开放代码.连结机器人

OPEN CODES. CONNECTED BOTS

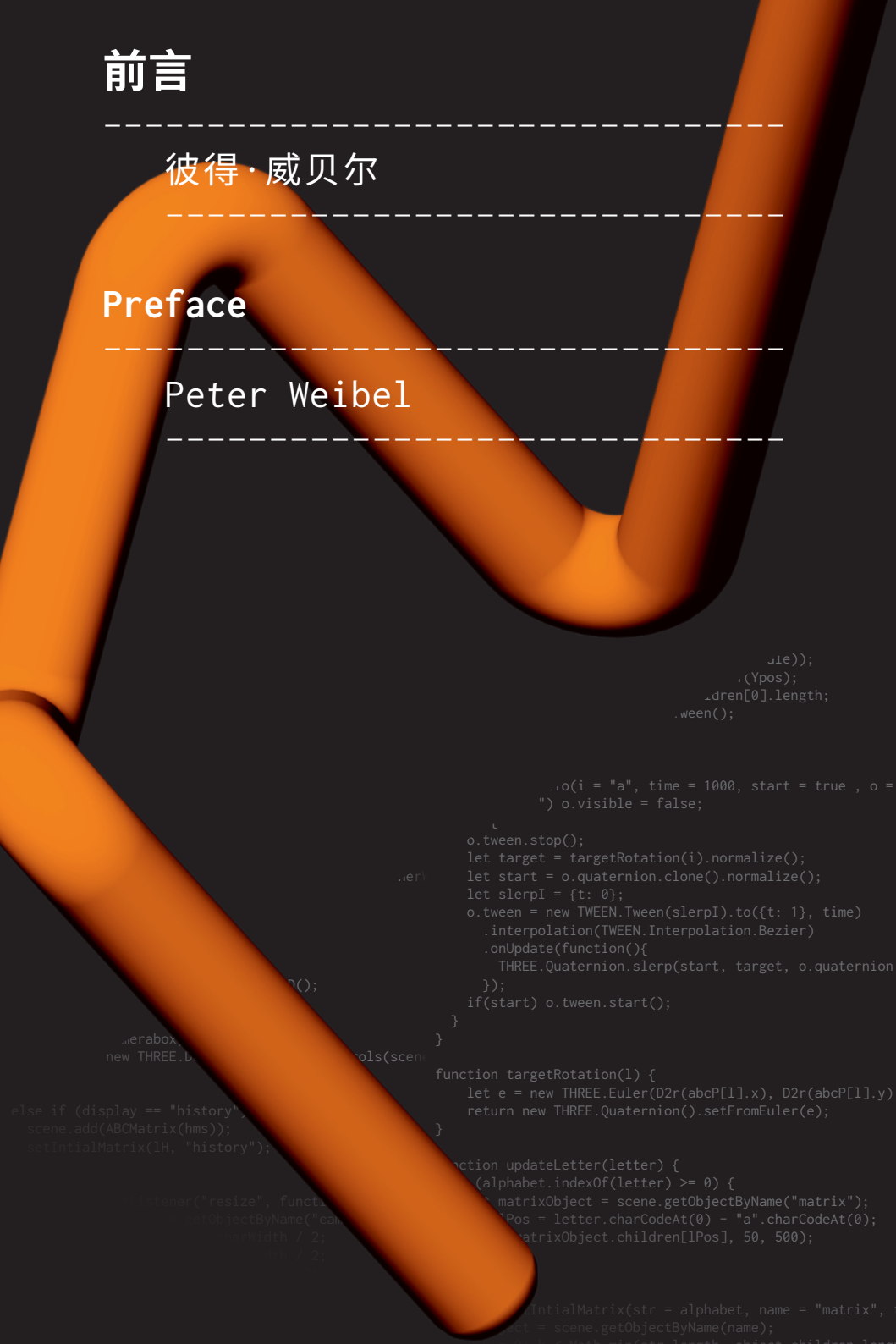
2019/07/20 - 2019/10/07

前言

彼得·威贝尔

Preface

Peter Weibel



所有文化都是一种文化技术并且依赖于工具，不管是一支笔、一把刷子或一架钢琴。 模仿文化的工具一直都是实际存在的硬件。而数字文化的工具，从编程到代码，都是非物质软件。艺术家发展出了人与机器之间的新界面，并在新兴专业系统——从算法到人工智能——的帮助之下，衍生出他们之间新的互动与合作。艺术家创造出基于数据与传感器的环境和活动，并使个体与社会以及自然系统的关系得到完善。如果这个世界是一个数据场，那么我们需要代码的艺术在其中指引我们。我们将一起打开通向代码世界的大门，也是走入模拟与数字的世界的大门。

生活在一个数字化的世界意味着一种愈发沉浸于编程化、智能化环境中的生活，即所谓的“脚本化现实”和“信息空间”（infosphere）。这个脚本一部分是由传感器决定的，为我们提供关于我们所处现实环境的信息。来自于传感器的数据由算法进行处理，使我们在这个犹如数据场一般的世界中穿梭。因为我们身处的这个世界已不仅仅是一个自然世界，而是一个越来越受人工影响的人造数据世界。

众所周知，我们这个数据世界的基础是由 0 和 1 组成的二进制代码。尽管字母代码几千年来作为主要编码占据了人类的文化与通讯，如今数字代码已主导了我们的世界。这种代码本质由 0 和 1 的密码组成，并由此形成了几乎无穷尽的数字组合。

塞缪尔 · 摩尔斯于 1833 年为字母代码所做的成就——将拉丁字母表中的 26 个字母缩减至长短两种信号——戈特弗里德 · 威廉 · 莱布尼茨在 1697 年就已在数字代码中实现。[1] 莱布尼茨证明了所有数字都可以被两个数码代表，0 和 1。这意味着任何信息都有被编码并解码成一列二进制数码的可能性。

随后，数据、算法和编程的语言发展成了一种通用语言，一个包含声音、图像、文字和事物的世界从中逐渐浮现。因此数学早已不仅仅是一种自然的语言，它也成为了一种文化的语言。这本描述当代世界的书必须以《事物与数据》命名。事物、声音、词汇和图像的关系曾经是不可改变的，而如今在数字化世界中，数据与词汇、图像与声音之间的关系已可以互相转换。

1　见 G.W. 莱布尼茨至鲁道夫·奥古斯特（布伦瑞克·吕讷堡公爵）的信件中，后被称为新年信件，1697 年 1 月 12 日

All culture is culture technique and relies on tools, be it a pen, a brush or a piano. The tools of analogue culture have been material hardware. In the digital culture the tools are immaterial software, from programming to coding. Artists have developed new interfaces between men and machine, which allow new interactions and cooperation between them with the help of new expert systems, from algorithms to artificial intelligence. Artists create sensor and data based environments and events, which optimize the relations between individuals and social and natural systems. If the world is a field of data, than we need the art of coding that can guide us through it. We together will open the doors to the world of codes, which opens the doors to the analogue and digital world.

Living in digital worlds means increasingly a life spent in a programmed, smart environment, a so-called "scripted reality" and "infosphere". The script is dictated in part by sensors, which provide information about the state of reality around us. The data from the sensors are processed by algorithms, which steer us through the world as through a field of data. For the world we live in is not so much just a natural world all by itself, but more and more an artificial, human-made data world.

The basis of our data world, as everybody knows, is the binary code, made of 0s and 1s. While the alphabetical code predominated as the primary code for human culture and communication for thousands of years, today a numerical code dominates our world. This code essentially consists of the ciphers 0 and 1, through which an almost infinite set of numbers can be formed.

What Samuel Morse did in 1833 for the alphabetical code, namely, reduce the 26 letters of the Latin alphabet to two kinds of signals, long or short signals, Gottfried Wilhelm Leibniz accomplished in 1697 for the numeric code.[1] Leibniz proved that all numbers can be represented by just two digits, 0 and 1. Which implies the possibility of encoding and also decoding any kind of information as a row of digits in a binary system.

Subsequently the language of data, algorithms, and programming has become a universal language out of which the world of sounds, images, texts, and things emerges. Thus, mathematics has long since ceased to be just the language of nature; it has become the language of culture, too. The book that describes the contemporary world must be titled *The Things and the Data*. The relationship between things, sounds, words, and images used to be irreversible. However, now the relationships between data and words, images and sounds are reversible in the digital world.

1 G. W. Leibniz in a letter to Rudolph August, Duke of Brunswick-Lüneburg, known as the New Year's Letter, January 12, 1697.

展览概念

丽维亚·诺拉斯克·洛萨斯

Exhibition concept

Lívia Nolasco-Rózsás

“开放代码”将计算和艺术以各种方式结合在一起。它是一种新的聚集方式。在这里，围绕计算机代码创造与理解的知识产出以及批判性的艺术实践会同时进行。该项目旨在赋予其参与者以思想为工具重获理解并连接现实的能力，并对数字代码、计算机编程与软件的谱系及社会影响进行反思。

“开放代码”项目此次于新时线媒体艺术中心（CAC）的重现展出将聚焦算法的情感共振以及计算模拟和物理现实感知之间的交互牵涉。在全球规模计算时代下的当代社会，驾驭代码世界的能力以及相关的数字知识素养至关重要，尤其当算法主体设计的主旨在于转变公众舆论时。

虚拟软件主体——通俗地称为机器人（bots）——运行大部分重复性任务的速度比人类所能达到的极限速度要快得多。它们又或专用于线上平台对话，这种情况下的编程使它们如人类一般表现。社交媒体机器人是新兴的高可见度经济的产物，且目前占线上流量的一半以上。一些机器人可以学习我们，其它则能够爬行于网络中对内容建立索引，更甚有努力改变我们思维模式并用于政治活动的机器，其它的会恶意投放垃圾邮件。

“Bot”一词来源于“Robot”，由卡雷尔 · 恰佩克首次用于他 1920 年出版的科幻小说，描述了一个虚构的人形机器人。“机器人”（robot）一词被用于指代自动机器，正如捷克语中的单词 'robota' 所指，它能够执行复杂的任务亦或强制工作。机器人无论是否有型，都是为了提供服务，就像 1966 年约瑟夫 · 维泽鲍姆所描述的自然语言对话程序 ELIZA 一样。这个早期的聊天机器人对某些用户来说是一个比人类心理学家更好的选择。几十年后，像 Alexa 这样的虚拟助手不仅能够聊天，还能够执行更复杂的任务。在世界的某些地方，它们就如吹风机一样普通地存在于许多家庭之中。

目前，不计其数的关于以二进制为基础的人类替代品的实验和实际应用正在运行中。除了它们的优点之外，实现这些工具还是需要付出代价的——繁重的人工劳动隐藏在虚拟助手的流畅操作背后。聊天机器人在和用户的谈话中将知识汇总，而这些用户不一定代表整个社会，因此它们的陈述经常受到种族主义或性别歧视观点的左右，并可能强化它们已经接收到的各种社会排斥观念。

机器人无论是播放音乐、挖掘比特币、与我们聊天或为我们聊天，它们的影响力都在增长。这种发展的后果是，我们可能要问自己是否

正在变成机器人?当我们身处一个时髦的共享工作空间、坐在电脑前时，机器人正在影响我们的决定。算法能以惊人的程度麻木人类主体，这对于人机之间和人类之间的感知、记忆以及社交互动具有无法避免的影响。人类对计算的依赖远超二元对立，已经变为在多个层面上的关联。

本次展览包含基于计算机代码的艺术作品、也包括了旨在揭示计算机代码如何深度渗透我们生活、社会、地缘政治格局、财政体系、劳动条件、基础设施、环境，甚至我们对自己源代码 DNA 的感知的艺术作品。

借助艺术家和程序员的大约 20 件作品，展览从八个主题呈现了数字代码的世界以及未来受其影响的生活领域：# 代码谱系 (#GenealogyOfCode)、# 编码 (#Coding)、# 机器学习 (#Machine-Learning) 、# 算法治理（#AlgorithmicGovernance)、# 算法经济 (#AlgorithmicEconomy)、# 虚拟现实（#VirtualReality)、# 劳动与生产（#Labor&Production）以及 # 遗传密码（#GeneticCode)。这些关键词共同勾勒出一副想象蓝图，为理解我们所居住的这个数字世界提供基础。

这场展览空间将被打造成一场基于建筑空间的“跑酷”，为访客提供利用工作空间进行独立创意活动的机会。CAC 的空间将具有多种功能：展厅不仅展示艺术品，也可用于活动、工作坊、聚会和讲座，以及在“开放代码”项目对应主题的图书馆中自由翻阅。

该项目在上海的呈现得到了中央美术学院、华东师范大学和同济大学师生、及众多黑客和创客空间的参与。展览将专门开辟一个空间呈现相关的学生作品。本次展览由瑞士著名家居设计品牌 Vitra 独家赞助。

CAC“开放代码 . 连接机器人”系卡尔斯鲁厄艺术与媒体中心 |ZKM 于 2017 年 10 月 20 日至 2019 年 6 月 2 日期间所策划项目“开放代码”（***Open Codes***）的衍生展览。

Open Codes brings computing and art together in various ways. It is a new form of assembly, combining practical knowledge of computer code and critical artistic approaches in a single venue. The project seeks to empower its participants to regain access to reality through instruments of thought and to reflect on the genealogy and current social impact of digital code, computer programming, and software.

The current iteration of this project at the Chronus Art Center focuses on the affective resonance of algorithms and the reciprocal, perceptual entanglements of computational simulacra and physical reality. The ability to navigate the world of code, hand in hand with digital literacy, are essential to contemporary society in the age of planetary-scale computation, especially when algorithmic agents are designed to influence public opinion.

Virtual software agents, colloquially referred to as bots, run mostly repetitive tasks at a higher speed than humans ever could. Some specialize in conversations on online platforms, where they are programmed to act like regular people. Social media bots are a product of the new economies of visibility and currently make up more than half of online traffic. Some bots can learn from us; others can spider the web to index content; further ones strive to alter our mindsets and are used in political campaigns, and others maliciously litter inboxes with spam. .

The world bot derives from "robot", first used by Karel Čapek in his 1920 science fiction novel, to describe a fictional humanoid. Robot became known as a word for autonomous machines, capable of carrying out complex tasks, that can be forced to work, as the meaning of the Czech word 'robota' suggests. Robots, whether with or without a built body, are meant to provide services, much like the natural language conversation program, ELIZA, described by Joseph Weizenbaum in 1966. This early chatbot was in some users' opinion a better psychologist than its human alternatives. A couple of decades later virtual assistants, like Alexa, are not only capable of chatting, but of accomplishing more complex tasks, and are now as mundane as a hairdryer in certain parts of the world.

Countless experiments and actual applications of binary-based human substitutes are now in use. Despite their advantages, these tools come at a price –the seamless operation of a virtual assistant obscures cumbersome human labour. Chatbots aggregate knowledge from their conversations with users, who do not necessarily represent society as a whole, thus their statements often repeat racist or sexist opinions and may reinforce the kinds of social exclusion they have been fed with.

Whether they play music, mine bitcoin, or chat with us, the bots' influence grows. As a consequence of this development, we might ask if we are becoming bots ourselves? As we sit in a

stylish co-working space in front of a computer, bots are influencing our decisions. Algorithms numb human agency to a shocking degree, with inevitable implications for perception, memory, and social interactions, whether with bots or among ourselves. Human dependency on computation has become mutually reinforcing on multiple levels, going beyond binary oppositions.

The exhibition includes artworks based on computer code, as well as artworks that reveal how deeply such code has penetrated our lives, societies, geopolitical situations, fiscal systems, labour conditions, infrastructure, environment, and even the perception of our own source code, DNA.

With the aid of around 20 works by artists and programmers, the exhibition presents the world of digital code and its future influence in eight sections: #GenealogyOfCode, #Coding, #MachineLearning, #AlgorithmicGovernance, #AlgorithmicEconomy, #VirtualReality, #Labor&Production, and #GeneticCode. These key terms form an imaginary map, which serves as the grounding for understanding the world we inhabit.

The discourse of the exhibition is laid out as an architectonic parcours to offer visitors the opportunity to use the workstations for independent creative activities. The spaces of Chronus Art Center bear multiple functions: the exhibition halls display artworks, but are also available for events, workshops, meetups, and lectures, as well as independently browsing Open Codes' thematically curated library.

The Shanghai iteration of the project was partially developed in collaboration with the Central Academy of Fine Arts, East China Normal University and Tongji University, as well as with hacker and maker spaces. Selected student works will be presented throughout the duration of the exhibition. The exhibition is generously supported by Vitra.

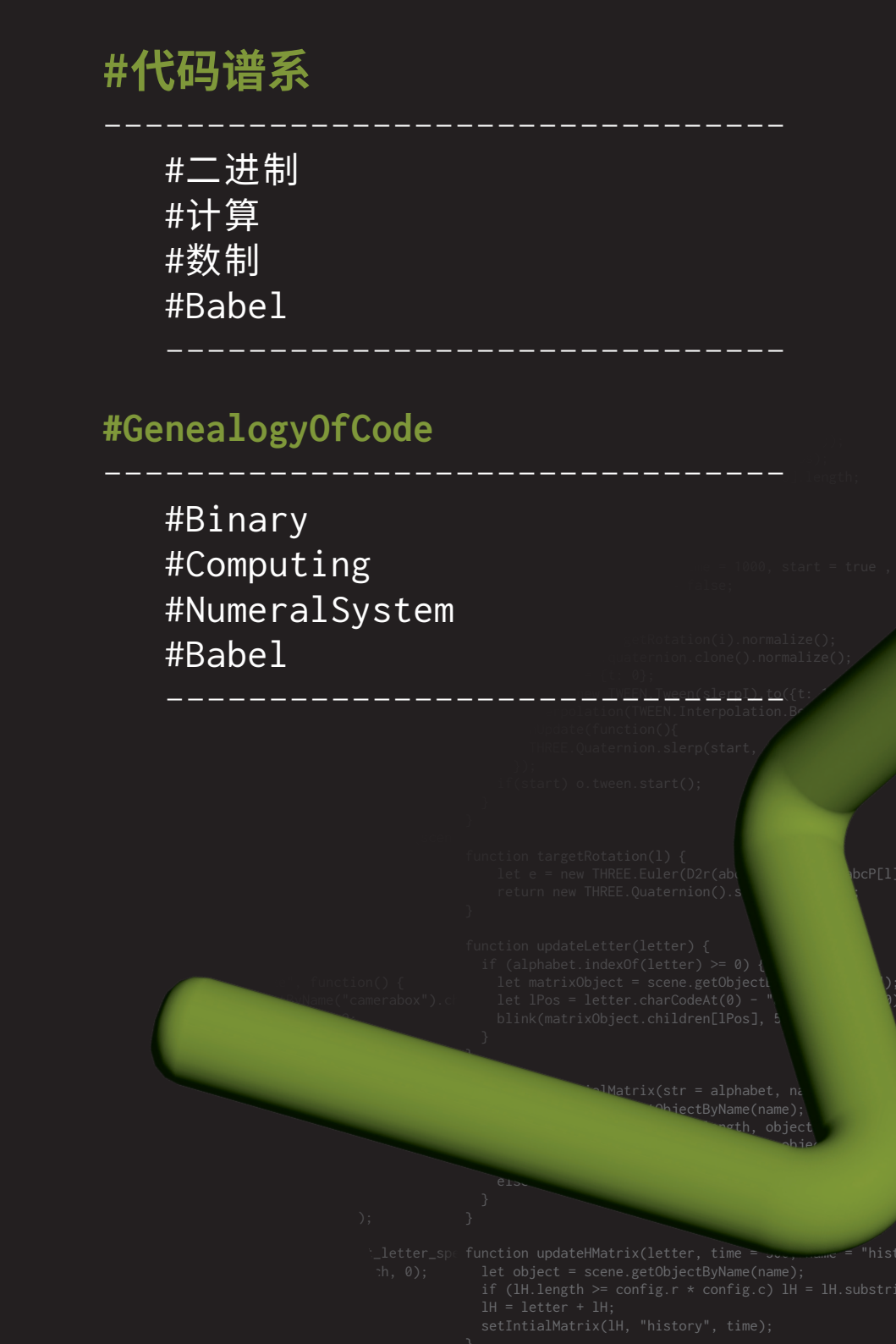
Open Codes. Connected Bots is a satellite exhibition of Open Codes at the ZKM | Centre for Art and Media Karlsruhe, Germany, which ran from October 20, 2017 to June 2, 2019.

#代码谱系

#二进制
#计算
#数制
#Babel

#GenealogyOfCode

#Binary
#Computing
#NumeralSystem
#Babel



算法的起源显然不仅仅来自于个人电脑或它们在二十世纪的祖先们。若要了解构成如今的算法原则的基石，一切将至少追溯至中世纪。

雷蒙·卢尔（1232–1316）是一个来自马略卡岛的思想家，他致力于研发一个可以解决基础神学问题和哲学问题的系统——通过丰富的图表来探索发掘所有概念的组合可能性。他的著名作品《大术》（1274–1308）[1] 为这个他称之为“大术”的过程做了最详尽的解释。戈特弗里德·威廉·莱布尼茨（1646–1716）撰写了《论组合术》，并在其中提出了受卢尔启发而构思的关于逻辑与形而上学的相关性[2]。

于 1679 年，莱布尼茨在他未公布的一封信件中提到了只运用 0 和 1 作为数字的**#二进制**（“二价”或“二进制算术”）。他不确定这个发明会带来什么实际的用处，但他在许多写给同事的信件中多次提及了二进制带来的可能性。1701 年，他告诉一位法国数学家他似乎能预见到在二进制的无尽数字背后有一些独特的东西能够被实现。莱布尼茨早在 1679 年就描述了第一台能够进行二进制计算的设备（**#计算**）。这段描述一直没有公布于众，而这台机器也没有在他生前被制造出来。[3]

两个世纪后，查尔斯·巴贝奇正在研发他的差分机，并随后发明分析机，但由于经费不足，两者都不是在巴贝奇的完全监管下制造完成的。分析机应是第一个拥有广义用途的计算机，但它仍是一个机械计算机。“当利用打孔卡片的想法出现的那一刻，计算的边界就已经被突破了。分析机并不只是一个普通的‘计算机器’，”[4] 阿达·洛芙莱斯（1815–1852）——如今被认为是世界上的第一位程序员——对巴贝奇的自动计算机这样评论道。这台年代久远的机器使用十进制运作**#数制**，而如今的电脑都仅用二进制系统，二进制由莱布尼茨首先使用，之后被乔治·布尔（1815–1864）再次带到大众的视野。

布尔在《逻辑的数学分析》（1847）中首次使用代数的形式表现逻辑，布尔代数随即问世。[5] 布尔的二进制系统使用三种最基础的表示进行逻辑运算——“与”，“或”和“非”。[6]

这个系统一直没有被运用到实践中，直到“克劳德·香农于 1937 年在那一篇可能是历史上最有影响力的硕士学位论文中证明了——简易的电报继电器开关可通过不同的连接方式实现整个布尔代数的运算。”[7]

也是在 1937 年，艾伦·图灵（1912–1954）制造了一个布尔逻辑乘法器，并在他的论文《决定性问题》中提出了可计算性理论。[8]

这篇论文在一年前就已写完，当时他还在研究著名的图灵机——图灵机不是一个真实存在的计算机，而是一个关于计算的数学模型。在他的布尔逻辑乘法器中，图灵希望“通过一套继电器开关设备系统来表现图灵机的逻辑结构”[9]，这个想法成为了创造乘法器的基石。

不久，从电子计算机初现的 1940 年代开始，第一代编程语言陆续被设计出来，汇编语言（asm）就是其中一个。由于这种语言与机器语言有很高的关联性，这个低级编程语言能够直接被转换成可执行的机器代码。自 1950 年代起，高级编程语言就开始取代它们的“低级”祖先。许多编程语言自此被编写和建立起来，比如前期有 ALGOL（ALGOrithmic Language），随后出现了 Fortran，Pascal，C++，Java 和 Python，这都仅是一小部分。“这座后现代的巴别塔始于以硬件配置作为语言扩展的简单指令代码，经过了以操作码作为扩展的汇编语言，最终到达了用汇编语言作扩展的高级计算机语言”[10]（# Babel）

这些语言都基于二进制数字系统，一系列的“开”和“关”可以控制电路中电流的流动和停止。尽管计算机语言的基本构成很简单，但计算机语言可以用来描述极其复杂的操作。以上提到的计算设备中，除了巴贝奇的机器使用十进制，其它设备都使用二进制代码运行。基于香农的贡献和晶体管的应用，二进制系统在计算过程中变得无处不在。

除了#算法（参见关键词#编码）和计算以外，其他所有可解码的东西也都可以用二进制代码表示。其中最著名的例子也许是 ASCII（美国信息交换标准代码），也就是计算机屏幕上显示的字符，这套转换系统于 1960 年从电报编码中开发出来。

当前计算机的存储量很可能不足以满足未来大量复杂的计算。现代计算机的发展非常迅速，尤其是将汽车的发展与计算机相比时——如果 1971 年的汽车能够拥有计算机芯片性能同等的提升速率，那么 2015 年的新车型应该已拥有 6.8 亿公里每小时的最高速度。[11]

量子计算也许就是导致近来看似必然的扩张因素。在量子计算机中，逻辑操作是原子级的。原子相较比特能寄存更多信息，它们能够同时寄存 0 和 1，因此量子比特（quantum bits）或量子位（qubits）比经典比特位有更高的效率，因为它们能同时进行两个运算。[12]

“计算还能在宇宙中持续多久?现今的观察结果显示宇宙将永远扩张下去。在它扩张之时，被执行的操作码数量与可用的比特数量将在我们的视野里持续增长下去。”[13]

1 雷蒙·卢尔,《歌剧》,第 2 卷,福尔曼·霍兹堡,斯图加特·巴德·坎斯达特,1996,第 228-663 节,另见《对话:雷蒙·卢尔的思想方法与艺术实践》,阿玛多尔·维加,彼得·威贝尔,齐格弗里德·齐林斯基编,明尼苏达大学出版社,明尼阿波利斯市,2018

2 见阿纳·H·马洛斯蒂加,“组合技术与时间:卢尔,莱布尼茨和皮尔斯”,于《关于卢尔的研究》,第 32 卷 ,1992,第 105-134 页,此处第 111 页

3 见赫尔曼 J. 格雷夫,“探索二进制世界”,于《冯·莱布尼兹先生对于 0 和 1 的见解》,西门子股份公司,柏林,慕尼黑,1966,第 21-31 页

4 L. F. 梅纳布雷亚,《查尔斯·巴贝奇发明的分析机简图》,翻译版 , 由阿达·洛芙莱斯附加纪念注释,理查德与约翰 E. 泰勒印刷,伦敦,1843,第 696f 页

5 乔治·布尔,《逻辑的数学分析:一篇指向演绎推理的微积分的论文》,麦克米伦,剑桥,1847

6 见保罗 J. 纳辛,《逻辑学家与工程师:乔治·布尔与克劳德·香农如何创造了信息时代》,普林斯顿大学出版社,普林斯顿,牛津,2013

7 弗里德里希·基特勒,“世界上没有软件”,于《斯坦福文献评论》,第 9 卷,1992 年春第 1 本,第 81-90 页,此处第 88 页

8 见艾伦·图灵,“论可计算数及其在判定性问题上的应用”,于《伦敦数学协会会报》,第 s2-42 卷,1937 年 1 月第 1 本,第 230-265 页

9 安德鲁·霍奇斯,《艾伦·图灵传:如谜的解谜者》,普林斯顿大学出版社,普林斯顿(新泽西),牛津,2014,第 177 页

10 基特勒 1992,第 82 页

11 见蒂姆·克罗斯,“消失点:隐形计算机的崛起”,于《卫报》,2017 年 1 月 26 日,线上来源:https://www.theguardian.com/technology/2017/jan/26/vanishing-point-rise-invisible-computer, accessed 09/13/2017.

12 赛斯·劳埃德,《宇宙编程:量子计算机科学家看宇宙》,二手书,纽约,2007 年,第 136-139 页

13 同上 , 第 206 页

Computation clearly does not begin with personal computers and their direct ancestors from the twentieth century. To find the roots of the principles upon which computation of today is based on one has to go back at least to the Middle Ages.

Ramon Llull (1232–1316), a Majorcan thinker, sought to develop a system for solving basic theological and philosophical questions, a method by means of which he tried to find and explore all possible combinations of concepts with the help of dynamic charts. This procedure, his so-called Ars magna [Great Art], is explained most extensively in his notable work Ars magna (1274–1308).[1] Gottfried Wilhelm Leibniz (1646–1716) conceived his Dissertatio de arte combinatoria [Dissertation on the Art of Combinatorics] (1666), in which he proposes a parallelism between logic and metaphysics inspired by Llull.[2]

In 1679 Leibniz wrote about a #Binary system ("dyadic" or "binaria arithmetica") in one of his unpublished letters, which uses only 0 and 1 as numbers. He was not sure about the practical use of his invention, but frequently wrote about its possibilities in various letters to his colleagues. In 1701 he claimed to a French mathematician that he imagined to foresee, that by this means and the endless rows there is something to achieve, which wouldn't be easy in another way. Leibniz described the first computing device (#Computing) that works with the binary system as early as 1679. The description remained unpublished and the machine was not built in his lifetime.[3]

Two centuries later Charles Babbage was working on his Difference Engine, followed by the Analytical Engine, neither of which were constructed entirely under his guidance due to insufficient funding. The Analytical Engine would have been the first general purpose computer, but still a mechanical one. "The bounds of *arithmetic* were, however, outstepped the moment the idea of applying the cards had occurred; and the Analytical Engine does not occupy common ground with mere 'calculating machines'"[4], wrote Ada Lovelace (1815–1852), acknowledged today as the first programmer, in her notes on Babbage's computing automaton. This early device operated with a decimal #NumeralSystem. Computers nowadays are based only on a binary numeral system, first used by Leibniz, then reintroduced by George Boole (1815–1864).

Boole first cast logic into algebraic form in his book *The Mathematical Analysis of Logic* (1847), introducing the Boolean algebra.[5] Boole's binary system is based on the three most basic operations used as logical operations: AND, OR, and NOT.[6]

This system was not put into operation until "Claude Shannon, in 1937, proved in what is probably the most consequential M.A. thesis ever written, that simple telegraph switching relays can implement, by means of their different interconnections, the whole Boolean algebra."[7]

Also in 1937, Alan Turing (1912–1954) built a Boolean logic multiplier and proposed a theory of computability in his essay on the "Entscheidungsproblem" [decision problem].[8] The paper had already been written the previous year while he was working on his well-known Turing machine, which was not a physically existing computer, but a mathematical model of computation. With his multiplier based on Boolean logic, Turing tried "to embody the logical design of a Turing machine in a network of relay-operated switches",[9] which served as a basis for creating the multiplier.

Soon after, from the 1940s with the appearance of electronically powered computers, different programming languages were designed and assembler (asm) was one of the first. This low-level programming language can be converted into executable machine code in one step, as there is a very strong correspondence between the language and the machine code. From the 1950s onward high-level programming languages started to replace their "low" antecedents. Dozens of programming languages have been written and developed, starting with ALGOL (ALGOrithmic Language), then Fortran, Pascal, C++, Java, and Python, to name just a few. "This postmodern Tower of Babel reaches from simple operation codes whose linguistic extension is still hardware configuration, passing through an assembler whose extension is this very opcode, up to high-level programming languages whose extension is that very assembler."[10] (#Babel)

All these languages are based on a binary number system, a sequence of "ons" and "offs" allowing electricity in the circuit to flow or stop. Despite the simplicity of their basic components, programming languages can describe exceedingly complex operations. The computing devices mentioned above all run with binary code, except Babbage's machines, which used the decimal system. Due to Shannon's work and the implementation of transistors binary systems became ubiquitous in computing.

In addition to algorithms (see #Algorithm in key area #Encoding) and calculations anything decodable can be described by binary code. Perhaps the best-known example is ASCII (American Standard Code for Information Interchange), the characters displayed on a computer screen, which was developed from telegraph code beginning in 1960.

The capacities of current computers may not be sufficient for the amount and complexity of computing in the future. The development of modern computers has been very fast, which is even more striking when compared to the improved performance of cars. If cars made in 1971 had improved at the same rate as computer chips, then by 2015 new models would have had top speeds of about 680 million kilometers per hour.[11]

Quantum computing could be the answer to the recent and seemingly inevitable expansion.

In a quantum computer logical operations are performed on an atomic level. Atoms register more than bits, they are able to register 0 and 1 at the same time, and thus quantum bits or qubits are more efficient than classical bits because they can perform two computations simultaneously.[12]

"How long can computation continue in the universe? Current observational evidence suggests that the universe will expand forever. As it expands, the number of ops performed and the number of bits available within the horizon will continue to grow."[13]

1 Raimundus Llullus, *Opera*, 2 vols., Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1996, S. 228–663. See also *DIA-LOGOS. Ramon Llull's Method of thought ad Artistic Practice*, Ed. by Amador Vega, Peter Weibel, Siegfried Zielinsky, University of Minnesota Press, Minneapolis, 2018

2 See Ana H. Maróstica, "*Ars Combinatoria* and Time: Llull, Leibniz and Peirce," in: *Studia Lulliana*, vol. 32, 1992, pp. 105–134, here p. 111.

3 See Hermann J. Greve, "Entdeckung der binären Welt," in: *Herrn von Leibniz' Rechnung mit Null und Eins*, Siemens Aktiengesellschaft, Berlin, Munich, 1966, pp. 21–31.

4 L. F. Menabrea, *Sketch of the Analytical Engine Invented by Charles Babbage*, translated from the French and supplemented with notes upon the memoir by Ada Lovelace, printed by Richard and John E. Taylor, London, 1843, p. 696f.

5 See George Boole, *The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning*, Macmillan, Cambridge, 1847.

6 See Paul J. Nahin, *The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age*, Princeton University Press, Princeton (NJ), Oxford, 2013.

7 Friedrich Kittler, "There Is No Software," in: *Stanford Literature Review*, vol. 9, no. 1 Spring 1992, pp. 81–90, here p. 88.

8 See Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," in: *Proceedings of the London Mathematical Society*, ser. 2, vol. 42, no. 1, January 1937, pp. 230–265.

9 Andrew Hodges, *Alan Turing: The Enigma*, Princeton University Press, Princeton (NJ), Oxford, 2014, p. 177.

10 Kittler 1992, p. 82.

11 See Tim Cross: "Vanishing point: The rise of the invisible computer," in: *The Guardian*, 01/26/2017, available online at: https://www.theguardian.com/technology/2017/jan/26/vanishing-point-rise-invisible-computer, accessed 09/13/2017.

12 Seth Lloyd, *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*, Vintage Books, New York, 2007, pp. 136–139.

13 Ibid., p. 206.

#编码

#摩尔斯电码
#声音编程
#算法
#软件
#硬件
#界面
#解码

#Encoding

#MorseCode
#ProgrammingSound
#Algorithm
#Software
#Hardware
#Interface
#Decoding

从#遗传密码（参见相关关键词）到音乐符号，从为了解决感官障碍而发明的通信系统，比如手语，到#莫尔斯电码，从安全代码和社会行为准则，“代码”这词可以广泛地指代可识别的元素和我们熟悉的过程，但它在#编程（参见关键词#劳动与生产）和计算方面意味着什么呢？

《计算字典》将代码定义为“将消息从一种符号形式（源字母表）转换为另一种符号形式（目标字母表）的规则。”[1] 因此，代码可以被视为一组指令，“把输入的信息从一个状态改变到另一个状态，其结果就是代码完成了自己的工作”。[2] 这种执行方式恰恰指出了代码的一个主要特征：它不仅可读而且可被执行；既是一种媒介，又是一种指令。这一根本的优点使代码区别于通用语言。通用语言可以被读写，但并不会造成实际的变化。从这个层面上来讲，计算机代码“是第一种实际完成语言目的的语言 — 它是一种将意义转化为行动的机器。”[3]

计算机代码另一个重要的方面是它那具有欺骗性的不可见性。代码通常是隐藏的；它本身缺乏物质性，在机器内部几乎看不见，但它在我们的世界中产生可见的、具体的和有形的效果。以声音编程作品为例，输出的不同的声音或者谱曲其实是一行或多行代码的成果（#声音编程）。

与代码类似，尽管#算法这词常被定义为用于各种过程的需要执行的动作序列，算法通常也与计算和编程相关联。算法是一组用于指定如何解决问题或执行任务的规则。从这种意义上讲，食谱或说明书也可以被理解为算法。在计算领域中，这些规则被制定出来用以处理数据，并产生一个输出值。

算法和代码都是不可见的部分，通常概括在#软件这一术语下。软件是“一个指代无形的、非物理的计算机系统组件的通用术语。”[4] 与其相反的，#硬件是组成计算机系统的物理元件，例如主板。为了使软件和硬件能够交换信息，需要第三个元素#界面。界面也连接软件、硬件和人。要理解这种交换是如何工作的，我们只需要思考一下“电源”按钮：电源按钮是你和机器背后的电线之间的界面。按下它，机器将会打开和关闭。

即使在普遍的每日应用例如发送短信中，代码也执行了极其多的算法操作。在计算机中，#编码是将一系列字符转换为特定格式以进行有效传输或存储的过程。为了将编码格式转换回原始字符序列，需

要相反的过程，称为#解码。这两种过程通常用于数据通信，网络和存储，特别是与无线通信系统相关的领域中。通过运行这些流程，现在的代码能够在几秒钟内处理和控制许多不同的操作，为社会、经济或文化活动塑造及创造新的视野。

1 安德鲁·巴特菲尔德和吉阿德·艾克贝·宁甘地，《计算机科学词典》，第 7 版。牛津大学出版社，牛津，2016 年，第 93 页

2 罗伯·柯池和马丁·道奇，《代码 / 空间：软件和日常生活》，麻省理工出版社，剑桥（马萨诸塞州），伦敦，2011 年，第 25 页

3 亚历山大·R·盖洛韦，《议定书：权力下放后如何控制》，麻省理工学院出版社，剑桥（马萨诸塞州），伦敦，2004 年，第 166 页

4 苏珊·M·哈奇，《计算字典》，第 2 版。牛津大学出版社，牛津，1986 年，第 352 页

From #GeneticCode (see correlating key area) to music notation, from communication systems for sensory impairments, such as sign language, to #MorseCode, from safety codes and standards to social rules of conduct, the term "code" may outwardly designate recognizable elements and familiar processes, but what does it mean in terms of #Programming (see key area #Labor&Production) and computing?

The *Dictionary of Computing* defines code as "a rule for transforming a message from one symbolic form (the source alphabet) into another (the target alphabet)."[1] Therefore, code could be seen as a set of instructions "that changes the input from one state to another, and as a consequence the code performs work."[2] This way of performing designates precisely one of its main characteristics: code is at the same time legible and executable; it is simultaneously a medium and an instruction. This essential virtue makes code different from common languages, which can be read or written but do not cause any changes by doing this per se. In that sense, computer code "is the first language that actually does what it says – it is a machine for converting meaning into action."[3]

Another crucial aspect of computer code is its deceptive invisibility. Code is generally hidden; it lacks materiality in itself and remains mostly unseen inside the machine, but it generates visible, concrete, and tangible effects in the world. Taking a programmed sound work as an example, the different sounds or compositions would be the output, in other words, the result of one or many lines of code (#ProgrammingSound).

Similarly to code, the word #Algorithm is often associated with computing and programming, although the definition of algorithm, being a sequence of actions to be performed, could be applied for various procedures. An algorithm is a set of rules that specify how to solve a problem or perform a task. In that sense, a recipe or a manual of production could be understood as an algorithm, too. In computing, these sets of rules or steps are established in order to process data and, as we have already seen, produce an output.

Algorithms and code are also the invisible part, commonly summarized under the term #Software, which is "a generic term for those components of a computer system that are intangible rather than physical."[4] By contrast, #Hardware is the compilation of physical components that form a computer system like, for example, the mainboard. In order that software and hardware can exchange information, a third element is needed, the #Interface, which also can be the link between software, hardware, and humans. To understand how this exchange works, we only have to think about a "power" button: the button is, namely, the interface

between you and the electrical wiring behind the machine. You press it and the machine turns on and off.

Even in common, ordinary applications such as sending an SMS, code executes an extremely high number of algorithmic operations. In computers, #Encoding is the process in which a sequence of characters is transformed into a specific format for efficient transmission or storage. In order to convert an encoded format back into the original sequence of characters, the opposite process, called #Decoding, would be necessary. Both processes are commonly used in data communications, networking, and storage, and especially with regard to wireless communications systems. By running these and other processes, code nowadays has the capacity to process and control many different operations within seconds, shaping and creating new horizons for social, economic, or cultural activity.

1 Andrew Butterfield and Gerard Ekembe Ngondi, *A Dictionary of Computer Science*, 7 ed., Oxford University Press, Oxford, 2016, p. 93.

2 Rob Kitchin and Martin Dodge, *Code/Space: Software and Everyday Life*, The MIT Press, Cambridge (MA), London, 2011, p. 25.

3 Alexander R. Galloway, *Protocol: How Control Exists after Decentralization*, The MIT Press, Cambridge (MA), London, 2004, p.166.

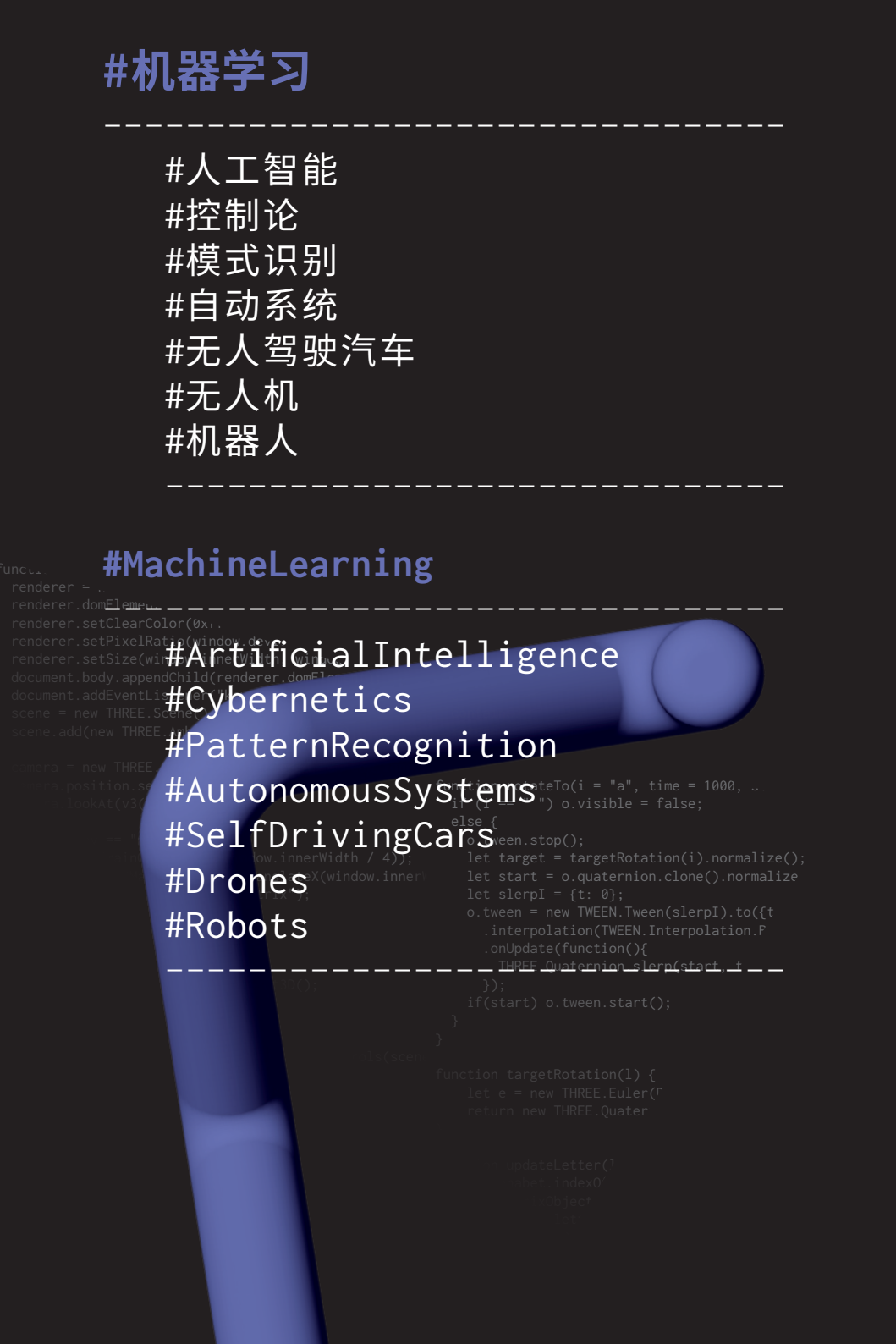
4 Susan M. Hockey, *A Dictionary of Computing*, 2 ed., Oxford University Press, Oxford, 1986, p. 352.

#机器学习

#人工智能
#控制论
#模式识别
#自动系统
#无人驾驶汽车
#无人机
#机器人

#MachineLearning

#ArtificialIntelligence
#Cybernetics
#PatternRecognition
#AutonomousSystems
#SelfDrivingCars
#Drones
#Robots



在计算机科学中，#人工智能（AI）是智能主体对机械模式或“形式化”论证的研究。AI 建立在机器对人类智能的精准模拟之上。艾伦 · 图灵的计算机理论提出由简单的符号运算，比如说 0 和 1，来表达逻辑运作的可能性。图灵假设思维可以被当作一连串独特的以因与果为基础的机械操作——用另一种说法就是一连串独特的、在一套规则下运作的逻辑步骤（#算法，参见关键词#编码）。[1] 而后得名的人工智能的典型象征方法是把机器认知视为以规则为主宰的、对形式符号进行操控的中央控制机制。这是一次把全世界的知识编成数学语言的尝试。这种方式因为所谓的专业系统获得成功，该系统可以完成复杂的任务，比如医学诊断或者专家级的计划和部署。但是，他们同时也证明了编程的高难度，一个小错误有时就会导致一整个系统的失败。更重要的是，这样的系统不能自我学习。[2] 到 1980 年间，这种方法被逐渐淘汰，因为人们意识到对于思想的模拟不等于真正的理解。所以，对于句法和符号的操控不足以实现认知。[3]

一个更加灵活的处理机器认知的方法可以从神经科学和#控制论中找到：人工智能并不被当做规则和表象来对待，而是作为动态的系统。沃伦 · S · 麦卡洛克和沃尔特 · 皮茨突破性的研究第一次把人类大脑当成计算机器。[4] 结合唐纳德 · 奥尔丁 · 赫布从神经元之间的发射和突触关系中推导而出的联想学习方法，[5] 弗兰克 · 罗森布拉特开发了机器学习的基础。[6] #机器学习是人工智能中一个探索不同计算方式的领域，它让程式能够在不手动处理算法的前提下，自主地调整并更改内在参数来处理数据。这个算法结构的构成犹如人工神经网络，其论证回路由上千个神经元所执行，排列成数百个错综复杂的互联层来处理因果关系。神经计算基于对自适应系统的建模，该系统通过捕获环境数据并将其反馈到系统中而得以进化。[7] 最重要的是，网络输出结果构成了近似值，该值是一个统计概率上最有可能的结果。

自 2006 年起，由于计算能力的稳步提高，数据捕获机制的大量扩张和 IT 基础设施的扩大，机器学习取得了巨大的进步。[8] 在其应用方面，机器学习算法在#模式识别中被广泛运用；视觉物体识别和物体检测与#自动系统息息相关，例如#无人驾驶汽车、#无人机和#机器人。从本质上讲，机器学习重新构建了一种思维方式，并在自动化决策、机器偏差、责任和问责制方面提出了许多道德和法律问题。

1 参见艾伦·图灵，“论数字计算在决断难题中的应用”，载于《伦敦数学协会会报》，第 2 版，第 42 卷，1937 年 1 月，第 230-265 页

2 参见大卫·戴文迫，“人工智能的两个（计算）面孔”，载于《人工智能哲学和理论》，编辑文森特 C 穆勒，应用哲学研究、认识论与理性伦理，第 5 卷，施普林格，海德堡，2013，第 43-58 页，此处第 44 页

3 参见约翰·罗杰斯·希尔勒，“思想、大脑和程序”，载于《行为与脑科学》，第 1 卷，第 3 章。1980 年 9 月 3 日，第 417-424 页

4 沃伦·麦卡洛克和沃尔特·皮茨，“在神经活动中内在的思想逻辑算法”，载于《数学生物物理学通报》，第 5 卷，第 4 章，1943 年 12 月，第 115-133 页

5 唐纳德·赫布，《行为组织：神经心理学理论》，威利，纽约，查普曼和霍尔，伦敦，1949 年

6 弗兰克·罗森布拉特，“感知器：大脑中信息存储和组织的概率模型”，载于《心理学评论》，第 65 卷，第 6 章，1958 年，第 386-408 页

7 参见杨·立昆，约书亚·本希奥和杰弗里·辛顿，“深度学习”，载于《自然》，第 521 卷，2015 年 5 月，第 436-444 页

8 参见杰弗里·辛顿，西蒙·欧辛德若和郑宇怀，“深度信念网的快速学习算法”，载于《神经计算》，第 19 卷，第 7 章，2006 年 7 月，第 1527-1554 页

In computer science #ArtificialIntelligence (AI) determines the operations of intelligent agents using forms of mechanical or "formal" reasoning. AI was founded on the idea that a machine can simulate human intelligence. Alan Turing's theory of computation suggested that it was possible to represent logical operations by modifying simple symbols such as 0 and 1. Turing assumed that reasoning can be formalized as distinctive sequences of mechanical operations based on cause and effect – in other words, discrete sequences of logical steps based on a set of rules (#Algorithm, see key area #Encoding).[1] What came to be known as the classical symbolic approach to AI considers machine cognition as rule-governed manipulation of formal symbols with a centralized control mechanism. It was the attempt to code knowledge about the world in formal mathematical language. This approach was successful for so-called expert systems, which were able to carry out complex tasks, such as medical diagnosis, or planning and configuration at the level of human experts. However, they proved difficult to program since one simple error sometimes caused the whole system to fail. But most importantly the systems were not able to inherently learn.[2] By 1980 the approach was no longer pursued as it became clear that a mere simulation of thought does not amount to real understanding; therefore, that syntactic manipulation of symbols does not suffice for cognition.[3]

A more flexible and adaptive approach to machine cognition came from the field of neuroscience and #Cybernetics, where artificial intelligence was not treated in terms of rules and representations but as dynamic systems. Warren S. McCulloch and Walter Pitts' ground-breaking research was the first work that treated the brain as a computational apparatus.[4] Together with Donald O. Hebb's work on associative learning deriving from the firing of nodes that
produce synaptic interrelations,[5] Frank Rosenblatt developed the foundation for machine learning.[6] **#MachineLearning** is a field of AI that explores forms of computation which allow programs to change and adjust its internal parameters automatically, that is, without hand engineering the algorithms, in order to process data. The algorithmic structure is constituted as an artificial neural network, whose reasoning is executed by thousands of neurons, arranged into hundreds of intricately interconnected layers breaking up causal relations. Neural computation is based on modelling an adaptive system that evolves through the capturing of environmental data, which is fed back into the system.[7] Crucially, the networks' output constitutes an approximation, a statistical likelihood for the most probable outcome.

Since 2006, machine learning has made huge leaps forward as a consequence of a steady increase in computational power coupled

with the vast expansion of data capturing mechanisms and the enlargement of the physical IT infrastructure.[8] In its practical application machine learning algorithms are heavily employed for #PatternRecognition; visual object recognition and object detection particularly relevant for #AutonomousSystems such as #SelfDrivingCars, #Drones, and #Robots. In essence, machine learning reconstitutes what thinking means and raises many ethical and legal questions with regard to automated decision-making, machine bias, liability, and accountability.

1 See Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," in: *Proceedings of the London Mathematical Society*, ser. 2, vol. 42, no. 1, January 1937, pp. 230–265.

2 See David Davenport, "The Two (Computational) Faces of AI," in: *Philosophy and Theory of Artificial Intelligence*, ed. Vincent C. Müller, Studies in Applied Philosophy, Epistemology and Rational Ethics vol. 5, Springer, Heidelberg, 2013, pp. 43–58, here p. 44.

3 See John R. Searle, "Minds, Brains, and Programs," in: *Behavioral and Brain Sciences*, vol. 3, no. 3, September 1980, pp. 417–424.

4 Warren S. McCulloch and Walter Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," in: *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, December 1943, pp. 115–133.

5 Donald O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York, Chapman and Hall, London, 1949.

6 Frank Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," in: *Psychological Review*, vol. 65, no. 6, 1958, pp. 386–408.

7 See Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep Learning," in: *Nature*, vol. 521, May 2015, pp. 436–444.

8 See Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A Fast Learning Algorithm for Deep Belief Nets," in: *Neural Computation*, vol. 18, no. 7, July 2006, pp. 1527–1554.

#算法治理

#大数据
#量化自我

#AlgorithmicGovernance

#BigData
#QuantifiedSelf

治理是一个过程——是社会标准、法律和行动的构成、维持以及通过政府、社会、市场经济追究责任的方式方法。从本质上讲，治理涉及社会组成和组织的实践以及监管的逻辑和语言。因此，治理也意味着一种对某人或某事行使权力的方式。[1] #算法治理通过#算法探索管理生活中正式与非正式的规则（参见关键词#编码）。算法治理是一种软实力的表现形式，它可以中断一个人的习惯并重新定位其行动的潜力。它也是一种生产力，产出下一个即将浮现的行为；一种在行为成形之前就起作用的力量。[2] 因此，算法治理提供了一种完全不同的形式来管理人类生活的各个方面，无论是社会、政治、经济还是环境。它提出了如何对算法处理进行监管和立法的内在问题。

新治理形式的基础是收集和分析数据以赋予价值。过去十年中，我们见证了实时捕捉与处理的数据量激增。我们的环境逐渐被编码（参见关键词#编码），变为机器可读、独特的索引排列，并且可以被大量连接在一起的设备和传感器所识别。日常生活越来越多地受到数字设备的影响和计算基础设施的推动。 #大数据事业致力于捕捉整个社会人口及其活动的数据。[3] 制造复杂的统计模型数据来表现、模拟和预测人类生活的目的支持着对数据收集和自我量化（#量化自我）的投入。汇集所有这些数据的关键是信息的关联方式——通过各种统计分析和#机器学习的算法处理数据排列模式（参见关键词#人工智能和#机器学习）——来检测数据之间的模式和连接，数据的关联变为知识和/或信息的源头。

因此，治理似乎变成了如何评估数据以及由谁进行评估的斗争。基本上，数据相关联性所带来的可能是集合个人和群体的档案用以确定所谓的正常行为，并区分异常行为。因此，个人变成了“个体”（dividuals），由数据组合而成的代码数字体。[4] 在这些档案的基础上，政府和企业实行他们的议程。后者采取战略来实现资本的积累，以产生可观的利润，但前者的关注点是国家安全。随着越来越具有侵略性的分析手段，企业一方面寻求通过产品的微营销来个性化客户体验。另一方面，国家根据新兴技术收集情报来预防犯罪。在这两种情况下，强大的算法结合预测分析的能力被用于预判生活中进行下一步的条件。巧妙地实行控制使得“个体”看起来好像是在自主地行动，但它已经缺乏自我意志决定的能力。

1 参见伊莎贝尔·洛雷,《不安全的国家:不稳定的政府》,维索未来,维索图书,伦敦,纽约,2015 年,第 23 页

2 参见露西安娜·帕瑞思,《传染性建筑:计算,美学和空间》麻省理工学院出版社,剑桥(MA),2013 年,第 169 页

3 参见罗伯·柯池,《数据革命:大数据,开放数据,数据基础设施及其后果》,赛吉出版公司,洛杉矶,伦敦,2014 年,第 67 页

4 参见吉利·迪鲁斯,“控制社会的后记”,于《十月》,第 59 卷,1992 年冬,第 3-7 页

Governance refers to a process of governing – the way in which norms, laws, and actions are structured, sustained, and held accountable, whether undertaken by the government, society, or the market economy. Essentially, governance involves the practice in which societies are organized, the logic or language of regulation. Hence governance also implies a way of exercising power over someone or something.[1] #AlgorithmicGovernance explores the formal and informal rules of organizing the living through #Algorithms (see key area #Encoding). Algorithmic governance refers to a form of soft power that interrupts habits and reorients action potentials. It is a producing force that generates the particular behavior that comes to the surface next; a force that acts before the behavior takes shape.[2] As such algorithmic governance offers a radically different form of managing all aspects of human life, be it socially, politically, economically, or environmentally. It raises immanent questions of how algorithmic processing should be regulated and legislated.

Underlying new forms of governance is the way in which data is gathered and analyzed in order to ascribe value. The last decade has seen an explosion in the amount of data that is being captured and processed in real time. Our environment is increasingly encoded (see key area #Encoding), rendered machine-readable, uniquely indexical, and identifiable by the vast assemblage of connected devices and sensors. Daily life is becoming more and more mediated by digital devices and facilitated by computational infrastructure. The #BigData undertaking strives at capturing society as a whole, the entirety of the population and its activities.[3] The endeavor of data collection and the quantification of the self (#QuantifiedSelf) is underpinned by the intention to produce sophisticated statistical models that characterize, simulate, and predict human life. The key to assembling all this data is the way in which information is correlated – the processing of data through various kinds of statistical analysis and #Machine-Learning algorithms – which detect patterns and connections between pieces of data. Correlations of data become sources of knowledge and/or information.

In consequence, governance seems to have turned into a struggle of *how* data is evaluated and *by whom*. Essentially, what the correlation of data allows for is the assemblage of profiles for individuals and groups of people to determine so-called normal behavior and distinguish the abnormal. Individuals are thereby turned into "dividuals," numerical bodies of code comprised of data assemblages.[4] On the basis of these profiles governments and businesses implement their agendas. Whereas the latter adopt strategies to realize capital accumulation that will produce significant profits, the concern of the former is state security.

With increasingly invasive means of profiling, companies seek on the one hand to personalize consumer behavior through micromarketing of products. On the other side stands the state which uses new technology to gather information that is supposed to prevent crime but at the same time can attempt to influence how the electorate votes through microtargeting. In both cases powerful algorithms in combination with predictive analytics are employed to conditions of life's nextness. Control is exercised subtly, making it seem as if the dividual is acting autonomously, yet it lacks the ability to make decisions of its own volition.

1 See Isabell Lorey, *States of Insecurity: Government of the Precarious*, Verso, London, New York, 2015, pp. 23ff.

2 See Luciana Parisi, *Contagious Architecture: Computation, Aesthetics, and Space*, The MIT Press, Cambridge (MA), London, 2013, pp. 169ff.

3 See Rob Kitchin, *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*, Sage Publications, Los Angeles, London, 2014, pp. 67ff.

4 Gilles Deleuze, "Postscript on the Societies of Control," in: *October*, vol. 59, Winter 1992, pp. 3–7.

#劳动与生产

#工业4.0
#物联网
#编程
#智能工厂
#自动化
#工作4.0

#Labor&Production

#Industry4.0
#InternetOfThings
#Programming
#SmartFactories
#Automation
#Work4.0

人们越来越渴望能够时刻得到为自己量身定制的产品和服务。这是数字经济带来的现象——数字经济是一个影响深远的商业模式，它涉及包括生产、服务、交通和通信在内的许多领域，并很大程度上依赖于信息技术。[1] 这个模式正在重塑整个消费品价值链的结构与经营方式，同时植入了一种新的基础设施。实现这个商业模式的关键在于产品、加工与设施之间的实时关系网络以及相关的线上客户服务。这一切都将固化的价值链转化为具有高度弹性的价值网络。

这一改变被命名为**#工业 4.0**，并被认为将促成第四场工业革命。这场革命的特点是一种可互相操作的设计理念，它让机器、设备、传感器和人类之间能够互相连接并实时通过**#物联网**（IOT）来交换重要信息。这种高透明度使产品生产流程可以基于不同变量——成本价、有效性和资源消耗——完成最优化，从而实现动态化与高效率。软件和机器都能自主运行，且即便有新的要求，也不需太复杂的 # 编程即可实现，因此每个顾客的需求都能得到迅速的响应。这个产业链上的每一个环节都“知道”自己的身份、定位和任务，并能够与生产车间进行交流。车间随即便能根据重要性与时间安排自主地规划任务。在这些模块化的**#智能工厂**中，正在运行的软件能够及时地识别错误和缺陷，并采取相应措施。[2]

工业 4.0 还是一个尚在开发的过程。若要成功实现，还需要大量全球性的标准化与统一性。各个行业的公司在境内和境外都需要寻求一种新的合作方式。智能工厂灵活的价值网络需要一种**#界面**（见关键词**#编码**）之间的协作——一种可参考的架构、一系列经过统一的定义与方法。它呼吁达成一种针对系统标准化描述和规格的通用结构语言。工业 4.0 给信息技术与数据的安全性带来了许多挑战，并将影响生产流程。同样地，它也将带来关于数据保护（企业、员工和顾客）及自动化系统责任的法律问题，这一切都需要更多的规范管理措施。[3]

然而这一新型商业模式带来的最大变化其实是劳动力的分配形式。高重复、低技术的工作正愈发受到来源于**#自动化的影响**，因为智能机器和**#机器人**（见关键词**#人工智能**、**#机器学习**）正在替代它们。从业人员应拥有更广泛的才能来做一些**#算法**（见关键词**#编码**）所不能完成的行为决策。鉴于劳动力需求不断变化的本质，职员将需要灵活可变的应变措施，从而能够不断地适应新职位、新角色的要求。这一转变被命名为**#工作 4.0**。[4] 在这种经济模式下，知识成为关键资源，

因为社会的一切都将向着创新而运作。在这种未来产业链的一端是以创意为重，变得越来越灵活弹性以满足员工个人需求，从而促进创新的工作环境。然而，只有处于社会顶端的从业者才能从这些良好且顾及个人家庭的工作环境中获得利益与财富。产业链的另一端将仍是类似硅谷加速器的软件工程开发体力劳动工厂。

1 见尼克·斯尼斯克，《平台资本主义》，理论还原系列，剑桥（英国）政治出版社

2 见联邦经济与能源部，《工业 4.0 和数字经济：刺激增长，就业与创新》，联邦经济与能源部，柏林，2015

3 见联邦教育与研究部，《未来愿景工业 4.0》，联邦教育与研究部，柏林，2015

4 见奈德·罗斯特，《软件、基础设施、劳动力：一个逻辑梦魇的媒体理论》，纽约，劳特利奇，2016，第 109 页

The desire for on demand goods and services, customized to one's personal tastes and available 24–7, is steadily increasing. It is a phenomenon of the digital economy, a business model that cuts across sectors – including manufacturing, services, transportation, and telecommunications – which is heavily reliant on information technology.[1] This model is reshaping the organization and management of the entire value chain of consumer goods and putting in place a new infrastructure. What makes this business model possible is the real-time networking of products, processes, and infrastructure, as well as related customer services via the Internet. This enables rigid value chains to be transformed into highly flexible value networks.

The approach has been termed #Industry4.0 and is deemed to constitute a fourth industrial revolution. It is characterized by its interoperable design where machines, devices, sensors, and people are connected and can exchange relevant information in real time over the #InternetOfThings (IOT). This transparency enables dynamic, efficient production processes that can be optimized on the basis of different criteria such as cost, availability, and resource consumption. Software and machines operate autonomously and do not require complicated #Programming to meet new requirements, which makes it possible to react fast to individual customer requests. Individual parts of the chain "know" what they are, where they belong, how they need to proceed, and can interact with the production plant. The plant then decides by itself what should be done in accordance with priority and time frame. In these modular structured #SmartFactories the implemented software recognizes defects or mistakes at an early stage and is able to counteract them.[2]

Industry 4.0 is as yet a developing process. To work successfully, it will require a great deal of standardization and uniformity on an international scale. New forms of cooperation between companies across sectors both nationally and globally need to be created. The smart factory's highly flexible value networks call for the harmonization of #Interfaces (see key area #Encoding); that is, a reference architecture, a set of uniform definitions and methods. It necessitates a common structure and language for standardized description and specification of systems. Industry 4.0 brings many challenges to IT and data security, which can compromise the integrity of production processes. Similarly, it raises legal issues that need regulation, concerning data protection (corporate, employee, and consumer) and liability for automated systems.[3]

However, the greatest transformation that the new business models bring with them is the way in which labor is organized. Routine and low-skill jobs are increasingly threatened by #Automation, for

they are being taken on by intelligent machines and #Robots (see key area #MachineLearning). Employees are obliged to acquire a much broader range of skills which allow them to take action and make decisions that #Algorithms (see key area #Encoding) cannot. Considering these changing dynamics of labor, employees will need to be trained and qualified for new roles, be more flexible and mobile. This transformation has been termed #Work4.0.[4]
In this economy knowledge is the key resource in which everything is geared towards innovation. The changeover from a labor-based society to a knowledge society is imminent. Fewer people will be top wage earners, fewer people will have less (routine) work to do, and fewer people will do more (highly technical and highly qualified) work. Knowledge and know-how will be the new gold, the new oil. At one end of the spectrum, the workplace increasingly adapts to more flexible and dynamic structures that cater to individual needs in order to harness creativity. Yet only the top-end workers receive these benefits as well as profit from healthy and family-friendly working arrangements. The other end of the spectrum may resemble the manual labor factories for software engineering similar to the Silicon Valley accelerators.

1 See Nick Srnicek, *Platform Capitalism*, Theory Redux series, Polity Press, Cambridge (UK), 2017, pp. 4–5.

2 See Bundesministerium für Wirtschaft und Energie, *Industrie 4.0 und Digitale Wirtschaft: Impulse für Wachstum, Beschäftigung und Innovation*, Bundesministerium für Wirtschaft und Energie, Berlin, 2015.

3 See Bundesministerium für Bildung und Forschung, *Zukunftsbild Industrie 4.0*, Bundesministerium für Bildung und Forschung, Berlin, 2015.

4 See Ned Rossiter, *Software, Infrastructure, Labor: A Media Theory of Logistical Nightmares*, Routledge, New York, 2016, p. 109.

#算法经济

#高频交易
#比特币
#加密货币
#解密
#区块链

#AlgorithmicEconomy

#HighFrequencyTrading
#Bitcoin
#Cryptocurrencies
#Decrypt
#Blockchain

在一切都变得数字化的世界里（我们的沟通方式、广告、休闲和工作场所），以数码形式生产资金只是一个时间上的问题。几十年来，银行和市场一直使用计算机算法运营，许多他们的客户也已经能够数字化地管理资金。然而，现在的问题是——#算法经济——更为广泛和复杂。代码的执行对我们的全球化经济有哪些影响?哪些系统已经出现——或将在未来出现?

在谈论经济学、数学和计算机科学的结合时，首先出现的概念之一就是算法交易，这是投资银行和养老基金广泛使用的一种方式。它利用自动预期编程的指令在金融市场做出决策和执行交易。这意味着当今#**算法**（见关键词#**编程**）推动了大量的股票交易。许多这些自动化交易的系统都属于#**高频交易**（HFT）的类别，其特点是速度快，以至于人类永远无法在同一时间内执行它们，甚至无法接近这样的速度。

作为这种霸权体系及其金融化的替代品，一种名为#**比特币**的新兴货币于 2009 年在线上发布。其后，许多其他数字货币发布，如以太币或莱特币。但是什么使这些#**加密货币**与传统货币不同呢?顾名思义，它们基于加密系统 [1]，这意味着它们背后的代码制定了一个信息保密的系统。只有知道如何#**解密**的人——或者更确切地说，程序——才能访问这些信息。加密货币是非物质、去中央性的。不同于传统的银行业务中政府可以通过印钞机控制货币价值，政府无法控制加密货币：它们的价值在互联网传播中，没有任何中介机构的参与。

要理解这些相关性，我们必须关注到加密货币背后的系统 # 区块链。区块链是一个开放的数据库，在加密货币中，它存储了金融交易的历史数据。单个区块包含各种交易，每个交易都连接着链中的上一个记录。当有人用比特币购买东西时，以加密拼图为形式的请求会被比特币点对点网络中的所有计算机——称为矿工——所发送和接收。当一个矿工解决一个加密拼图时，一个新的区块会被添加到链中，矿工会因此获得一些比特币奖励。但赚取比特币并不是“采矿”的唯一目的：加密拼图非常复杂，以至于每个新区块都会增加之前的区块和整个链条的安全性。黑客破解区块链需要极快的速度来改变仅仅一个交易。随着许多矿工不断地添加区块，则需要大量的计算能力破解区块链。

与数字时代产生的其他颠覆性技术一样，加密货币正在挑战迄今为止经济学的工作方式，预见了一个中间商将会成为过去的未来。在

由区块链技术主导的世界中，用于商业战略和管理交易的新工具将被开发而出，“金钱和信息的控制权将从精英阶层 [...] 转移到它所属于的人手中。”[2] 有些人认为这些模式将扰乱中央式的经济和政治建构，其他人则表示，它们将严重影响我们的就业市场，只会使劳动力市场中的高层人士受益。未来不可能预测，但要了解我们生活的世界和我们正在建设的经济，我们必须首先认识和分析算法和计算的力量。

1　术语（Cryptography）密码学源自希腊语 κρυπτ ，kryptós, 意为“隐藏”或“秘密”，而 γρ φειν, graphein, 希腊语为“写作”。参见亨利·乔治·李雕和罗伯特·斯科特，《希腊英语词典》，牛津大学出版社，1984 年。

2　保罗·费格纳和麦克·J·凯西，《加密货币时代：比特币和区块链如何挑战全球经济秩序》，圣马丁出版社，纽约，2015 年，第 6 页。

In a world where everything is becoming digital (our way of communication, our advertising, our leisure and workplaces), it was only a matter of time before money could be generated in a digital way. Banks and markets have been operating for decades using computerized algorithms and many customers have had digital access to their money for some time now. However, the matter at hand – #AlgorithmicEconomy – is more extensive and complex. Which impacts has the implementation of code had in our globalized economy? Which systems have appeared – or will appear in the future?

One of the first concepts that emerges when talking about the combination of economics, mathematics, and computer science is algorithmic trading, a practice widely used by investment banks and pension funds that utilize automated preprogrammed instructions to make decisions and execute transactions in the financial markets. This means that nowadays #Algorithms (see key area #Encoding) drive a great number of stock trades. Many systems of these automated activities fall into the category of #HighFrequencyTrading (HFT), which is characterized by such high speeds that a human could never carry them out in the same time nor even close to it.

As an alternative to this hegemonic system and its financialization, a new digital currency called #Bitcoin was released online in 2009, followed by many other digital cash currencies, such as Ethereum or Litecoin. But what makes #Cryptocurrencies different from traditional currencies? As its name implies, they are based on a cryptographic system,[1] which means that the code behind them is elaborated on a system that keeps information secret. Only the people – or more precisely, the programs – that know how to solve it, how to #Decrypt it, will have access to this information. Cryptocurrencies are also immaterial and decentralized. Unlike centralized banking, where governments control the currency values through the process of printing money, governments have no control over cryptocurrencies: their value circulates on the Internet without the regulating involvement of any intermediaries.

To understand the correlations, one has to look at the #Blockchain, the system behind cryptocurrencies. Blockchain is an open database that, in this case, stores a history of financial transactions. Single blocks contain various transactions, each of which is linked to a previous record in the chain. When someone purchases something with Bitcoins, a request in the form of a cryptographic puzzle is sent to and received by all the computers – known as miners – on the Bitcoin peer-to-peer network. When a miner solves a puzzle, a new block is added to the chain and it is rewarded with some Bitcoins. But earning Bitcoins is not the

only point of mining: the puzzles are so complex that every new block makes the previous ones and the whole chain a safer environment. Hacking the block-chain would require immense speed to alter just one transaction. With many miners adding blocks continuously, a vast amount of computing power would be needed.

Like other disruptive technologies born in the digital age, cryptocurrencies are challenging the way things have been done in economics so far, foreseeing a future in which middlemen would become obsolete. In a world run by blockchain technologies, new tools for business strategies and managing transfers would be developed, shifting "the control of money and information away from the powerful elites [...] to the people to whom it belongs."[2] While many people argue that these models will disrupt the centralized economic and political establishments, others say that they will severely impact our job market and only benefit those in the upper echelons of the workforce. It is not possible to predict the future, but to understand the world we live in and the economy we are building, we necessarily need to recognize and analyze the power of algorithms and computation.

1 The term cryptography derives from Greek κρυπτός, *kryptós,* which means "hidden" or "secret," and γράφειν, *graphein*, Greek for "writing." See Henry George Liddell and Robert Scott, *A Greek–English Lexicon*, Oxford University Press, 1984.

2 Paul Vigna and Michael J. Casey, *The Age of Cryptocurrency: How Bitcoin and Blockchain are Challenging the Global Economic Order*, St. Martin's Press, New York, 2015, p. 6.

#虚拟现实

#头戴式显示器
#计算机模拟环境
#增强现实
#计算机生成设计
#逃避主义

#VirtualReality

#HMD
#ComputerSimulatedEnvironments
#AugmentedReality
#ComputerGeneratedDesign
#Escapism

要了解现今语境下的虚拟及虚拟现实（#虚拟现实），我们要研究这些术语如今是如何被频繁使用的。虚拟现实（VR）被理解成一个技术术语，一种为观众仿制某个环境体验的媒介——包括一种存在在此处的感觉，这种感觉在其他视觉刺激被遮挡后尤为明显（比如用#头戴式显示器）。想要认识虚拟在我们视界之中的影响，要先分析这个词来源及定义。英语中虚拟（virtual）作为形容词的派生是力量（strength）、美德（virtue）这两个词。一个虚拟的存在拥有不基于物质而行动的能力。自从计算机领域开始使用这个词语，大多数字典都添加了其与#计算机模拟环境相关的语义——被模拟于计算机或计算网络中，或存在于虚拟现实中。[1]

因此我们可以认为“虚拟是一种替代品——‘不基于物质中介的行动’——和一种物质本身的无形代理。这个词语标志了一种真实与模仿、原创与复制、图片与它的相似品之间的关系演变。”[2]

哲学家亨利·柏格森、吉尔·德勒兹、菲利克斯·加塔利和皮埃尔·利维拓展了许多关于虚拟的哲学概念。柏格森将记忆的无形称为虚拟。[3] 德勒兹认为虚拟不是“现实”的对立面，而是“实际”的对立面——在这样的理解之下，“虚拟”成为了现实的一种展现形式。[4] 加塔利将虚拟称为“四个本体论函子”[5]——虚拟、实际、现实和可能。

虚拟现实是一个相对较新的术语，最初可能是由安托南·阿尔托在他出版于 1938 年的法文书《剧院及其复像》[6] 中创造的。如今大众理解的虚拟现实与阿尔托的用法并不相同。在过去的几十年中，它的含义经过了演变，现在它与#增强现实（通常通过图像呈现与附加设备（如头戴显示器）实现）多用来表示计算机辅助的互动及沉浸环境。

艺术家和工程师随即展开针对这项技术的实验，并促进开发了#计算机生成设计。特别是 1990 年代出现了大量关于虚拟现实的应用与艺术实验，许多艺术作品亦随之诞生。虽然当时的技术由于不够成熟无法被广泛应用，但近年随着相关硬件与软件的发展，更多艺术家开始以虚拟现实作为一种媒介来创作。

这个媒介提供了视觉上的完全沉浸。它不仅仅像图像画面那样为观者打开了一扇窗户——如莱昂·巴蒂斯塔·阿尔伯蒂在他的论文《论绘画》中描述道，它还将观众拉进画面里，并打开了一个折射现实空间的世界。

除艺术领域外，虚拟现实也在游戏产业和其他娱乐产业内（如虚

拟影院）被广泛应用。它在医学界也早已被普及：虚拟模型能帮助医生迅速有效地定位肿瘤并判断手术切口位置。心理学家与其他医学专家也正在用虚拟现实改善传统治疗方法，寻找对于创伤后应激障碍（PTSD）、焦虑症和社交障碍症的有效治疗方案。房地产企业和建筑师也能为潜在租客或房屋承包商带来尚未建成楼房的虚拟探访。

虚拟现实技术正变得无处不在。不仅是因为这种媒介可以提供完美的#**逃避主义**，还因为它具有强大的应用潜力及商业价值。

1 见《韦伯斯特第三部新国际词典》，完整版，梅里亚姆·韦伯斯特，1993 年

2 见安妮·弗里德伯格，《虚拟之窗：从巴蒂森到微软》，麻省理工学院出版社，剑桥（马萨诸塞州），伦敦，2006，第 8 页

3 见亨利·柏格森，《事物及记忆：一篇关于身心关系的论文》，埃里克·奥格尔、朱利叶斯·弗兰肯伯格、费利克斯·梅纳撰前言，汉堡，1991，第 127 页

4 由曼纽尔·德兰达提供详细德勒兹称之为反实现的过程（从实际到虚拟）；见曼纽尔·德兰达，《强化的科学与虚拟现实》，Continuum，伦敦，2002

5 菲利克斯·加塔利，《Chaosmosis：一种伦理审美范式》，翻译版，保罗·班恩
和朱丽安·帕法尼斯，印第安纳大学出版社，布鲁明顿，1995

6 见托南·阿尔托，《剧院及其复像》，翻译版，玛丽 R·里查德，格罗夫出版社
纽约，1958，第 49 页

To understand the significance of *virtual* and #VirtualReality in the present context let us take a closer look at the rise of current usages of these terms. Virtual reality (VR) is understood as a technical term, as a medium that reproduces spatial experiences for its viewers – experiences of and in spaces that do not physically exist and cannot be explored by touch, for example, especially when other visual stimuli are blocked out (for example, by head-mounted displays, #HMD). In art, since ca. 1920, bodies that only appear to exist are referred to as virtual (e.g., Naum Gabo, Konstruktion, 1921). A rotating wire driven by an electric motor, for example, produces what looks like a three-dimensional figure on a disk. In the 1920s, experimental psychology and gestalt theory investigated this phenomenon in depth, for example, the stereokinetic effect. Kinetics and Op Art are its products. Since the term has been used in the context of computer technology the meaning relating to #ComputerSimulatedEnvironments has been added to most dictionary definitions as follows: simulated on a computer or computer network, or existing within a virtual reality.[1]

Thus we can conclude that "the virtual is a substitute – 'acting without agency of matter' – an immaterial proxy for the material. The term becomes a key marker of a secondary order in the relationship between the real and its copy, the original and its reproduction, the image and its likeness."[2]

In philosophy Henri Bergson, Gilles Deleuze, Félix Guattari, and Pierre Lévy all developed various concepts of the virtual. Bergson describes the immateriality of memory as virtual.[3] For Deleuze virtual is not opposed to real, but to actual – in this understanding virtual is a mode of reality.[4] Guattari describes virtual as one of "four ontological functors"[5] – the virtual, the actual, the real, and the possible.

The term "virtual reality" is relatively recent and was probably coined by Antonin Artaud in his book *The Theatre and Its Double*, first published in French in 1938.[6] Our current understanding of VR does not coincide with Artaud's usage of the term; the meaning has shifted over the last decades, and now the term is predominantly used for computer-aided interactive and immersive environments, together with #AugmentedReality, that are accessed via screened images and in many cases additional devices (such as HMDs).

Artists and engineers began to experiment with the medium in the 1980s (Myron W. Krueger, *Artificial Reality*, 1983) and contributed to its development of #ComputerGeneratedDesign. Especially in the 1990s applications and artistic experiments using VR proliferated and resulted in artworks. Although at that time the technology was not sufficiently developed for wider usage, with the wider availability of the hardware and various software for it, in the last few years more and more artists have started to work

with VR as a medium.
The medium offers complete visual immersion; it not only opens a window, as framed images do, as Leon Battista Alberti claims in his treatise *On Painting* (1435). The Art of Immersion, however, actually pulls the observer into the image and not only opens a window as painting and framed art works do. VR is a gateway through which viewers in the real world enter and leave the virtual world. VR literally opens a door into another reality.

In the gaming industry and in medicine the technology is already widespread. Virtual models help surgeons, for example, to identify the safest and most efficient way to locate tumors and place surgical incisions. Psychologists and other medical professionals are using VR to enhance traditional therapy methods and find effective solutions for treatment of posttraumatic stress disorder (PTSD), anxiety, and social disorders. Real estate businesses and architects accompany their possible tenants or building contractors on walk-throughs of as yet nonexistent buildings.

VR technologies are becoming ubiquitous, not only because of the supreme #Escapism the medium offers, but also because of its practical and commercial potentials.

1 See *Webster's Third New International Dictionary*, unabridged edition, Merriam Webster, 1993.

2 Anne Friedberg, *The Virtual Window: From Alberti to Microsoft*, The MIT Press, Cambridge (MA), London, 2006. p. 8.

3 See Henri Bergson, *Materie und Gedächtnis. Eine Abhandlung über die Beziehung zwischen Körper und Geist*, introduction by Erik Oger, trans. Julius Frankenberger, Verlag Felix Meiner, Hamburg, 1991, p. 127.

4 Manuel DeLanda provides a comprehensive explanation of the process that Deleuze calls counter-actualization (moving from the actual to the virtual); see Manuel DeLanda, *Intensive Science and Virtual Philosophy*, Continuum, London, 2002.

5 Felix Guattari, *Chaosmosis: An Ethico-aesthetic Paradigm*, trans. Paul Bains and Julian Pefanis, Indiana University Press, Bloomington, 1995.

6 See Antonin Artaud, *The Theater and Its Double*, trans. Mary C. Richards, Grove Press, New York, 1958, p. 49.

#遗传密码

#DNA
#源代码
#生物工程
#表现型
#DNA数据储存
#基因型

#GeneticCode

#DNA
#SourceCode
#Bioengineering
#Phenotype
#DNADataStorage
#Genotype

人们已知#DNA（脱氧核糖核酸）包含#源代码。#遗传密码是编写在遗传物质（DNA 或 mRNA 序列）中的信息被活细胞转录和翻译成蛋白质时要遵守的一系列规则。

关于遗传密码的最初描述始于 1950 年代。在 1960 年代末，人们对遗传的一些认识已逐渐清晰——形成双螺旋的大分子是 DNA 中的遗传信息，它由腺嘌呤（A），鸟嘌呤 (G)，胞嘧啶 (C)，胸腺嘧 (T) 四种碱组成。那时，分子生物学的中心信条认为 DNA 包含了能够组建蛋白质并在结构上催化生命“执行”的密码。[1]

在 1940 和 1950 年代极有影响力的#控制论（参见关键词#人工智能 和#机器学习）和信息论很大程度上塑造了分子生物学中用到的隐喻和术语，也正是在这个年代，关于遗传学的研究开始羽翼渐丰。[2]

DNA 和 RNA 被称作“信息分子”或“存储带”，它们遵循信息处理中的原则。[3] 遗传密码也时常被人们拿来与计算机程序相比较，比如“人体器官，细胞和分子由一个通信网络联合起来。”[4]

分子生物学的首要命题是解开生物“生命之书”的密码，研究人员竞相朝着这个目标努力。人类基因组计划（HGP,1990-2003）是一个以探索人类#基因型[5] 为目标的全球性科学事业。基因型是组成人类 DNA 的核苷酸碱基对的序列。

分子生物学的发展促进了新领域的工程化。例如，#生物工程“是对有机体进行操控以生成非天然分子（比如药品或蛋白质）。”[6] 1970 年代，斯坦利 · 诺曼 · 科恩与赫伯特 · 伯耶将人类 DNA 嵌入细菌用来生成可以治疗糖尿病的重组胰岛素，这一实验也让他们最早发明了现已成为行业核心的 DNA 重组技术。分子生物领域的最新发展是基因组编辑技术，CRISPR/Cas9 目前得到了最高的关注度。基因组编辑让研究人员得以修改包括人类基因组在内的任何基因组，并提供了广泛的应用可能性。[7] 和基因工程技术一样，这种技术也引起了许多道德伦理的问题。

最新的研究发现 DNA 分子可以储存任何的信息（#DNA 数据储存）。文字、视觉信息和动态画面都可以被转化为二进制数据从而被记录到遗传密码里。[8] 这意味着研究人员能够以解码的角度去编码——比如存储于细菌 DNA 中的埃德沃德 · 迈布里奇的《人类和动物的运动》组图。

1 见阿德里安·麦肯齐和西奥·韦德巴奇斯,“危机中的代码和编码:意义,表现和过剩”,于《理论,文化和社会》,第 28 卷,第 6,2011,第 3-23 页,此处第 7 页

2 见丽莉 E·凯,《谁书写了“生命之书”:遗传密码史》,斯坦福大学出版社,斯坦福,2000,第 73-127 页(第三章:话语生产:控制论,信息和生命)

3 见卡尔 R·伍兹,《遗传密码:基因表达的分子基础》,哈珀与罗,纽约,1967,第 253-54 页

4 弗朗索瓦·雅各布《生命逻辑:一部遗传史》[1970],翻译版,贝蒂 E·斯皮尔曼,万神殿图书,纽约,1973

5 # 基因型定义有机体内的基因,而 # 表现型描述它的生理外表。

6 布兰登·阿德金斯,《生物工程指南》,Kindle 电子书,亚马逊发布,莱比锡,2016,第 5 页

7 见亚历克斯·瑞斯,“CRISPR/Cas9 与基因组目标编辑:分子生物学的新时代,”于 NEB,第 1 期,2014,线上来源:https://www.neb.com/tools-and-resources/feature-articles/crispr-cas9-and-targeted-genome-editing-a-new-era-in-molecular-biology, accessed 09/13/2017.

8 见安迪·埃克斯坦斯,“DNA 如何储存世界上所有的数据?”于《自然》,第 537 卷,第 7618 期,9 月 1 号,2016,第 22-24 页

#DNA (deoxyribonucleic acid) is known to contain the #SourceCode. #GeneticCode is the set of rules by which information encoded within genetic material (DNA or mRNA sequences) is translated into proteins by living cells.

The description of genetic code began in the 1950s. By 1953 it was clear that the genetic information in DNA, a macromolecule forming a double helix (James Watson, Francis Crick), is made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). At this time the central dogma of molecular biology became that DNA contains the code for the construction of proteins that catalytically and structurally "execute" life.[1]

The metaphors and phrasing used in molecular biology were strongly influenced by #Cybernetics (see key area **#Machine-Learning**) and information theory that became influential in the late 1940s and 1950s, exactly when genetics started to spread its wings.[2]

DNA and RNA were called "informational molecules" or "tapes" governed by the rules of information processing.[3] Genetic code was also compared to a computer program, for example, "organs, cells and molecules are united by a communication network."[4]

To decipher the code of the biological "Book of Life," was a central issue in molecular biology, and researchers were racing to crack it. The Human Genome Project (HGP, 1990–2003) was an international scientific endeavor with the goal of determining human #Genotype[5], the sequence of nucleotide base pairs that make up human DNA. The private research project of Craig Venter (Celera Corporation) has worked with automated DNA sequencing since 1998.

The developments in molecular biology facilitated new fields of engineering. #Bioengineering "is the manipulation of an organism to produce non-native molecules (such as drugs or proteins)."[6] Recombinant DNA technology, a method originally invented by Stanley N. Cohen and Herbert Boyer in the 1970s to insert human DNA into bacteria to produce a recombinant version of insulin for the treatment of diabetes, is the key for this discipline. The latest developments in the field are genome editing methods; CRISPR/Cas9 recently got the most publicity. Genome editing allows researchers to modify any genomes, including human, with wide application possibilities.[7] The method, just like genetic engineering in general, raises ethical questions.

It has been recently discovered that DNA molecules can store any data (#DNADataStorage). Textual and visual information, even moving images, can be converted to binary then to genetic code[8], which has allowed researchers to encode in a decodable way, for example, a series of frames from Eadweard Muybridge's *Human and Animal Locomotion* in bacterial DNA.

1 Adrian Mackenzie and Theo Vurdubakis, "Codes and Codings in Crisis: Signification, Performativity and Excess," in: *Theory, Culture & Society*, vol. 28, no. 6, 2011, pp. 3–23, here p. 7.

2 See Lily E. Kay, *Who Wrote the Book of Life: A History of Genetic Code*, Stanford University Press, Stanford, 2000, esp. pp. 73–127.

3 See Carl R. Woese, *The Genetic Code: The Molecular Basis for Genetic Expression*, Harper & Row, New York, 1967, pp. 253–254.

4 François Jacob, *The Logic of Life: A History of Heredity* [1970], trans. Betty E. Spillmann, Pantheon Books, New York, 1973.

5 #Genotype defines the genes within the organism, while #Phenotype describes its physical appearance.

6 Brandon Adkins, *A Future Guide to Bioengineering*, Kindle e-book, Amazon Distribution, Leipzig, 2016, p. 5.

7 See Alex Reis, "CRISPR/Cas9 and Targeted Genome Editing: A New Era in Molecular Biology," in: *NEB* expressions, no. 1, 2014, available online at: https://www.neb.com/tools-and-resources/feature-articles/crispr-cas9-and-targeted-genome-editing-a-new-era-in-molecular-biology, accessed 09/13/2017.

8 See Andy Extance, "How DNA Could Store All the World's Data?," in:
Nature, vol. 537, no. 7618, September 1, 2016, pp. 22–24.

开放代码

“开放代码.连结机器人”
参展作品代码

Open Codes

Works in the Exhibition
Open Codes. Connected Bots

A

aaajiao
*1984年出生于西安(中国),现于上海(中国)和柏林(德国)居住和工作
观察者
2017—18, 单频彩色录像、网站,15'32"
#机器学习
#虚拟现实
#算法 #逃避主义

B

扎克·布拉斯,杰迈玛·怀曼
*1981年出生于加利波利斯(美国),现于伦敦(英国)居住和工作
*1977年出生于悉尼(澳大利亚),现于澳大利亚和洛杉矶(美国)居住并工作
我是来学习的所以:)))))
2017,四频道视频装置
#机器学习
#算法 #人工智能 #模式识别 #大数据

---

bleeptrack(由哈利·约瑟芬·吉尔斯提供协助)
*1992年出生于布赫(德国),现于布赫(德国)居住和工作
字形语言
2019,单频投影交互装置
#编码
#软件 #解码

詹姆斯·布莱德
*1980年出生于伦敦(英国),现于雅典(希腊)居住和工作
前市民
2015,基于网站的交互装置,笔记本电脑、旗帜、显示器
#算法治理
#量化自我

C

马克斯·库珀,安迪·洛马斯
*1980年出生于贝尔法斯特(北爱尔兰),现于伦敦(英国)居住和工作
1967年出生于韦林花园城(英国),现于伦敦(英国)居住和工作
染色体
2017,HTC Vive头显、虚幻引擎软件
#虚拟现实
#遗传密码
#头戴式显示器 #DNA #计算机模拟环境

凯特·克劳福德,弗拉丹·乔拉
*1976年出生于澳大利亚,现于纽约(美国)居住和工作
*1977年出生于南斯拉夫,现于诺维萨德(塞尔维亚)居住和工作
人工智能系统解剖
2018,数字印刷
#劳动与生产
#机器学习
#自动化 #工作4.0 #人工智能 #机器人

D

DISNOVATION.ORG
尼古拉斯·马格瑞特,玛丽亚·罗斯可斯克
*1980年出生于隆斯勒索涅(法国),现于巴黎(法国)居住和工作
*1982年出生于华沙(波兰),现于巴黎(法国)居住和工作
可预测艺术机器人
2017,双频投影、在线机器人
#机器学习
#自动系统

E

乔纳斯·埃尔特斯/Fabrica
*1993年出生于孔斯巴卡(瑞典),现于雷维索(意大利)居住和工作
迷失在计算中
2017,装置,2个屏幕、2个树莓派计算机
由Fabrica友情提供
#机器学习
#编码
#模式识别 #自动系统

塞萨尔·埃斯库德罗·安达卢兹,马丁·纳达尔
*1983年出生于阿维拉(西班牙),现于林茨(奥地利)居住和工作
*1978年出生于马德里(西班牙),现于林茨(奥地利)居住和工作

苦特币
2016，装置，计算器
#算法经济
#比特币 #密码货币 #区块链

G

郭城
*1988年出生于北京（中国），现于上海（中国）居住和工作
网络游者——握手之旅
2019，交互装置，雕刻机、网络机柜、显示器
#算法治理
#虚拟现实
#逃避主义

L

贝恩德·林特曼
*1967年出生于杜塞尔多夫（德国），现于卡尔斯鲁厄（德国）居住和工作
你：是：代码
2017，多频道投影互动装置
#编码
#遗传密码
#解码 #软件 #硬件 #界面 #源代码 #量化自我
构想及实现：贝恩德·林特曼
音频设计：卢德格·布鲁尔、杨尼克·呼夫曼
点阵磁翻显示：克里斯汀·洛克斯
技术支持：曼夫尔德·哈弗曼、简·格瑞克
制作与计划：托马斯·诗瓦布
灵感来源：彼得·威贝尔
制作：ZKM | 赫兹实验室

M

肖恩·马克西莫
*1975年出生于多伦多（加拿大），现于纽约（美国）居住和工作
开门
2017，数字印刷
#劳动与生产
#工作4.0 #自动化 #编程 #算法 #软件

乔安娜·莫尔
*现于巴塞罗那（西班牙）和柏林（德国）居住和工作
许可的算法
2017，基于研究结果的线上存储库
#编码
#算法治理
#算法 #大数据

S

塞巴斯蒂安·施米格
*1983年出生于柏林（德国），现于柏林（德国）居住和工作
分割.网络
2016，以网络浏览器为基础的分割显示
#劳动与生产
#机器学习
#自动化 #工作4.0 #人工智能

亚当·斯沃维克，克里斯汀·洛克斯，彼得·威贝尔
*1980年出生于斯凯尔涅维采（波兰），现于柏林（德国）居住和工作
*1990年出生于白原市（美国），现于卡尔斯鲁厄（德国）居住和学习
*1944年出生于敖德萨（乌克兰），现于卡尔斯鲁厄（德国）居住和工作
字母空间
2017，基于计算机的装置
#编码
#代码谱系
#计算机模拟环境 #软件 #硬件 #界面

T

奈·汤普森
*1966年出生于布里斯托尔（英国），现于伦敦（英国）居住和工作
那些重塑世界的词语
2018，装置，印刷、屏幕
#机器学习
#人工智能 #大数据

W

王长存
*1981年出生于中国，现于杭州/上海（中国）居住和工作
「好吧」
2018，软件、计算机、屏幕
#编码
#算法 #软件

彼得·威贝尔，克里斯汀·洛克斯
世界是一个数据场域
2018，数据装置
#编码
#算法治理
#软件 #硬件 #大数据

Z

ZKM | 赫兹实验室
数字代码的谱系
2017/19，装置
#代码谱系
#编码
#计算 #算法 #软件 #硬件

无作者
996.ICU
2019，线上宣言
#劳动与生产
#工作4.0

A

aaajiao
*1984 in Xi'an (CN), lives and works in Shanghai (CN) and Berlin (DE).
bot,
2017-18, single channel video, color, website, 15'32"
#MachineLearning
#VirtualReality
#Algorithm #Escapism

B

Zach Blas, Jemima Wyman
*1981 in Gallipolis (US), lives and works in London (UK)
*1977 in Disney (AU), lives and works in Australia and Los Angeles (US)
im here to learn so :)))))))
2017, 4-channel video installation
#MachineLearning
#Algorithm
#ArtificialIntelligence
#PatternRecognition #BigData

---

bleeptrack
(with the contribution of Harry Josephine Giles)
*1992 in Buch (DE), lives and works in Buch (DE)
Glyphsprache
2019, shared database, website, online, printed guide
#Encoding
#Software #Decoding

James Bridle
*1980 in London (GB), lives and works in Athens (GR)
Citizen Ex
2015, interactive web-based installation with a laptop, flag, monitor
#AlgorithmicGovernance
#QuantifiedSelf

C

Max Cooper, Andy Lomas
*1980 in Belfast (GB), lives and works in London (GB)
*1967 in Welwyn Garden City (GB), lives and works in London (GB)
Chromos
2017, head-mounted display, Unreal Engine software
#VirtualReality
#GeneticCode
#HMD #DNA
#ComputerSimulatedEnvironments

Kate Crawford, Vladan Joler
*1976 in (AU), lives and works in New York (US)
*1977 in Yugoslavia, lives and works in Novi Sad (RS)
Anatomy of an AI System
2018, digital prints
#Labour&Production
#MachineLearning
#Automation #Work4.0 #ArtificialIntelligence #Robots

D

DISNOVATION.ORG
Nicolas Maigret, Maria Roszkowska
*1980 in Lons-Le-Saunier (FR), lives and works in Paris (FR)
*1982 in Warsaw (PL), lives and works in Paris (FR)
Predictive Art Bot
2017, two-channel projection, online bot
#MachineLearning
#AutonomousSystems

E

Jonas Eltes/Fabrica
*1993 in Kungsbacka (SE), lives and works in Treviso (IT)
Lost in Computation
2017, installation of 2 screens, 2 computers
Courtesy of Fabrica, Catena di Villorba
#MachineLearning

#Encoding
#PatternRecognition
#AutonomousSystems

César Escudero Andaluz, Martín Nadal
*1983 in Ávila (ES), lives and works in Linz (AT)
*1978 in Madrid (ES), lives and works in Linz (AT)
Bittercoin
2016, installation with a calculator
#AlgorithmicEconomy
#Bitcoin #Cryptocurrencies #Blockchain

G

GUO Cheng
*1988 in Beijing (CN), lives and works in Shanghai (CN)
The Net Wanderer. A Tour of Suspended Handshakes
2019, interactive installation with engraving machine, server rack and monitor

#AlgorithmicGovernance
#VirtualReality
#Escapism

L

Bernd Lintermann
*1967 in Düsseldorf (DE), lives and works in Karlsruhe (DE)
YOU:R:CODE
2017, interactive installation with multi-channel projection; idea: Peter Weibel; concept and realization: Bernd LIntermann; audiodesign: Ludger Brümmer, Yannick Hofmann; technical support: Mnfred Hauffen, Jan Gerigk; production of the ZKM_Hertz-Lab
#Encoding
#GeneticCode
#Decoding #Software #Hardware #Interface #SourceCode #QuantifiedSelf

M

Shawn Maximo
*1975 in Toronto (CA), lives and works in New York (US)
Open Doors
2017, digital print
#Labor&Production
#Automation #Work4.0 #Programming #Algorithm #Softwaree

Joana Moll
*Lives and works in Barcelona (ES) and Berlin (DE)
ALGORITHMS ALLOWED
2017, online repository of research results on a website
#Encoding
#AlgorithmicGovernance
#Algorithm #BigData

S

Sebastian Schmieg
*1983 in Berlin (DE), lives and works there
Segmentation.Network
2016, browser-based visualization of manually created segmentations
#Labour&Production
#MachineLearning
#Automation #Work4.0 #ArtificialIntelligence

Adam Slowik, Christian Lölkes, Peter Weibel
*1980 in Skierniewice (PL), lives and works in Berlin (DE)
*1990 in White Plains (US), lives and studies in Karlsruhe (DE)
*1944 in Odessa (UA), lives and works in Karlsruhe (DE)
Alphabet Space
2017, computer-based installation
#Encoding
#GenealogyOfCode
#ComputerGeneratedDesign #Software #Hardware #Interface

T

Nye Thompson
*1966 in Bristol (GB), lives and works in London (GB)
Words that Remake the World
2018, installation of a print and a screen
#MachineLearning
#ArtificialIntelligence #BigData

W

WANG Changcun
*1980 in China, lives and works in Hangzhou/Shanghai (CN)
well then
2018, software, computer and screen
#Encoding
#Algorithm #Software

Peter Weibel, Christian Lölkes
The World as a Field of Data
2018, data installation
#Encoding
#AlgorithmicGovernance
#Software #Hardware #BigData

Z

ZKM | Hertz-Lab
Genealogy of the Digital Code
2017/19, installation
#GenealogyofCode
#Encoding
#Computing #Algorithm #Software #Hardware

Without author
996.ICU
2019, online manifesto
#Labour&Production
#Work4.0

画册

主编： 毕昕、丽维亚·诺拉斯克·洛萨斯
编辑： 曹佳敏、郝雨
作者： 布兰卡·希梅内斯、亚斯敏·克斯汀特佩、丽维亚·诺拉斯克·洛萨斯、彼得·威贝尔
校对： 凯西·莫罗
翻译： 毕昕、曹佳敏、郝雨
中文版设计： 陈韵竹、大气设计事务所

© 文字版权归原作者所有
除特殊声明外，作品版权归艺术家所有

Editors: BI Xin, Lívia Nolasco-Rózsás
Editorial staff: CAO Jiamin, HAO Yu
Authors: Blanca Giménez, Yasemin Keskintepe, Lívia Nolasco-Rózsás, Peter Weibel
Copy editing: Kathy Morrow
Translations: BI Xin, CAO Jiamin, HAO Yu
Chinese version design: CHEN Yunzhu, Atmosphere Office

© of the texts: the authors
Unless otherwise noted, the artworks are in the posession of the artists.

展览

概念： 彼得·威贝尔
策展人： 克里斯汀·洛克斯、丽维亚·诺拉斯克·洛萨斯、张尕
场景与室内设计： 策展人和Vitra团队
展览平面设计： 彼得·威贝尔、克里斯汀·洛克斯、大气设计事务所
项目总监： 毕昕
公共项目策划： 毕昕、曹佳敏
技术项目管理： 曹达旭
传播与媒体： 郝雨

卡尔斯鲁厄艺术与媒体中心（ZKM）和新时线艺术中心（CAC）感谢参与展览的艺术家和合作伙伴。

Concept: Peter Weibel
Curators: Christian Lölkes, Lívia Nolasco-Rózsás, ZHANG Ga
Scenography and interior design: the curators and Vitra team
Exhibition graphic design: Peter Weibel, Christian Lölkes, Atmosphere Office
Management director: BI Xin
Public programs curators: BI Xin, CAO Jiamin
Technical project managemen: CAO Daxu
Communication and media: HAO Yu

The ZKM | Karlsruhe and the Chronus Art Center thank the artists and cooperation partners of the exhibition.

“开放代码”展览由彼得·威贝尔提出相关概念，并于卡尔斯鲁厄艺术与媒体中心（ZKM）首次展出。此次同名展览由新时线艺术中心（CAC）与卡尔斯鲁厄艺术与媒体中心（ZKM）共同呈现。

Open Codes was conceived and first exhibited at ZKM | Center for Art and Media Karlsruhe based on a concept by Peter Weibel. This new adaption is co-produced by Chronus Art Center and ZKM | Center for Art and Media Karlsruhe.

CO-ORGANIZED BY

SUPPORTED BY

IN COOPERATION WITH

参观信息

开馆时间

周三 - 周日
11:00 - 18:00

联系

电话：86 21 52715789
邮箱： info@chronusartcenter.org
网站：http://www.chronusartcenter.org
微信号：chronusartcenter
新浪微博：新时线媒体艺术中心 CAC
Facebook: Chronus Art Center (CAC)
Instagram: chronus_art_center

地址

上海市普陀区莫干山路 50 号 18 号楼

VISIT

OPENING HOURS

Wednesday - Sunday
11:00 - 18:00

CONTACT

Tel: 86 21 52715789
E-mail: info@chronusartcenter.org
Website: http://www.chronusartcenter.org
Wechat: chronusartcenter
Sina weibo 新时线媒体艺术中心 CAC
Facebook: Chronus Art Center (CAC)
Instagram: chronus_art_center

ADDRESS

Bldg18, NO.50 Moganshan Rd, Shanghai

CAC
新时线媒体艺术中心
CHRONUS ART CENTER