Design zum weltweiten Design-Standard, der reproduziert und nachgebaut wird. Diese Ästhetik wird im akademischen Kontext als kultur- und identitätslos wahrgenommen, wenn sie vom Globalen Süden verwendet wird — als Nachahmung.

*„Why are we so fascinated with ‘history’ and with the ‘native’ in ‘modern’ times? What do we gain from our labor on these ‘endangered authenticities’ which are presumed to be from a different time and a different place?“<sup>94</sup>*

Velleicht kommt die Kritik aus dem bewussten oder unbewussten Bedürfnis für die ‚universelle‘ weiße Ästhetik ein Anderes, einen Gegenpol zu brauchen. Das Unschöne, Unästhetische, nicht Durchdachte und nicht Reale im Globalen Süden sehen zu können.

Der voreingestellte Hintergrund einer Website basierend auf ein nicht beschriebenes HTML Dokument ist immer ‚transparent‘. Dahinter ist der Hintergrund weiß — by default. Die Unsichtbarkeit und Transparenz von weiß sein wird bei den meisten Medien als Ausgangspunkt anerkannt. Ob HTML-Datei, White Cube oder der weiße Leinwand — weiß führt das dazu, dass Schwarz sein als das Andere verurteilt wird.

92 Gikandi, Simon: *Aesthetic Reflection and the Colonial Event: The Work of Art in the Age of Slavery*. Michigan Publishing, 1997. <https://quod.lib.umich.edu/j/jii/4750978.0004.306?view=text;rgn=main> (Stand: 05.03.2024).

93 Webhosting: Macht eine Website im Internet verfügbar, indem es Server für die Dateien zur Verfügung stellt.

94 Chow, Rey: *Writing Diaspora. Tactics of Intervention in Contemporary Cultural Studies*. Indiana University Press, 1993.

Eine Design-Identität im Web losgelöst vom europäischen und Us-amerikanischen Standard ist nicht einfach zu finden, da die digitale Ästhetik aus einem tief kapitalistischen System des Globalen Nordens entstanden ist. Unlearning dieser Ästhetik und der Grundregeln des Designs, die hier in Form von Code schwer aufzubrechen sind — Pixel, Grids, Zahlen, Gleichungen —, sind Entscheidungen, die nur Gestalter:innen und Coder:innen selbst treffen können und aus eigener Initiative entstehen müssen. Wenn das nicht getan wird, laufen wir im Globalen Norden Gefahr, dass Regeln dem Globalen Süden immer wieder aufgedrungen werden und Dominanz aufrechterhalten wird. Wenn Personen oder Institutionen des Globalen Nordens unauthentische Websites des Globalen Südens als Nachahmung sehen, statt eine Notwendigkeit, mit der Modernisierung Schritt halten und sich zwangsläufig dem Globalen Norden zuwenden zu müssen, ist das Überheblich.

Grundsätzlich wäre es eher ein Lösungsansatz, die eigene europäische oder Us-amerikanische Designgeschichte zu analysieren und dann festzustellen, dass die Identitäts- und Kulturlosigkeit eine Folge des Kolonialismus, Rassismus, Sexismus und Klassismus war und ist. Macht kann in diesem Sinne nur aufgebrochen werden, wenn wir erst einmal akzeptieren, dass die Machtverhältnisse in digitalen Systemen und in der Ästhetik beibehalten und durch globale Plattformen verstärkt werden — manchmal sogar genau durch die Videos und Foren, die uns beibringen zu coden.

Das Ziel des Globalen Nordens sollte nicht sein, sich der Praxis des Globalen Südens anzunähern oder gar kulturell anzueignen, einen Kompromiss oder eine Inspiration zu finden, da wir Gefahr laufen würden, dass wir etwas neues erschaffen wollen und eine bessere Lösung herbeiführen wollen. Es geht eher darum, eine Eigenkritik zu schaffen und zu hinterfragen, woher die vorliegende Ästhetik kommt und ihre Verbindungen in der Vergangenheit zu erforschen. Der Ausmaß der Gewalt westlicher Ästhetik ist sehr lange ungestört geblieben, trotz vorgeschlagener, nicht westlicher Ansätze, die eher als primitiv, undifferenziert und minderwertig galten.<sup>95</sup>

95 Sonderegger, Ruth: Zur Kolonialität der philosophischen Ästhetik. Ruth Sonderegger im Gespräch mit Maja Figge und Wilma Lukatsch. In: Künste dekolonisieren. Ästhetische Praktiken des Lernens und Verlernens. Transcript, 2023. S.55.

# 14. Academic Coding a. k. a. Inaccessible Contemporary Sites

*„When you’re constantly diving through the world wide web like I am, you can’t help but recognize that there is a new trend in graphic design and technology at every corner. But driven by the fear of missing out, I learned that following these trends won’t help you expand your creative potential. Always cultivating proactive instead of reactive behavior, in 2013 I decided to follow my principles, overcome my fear of programming and explore the realms of Creative Coding. This bold decision guided me into unexplored territories [...]“*<sup>96</sup>

Bei genauer Betrachtung von Creative Coding-Projekten wird schnell deutlich, dass eine Gegenbewegung oder sogar eine kritische Haltung gegenüber konventionellen Websites erkennbar ist. Creative Coding-Projekte sind oft von Hand gemacht, stark grafisch ausgerichtet und dekonstruktiv. [Abb.13] Sie haben weniger den Zweck, bestimmte Informationen zu vermitteln, sondern jegliche Art von Daten in Form von Bild, Text oder Sound so zu verarbeiten, dass etwas Abstraktes entsteht. [Abb.14] Wie im oben angeführten Zitat eines Creative Coders, wird hier gegen Trends eigenen Prinzipien gefolgt. Assoziationen werden nicht in aktuellen gesellschaftlichen Tendenzen gesucht, sondern im eigenen individuellen geübten Wissen und Verständnis. Bei diesen Projekten setzt sich ein Individualismus durch, der die professionelle Bildung perfektioniert und manifestiert. Skeptisch sein klingt hier oft im Hintergrund nach. Je besser geglaubt wird kritisch in der Theorie als auch in der Praxis zu arbeiten, desto größer wird die Kluft zwischen dem privaten Individuum und dem des sozialen Wesen.<sup>97</sup> Ähnlich ist es bei Webseiten, die im Kontext von Kunsthochschulen, Ausstellungen, Kunstinstitutionen, Designbüros und Portfolios entstehen. [Abb.15]

Die Idee, etwas unzugänglich und kryptisch zu gestalten, ist vor allem im akademischen Rahmen verbreitet, wo davon ausgegangen werden kann, dass sich User:innen lange und ausführlich mit dem Produkt auseinandersetzen und Verknüpfungen zu bereits erlernten Theorien herstellen können. Diese Praxis ist jedoch ein Privileg und hängt eher mit finanziellen Ressourcen und früh erlernten sozialen Codes zusammen als mit einer vermeintlichen Underground-Idee. Kritik am

<sup>96</sup> <https://timrodenbroeker.de/about/> (Stand: 01.03.2024)

<sup>97</sup> Harney, Stefano; Moten, Fred: Die Undercommons. Flüchtige Planung und schwarzes Studium. Transversal Texts, 2016. S.19.

,Mainstream‘ kann hier als Form der Diskriminierung betrachtet werden.

Problematisch an diesen Ansätzen ist insbesondere, dass durch eine kryptische Designsprache und das Aufbrechen von Designregeln Barrieren geschaffen werden können. Vor allem Schrift, Farbkontraste und Aufbau sollten im Design und Code für Barrierefreiheit berücksichtigt werden. Sich schnell bewegende Objekte, Animationen und Flackern im Design können die Lesbarkeit beeinträchtigen. Assoziative Symbole und Elemente können je nach kulturellem Hintergrund oder Sozialisierung nicht verständlich sein. Anstatt mit klar verständlichen Mitteln im Design und Code zu arbeiten, wird bewusst abstrahiert und verschachtelt.

Es gibt jedoch auch handgemachte Websites mit künstlerischen Ansätzen, die bewusst dem ,Mainstream‘ entgegenwirken und darauf bedacht sind, verständlichen Code zu schreiben und zugängliches Design zu gestalten. Künstler:innen im Web sollten sich bewusst sein, dass Abstraktion, Stil und Trends Barrieren darstellen können. Etwas Wertvolles, was künstlerische Websites bieten können, ist, dass sie neue Möglichkeiten durch Experimente vorschlagen können. Neue Verknüpfungen, Zusammenhänge und Perspektiven an der Schnittstelle des technischen und künstlerischen sind für den Fortschritt Richtung dekoloniale, intersektionale Websites notwendig. Ein kreativer und technischer Hack<sup>98</sup> kann etablierte problematische Strukturen aufbrechen, erfordert jedoch oft mehr Aufmerksamkeit und Auseinandersetzung der User als ein standardisiertes Webdesign. Künstlerische Projekte, die zum Beispiel bei esolangs.org hochgeladen werden, bei denen neue Programmiersprachen oder Funktionen erschaffen werden, sind ebenfalls auf den ersten Blick eine Chiffrierung von Code. Allerdings funktioniert diese Plattform ,Open Source‘ und erläutert die Vorgänge und Anwendungsbereiche.

Es lassen sich hauptsächlich drei Tendenzen identifizieren: verständliches Design und unzugängliche Codes versus unzugängliches Design und verständliche Codes versus Websites, bei denen beides unzugänglich und unverständlich ist. Alle drei sind wahrscheinlich mindestens stückweise letztlich kapitalistische Produkte, die auf die Vermarktung von Artikeln oder des eigenen Individiums und dessen Umfeld abzielen. Es ist fragwürdig, dass eines dieser Modelle an einer inklusiven Arbeitsweise und gegenseitigem Verständnis interessiert ist, vor allem nicht letzteres.

98 Hack: Das Produkt einer Funktionserweiterung oder Problemlösung, bei der das Ziel auf eine ungewöhnliche, originelle Weise erreicht wird.

Nostalgische Websites, die die Ästhetik des Internets der 90er Jahre aufgreifen, sind im Internet weit verbreitet und können heute ebenfalls als eine Art Protest verstanden werden. Sie zeichnen sich durch GIF's<sup>99</sup>, Glitzer, Hintergrundbilder und penetrante Pop-Up-Fenster aus — eine Ästhetik, die von zahlreichen Internetkünstler:innen seit drei Jahrzehnten vorangetrieben wird. [Abb.16] Viele dieser kleinen, süßen und skurrilen Details sind jedoch aus dem Internet verschwunden, seit Social-Media-Seiten aufgekommen und All-Inclusive-Webangebote gegründet wurden. Heute kann selbstgecodetes Web unabhängig zu publizieren eine Hürde sein, weil FTP's<sup>100</sup>, Berechtigungen und Hosting komplizierter sind als in den 2000ern. All-Inclusive-Webangebote machen das jedoch so einfach, sodass jede Person ohne großes Know-How theoretisch publizieren kann [Abb.17], wenn sie ihre Angebote nutzen. Es stellt sich die Frage, warum es bei diesen Projekten so wichtig ist, dass jedes Detail selbst entschieden und Selbst gemacht aussieht, wenn der Grundgedanke von Programmierbarkeit ist, dass Automatisierungen entstehen und sowohl dem Nutzer als auch dem Entwickler das Leben erleichtern. Der Protest liegt gegebenenfalls genau in dieser Aktion: Es fühlt sich natürlich nicht handgemacht an, wenn eine Website von Anfang an vollkommen makellos standardisiert gestaltet ist, sondern nur dann, wenn Elemente sich ‚anfühlen‘ als wären sie von einer echten Person bestimmt worden.<sup>101</sup> Vielleicht gibt es eine feine Grenze, bei der eine Website noch liebevoll genug gestaltet ist und nicht lieblos generiert wirkt, und so dem geregelten Verhalten und vorhersehbaren Designs der Maschine neuen frischen Wind verleiht.

Den User:innen und Programmierer:innen, die in den 90ern im Web unterwegs waren, wird die damalige Ästhetik bekannt sein. Dennoch handelt es sich bei heute publizierten 90er Websites um eine Design-Ästhetik und gelegentlich um technische Tricks aus einer Nischenblase, die hauptsächlich an sich selbst adressieren. Wer diese Ästhetik und insbesondere die Referenzen verstehen und reproduzieren kann, benötigt nicht nur eine Community, die über die konzeptuelle technische Substanz und deren Tricks und Hacks Bescheid weiß, sondern auch ein intellektuelles Netzwerk, das die Meta-Ebene, die Überspitzung und die Ironie eines solchen Konzepts versteht. Darüber hinaus ist der Preis und die Zeit für eine solche handgemachte Website höher und bietet höchstwahrscheinlich weniger Leistung.

99 GIF: Das Graphics Interchange Format verknüpft mehrere Bitmap-Dateien zu einem animierten Bild. Das erste Bildformat, dass von diversen Webbrowsern unterstützt wurde.

100 FTP: Das File Transfer Protocol ist ein Dateiübertragungsprotokoll um z. B. lokale Dateien auf ein Server hochzuladen.

101 Lorusso, Silvio: The user condition: Computer Agency and behaviour. 2020.  
<https://theusercondition.computer/> (Stand: 01.03.2024)

An dieser Stelle wird der Widerspruch dieses Kapitels exemplarisch, denn besonders in diesem Kontext sind Codes im Inspizier-Modus<sup>102</sup> spannend, weil sie sind experimentell und gleichzeitig leichter verständlich sind. Es stellt sich nun trotzdem wiederholt die Frage, wo die Ästhetik einzuordnen ist und wer sich damit identifizieren kann. Es scheint bei den genannten Beispielen von Web-Design-Ästhetiken und dem Code dahinter keinen fairen Kompromiss in der Zugänglichkeit, Lesbarkeit und Identifizierbarkeit zu geben.

## 15. Hacking, Decolonizing, Unlearning

Jenseits kryptischer Codes und Designs gibt es Ansätze und Projekte, die sich bemühen, die große Informationskluft zwischen den Intellektuellen, Akademischen, weißen und den Marginalisierten, Unterdrückten, Subalternen, Schwarzen durch Hacking zu verkleinern.

Sowohl Design als auch Code befinden sich in einem ständigen Wandel und durch einen fortlaufenden, dauerhaften Dialog, der jedes Scheitern und jede Kritik einzelner Ideen einbezieht, können neue Ansätze und Diskurse entstehen. In diesem Kontext bedeutet Aktivismus vor allem, mit Querverweisen, Links, Bibliotheken, Materialien, Codes und Verknüpfungen ‚Open Source‘ zu arbeiten.

*„So-called intellectual property is property that not merely has a separate legal existence to other property, but is different in kind. Land need only occupy different space to other land, capital’s property likewise need only be distinct in space and time. The vectoralist class<sup>103</sup> depends on the hacker class<sup>104</sup> to produce the qualitative differences of intellectual property that it comes directly to own, and indirectly to profit by, and the owner of the vectors of its distribution. It depends on the very class capable of hacking into actuality the very virtuality it must control to survive.“<sup>105</sup>*

McKenzie Warks Zitat über geistiges Eigentum betont, dass der Zugang zu Informationen und Wissen die Möglichkeit bietet, zu besitzen, zu betreiben und zu kontrollieren. Die Konzentration der Produktion und

102 Inspizier-Modus: Webbrowsere Funktion Code sichtbar zu machen.

103 Vectoralist Class: McKenzie Warks Begriff für die Klasse, die Informationsnetzwerke und Plattformen in einer informationspolitischen Wirtschaft besitzt, betreibt und kontrolliert.

104 Hacker Class: McKenzie Warks Begriff für die Klasse von kreativen Produzent:innen kultureller Formen in einer informationspolitischen Wirtschaft.

Effizienz durch Technologie und Computern ist eine tiefgreifende Mutation des Kapitalismus<sup>106</sup>, deren Ursprünge im Kolonialismus und in der Idee der Expansion und Produktivität liegen. Die Idee der Algorithmen im kapitalistischen Sinne entstand mithilfe von Diagrammen, Daten mit Kalkulationen und Prognosen mit dem Ziel, auf das „Subjekt ganz und gar [zu] verzichten“ und Arbeit zu mehr Arbeit zu machen.<sup>107</sup> Die einzigen natürlichen Feinde einer Computer Störung sind Hacker und Viren.<sup>108</sup>

In McKenzie Warks „Hacker Manifesto“ ist ‚the Hacker‘ eine politische Figur, die einen freien Informationsaustausch ohne Kontroll- und Machtstrukturen anstrebt. Hacker sind in der Lage, in geschlossene Informationssysteme einzudringen und Schwachstellen aufzudecken, was sie zu Gegnern von großen Tech-Unternehmen und der Regierungen macht, die das Internet kontrollieren und kommerzialisieren, um ihre eigene Macht zu bewahren.

Hacker spielen eine wichtige Rolle dabei, Wissen zu verbreiten, ohne es zu einem Produkt zu machen oder künstliche Verknappungen zu erzeugen — Wissen und Informationen viral zu machen. Sie eröffnen oder etablieren Schlupflöcher und Unterführungen zwischen den Klassen<sup>109</sup>. Ihr Produkt ist ein Hack: Eine Funktionserweiterung oder Problemlösung, bei der das Ziel auf eine ungewöhnliche, originelle Weise erreicht wird. Unlearning bedeutet hier nicht, den Code ungeschrieben zu lassen, sondern ihn zu nutzen, um der digitalen Kolonialisierung entgegenzuwirken, indem Macht umgelagert und -verteilt, eigene Inhalte geschaffen und Tools gebaut werden.

Jeder Riss im System unserer kapitalistischen, effizienten und kontrollierenden Gesellschaft durch die Hacker macht die Kluft zwischen Coder und User kleiner. Denn jede Interaktion mit einem Computer beinhaltet eine Form der Umprogrammierung. Jede Mausbewegung, jeder eingetippte Buchstabe, jedes gespeicherte Bild übersetzt ein Befehl und ein Binärcode in den Tiefen der Maschine. Ein:e Coder macht nichts anderes, als sich vorprogrammierter Bibliotheken zu bedienen — Jeder:r Coder ist ein:e User. Ob wir uns an einem Klick oder einer Code-Zeile bedienen, ist grundsätzlich erst einmal irrelevant, wenn unser Befehl wie gewünscht ausgeführt wird.<sup>110</sup> Hacking=Unlearning=Decolonizing.

<sup>106</sup> Deleuze, Gilles: Postskriptum über die Kontrollgesellschaften. In: *Unterhandlungen. 1972–1990*. Suhrkamp, 1993. S 258–259.

<sup>107</sup> Harney, Stefano; Moten, Fred: *Die Undercommons. Flüchtige Planung und schwarzes Studium*. Transversal Texts, 2016. S. 103–104.

<sup>108</sup> Deleuze, Gilles: Postskriptum über die Kontrollgesellschaften. In: *Unterhandlungen. 1972–1990*. Suhrkamp, 1993. S 258–259.

<sup>109</sup> Vectoralist Class und Hacker Class, s. Zitat oben.

<sup>110</sup> Hoff, Melanie: *Always Already Programming*. GitHub, 2020.

<https://gist.github.com/melaniehoff/95ca90df7ca47761dc3d3d58fead22d4> (Stand: 01.03.2024)

Im Essay „Can the Subaltern Speak?“ hat die Theoretikerin Gayatri Spivak die Unhörbarkeit und Unterdrückung der Subalternen in kolonialen und postkolonialen Kontexten thematisiert. Ihr berühmter Schlussatz, „The subaltern cannot speak“<sup>111</sup>, unterstreicht die Unmöglichkeit für unterdrückte und marginalisierte Gruppen zu sprechen oder gar gehört zu werden, selbst wenn sie sprechen könnten. Dieser Satz verweist auf die strukturellen Ungleichheiten, denen subalterne Gruppen gegenüberstehen, wenn es darum geht, ihre eigenen Geschichten und Perspektiven dem Globalen Norden zu teilen. Es ist eben nicht möglich oder realistisch, dass alle Länder, Regionen und Orte der Welt das westliche historische epistemologische Modell revidieren und neu hinterfragen, um an der Erfassung des lokalen Wissens in der Welt teilzunehmen<sup>112</sup>. Im Kontext der Dekolonialisierung von Coding und Design können wir die Erkenntnisse von Spivak heranziehen, um zu betonen, dass viele der gegenwärtigen Strukturen im digitalen Raum die Stimmen, Beiträge und Bedürfnisse der Subalternen ignorieren, unterdrücken, ausbeuten, nicht sehen und ausnutzen.

Es ist hier wichtig zu betonen, dass ‚the Hacker‘ schon alleine aus dem Grund, dass das Produkt gehört, gelesen und gesehen wird, nicht ‚the Subaltern‘ sein kann. Die Coding-Praxis ist immer geprägt von Machtstrukturen und kann nicht unabhängig davon betrachtet werden. ‚The Hacker‘ kann versuchen, eine Verbindung zum ‚Subaltern‘ zu knüpfen, wird jedoch daran scheitern, denn ‚the Hacker‘ kann ‚the Subaltern‘ nicht hören. ‚Subalterne‘, die gehört, gelesen und gesehen werden, sind ‚Hacker‘.

Coding ist nicht nur ein technischer Prozess, sondern auch eine gestalterische Praxis, die maßgeblich die digitale Welt formt. Die Diskussionen über Kolonialisierung sind im Design und Coding relevant, weil Information, Produkte und Gewinn besonders einseitig dominante Perspektiven zeigen und bereichern – selbst wenn sie ein Produkt der Hacker Class sind.

Plattformen haben sich längst als fundamentaler Bestandteil des kapitalistischen Systems etabliert und sind praktisch unverzichtbar geworden. Selbst die 90er Websites, die Creative Coding Websites und die Websites, die im akademischen Kunsthochschulkontext gecodet und gestaltet werden, referieren sich selbst. Die Arbeit der Dekolonialisierung besteht nicht nur darin, alternative, künstlerische oder esoterische Designs und

111 Spivak, Gayatri Chakravorty: Can the Subaltern Speak? S. 104.  
[https://abahlali.org/files/Can\\_the\\_subaltern\\_speak.pdf](https://abahlali.org/files/Can_the_subaltern_speak.pdf)

112 Wildi Merino, Ingrid: Arquitectura de las Transferencias. Arte, Política y Tecnología. 2012–2014.  
[http://www.ingridwildimerino.net/images/recherche/ADT\\_version\\_ESP.pdf](http://www.ingridwildimerino.net/images/recherche/ADT_version_ESP.pdf). Übersetzung: FBS.

Codes zu schaffen, sondern auch darin, die Bedingungen zu verändern, für diejenigen, die am meisten unterdrückt werden.

Durch heutige Machtstrukturen von Technologie und Digitalisierung, die mit Code und Design aufrechterhalten werden können, ist es nicht möglich, angemessen und respektvoll Subalterne zu berücksichtigen. Hacking=Unlearning=Decolonizing sind Prozesse, die nur mit Zugang zu Wissen, zu Informationen und Infrastruktur erfolgreich umgesetzt werden können. Das Subjekt definiert sich durch das, was es nicht ist und unmöglich sein kann — 0/1, wahr/falsch, ja/nein, ein/aus, Schwarz/weiß, Hacker/Subaltern.

Ein Unlearning-Hack könnte darin bestehen, Coden mittels Design als Praxis für alle zu betrachten und anzuwenden. Dafür müssen wir uns bewusst machen, dass das Web tiefe koloniale Strukturen hat, an denen wir nur rütteln können, wenn wir weiter fragen und mit offenen Augen neu entdecken, was schon lange hinter dem transparenten Hintergrund lag, der fälschlicherweise von Anfang an weiß hinterlegt wurde.

## 16. Nachwort

Unlearning und Decolonizing Code werden hier hauptsächlich mit kritischen Theorie Ansätzen der Geisteswissenschaften statt wissenschaftlicher Theorien der Informatik behandelt. Die meisten einbezogenen Theorien wurden in Europa und im Globalen Norden entwickelt, wobei nur einige wenige, jedoch wichtige, marginalisierte Stimmen aus dem Globalen Süden stammen. Obwohl alle behandelten Texte teilweise in einzelnen Kapiteln stark unabhängig voneinander behandelt werden, glaube ich, dass sie im Nachklang eine intersektionale Perspektive und Denkweise zeigen können. Es ist eine Herausforderung, in einer Disziplin, die nur und ausschließlich aus der Hegemonie des Globalen Norden hervorgegangen ist und agiert, eine kritische Arbeit zu schreiben und in angemessenen Kontext zu setzen. Vor allem weil es die Perspektive einer Coderin ist, die Infrastruktur, Informationen und Werkzeuge hat und deshalb keinen Lösungsansatz bieten kann.

Für den ‚the Hacker‘ und ‚the Subaltern‘:

*„I believe that the differences in our practices are not ideological differences. We are simply in different places from where we read and create a dialectical sense of diversity. Sometimes we are very far away from each other and then again very close as far as practices and attitudes are concerned. We walk together, one with the other, looking at each other, never in parallel lines but connected, like in a net.“*<sup>113</sup>

Unlearning, Hacking und Decolonizing ist ein Prozess.

## Literaturverzeichnis

- S.7,8,9,11 Evans, Claire L: BROAD BAND. The Untold Story of the Women Who Made the Internet. Portfolio/Penguin, 2018.
- S.12,27,28 Mustafa Ali, Syed: A brief introduction to decolonial computing. Research Gate, 2016.  
[https://www.researchgate.net/publication/303980379\\_A\\_brief\\_introduction\\_to\\_decolonial\\_computing](https://www.researchgate.net/publication/303980379_A_brief_introduction_to_decolonial_computing)
- S.32,36 Harney, Stefano; Moten, Fred: Die Undercommons. Flüchtige Planung und schwarzes Studium. Transversal Texts, 2016. S.19.
- S.21,23 Wajcman, Judy: Technofeminism. Polity Press. 2004. S.13.
- S.28,37 Spivak, Gayatri Chakravorty: Can the Subaltern Speak?
- S.35 Wark, McKenzie: A Hacker Manifesto. Harvard University Press, 2004. S. 89.
- S.17 Hayles, N. Katherine: My Mother Was a Computer. Digital Subjects and Literary Texts. University of Chicago Press, 2005. S. 39.
- S.25 Srnicek, Nick: Platform Capitalism. Polity, 2017. S.53.  
[https://abahlali.org/files/Can\\_the\\_subaltern\\_speak.pdf](https://abahlali.org/files/Can_the_subaltern_speak.pdf)
- S.28 Birhane, Abeba; Guest, Olivia: Towards decolonising computational sciences. Women, Gender & Research, 2021.
- S.30 Gikandi, Simon: Slavery and the Culture of Taste. Princeton University Press (Reprint Edition), 2014. S. 42
- S.30 Gikandi, Simon: Aesthetic Reflection and the Colonial Event: The Work of Art in the Age of Slavery. Michigan Publishing, 1997.
- S.30 Chow, Rey: Writing Diaspora. Tactics of Intervention in Contemporary Cultural Studies. Indiana University Press, 1993.
- S.31 Sonderegger, Ruth: Zur Kolonialität der philosophischen Ästhetik. Ruth Sonderegger im Gespräch mit Maja Figge und Wilma Lukatsch. In: Künste dekolonisieren. Ästhetische Praktiken des Lernens und Verlernens. Transcript, 2023. S.55.
- S.13,14,23 Gheverghese Joseph, George: A Brief History of Zero. Research Gate, 2008.  
[https://www.researchgate.net/publication/228783983\\_A\\_Brief\\_History\\_of\\_Zero](https://www.researchgate.net/publication/228783983_A_Brief_History_of_Zero)
- S.36 Hoff, Melanie: Always Already Programming. GitHub, 2020.  
<https://gist.github.com/melaniehoff/95ca90df7ca47761dc3d3d58fead22d4>
- S.19 Fouché, Rayvon: Race After the Internet From Black Inventors to One Laptop Per Child. Routledge, 2012.
- S.4,37 Wildi Merino, Ingrid: Arquitectura de las Transferencias. Arte, Política y Tecnología. 2012–2014.  
[http://www.ingridwildimerino.net/images/recherche/ADT\\_version\\_ESP.pdf](http://www.ingridwildimerino.net/images/recherche/ADT_version_ESP.pdf). Übersetzung: Fernanda Braun Santos.
- S.6 Cairns, Anna: Code, Text, Type. c0da, 2023. <https://c0da.org/contributions/code-text-type>
- S.13 Meyer, Eva: Zählen und Erzählen. Für eine Semiotik des Weiblichen. Vittorio Klostermann, 2013. S. 69.
- S.16 Nasser, Ramsey: بـ . 2013. <https://nas.sr/%D9%82%D9%84%D8%A8/>
- S.17 Cunningham, Claire: Equations of Care & Responsability. DANCEOLITICS. Uferstudios GmbH, 2021. S. 74.  
<https://online.fliphtml5.com/zgewh/lssv/#p=1>
- S.18 Leichtfuß, Anne: Leichte Sprache.(Vortrag, Januar 2023). Klasse Digitale Grafik, Field trip: Simple Language.
- S.24 Kwet, Michael: Digital colonialism. US empire and the new imperialism in the Global South. Institute of Race Relations, 2019.
- S.34 Lorusso, Silvio: The user condition: Computer Agency and behaviour. 2020.  
<https://theusercondition.computer/> (Stand: 01.03.2024)
- S.36 Deleuze, Gilles: Postskriptum über die Kontrollgesellschaften. In: Unterhandlungen. 1972–1990. Suhrkamp, 1993. S 258–259.
- S.39 E-Mail-Korrespondenz zwischen Spideralex und Anamhoo. In: The Beautiful Warriors. Technofeminist Praxis in the Twenty-First Century. Edited by Cornelia Sollfrank. Minor Compositions, 2020. S. 38.

Links (Stand: 01.03.2024):

- S.6 [https://en.wikipedia.org/wiki/Computer\\_programming](https://en.wikipedia.org/wiki/Computer_programming)  
S.6 [https://pure.mpg.de/rest/items/item\\_3287104\\_1/component/file\\_3287105/content](https://pure.mpg.de/rest/items/item_3287104_1/component/file_3287105/content)  
S.7 [https://de.wikipedia.org/wiki/Alan\\_Turing](https://de.wikipedia.org/wiki/Alan_Turing) / <https://de.wikipedia.org/wiki/Turingmaschine>  
S.8,10,12 [https://en.wikipedia.org/wiki/Computer\\_programming#History](https://en.wikipedia.org/wiki/Computer_programming#History)  
S.8,9 [https://en.wikipedia.org/wiki/History\\_of\\_programming\\_languages](https://en.wikipedia.org/wiki/History_of_programming_languages)  
S.10 <https://de.wikipedia.org/wiki/Internet#Geschichte>  
S.10 <https://de.wikipedia.org/wiki/Hypertext>  
S.11 <https://de.wikipedia.org/wiki/Internet#Geschichte>  
S.12 <https://www.bpb.de/kurz-knapp/lexika/politiklexikon/17718/kolonialismus/>  
S.13 <https://twitter.com/SavvyArtSpace/status/718837212838764545>  
S.14 [https://de.wikipedia.org/wiki/APL\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/APL_(Programmiersprache))  
S.14 <https://c0da.org/contributions/ich-ein-transformer-oder-how-to-code-a-feminist-transformer>  
S.15 <https://www.washington.edu/accesscomputing/sites/default/files/doit-sync/files/QuorumCSP.pdf>  
S.15 <https://analyticsindiamag.com/6-popular-non-english-programming-languages/>  
S.16 <https://timrodenbroecker.de/yehwan-song/>  
S.18 [https://esolangs.org/wiki/Main\\_Page](https://esolangs.org/wiki/Main_Page)  
S.20 <https://survey.stackoverflow.co/2023/#developer-profile-education>  
S.21 <https://www.codecademy.com/resources/blog/gender-pay-gap-in-tech/>  
S.21 <https://www.census.gov/content/dam/Census/library/publications/2020/demo/p60-270.pdf#page=18>  
S.21 <https://www.statista.com/chart/4467/female-employees-at-tech-companies/>  
S.21 <https://punkt.de/de/blog/2018/frauen-in-der-softwareentwicklung.html>  
S.21 <https://de.statista.com/infografik/13283/frauen-in-der-tech-branche/>  
S.21 <https://code.likeagirl.io/which-countries-have-the-most-skilled-female-developers-b532fc53c7c8>  
S.21 <https://www.pewresearch.org/science/2021/04/01/stem-jobs-see-uneven-progress-in-increasing-gender-racial-and-ethnic-diversity/>  
S.21 <https://about.fb.com/news/2015/06/driving-diversity-at-facebook/>  
S.21 [https://static.googleusercontent.com/media/about.google/en//belonging/diversity-annual-report/2022/static/pdfs/google\\_2022\\_diversity\\_annual\\_report.pdf?cachebust=1093852](https://static.googleusercontent.com/media/about.google/en//belonging/diversity-annual-report/2022/static/pdfs/google_2022_diversity_annual_report.pdf?cachebust=1093852)  
S.21 <https://blogs.microsoft.com/blog/2020/10/21/microsofts-2020-diversity-inclusion-report-a-commitment-to-accelerate-progress-amidst-global-change/>  
S.21 <https://de.statista.com/infografik/13283/frauen-in-der-tech-branche/>  
S.21 <https://code.likeagirl.io/which-countries-have-the-most-skilled-female-developers-b532fc53c7c8>  
S.21 <https://www.pewresearch.org/science/2021/04/01/stem-jobs-see-uneven-progress-in-increasing-gender-racial-and-ethnic-diversity/>  
S.22 <https://www.trustradius.com/buyer-blog/women-in-tech-report>  
S.22 <https://zeitfuerx.de/arbeit/wie-die-bro-kultur-einer-diversen-arbeitswelt-schadet/>  
S.22 [https://en.wikipedia.org/wiki/Bro\\_culture](https://en.wikipedia.org/wiki/Bro_culture)  
S.22 <https://www.deutschlandfunkkultur.de/frauen-informatik-programmieren-geschichte-100.html>  
S.23 [https://precolombino.cl/mods/expo\\_temporales/detalle/43/html/005a.htm](https://precolombino.cl/mods/expo_temporales/detalle/43/html/005a.htm)  
S.24 <https://www.documenta14.de/en/artists/22610/marilou-schultz>  
S.24 <https://www.righto.com/2019/07/software-woven-into-wire-core-rope-and.html>  
S.26 Zuazo, Natalia: #153 ChatGPT, la Carrera de la inteligencia artificial y la desigualdad. El Hilo de Radio Ambulante Estudios, 07.04.2023.  
S.27 <https://www.objectbay.com/blog/outsourcing-softwareentwicklung>  
S.32 <https://timrodenbroecker.de/about/>

# Bildteil: unlearning of a web code

Master Thesis von Fernanda Braun Santos  
Matrikel-Nummer: 2465357  
Hamburg, April 2024

Klasse Digitale Grafik  
HFBK Hochschule für bildende Künste Hamburg

Betreut durch Prof. Dr. Hanne Loreck  
(Studienschwerpunkt Theorie und Geschichte)  
Prof. Konrad Renner und Prof. Christoph Knoth  
(Studienschwerpunkt Grafik/Fotografie/Typografie)

# Abb.1

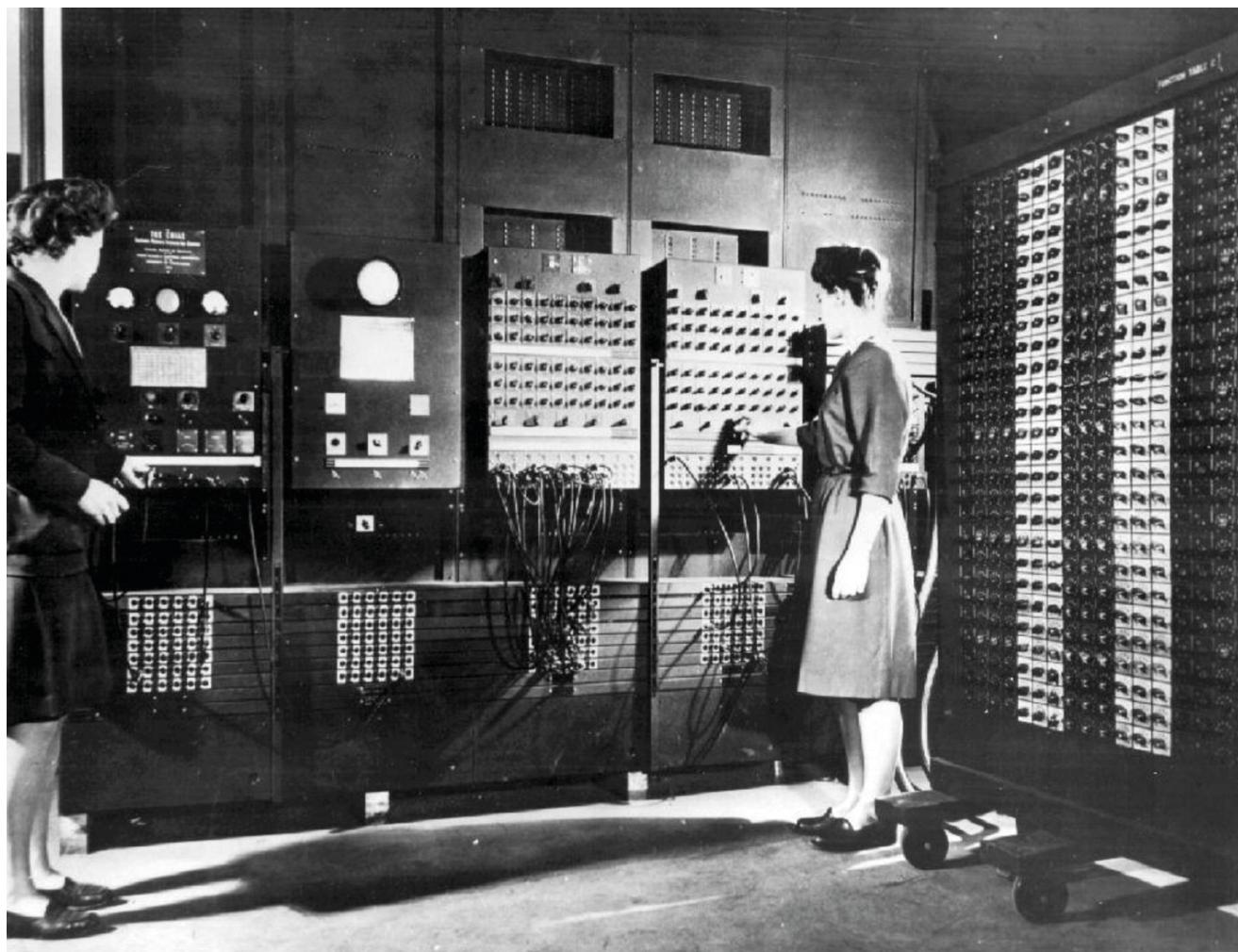


<http://www.ingridwildimerino.net/work-arquitectura-de-las-transferencias-2017.php>



[http://www.ingridwildimerino.net/pdf/ADT-Catalogo\\_D21\\_1.pdf](http://www.ingridwildimerino.net/pdf/ADT-Catalogo_D21_1.pdf)

# Abb. 2



<https://www.geo.de/wissen/weltgeschichte/geschichte-75-jahre-eniac-der-erste-universalrechner-der-welt-war-streng-geheim-30378318.html>

# Abb. 3

0	0
•	•
•	○
•	○
•	○
..	○
.	○
○	□
○	○
○	○
○	○
○	○
○	○
○	○

# Abb.4



<https://mayadecipherment.com/2012/06/15/the-calligraphic-zero/>

# Abb. 5



[https://de.wikipedia.org/wiki/APL\\_\(Programmiersprache\)](https://de.wikipedia.org/wiki/APL_(Programmiersprache))



<https://www.kleinanzeigen.de/s-anzeige/apl-tastatur-zu-verschenken/2684959265-225-3169>

Abb. 6

<u>أمثلة - النجدة - ما هذا؟</u>	<u>رمزي ناصر</u>	<u>قلب: لغة برمجة - مترجم ٢٠١٢، ٣</u>
<u>سهل: مرحبا يا عالم</u>	<u>متوسط: عدد فيبوناتشي</u>	<u>متقدم: لعبة الحياة لكونواي</u>
<<< (أمثلة)	<<< (حدد فيبوناتشي (لامدا (ن))	<<< (إذا (أصغر؟ ن ٢)
...	...	...
ن	...	...
(جمع (فيبوناتشي (طرح ن ١))	(فيبوناتشي (طرح ن ٢)))))	(قول (فيبوناتشي (١٠))
...	...	...
٥٥	٥٥ <==	٥٥ <<<

Abb. 7



Main Page Discussion Read View source View history Search Esolang

https://esolang.org/wiki/Main\_Page  
Create account Log in

Welcome to **Esolang**, the esoteric programming languages wiki!

Why not join us on IRC?

This wiki is dedicated to the fostering and documentation of programming languages designed to be unique, difficult to program in, or just plain weird.

**For readers**

You'll probably want to find out what on earth an esoteric programming language is in the first place.

Then, you might want to explore the complete list of languages, or find something more specific with the categories.

You could also visit the joke language list, which lists languages that can't even be programmed in.

Failing that, you could take a look at a random language.

You could also take a look at the list of special pages.

After getting bored, you could visit the Sandbox and have fun.

**Featured language**

**Thue** is an esoteric programming language based around the idea of a "semi-Thue system": a system which specifies strings that can be rewritten to certain other strings; a program is simply a list of search strings, and possible replacements for them. As a nondeterministic language, a program has the potential to halt if there is some way to reach an end state via applying replacements, even if rules such as "always apply the first replacement" would lead to an infinite loop. No data storage is necessary, apart from a single string that holds the entire state of the running program, although this often causes programs to run slowly due to delays in communicating information from one part of the string to another. ([more...](#))

Previously featured: Funciton · Brainfuck · Deadfish · Emmental · [more...](#)

**For creators**

If you've just created a language, you can create an article for it by typing its name into the search box, assuming the name is not already taken, but be sure to take a look at the help guide first. Then you should add it to the language list (or the joke language list, as appropriate). **DON'T** create pages with the Esolang: prefix for your esolang because that prefix is for general information about the wiki.

If you haven't got that far yet, take a look at the list of ideas for inspiration.

Otherwise, you could help out with a work in progress.

**Meta**

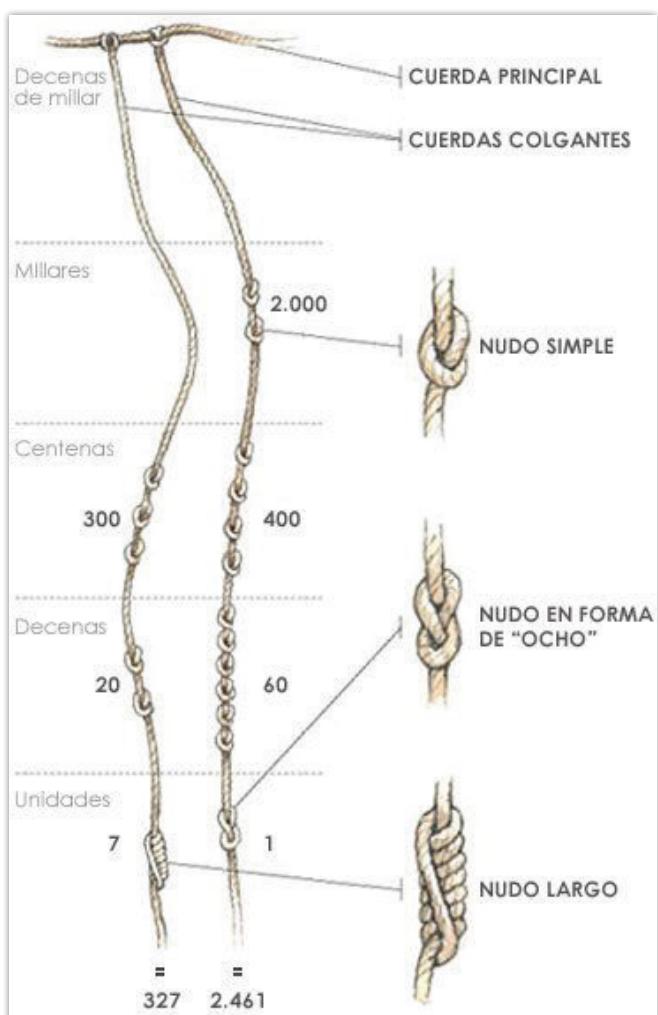
- Learn about this wiki
- Check out recent changes
- View the site policies
- Download an XML dump of the wiki's content
- Discuss the wiki on the community portal's talk page
- Talk with other esolang enthusiasts in the places listed in the community portal
- Go to the main page

# Abb. 8



[https://en.wikipedia.org/wiki/OLPC\\_X0](https://en.wikipedia.org/wiki/OLPC_X0)

# Abb. 9



<https://artsandculture.google.com/project/khipus>

Abb. 10

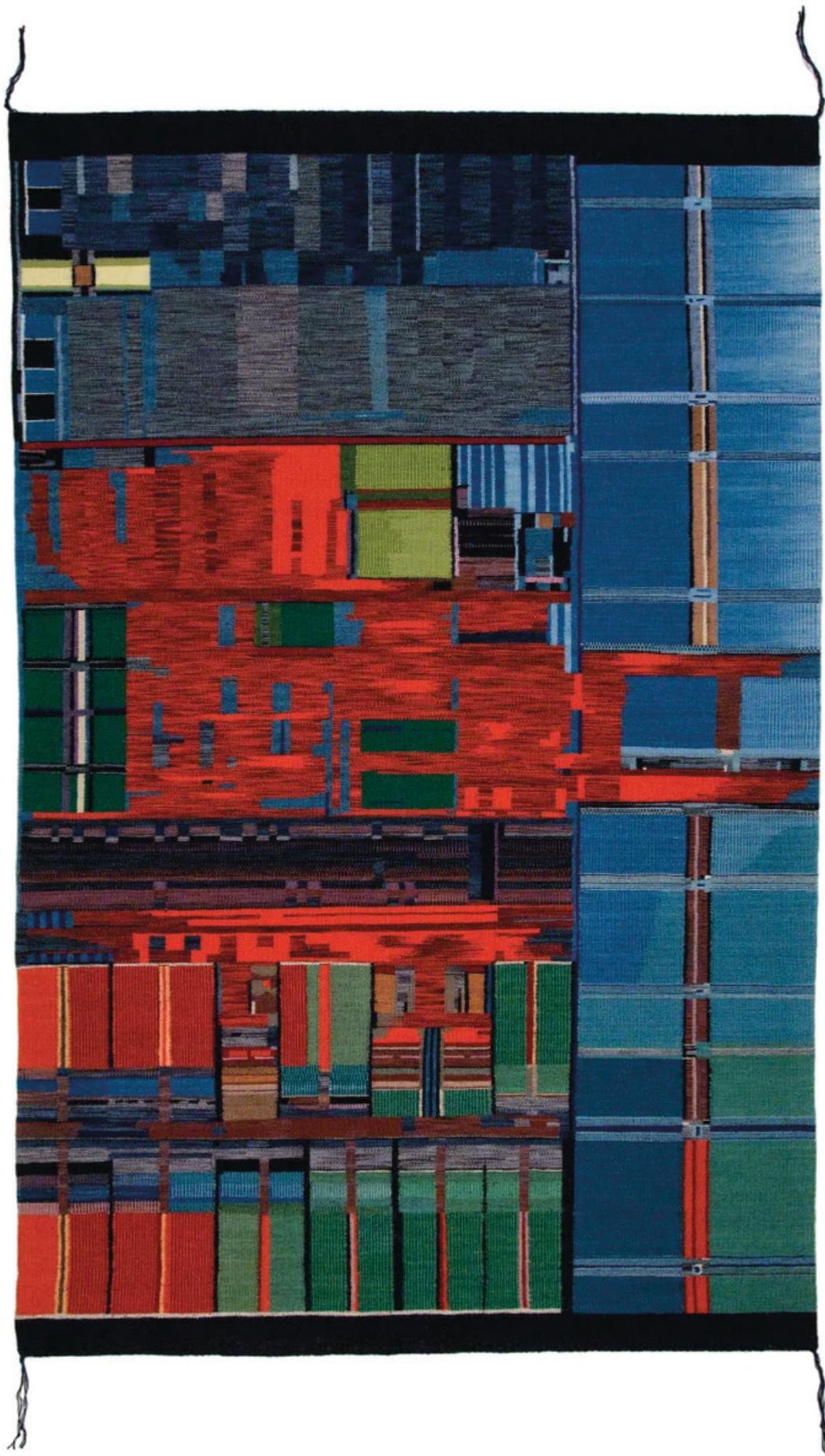
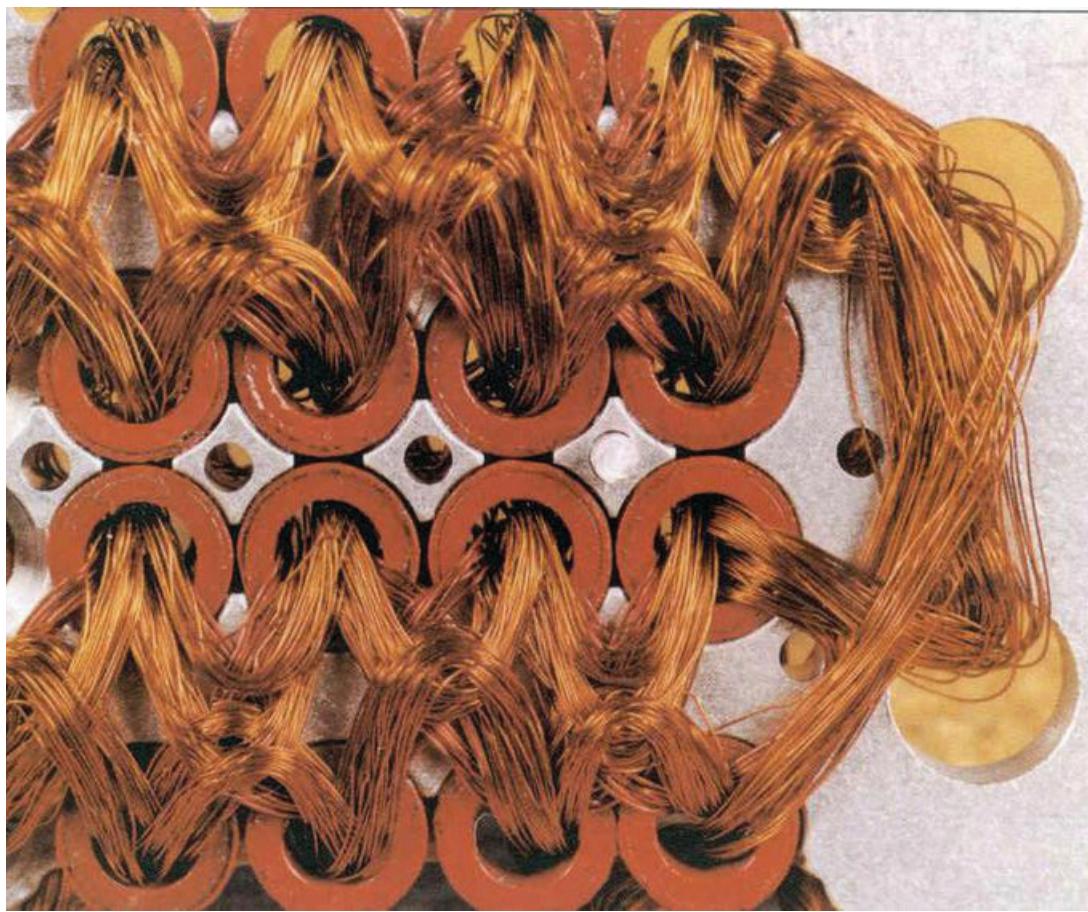


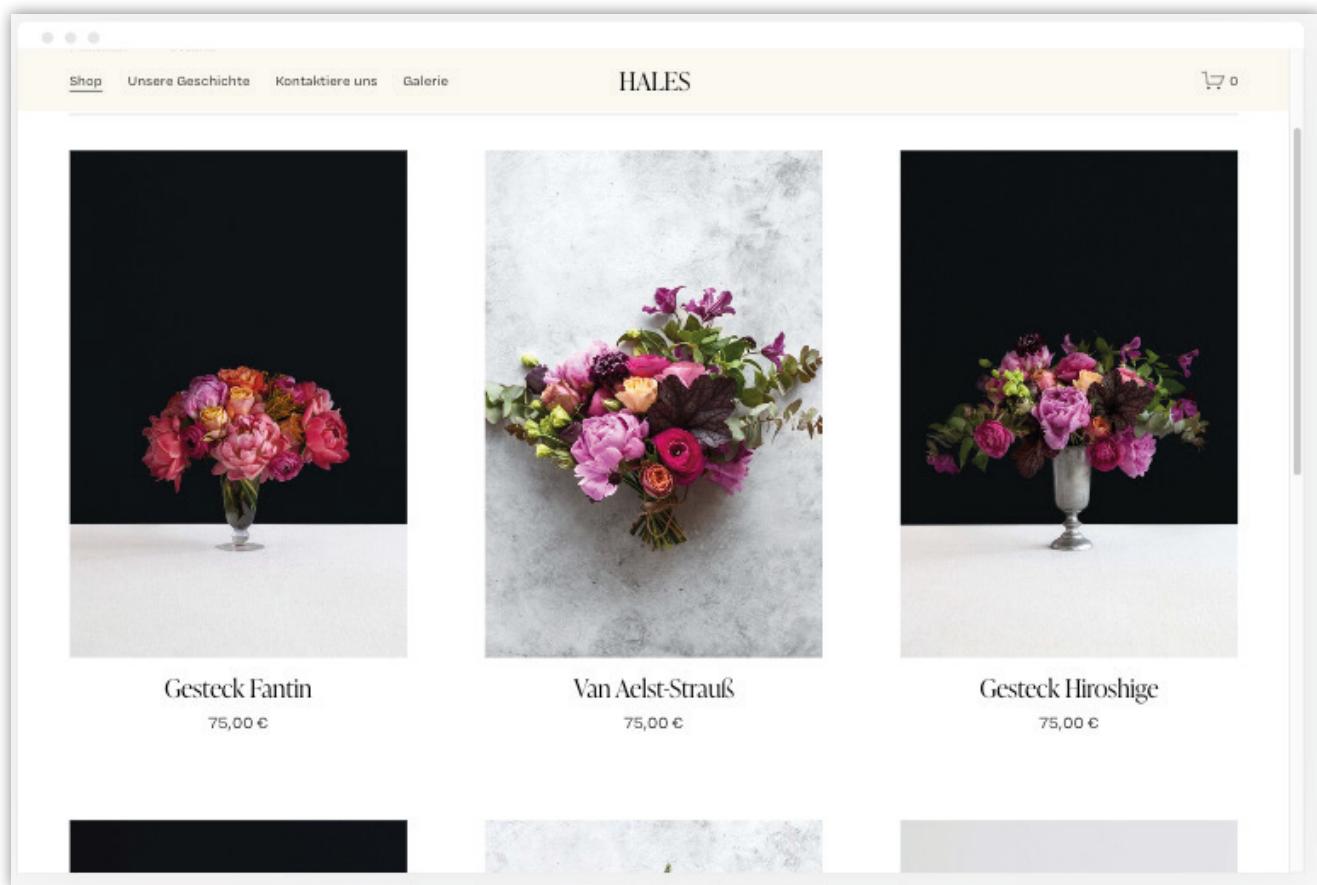
Abb. 11



<https://www.righto.com/2019/07/software-woven-into-wire-rope-and.html>



# Abb. 12

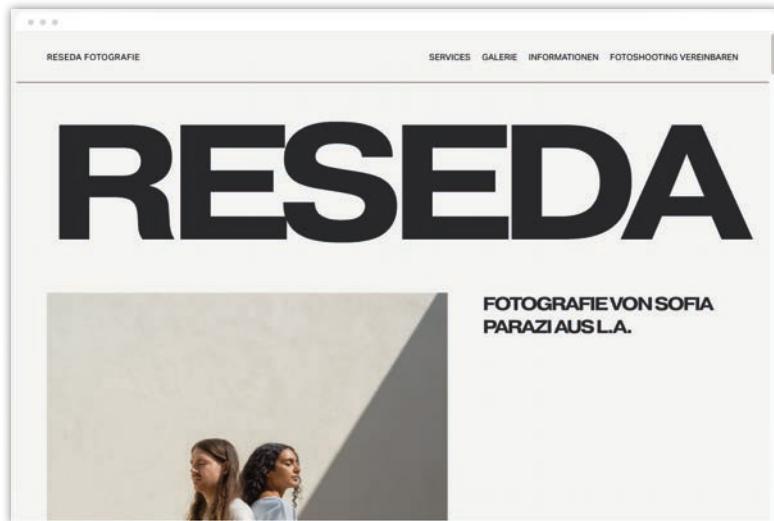
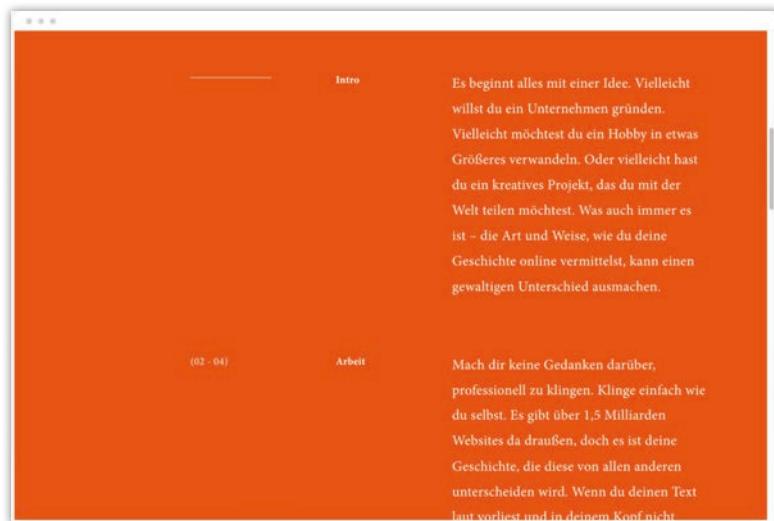
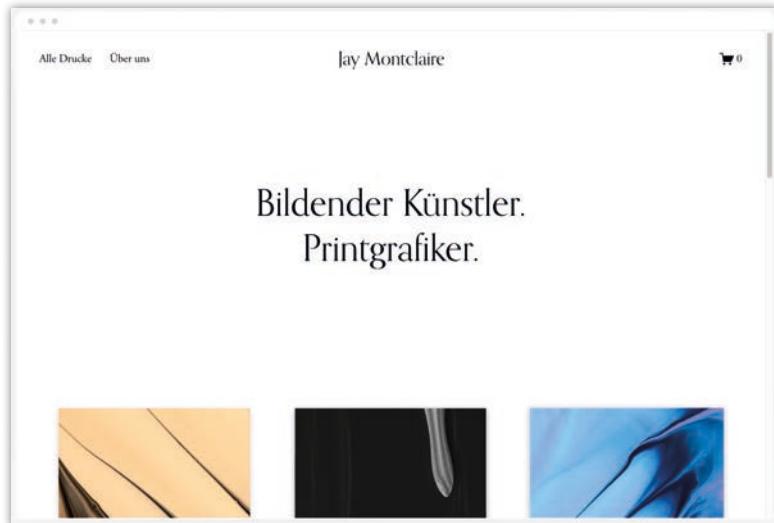


The screenshot shows the Hales website's homepage. At the top, there is a banner with a florist working on a large arrangement of white and pink flowers. The text on the banner reads: "Frische Blumen für jeden Geldbeutel." Below the banner is a dark green sidebar with the text: "Bei Hales arbeiten wir mit sorgfältig ausgewählten Gärtner:innen zusammen, von denen wir die frischsten Gewächshausbüumen, Wildblumen und saisonales Zierrgrün beziehen. Diese präsentieren wir in individuellen Keramik- und Glasgefäßen aus eigener Herstellung." The main content area features a section titled "Bestseller" with three flower arrangements: Gesteck Fantin, Van Aelst-Strauß, and Gesteck Hiroshige. Below this is a section titled "Feiertage & besondere" with a photo of colorful flower petals and leaves on a light surface. The Hales logo is at the bottom left, and a heart icon is at the bottom right.

<https://de.squarespace.com/templates>

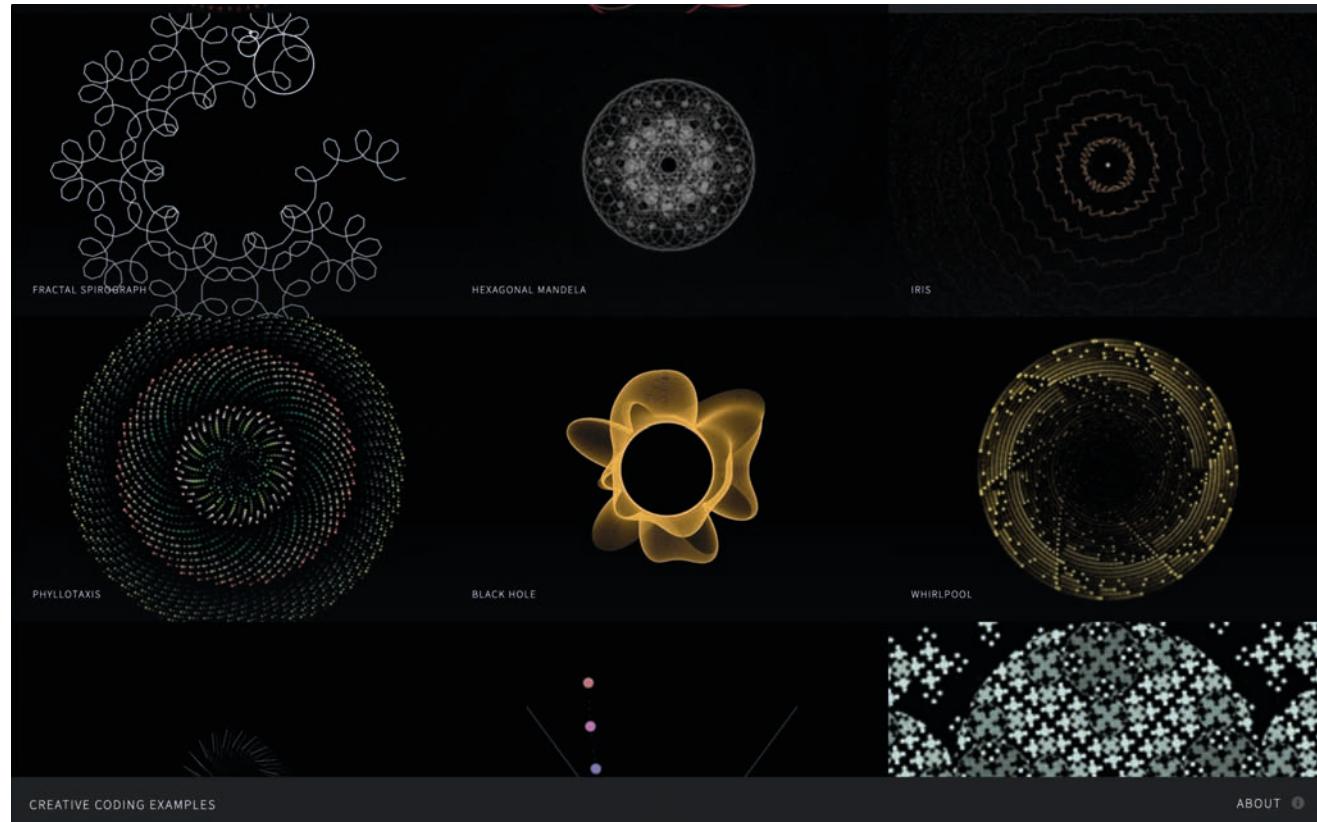
weiter auf der nächsten Seite...

# Abb. 12



<https://de.squarespace.com/templates>

# Abb. 13



Introduction to Creative Coding

Start Here   Glossary   Examples   Useful links   Put your code online

New background   Edit code

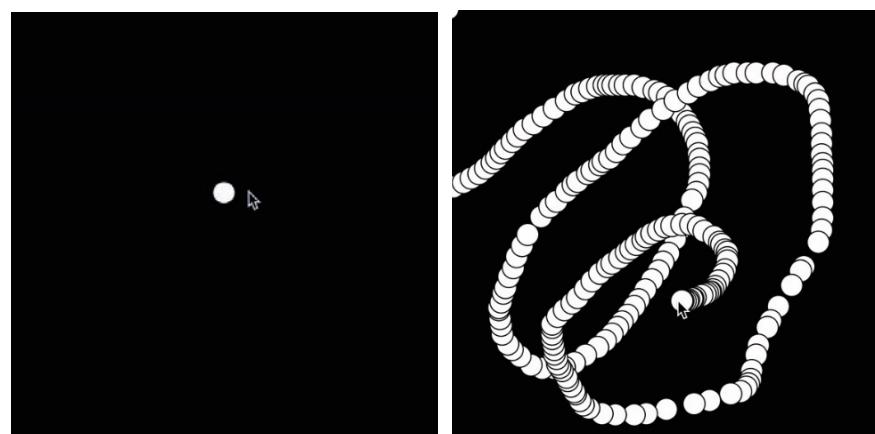
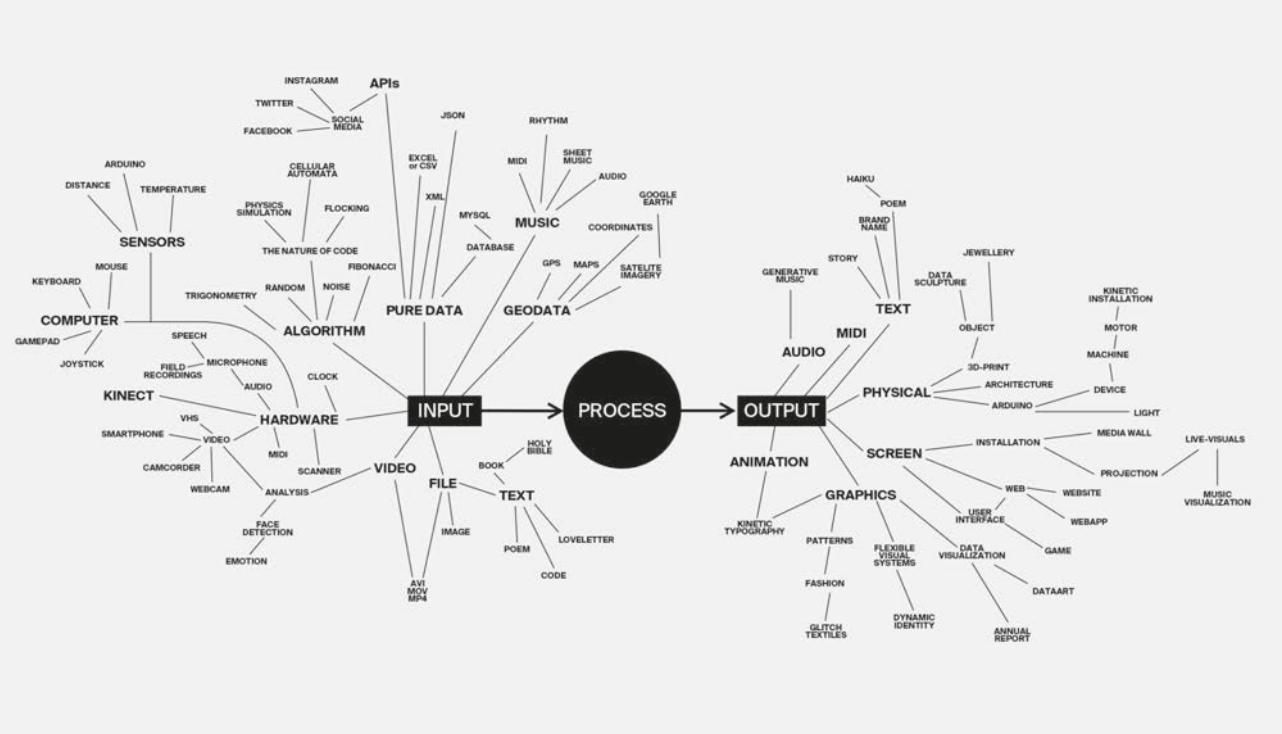


Abb. 14

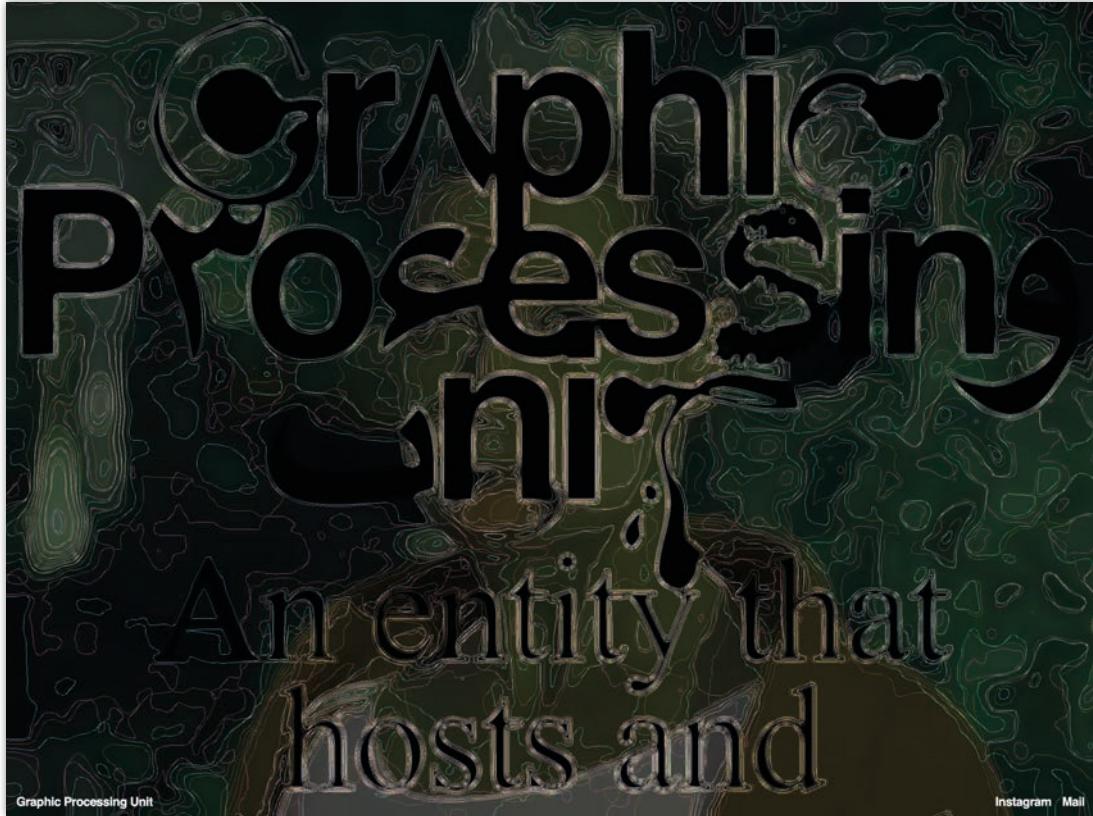


<https://www.patrik-huebner.com/method/>

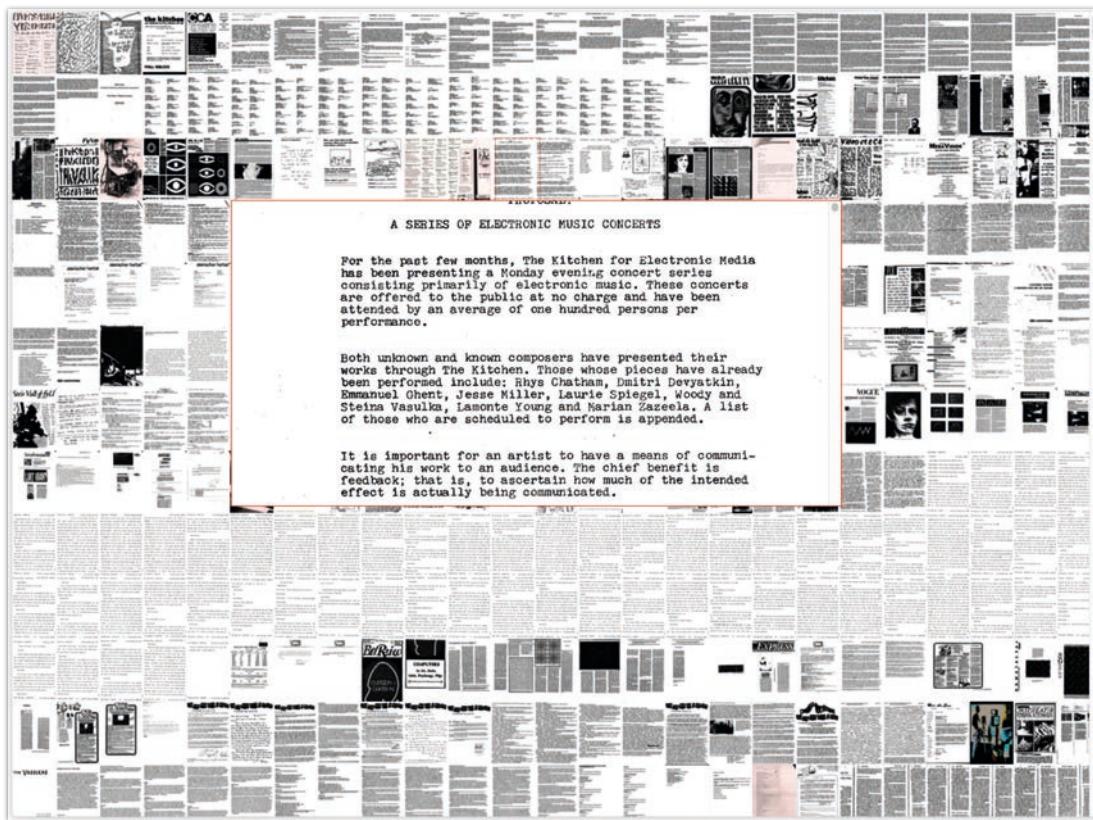
Abb. 15

<https://schemasofuncertainty.com/null>

weiter auf der nächsten Seite...



<https://gpu.gallery/index.html>



Archiv: <http://rmo.zkm.de/vasulka-pdf/#61>



<https://staedelschule.de/de/home>

Abb. 15

weiter auf der nächsten Seite...



<https://2020.swissdesignawardsblog.ch/>

## Extended Libraries

de en

Suchergebnisse:

[Publication of the final thesis in the library system \(pdf\)](#)

[Antrag auf Verlängerung des Promotionsstudiums\\_Application of extension of doctoral st...](#)

[HFBK Library A-Z \(pdf\)](#)

[Einrichtungen > Extended Library](#)

[Stories > Extended Libraries](#)

[Projekte > Extended Library als Proberaum](#)

[Kalender > Extended Bodies](#)

[Kalender > Eröffnung: Non-Extractivist Practices in the Dispossessed Lands - Arazi Asse...](#)

## März

Willkommen zu meiner Debattenrundschau vom 30.03.2020! Wer sich für die wirtschaftlichen Auswirkungen der Corona-Pandemie...

Rhizome

## Six Episodes on Non-Property

Six Episodes on Non-Property is a program emerging from a long-term study group on the question of Non-Property - unders...

Projekte

## What do you actually do? – Niclas Riepshoff

HFBK graduate Chloe Stead, together with the photographer and also HFBK graduate Jens Franke, met former HFBK students to talk about work, life and art.

Kunst und Leben, Projekte

## Bewerbung Malerei/Zeichnen, Bühnenraum, zeitbezogene Medien, Bildhauerei

Alle Informationen zum Bewerbungsverfahren für die Master-Studiengänge: Malerei/Zeichnen, Bühnenraum, zeitbezogene Medien, Bildhauerei  
[Bewerbung Master »Bildende Künste«](#)

## (K)eine Gruppe möchte ich sein

Filme zu produzieren war und ist schon immer ein kollektiver Prozess. Aber es ist etwas ganz anderes, eine kollektive Narration im Rahmen eines Kollektivfilms zu entwickeln – das verdeutlicht der HFBK-Filmprofessor Robert Bramkamp im Rückblick auf zwei langjährige Filmprojekte.

Aktuelles

## Programme: "Counter-

## Magazin »Lerchenfeld«

Die aktuelle Ausgabe hier auch online: das Magazin der HFBK Hamburg zu Strömungen in der Kunst, Kultur und Wissenschaft, zu Ausstellungen, Projekten, Positionsbestimmungen

Aktuelles

## Newsletter und myHFBK events

Newsletter bestellen, Kontakt zur Redaktion

Aktuelles

## → Akademischer Kalender

<https://academic-dates.hfbk.net>

Aktuelles

## → HISinOne

<https://hfbk.hispro.de/>

## → Online Repositorium

<https://projekte.hfbk.net/>

Aktuelles

Abb. 16



A horizontal banner with the words "UNDER CONSTRUCTION" in large, bold, black capital letters at the top. Below it is a yellow and black diagonal striped pattern. At the bottom, the words "coming soon..." are written in a smaller, black, sans-serif font.

<https://www.arngren.net/>

weiter auf der nächsten Seite...

Welcome

Appearing next at:

- 14.03.2023 - 19.03.2023 - Emerging Art Forum  
on display at BEAT. Wertheim Ausstellungsraum
- 18.03.2023 - ongoing - Emerging Art Fair Model of Tomorrow & A Different One (Germany)

**Books**

**Observations**

- *I'm full a smile, AI*,  
considering the last AI spring, 2022.
- *The user is OS and AI in China*,  
thinking of investigation of AI Pic, 2023.
- *Reyes's Home Page*,  
web history research outside of digital libraries, 2023.
- *Damien Chabon's Gom*,  
a monograph, 2022.
- *Internet Experience Discrepancy*,  
during Zuckerberg's homework, 2021.
- *An Infinite Silence*,  
monograph, 2022.
- *From Me to Me*,  
a monograph, 2020.
- *One Yearbook A Kidnapper And Family*,  
a monograph, part of every 20 minutes, since February 2021.
- *Shelly with site list in name*,  
her whole dozen? 2019.
- *Drop Again, The Darklands*,  
On Affluence, Forgiveness, and Ambiguity  
in Human Culture and Human History,  
a monograph, 2018.
- *In Jokes*,  
joke list, infinite scrolling, numbered  
days, 2018.
- *Animated GIFs Timeline*,  
Official and alternative, 2018.
- *Big Glitter*,  
with Mike Tolka, digging into Blingee.com's  
data and philosophies, 2017.
- *A social network for exhibiting net art*,  
that didn't get off, 2000-2017.
- *Armenian Remains*,  
Once again about traditional gifts, 2016.
- *Net Art Tech*,  
On the role of Media Theory at Universities  
of Applied art, Technology and Art and  
Designs, 2015.
- *On the Internet Everybody Knows You Bad*,  
Before posting someone's cuts, people made  
pages for their own dogs, 2013 -- on going.
- *Don't Know What You Are Thinking, Your  
Web Personalities as an Instruction*, 2015.
- *Fontaine Park Documentary*,  
Once again about URLs, 2015.

**Turing Complete User**

ISBN 978-3-9460101-0-1 (PDF)  
ISBN 978-3-94601072-1 (Softcover)

Monographs, part of the Interface Critique series ■■■

2021

Olia Lialina — Net Artist

ISBN : 978-3-23785616-2

Monograph edited by Valeria Perini ■■■

2020

Cory Arcangel &  
Olia Lialina

<https://art.teleportacia.org/#CenterOfTheUniverse>

# Abb. 17

Mach die ersten Schritte mit Squarespace

## Wie erstelle ich eine Webseite?

1. Wähle ein Template und starte eine kostenlose Probephase.
2. Hole dir einen kostenlosen benutzerdefinierten Domainnamen für das erste Jahr deines Website-Jahresabos.
3. Verwende unseren Website-BUILDER, um deinen eigenen Text und deine Fotos hinzuzufügen.
4. Passe die Website mit Hunderten von Schriftarten, Farben und Agenturbildern an deine Marke an.
5. Du hast kein Logo? Erstelle eines mit unserem [Online-Tool](#).
6. Veröffentliche deine Website und bewirb sie mit Social-Media- und [E-Mail-Marketing](#).

## So verkaufst du online

## Wie erstelle ich einen Blog?

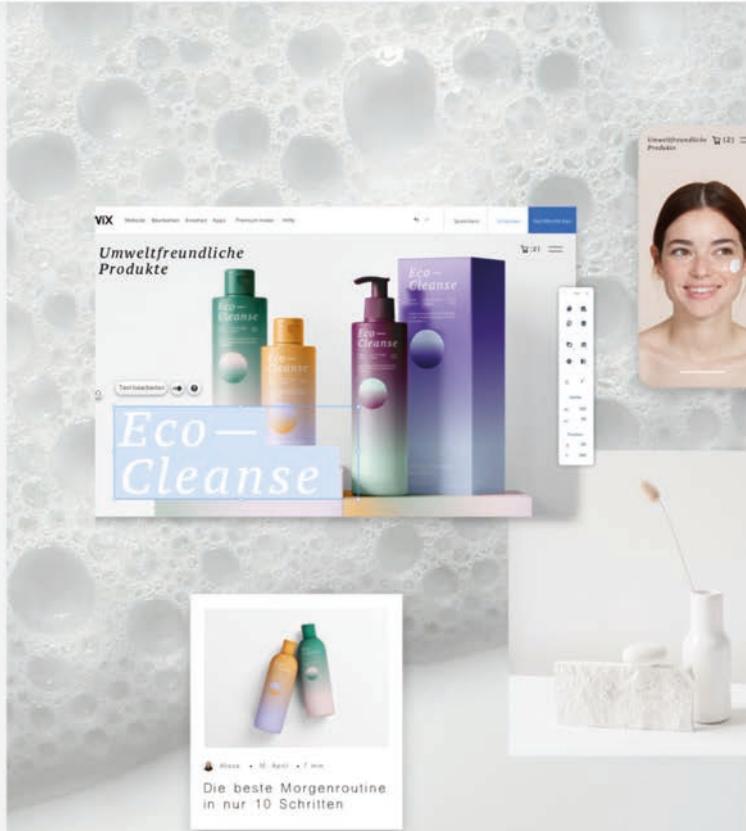
<https://de.squarespace.com/>

**WIX** Website ▾ Business ▾ Ressourcen ▾ Preise Support | Mehr ▾ [Anmelden](#)

# Deine Website Design-Komplettlösung

Mit den intuitiven Technologien von Wix und den integrierten Funktionen, hast du jeden Designspielraum, um eine Website zu erstellen, auf die du stolz sein kannst.

[Jetzt starten](#)



The screenshot shows the Wix website builder interface. At the top, there's a navigation bar with links for 'Website', 'Business', 'Ressourcen', 'Preise', 'Support', and 'Mehr'. On the right side of the header is a 'Anmelden' button. The main content area features a large, bold title 'Deine Website Design-Komplettlösung'. Below the title, a paragraph of text describes the benefits of using Wix. At the bottom left is a black button with white text that says 'Jetzt starten'. The central part of the screen displays a live preview of a website for 'Umweltfreundliche Produkte' (Eco-Cleanse). The preview shows several skincare products (bottles and boxes) arranged on a surface, with a woman's face smiling in the background. A sidebar on the right contains various design tools and options. At the bottom of the preview, there's a smaller inset showing two bottles and some text: 'Die beste Morgenroutine in nur 10 Schritten'.

<https://www.wix.com/>