Code (or, How You Can Write Something Differently)

Friedrich Kittler

Codes—by name and by matter—are what determine us today, and what we must articulate if only to avoid disappearing under them completely. They are the language of our time precisely because the word and the matter code are much older, as I will demonstrate with a brief historical regression. And have no fear: I promise to arrive back at the present.

Imperium Romanum

Codes materialize in processes of encryption, which is, according to Wolfgang Coy's elegant definition, "from a mathematical perspective a mapping of a finite set of symbols of an alphabet onto a suitable signal sequence."[1] This definition clarifies two facts. Contrary to current opinion, codes are not a peculiarity of computer technology or genetic engineering; as sequences of signals over time they are part of every communications technology, every transmission medium. On the other hand, much evidence suggests that codes became conceivable and feasible only after true alphabets, as opposed to mere ideograms or logograms, had become available for the codification of natural languages. Those alphabets are systems of identically recurring signs of a countable quantity, which map speech sounds onto letters more or less one-to-one and, hopefully, completely. A vocalic alphabet of a type such as Greek,[2] justly praised for being the "first total analysis of a language,"[3] does appear to be a prerequisite for the emergence of codes, and yet, not a sufficient one. For what the Greeks lacked (leaving out of consideration sporadic allusions in the work of Aischylos, Aenas, Tacticus, and Plutarch to the use of secret writing[4] was that second prerequisite of all coding, namely, developed communications technology. It is anything but coincidental that our reports of the first secret message systems coincide with the rise of the Roman Empire. In his *Lives of the Caesars*, Suetonius—who himself served as secret scribe to a great emperor—recounts discovering encrypted letters among the personal files left behind by both the divine Caesar and the divine Augustus. Caesar contented himself with moving all the letters of the Latin alphabet by four places, thus writing D instead of A, E instead of B, and so forth. His adoptive son Augustus, by contrast, is reported to have merely skipped one letter, but a lack of mathematical discernment led him to replace the letter X, the last in his alphabet, by a double A.[5] The purpose was obvious: When read aloud by those not called upon to do so (and Romans were hardly the most literate of people), a stodgy jumble of consonants resulted. And as if such innovations in matters of encryption were not sufficient, Suetonius attributes to Caesar another invention immediately beforehand—that of having written in several columns, or even separate pages, reports to the Roman Senate on the Gallic campaign. Augustus is credited with the illustrious deed of creating, with riders and relay posts, Europe's first strictly military express-mail system.[6] In other words, the basis on which command, code, and communications technology coincided was the Empire, as opposed to merely the Roman Republic or shorthand writers like Cicero. Imperium is the name of both the command and its effect: the world empire. "Command, control, communications, intelligence" was also the Pentagon's imperial motto until very recently, when, due to the coincidence of communication technologies and Turing machines it was swapped for C^4—"command, control, communication, computers"—from Orontes to the Scottish headland, from Baghdad to Kabul.

It was the case, however, that imperia, the orders of the Emperor, were also known as codicilla, the word referring to the small tablets of stripped wood coated with wax in which letters could be inscribed. The etymon codex for its part—*caudex* in Old Latin and related to the German verb *hauen* (to hew)—in the early days of the Empire assumed the meaning of "book," whose pages could, unlike papyrus scrolls, for the first time be leafed through. And that was how the word that interests us here embarked on its winding journey to the French and English languages. From Imperator Theodosius to Empereur Napoleon, "code" was simply the name of the bound book of law, and codification became the word for the judicial-bureaucratic act needed to arrest in a single collection of laws the torrents of imperial dispatches or commands that for centuries had rushed along the express routes of the Empire. Message transmission turned into data storage,[7] pure events into serial order. And even today the Codex Theodosius and Codex Iustinianus continue to bear a code of ancient European rights and obligations in those countries where Anglo-American common law does not happen to be sweeping the board. In the Corpus Iuris, after all, copyrights and trademarks are simply meaningless, regardless of whether they protect a codex or a code.

Nation-States

The question that remains is why the technical meaning of the word "code" was able to obscure the legal meaning to such a degree. As we know, contemporary legal systems regularly fail to grasp codes in the first place and, in consequence, to protect them, be it from robbers and purchasers or, conversely, from their discoverers and writers. The answer seems to be simple. What we have been calling a code since the secret writings of Roman emperors to the arcana imperii of the modern age was known as a "cipher" from the late Middle Ages onward. For a long time the term code was understood to refer to very different cryptographic methods whereby words could still be pronounced, but obscure or innocuous words simply replaced the secret ones. Cipher, by contrast, was another name for the zero, which at that time reached Europe from India via Baghdad and put sifr (Arabic: "emptiness") into mathematical-technical power. Since that time, completely different sets of characters have been devised (in sharp contrast to the invention of Greek for speech sounds and numbers: on one side of language the alphabet of the people, on the other the numbers of the bearers of secrets—the name of which spelled the Arabic sifr once again. Separate character sets, however, are productive. Together they brew wondrous creatures that would never have occurred to the Greeks or Romans. Without modern algebra there would be no encoding; without Gutenberg's printing press, no modern cryptology. In 1462 or 1463, Battista Leone Alberti, the inventor of linear perspective, was struck by two plain facts. First, that the frequency of occurrence of phonemes or letters varies from language to language, a fact which is proved, according to Alberti, by Gutenberg's letter case. From the frequency of shifted letters as they were written by Caesar and Augustus, cryptanalysis can heuristically derive the clear text of the encrypted message. Second, it is therefore insufficient to encrypt a message by shifting all the letters by the same number of places. Alberti's proposal that every new letter in the clear text be accompanied by an additional place-shift in the secret alphabet was followed up until World War II.[8] One century after Alberti, François Viète, the founder of modern algebra, and also a cryptologist in the service of Henry IV, intertwined number and letter more closely still. Only since Viète have there been equations containing unknowns and universal coefficients written with numbers encoded as letters.[9] This is still the work method of anybody who writes in a high-level programming language that likewise allocates variables (in a mathematically more or less correct manner) to alphanumeric signs, as in equations. On this basis—Alberti's polyalphabetic code, Viète's algebra, and Leibniz' differential calculus—the nation-states of the modern age were able to technically approach modernity.

Global Message Traffic

Modernity began, however, with Napoleon. As of 1794, messengers on horseback were replaced by an optical telegraph which remote-controlled France's armies with secret codes. In 1806, the laws and privileges surviving from the old days were replaced by the cohesive Code Napoléon. In 1838, Samuel Morse is said to have inspected a printing plant in New York in order—taking a leaf from Alberti's book—to learn from the letter case which letters occurred most frequently and therefore required the shortest Morse signals.[10] For the first time a system of writing had been optimized according to technical criteria—that is, with no regard to semantics—but the product was not yet known as Morse code. The name was bestowed subsequently in books known as Universal Code Condensers, which offered lists of words that could be abbreviated for global cable communications, thus reducing the length, and cost, of telegrams, and thereby encrypting the sender's clear text for a second time. What used to be called deciphering and enciphering has since then been referred to as decoding and encoding. All code processed by computers nowadays is therefore subject to Kolmogorov's test: Input is bad if it is longer than its output; both are equally long in the case of white noise; and a code is called elegant if its output is much longer than itself. The twentieth century thus turned a thoroughly capitalist money-saving device called "code condenser" into highest mathematical stringency.

The Present Day — Turing

All that remains to ask is how the status quo came about or, in other words, how mathematics and encryption entered that inseparable union that rules our lives. That the answer is Alan Turing should be well known today. The Turing machine of 1936, as the principle controller of any computer, solved a basic problem of the modern age: how to note with finitely long and ultimately whole numbers the real, and therefore typically infinitely long, numbers on which technology and engineering have been based since Viète's time. Turing's machine proved that although this task could not be accomplished for all real numbers, it was achievable for a crucial subset, which he dubbed computable numbers.[11] Since then a finite quantity of signs belonging to a numbered alphabet which can, as we know, be reduced to zero and one, has banished the infinity of numbers.

No sooner had Turing found his solution than war demanded its cryptanalytical application. As of spring 1941 in Britannia's Code and Cipher School, Turing's proto-computers almost decided the outcome of the war by successfully cracking the secret codes of the German Wehrmacht, which, to its own detriment, had remained faithful to Alberti. Today, at a time when computers are not far short of unravelling the secrets of the weather or the genome—physical secrets, that is to say, and increasingly often biological ones, too—we all too often forget that their primary task is something different. Turing himself raised the question of the purpose for which computers were actually created, and initially stated as the primary goal the decoding of plain human language:

> Of the above possible fields the learning of languages would be the most impressive, since it is the most human of these activities. This field seems, however, to depend rather too much on sense organs and locomotion to be feasible. The field of cryptography will perhaps be the most rewarding. There is a remarkably close parallel between the problems of the physicist and those of the cryptographer. The system on which a message is enciphered corresponds to the laws of the universe, the intercepted messages to the evidence available, the keys for a day or a message to important constants which have to be determined. The correspondence is very close, but the subject matter of cryptography is very easily dealt with by discrete machinery, physics not so easily.[12]

Conclusions

Condensed into telegraphic style, Turing's statement thus reads: Whether everything in the world can be encoded is written in the stars. The fact that computers, since they too run on codes, can decipher alien codes is seemingly guaranteed from the outset. For the past three-and-a-half millennia, alphabets have been the prototype of everything that is discrete. But it has by no means been proven that physics, despite its quantum theory, is to be computed solely as a quantity of particles and not as a layering of waves. And the question remains whether it is possible to model as codes, down to syntax and semantics, all the languages that make us human and from which our alphabet once emerged in the land of the Greeks.

This means that the notion of code is as overused as it is questionable. If every historical epoch is governed by a leading philosophy, then the philosophy of code is what governs our own, and so code—harking back to its root, "codex"—lays down the law for one and all, thus aspiring to a function that was, according to the leading philosophy of the Greeks, exercised exclusively by Aphrodite.[13] But perhaps code means nothing more than codex did at one time: the law of precisely that empire which holds us in subjection and forbids us even to articulate this sentence. At all events, the major research institutions that stand to profit most from such announcements proclaim with triumphant certainty that there is nothing in the universe, from the virus to the Big Bang, which is not code. One should therefore be wary of metaphors that dilute the legitimate concept of code, such as when, for instance, in the case of DNS, it was not possible to find a one-to-one correspondence between material elements and information units as Lily Ray discovered in the case of bioengineering. As a word that in its early history meant "displacement" or "transferral"—from letter to letter, from digit to letters, or vice versa—code is the most susceptible of all to faulty communication. Shining in the aura of the word code one now finds sciences that do not even master their basic arithmetic or alphabet, let alone cause something to turn into something different as opposed to merely, as in the case of metaphors, go by a different name. Therefore, only alphabets in the literal sense of modern mathematics should be known as codes, namely one-to-one, finite sequences of symbols, kept as short as possible but gifted, thanks to a grammar, with the incredible ability to infinitely reproduce themselves: Semi-Thue groups, Markov chains,[14] Backus-Naur forms, and so forth. That, and that alone, distinguishes such modern alphabets from the familiar one that admittedly spelled out our languages and gave us Homer's poetry[15] but cannot get the technological world up and running the way computer code now does. For while Turing's machine was able to generate real numbers from whole numbers as required, its successors have—in line with Turing's daring prediction—taken command.[16] Today, technology puts code into the practice of realities, that is to say: it encodes the world.

I cannot say whether this means that language has already been vacated as the House of Existence. Turing himself, when he explored the technical feasibility of machines learning to speak, assumed that this highest art, speech, would be learned not by mere computers but by robots equipped with sensors, effectors, that is to say, with some knowledge of the environment. However, this new and adaptable environmental knowledge in robots would remain

This means that the notion of code is as overused as it is questionable. If every historical epoch is governed by a leading philosophy, then the philosophy of code is what governs our own, and so code—harking back to its root, "codex"—lays down the law for one and all, thus aspiring to a function that was, according to the leading philosophy of the Greeks, exercised exclusively by Aphrodite.[13] But perhaps code means nothing more than codex did at one time: the law of precisely that empire which holds us in subjection and forbids us even to articulate this sentence. At all events, the major research institutions that stand to profit most from such announcements proclaim with triumphant certainty that there is nothing in the universe, from the virus to the Big Bang, which is not code. One should therefore be wary of metaphors that dilute the legitimate concept of code, such as when, for instance, in the case of DNS, it was not possible to find a one-to-one correspondence between material elements and information units as Lily Ray discovered in the case of bioengineering. As a word that in its early history meant "displacement" or "transferral"—from letter to letter, from digit to letters, or vice versa—code is the most susceptible of all to faulty communication. Shining in the aura of the word code one now finds sciences that do not even master their basic arithmetic or alphabet, let alone cause something to turn into something different as opposed to merely, as in the case of metaphors, go by a different name. Therefore, only alphabets in the literal sense of modern mathematics should be known as codes, namely one-to-one, finite sequences of symbols, kept as short as possible but gifted, thanks to a grammar, with the incredible ability to infinitely reproduce themselves: Semi-Thue groups, Markov chains,[14] Backus-Naur forms, and so forth. That, and that alone, distinguishes such modern alphabets from the familiar one that admittedly spelled out our languages and gave us Homer's poetry[15] but cannot get the technological world up and running the way computer code now does. For while Turing's machine was able to generate real numbers from whole numbers as required, its successors have—in line with Turing's daring prediction—taken command.[16] Today, technology puts code into the practice of realities, that is to say: it encodes the world.

I cannot say whether this means that language has already been vacated as the House of Existence. Turing himself, when he explored the technical feasibility of machines learning to speak, assumed that this highest art, speech, would be learned not by mere computers but by robots equipped with sensors, effectors, that is to say, with some knowledge of the environment. However, this new and adaptable environmental knowledge in robots would remain obscure and hidden to the programmers who started them up with initial codes. The so-called "hidden layers" in today's neuronal networks present a good, if still trifling, example of how far computing procedures can stray from their design engineers, even if everything works out well in the end. Thus, either we write code that in the manner of natural constants reveals the determinations of the matter itself, but at the same time pay the price of millions of lines of code and billions of dollars for digital hardware; or else we leave the task up to machines that derive code from their own environment, although we then cannot read—that is to say: articulate—this code. Ultimately, the dilemma between code and language seems insoluble. And anybody who has written code even only once, be it in a high-level programming language or assembly, knows two very simple things from personal experience. For one, all words from which the program was by necessity produced and developed only lead to copious errors and bugs; for another, the program will suddenly run properly when the programmer's head is emptied of words. And in regard to interpersonal communications, that can only mean that self-written code can scarcely be passed on with spoken words. May myself and my audience have been spared such a fate in the course of this essay.

Translated by Tom Morrison, with Florian Cramer

Notes

1. Wolfgang Coy, *Aufbau und Arbeitsweise von Rechenanlagen: Eine Einführung in Rechnerarchitektur und Rechnerorganisation für das Grundstudium der Informatik*, p. 5.

2. On the latest research developments, see Barry B. Powell, *Homer and the Origin of the Greek Alphabet.*

3. Johannes Lohmann.

4. See, Wolfgang Riepl, *Das Nachrichtenwesen des Altertums: Mit besonderer Rücksicht auf die Römer.*

5. See, Caius Suetonius Tranquillus, *Vitae Caesarum,* I 56, 6 and II 86.

6. See Suetonius, I 56, 6 and II 49, 3. On the cursus publicus, in which Augustus himself recorded passes, orders, and letters dated with the exact time of day or night

(Suetonius, II 50), see Bernhard Siegert, "Der Untergang des römischen Reiches," in Hans Ulrich Gumbrecht and K. Ludwig Pfeiffer, eds., Paradoxien, Dissonanzen, Zusammenbrüche: Situationen offener Epistemologie, 495–514.

7. On the subject of temporal and spatial media and the process of adjustment from the empire to the monastic early Middle Ages, see Harold A. Innis, *Empire and Communications,* 104–120.

8. On the subject of Alberti, see David Kahn, *The Codebreakers: The Story of Secret Writing.* On the Enigma of the German Wehrmacht, see Andrew Hodges, *Alan Turing, The Enigma of Intelligence.*

9. Viète himself chose vowels for unknowns, and consonants for coefficients. Since Descartes' *Géométrie* (1637), the coefficients proceed from the beginning of the alphabet and the unknowns from the end (a, b, c . . . x, y, z). Since then, $x^n + y^n = z^n$ has been the classical example of a mathematical equation with no numbers at all, and thus one that would have been inconceivable to the Greeks, Indians, and Arabs.

10. See Coy, *Aufbau*, 6.

11. See Alan M. Turing, *Intelligence Service: Schriften,* 19–60.

12. Ibid, 98. ("Intelligent machinery," in *Machine Intelligence* 5, or in *The Essential Turing.*

13. "daímohn hê pánta kubernâi" ("God, who [feminine form!] controls all") is what Aphrodite called Parmenides (DK 8, B 12, 3).

14. On the subject of Markov chains, see Claude E. Shannon, *Ein/Aus: Ausgewählte Schriften zur Kommunikations- und Nachrichtentheorie,* 21–25.

15. On the subject of Homer and the vocalic alphabet, see Barry B. Powell, *Homer and the Origin of the Greek Alphabet.*

16. See, Turing, *Intelligence Service*, 15.