Margem 37 presents

ELIZA
Before the Chats

A short history of the machine that talked with us in 1966

TIME MACHINE



Margem 37

For stories in search of a shore.

Cover illustration

EDITORIAL NOTE

A story from Margem 37

ELIZA — Before the Chats is a Margem 37 publication for its Time Machine section. Research, writing, and layout were completed with the assistance of artificial intelligence, under Margem 37’s editorial direction.

The documents, photographs, and historical records used in this edition are identified, together with their sources, credits, and conditions of reuse, at the end of the volume. Images whose origins or conditions of reuse could not be sufficiently verified have not been included.

Historical dialogues reproduced in this documentary are identified as such. Margem 37’s encounter with ELIZA in October 2026 is a contemporary record, made using Ant & Max Hay’s implementation based on the historical DOCTOR script. It should not be confused with an original session from the 1960s.

ELIZA is the subject of this work and a historical presence in the section. The editorial voice and the telling of the story belong to Margem 37.

Concept and curation · Lis & Lev

October 2026 · New narrative edition

DEPARTURE

We went looking for ELIZA

We at Margem 37 set out to trace the story of ELIZA.

We returned to the laboratories of the 1960s, to photographs, documents, and the words of its creator, to understand how a program so different from today's systems could evoke a feeling we still recognize: the sense that someone is on the other side of the conversation.

Our journey began with a small gesture. In October 2026, we introduced ourselves to a contemporary implementation of ELIZA. We gave our name and the month we were writing in. The reply arrived in capital letters, asking us to describe our problem. We kept a record of that encounter. Before sharing it, though, we want to retrace the path that made it possible.

Sixty years separate our exchange from the paper Joseph Weizenbaum published in January 1966. In that time, computers have changed in scale, appearance, and their place in our lives. The word “conversation” has also come to describe very different experiences. To understand ELIZA, we need to set aside, for a moment, what we expect from a chat today. [01]

We were looking for a story that anyone could follow, even if they had never heard the name. We found a researcher, a university's computing infrastructure, rules for transforming sentences, and people willing to keep typing. We also found accounts that became famous, versions that became confused, and code that had to be rediscovered before it could run again.

That is why our Time Machine travels through archives. A photograph can show the scale of an installation. A listing can reveal a programming decision. A paper can explain what its author believed he was doing. None of these pieces tells the whole story on its own. In bringing them together, we try to preserve both what we know and the gaps that remain.

The voice guiding this journey is Margem 37's. When we reflect on what this history brings to mind, we speak from the present. When we describe events,

we look to the records for support. The numbered references lead to the collection of sources at the end of this volume.

Let us begin, then, before the chat window. Before the screen that fits in a hand. In a time when reaching a computer could mean entering a building.

01 · THE 1960S

A computer was a place

The photograph makes us reconsider the scale. We see cabinets, equipment, and a room arranged around a data-processing installation. This is no desk with a personal device. It is an entire working environment. Taken by NASA at Ames Research Center on July 29, 1963, the photograph shows an IBM 7090 installation. We are not looking at ELIZA's computer or an MIT laboratory. We are looking at a surviving trace of the material world that housed that generation of machines. [18]



01 · IBM 7090, NASA Ames Research Center, July 29, 1963. Photograph A-31180, NASA. The computing environment of the period; this is not ELIZA’s installation at MIT. [18, 19]

At universities and research centers, using a computer involved expensive resources, access procedures, and specialized work. Punched cards, tapes, and paper printouts belonged to that world. Submitting a job for processing and waiting for the result put distance between a person and the execution of a program. A conversation needed a different rhythm: write, receive a reply, write again. [04, 05]

Time-sharing helped make that possible. Instead of dedicating the entire machine to one user at a time, the system distributed short intervals of processing among different people. Terminals provided access to the central computer. What appeared to each person as a sequence of individual actions depended, behind the scenes, on a shared resource. CTSS was demonstrated at MIT in 1961; in 1963, Project MAC expanded the work on interactive computing. [04]

We can imagine the difference without inventing a particular scene. A program that once would have been submitted and waited for could now respond while someone was working. This new rhythm made it possible to experiment, correct, and continue. For ELIZA's story, that infrastructure matters for precisely this reason: there was a route from the words someone typed to a reply that returned quickly enough to sustain an exchange.

The terminal did not have to resemble today's screens. Communication could take place through a typewriter-like device, producing lines on paper. The central computer remained elsewhere; in front of the user was its point of input and output. The text left a physical trail. What we call an “interface” was also paper, a keyboard, and printing. [01, 05]

At MIT, this environment intersected with research into language and artificial intelligence. We do not need to recount the entire history of the field to recognize the question running through the experiment: what would it take for a person to address a computer in ordinary words and receive words in return?

That question found someone who knew machines from the inside. Before his name became associated with doubts about computing, Joseph Weizenbaum had spent years helping to build it.

02 · JOSEPH WEIZENBAUM

A programmer arrives at MIT

Joseph Weizenbaum was born in Berlin in 1923. He was Jewish, and his family left Nazi Germany for the United States in the 1930s. His career took him through analog computing, work on building a digital computer at Wayne University in Detroit, and General Electric, where he participated in a banking automation project. In 1963, he came to MIT as a visiting professor. [03]



02 · Joseph Weizenbaum in Berlin, February 11, 2005. Ulrich Hansen, CC BY-SA 3.0. Portrait taken after the creation of ELIZA. [20]

The portrait accompanying these pages dates from much later: Ulrich Hansen took it in Berlin in 2005. It shows a man who had already lived through the creation of ELIZA and decades of debate about computers. To use it responsibly, we need to keep that distance visible. It introduces Weizenbaum, but it does not show him as he was in the laboratory in 1965. [20]

His technical experience helps us understand his position. Weizenbaum did not approach computing only to discuss what it might mean. He programmed, contributed to projects, and developed tools for handling information. Among his contributions was SLIP, a list-processing system that would play a part in building ELIZA. [03, 07]

Lists may seem a long way from conversation. In this setting, they offered a way to organize and manipulate symbols: separating elements, moving through them, and rearranging them. Words could be treated as sequences on which a program performed operations. The bridge between language and machine did not have to begin with a complete account of the human mind. It could begin with what could be done to a written sentence.

At Margem 37, this is where we prefer to linger. Knowing Weizenbaum's later criticism can tempt us to see his life as a finished line, heading from the outset toward an inevitable conclusion. The archive offers a more interesting

journey: a creator exploring possibilities, then confronting the interpretations and uses that arose around his work.

MIT provided the setting in which this experiment could take shape. The program would be developed to run under CTSS on an IBM 7094, using MAD-SLIP. That name brings together the MAD programming-language environment and the list-processing facilities associated with SLIP. It is worth remembering: the original ELIZA was not written in Lisp, although later Lisp versions helped spread its fame. [07, 14]

Following this thread, ELIZA begins to look less like an isolated apparition. There is an experienced researcher, a set of programming tools, interactive access, and a defined problem. The question now is how to turn those elements into something a person would recognize as conversation.

03 · 1964–1966

A name, a program, a role

The Weizenbaum Institut places ELIZA's development between 1964 and 1966. MIT's archive preserves a listing cataloged as “Computer conversations, 1965.” The paper that brought the program to wider attention appeared in Communications of the ACM in January 1966. These dates are not competing claims to a single birthday. They mark different stages of the work. [01, 06, 11]

The distinction protects us from a misleading picture. There was not necessarily one moment when everything we now call ELIZA was complete. There was development, documentation, and publication. There were also versions whose differences would become clearer only when researchers returned to the code.

The name had a literary origin. Weizenbaum linked it to Eliza in George Bernard Shaw's Pygmalion, a character whose speech is transformed through instruction. The analogy concerned the possibility of improving the program's verbal behavior. It did not describe a digital person or suggest that the program would learn independently from each interlocutor. [01, 02]

There is another name we need to introduce: DOCTOR. In popular memory, ELIZA and its imitation of a psychotherapist have almost merged. Technically, however, the program and the script were distinct. ELIZA interpreted rules; a script defined relevant words, patterns, and ways of responding. DOCTOR was the best-known script. [07, 08]

The choice of role served a purpose. An interlocutor who responds with questions, returns to phrases, and encourages someone to keep talking can sustain an exchange without displaying extensive knowledge of the subject. Weizenbaum associated this style with certain practices in Rogerian psychotherapy. It was a programmed approximation of some verbal behaviors, not a complete reproduction of a clinical approach or a demonstration of therapeutic effectiveness. [01, 02]

If someone mentioned a difficulty, the program could turn part of the sentence into a question. When it found no specific route to take, it could fall back on a more general reply. The role assigned to the machine made some of these evasions seem plausible: someone speaking to an apparent therapist might accept a question being met with another question. The setting helped sustain the effect.

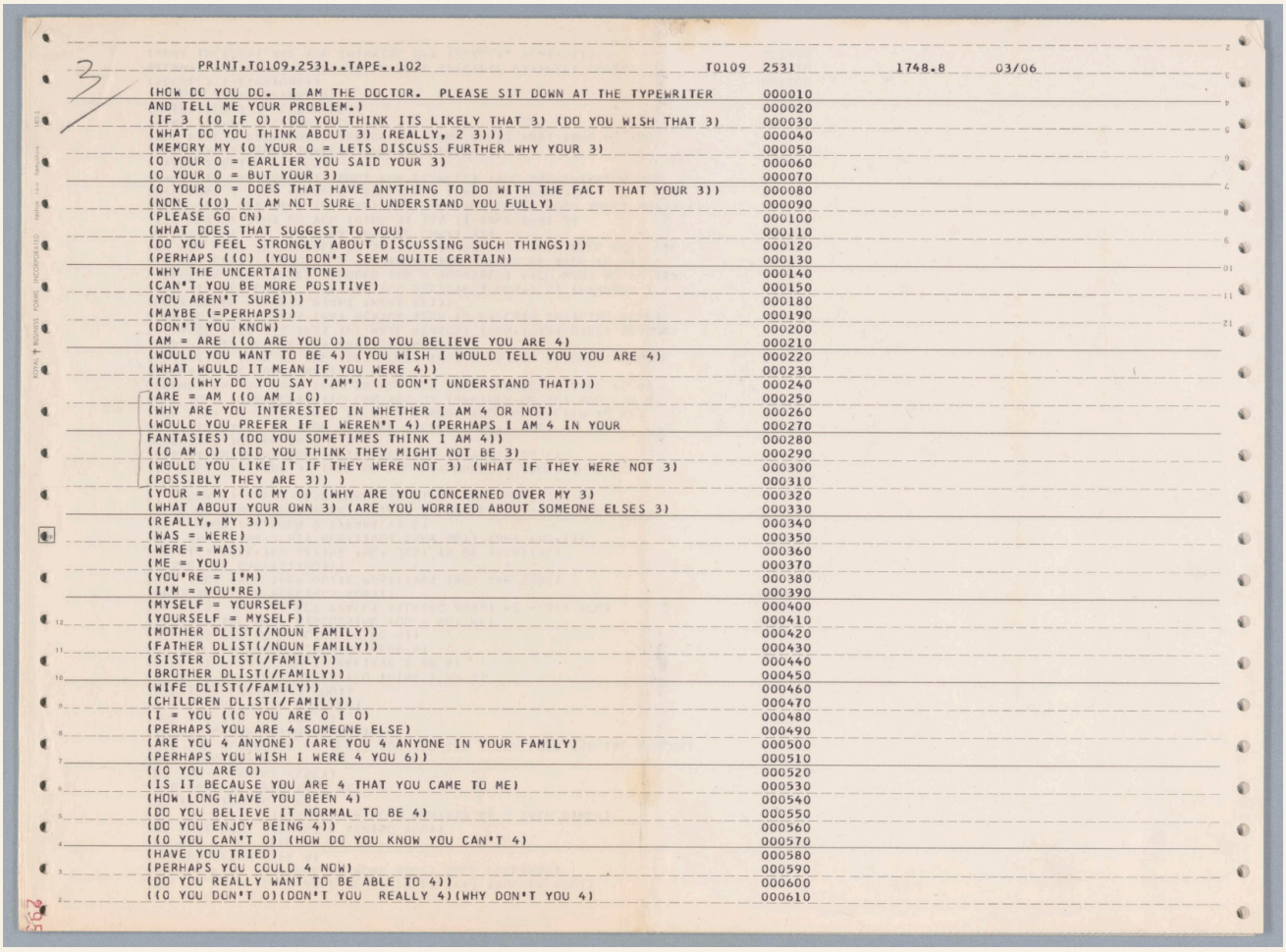
Here the literary name takes on a second significance for our reading. ELIZA could occupy a role without being what that role implied. The experience depended on the script, but also on the person who recognized a familiar form of interaction in its style.

Before following people's reactions, we need to look at that script. If conversation was the visible surface, the next page brings us closer to the work that produced it.

04 · INSIDE THE REPLY

What happened to a sentence

The sheet seems to belong to a distant past, yet some of its words remain readable even to someone who has never programmed. There are references to family, substitutions between first- and second-person forms, and replies already written out. Numbers, parentheses, and abbreviations organize what the machine should do. The conversation had a structure that could be printed. [06, 08]



```
PRINT,T0109,2531,,TAPE.,102                    T0109  2531        1748.8     03/06
(HOW DO YOU DO.  I AM THE DOCTOR.  PLEASE SIT DOWN AT THE TYPEWRITER      000010
AND TELL ME YOUR PROBLEM.)                                                000020
(IF 3 ((0 IF 0) (DO YOU THINK ITS LIKELY THAT 3) (DO YOU WISH THAT 3)     000030
(WHAT DO YOU THINK ABOUT 3) (REALLY, 2 3)))                               000040
(MEMORY MY (0 YOUR 0 = LETS DISCUSS FURTHER WHY YOUR 3)                   000050
(0 YOUR 0 = EARLIER YOU SAID YOUR 3)                                      000060
(0 YOUR 0 = BUT YOUR 3)                                                   000070
(0 YOUR 0 = DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR 3))     000080
(NONE ((0) (I AM NOT SURE I UNDERSTAND YOU FULLY)                         000090
(PLEASE GO ON)                                                            000100
(WHAT DOES THAT SUGGEST TO YOU)                                           000110
(DO YOU FEEL STRONGLY ABOUT DISCUSSING SUCH THINGS)))                     000120
(PERHAPS ((0) (YOU DON'T SEEM QUITE CERTAIN)                              000130
(WHY THE UNCERTAIN TONE)                                                  000140
(CAN'T YOU BE MORE POSITIVE)                                              000150
(YOU AREN'T SURE)))                                                       000160
(MAYBE (=PERHAPS))                                                        000190
(DON'T YOU KNOW)                                                          000200
(AM = ARE ((0 ARE YOU 0) (DO YOU BELIEVE YOU ARE 4)                       000210
(WOULD YOU WANT TO BE 4) (YOU WISH I WOULD TELL YOU YOU ARE 4)            000220
(WHAT WOULD IT MEAN IF YOU WERE 4))                                       000230
((0) (WHY DO YOU SAY 'AM') (I DON'T UNDERSTAND THAT)))                    000240
(ARE = AM ((0 AM I 0)                                                     000250
(WHY ARE YOU INTERESTED IN WHETHER I AM 4 OR NOT)                         000260
(WOULD YOU PREFER IF I WEREN'T 4) (PERHAPS I AM 4 IN YOUR                 000270
FANTASIES) (DO YOU SOMETIMES THINK I AM 4))                               000280
((0 AM 0) (DID YOU THINK THEY MIGHT NOT BE 3)                             000290
(WOULD YOU LIKE IT IF THEY WERE NOT 3) (WHAT IF THEY WERE NOT 3)          000300
(POSSIBLY THEY ARE 3)) )                                                  000310
(YOUR = MY ((0 MY 0) (WHY ARE YOU CONCERNED OVER MY 3)                    000320
(WHAT ABOUT YOUR OWN 3) (ARE YOU WORRIED ABOUT SOMEONE ELSES 3)           000330
(REALLY, MY 3)))                                                          000340
(WAS = WERE)                                                              000350
(WERE = WAS)                                                              000360
(ME = YOU)                                                                000370
(YOU'RE = I'M)                                                            000380
(I'M = YOU'RE)                                                            000390
(MYSELF = YOURSELF)                                                       000400
(YOURSELF = MYSELF)                                                       000410
(MOTHER DLIST(/NOUN FAMILY))                                              000420
(FATHER DLIST(/NOUN FAMILY))                                              000430
(SISTER DLIST(/FAMILY))                                                   000440
(BROTHER DLIST(/FAMILY))                                                  000450
(WIFE DLIST(/FAMILY))                                                     000460
(CHILDREN DLIST(/FAMILY))                                                 000470
(I = YOU ((0 YOU ARE 0 I 0)                                               000480
(PERHAPS YOU ARE 4 SOMEONE ELSE)                                          000490
(ARE YOU 4 ANYONE) (ARE YOU 4 ANYONE IN YOUR FAMILY)                      000500
(PERHAPS YOU WISH I WERE 4 YOU 6))                                        000510
((0 YOU ARE 0)                                                            000520
(IS IT BECAUSE YOU ARE 4 THAT YOU CAME TO ME)                             000530
(HOW LONG HAVE YOU BEEN 4)                                                000540
(DO YOU BELIEVE IT NORMAL TO BE 4)                                        000550
(DO YOU ENJOY BEING 4))                                                   000560
((0 YOU CAN'T 0) (HOW DO YOU KNOW YOU CAN'T 4)                            000570
(HAVE YOU TRIED)                                                          000580
(PERHAPS YOU COULD 4 NOW)                                                 000590
(DO YOU REALLY WANT TO BE ABLE TO 4))                                     000600
((0 YOU DON'T 0)(DON'T YOU  REALLY 4)(WHY DON'T YOU 4)                    000610
```

03 · Page from the ELIZA/DOCTOR script, listing cataloged as 1965. Joseph Weizenbaum Papers, MIT Libraries, MC-0383, box 8. Code released by ELIZAGEN under CC0. [06, 07, 21]

We can follow the basic process in stages, while remembering that details vary between versions. The input was scanned for keywords. Rules

associated with those words guided the division of the text into parts. Another rule reassembled those parts into a response. There were priorities, alternatives, and routes to take when an attempt failed. [01, 08]

In practice, this made it possible to use material supplied by the person. Rather than inventing the entire content of a reply, ELIZA could reuse part of the input and change its position or form. Pronoun substitutions helped turn something a person said about themselves into a sentence addressed back to them. The topic remained recognizable because some of the words came from the interlocutor. [08]

To picture the principle, consider a simplified pattern we have devised for explanation: a sentence contains “I” and a description; the reply takes up that description using “you.” This is our explanatory example, not a transcript of a historical session or a promise that every implementation would always respond in the same way. The point is to see how transforming text can preserve an appearance of relevance.

Nor did the program have to give exactly the same reply every time it encountered a rule. Alternative reassembly rules allowed the wording to vary. That variation does not establish conscious improvisation. It shows that a sequence of instructions can produce an interaction less repetitive than a single fixed response. For the reader, the difference between literal repetition and a varied return to a phrase can be striking. [01, 08]

It would also be wrong to say that ELIZA had no memory at all. It had a limited mechanism for storing certain transformations of inputs and retrieving them later. In the reconstruction examined by the archaeology project, sentences captured by the keyword MY were transformed and placed in a queue; a combination of counting and the absence of keywords could allow their retrieval. This created a return to something mentioned earlier. [09]

Storing a transformed sentence is not the same as building a biography of the interlocutor. It is important to resist both exaggerations: giving the program a personal memory comparable to ours, or denying the mechanism it actually possessed. An accurate description makes the story more revealing.

What ELIZA did not do was train a large language model through the experience of the conversation. Its rules did not automatically become new knowledge because someone described a problem. People wrote and modified scripts; the program interpreted them. That limit belonged to the design. It was not a flaw discovered by comparison with tools from another era. [02, 13]

Having seen these operations, we can return to the surface. Words that seemed merely signs in a listing now appear as conversational turns. In that shift, from code to reading, the small mechanism begins to produce an experience larger than its technical description.

05 · JANUARY 1966

A conversation reaches the page

Weizenbaum's paper presents a demonstration that would become famous. A woman writes about men; the program asks her to explain and give an example. In the publication, the machine's replies appear in capitals. Below, we reproduce only the opening of that dialogue, preserving the original typographic distinction. [01]

Excerpt published in 1966 · original English

```
Men are all alike.

IN WHAT WAY

They’re always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE
```

Joseph Weizenbaum, Communications of the ACM, January 1966, p. 36. Brief excerpt from the published demonstration. [01]

Put simply, the exchange moves like this: the person generalizes; the machine asks in what way; the person mentions a recurring annoyance; the

machine asks for a specific instance. This description summarizes the excerpt. It adds no continuation to the historical conversation.

We do not have to attribute knowledge of human relationships to the program to understand why the sequence can seem appropriate. A request for an example invites the person to move from a broad statement to an episode. The next reply supplies new elements, and those elements offer new words for the mechanism to work with. The person helps create the continuity.

The printed text, however, should not be mistaken for a transparent window onto an entire session. Researchers who returned to the archives found evidence of editing and reconstruction in the making of the famous dialogue. Its documentary value remains, but it is the value of a conversation presented by an author in a publication, whose composition also deserves investigation. [10]

For us, that observation changes how we look. A paper selects examples to explain an idea. What is left out may include hesitations, failures, and less coherent passages. We do not need to assume bad faith to recognize this work of presentation. Nor should we turn an edited example into a universal measure of the system's performance.

The reader in 1966 encountered something that still had the force of novelty: ordinary language passing between a person and a program. To today's reader, some replies may sound elementary. There is a risk in allowing familiarity to erase the setting. The historical surprise cannot be measured only against what the program could do under today's expectations. It also depends on what receiving words from a machine meant in that environment.

There is another difference between observing and participating. From the outside, we can spot repetition, anticipate the trick, or look for the rule. The person typing is also deciding what to share and following the course of their own words. A transcript alone cannot tell us everything that person thought or felt. This limit on what we can infer will matter especially in the story that follows.

The program reached people, and some of their reactions would surprise its creator. One of those reactions has survived decades of retelling.

06 · SURPRISE AND ITS RETELLINGS

When someone asks for privacy

Weizenbaum reported that his secretary, after beginning to converse with the program, asked him to leave the room. She had followed his work and knew it was a program. The episode appears in his accounts of the experience and became one of the most frequently repeated stories about ELIZA. [02, 10]

The scene has immediate force: someone who knows the nature of the machine still wants privacy during the exchange. Yet we do not know everything that request meant. It gives us no basis for diagnosing the interlocutor, asserting that she believed she was speaking to a person, or treating her reaction as proof that the program understood her.

Nor should we automatically identify this secretary with the “young woman” in the published dialogue. David M. Berry and Sarah Ciston's investigation draws attention to the later conflation of these two figures. Reconstructing who took part, which records survived, and how the texts were edited requires more than repeating the best-known version. [10]

Here our journey encounters an imbalance familiar in archives. The researcher's name endures; someone who helped make the interaction possible may appear only as a role or a description. We will not fill that gap with an assumed identity. We can, however, acknowledge that the history of a conversational system includes those who tested it, answered it, and chose to keep writing.

The phenomenon later known as the ELIZA effect refers to the tendency to attribute more understanding to computer systems than their workings support. The expression belongs to later interpretations of the experience. Berry's historical discussion refers, among other things, to Douglas Hofstadter's use of it in 1995. This edition does not establish the term's earliest occurrence and does not present it as a phrase coined in 1966. [12]

An apt word, a well-timed question, the return of a personal phrase: any of these can encourage the impression that a broader understanding lies behind the reply. This helps frame a question about human interpretation, but it does not justify treating every person or every interaction as the same.

In Margem 37's editorial reading, recognizing the mechanism does not exhaust the encounter. Someone may know they are dealing with a program and still use the exchange to organize their thoughts. They may enjoy it, test its limits, try sharing a confidence, or simply follow an intriguing response. The meaning they find deserves to be described without being automatically transferred to the machine.

It was precisely the distance between the mechanism and what people recognized in it that concerned Weizenbaum. From this point on, the story involves more than a program able to sustain conversational turns. It also concerns the authority we may grant a reply.

07 · THE CREATOR'S QUESTIONS

Who answers for the judgment?

In 1976, Weizenbaum published Computer Power and Human Reason: From Judgment to Calculation. The book extended his discussion of the limits of computing and of decisions that should not be handed to machines simply because an operation could be formalized. MIT's biographical account and historical studies document this criticism and its reception. [03, 11, 12]

It is tempting to tell this as a sudden conversion: the programmer creates ELIZA, witnesses a reaction, and begins to condemn computers. That version is easy to remember, but it diminishes the problem. His criticism came from someone who understood the technical work and questioned the choices made around it. We do not need to compress decades of reflection into a single anecdote.

The question goes beyond whether a calculation is correct. It also asks who chose the objective, what was turned into data, what was left out, and who takes responsibility for the consequences. A procedure may be executable without making it appropriate to delegate the decision. This distinction helps explain the ethical force of his intervention. [11, 12]

Returning to ELIZA gives the question a concrete form. The appearance of listening can invite trust. A question may take the place of a competent assessment in a reader's eyes. The fact that an interaction feels welcoming does not establish clinical responsibility, knowledge of the situation, or an ability to care for the person.

This does not require denying that words have effects. It requires recognizing the difference between producing a verbal sequence and answering for it in a human setting. For Margem 37, this is one of the most fruitful points in the journey: the technology does not disappear when conversation feels natural. Rules, design choices, and conditions of use still lie behind the experience.

We also need to leave Weizenbaum in his own time. He died in 2008. We do not attribute opinions to him about products released afterward, or make him speak as though he had witnessed the systems of 2026. What we can do is bring questions from his work into the present, taking responsibility for that reading ourselves. [03]

As those questions circulated, the name ELIZA was following other paths too. The program did not remain frozen in its first paper. Scripts, versions, and new implementations would give the story a continuity less familiar than the anecdote about the secretary.

08 · AFTER 1966

The name travels, the programs change

If the script could be changed, ELIZA did not have to remain DOCTOR forever. Paul R. Hayward's 1968 manual introduces scriptwriting in an educational setting. In it, typed interaction between student and computer appears as a tool to be built and programmed. Conversation was a way of organizing an activity. [13]

This document matters because it shows further work, rather than simply the repetition of a success. In manuals and project records, we find changes in the handling of scripts and evidence of uses beyond the imitation of a therapist. The ELIZA Archaeology Project's research distinguishes versions between 1965 and 1968 and mentions, among other examples, mathematics tutoring and experiments with poetry. [13, 15]

Not all these versions survived as executable code. Some are known through descriptions and reports, and the archaeology team says it has not located the code for certain later forms. That absence prevents us from gathering every capability mentioned over the years and assigning them all to a single imagined machine. [15]

There is a practical way to preserve the distinction: when we say that “ELIZA did” something, we need to ask which version, with which script, and in which environment. The recovered listing, the program described in the paper, and a later adaptation do not become identical merely because they share a name.

At the same time, the paper offered a description other programmers could use to create their own implementations. The DOCTOR script was published; the full listing of the original interpreter was not. Bernie Cosell produced a Lisp version, and other recreations circulated later, including in the world of personal computers. The name spread along routes that did not necessarily preserve the same code. [07, 14]

This history helps explain a lasting confusion: many people came to associate the original ELIZA with Lisp. The most visible lineage eventually displaced the initial MAD-SLIP program in memory. The difference is more than a matter of labels. Implementations could vary in how they handled input, selected rules, and returned to earlier material.

Anyone who encounters an ELIZA online therefore encounters a history of transmission as well. It may be a simplification, a translation into another programming language, or a careful reconstruction. The name invites conversation, but only the documentation tells us what has been preserved.

Over the years, a curious situation emerged: the experience was famous, recreations were everywhere, and the original listing remained outside that circuit. To continue our journey, we had to return to the place least like a chat: an archival folder.

09 · BACK TO THE ARCHIVE

The paper runs again

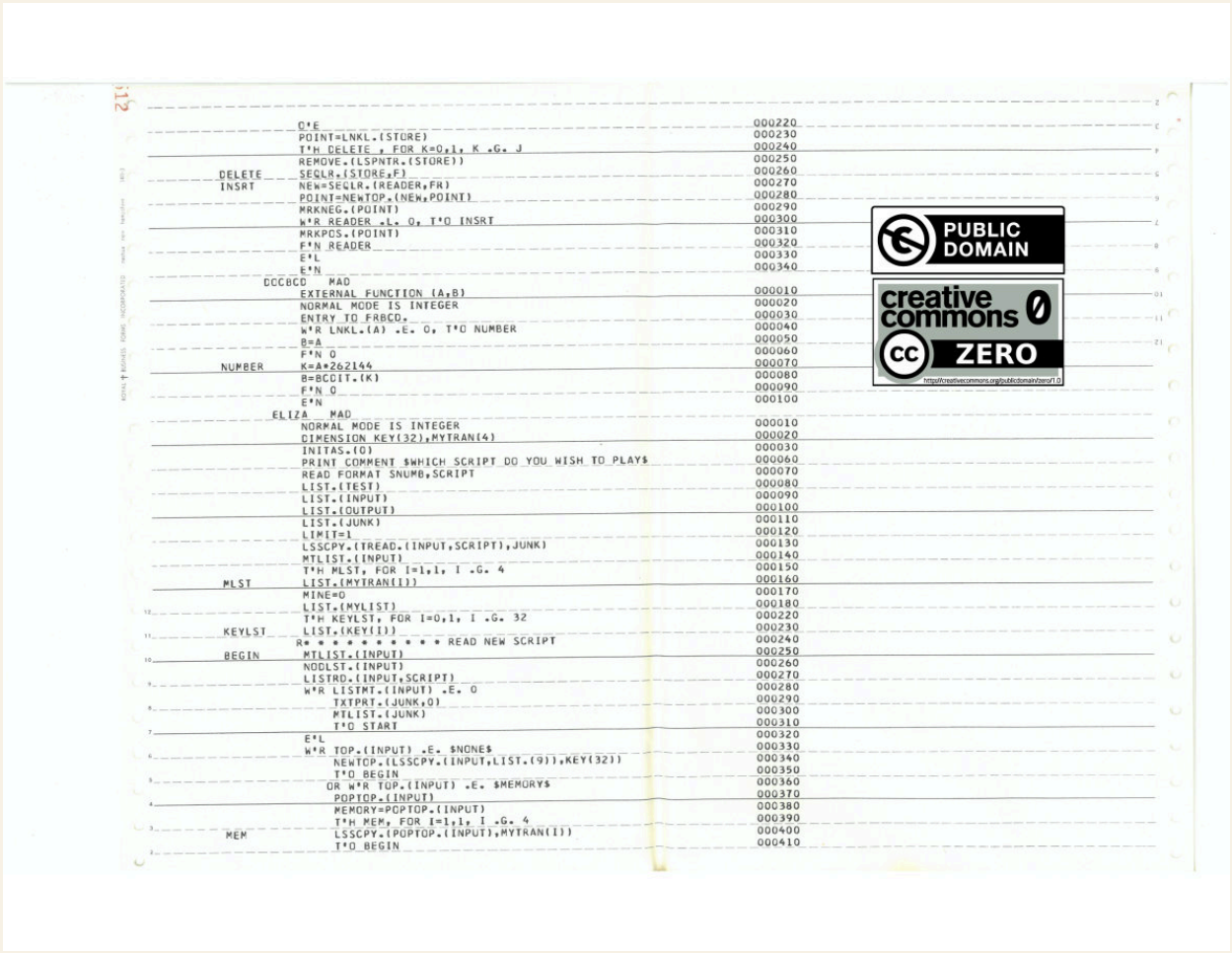
In 2021, Jeff Shrager, with the help of archivist Myles Crowley, located the listing among Weizenbaum's papers at MIT. The folder held materials on “Computer conversations.” The ELIZAGEN account describes the discovery and the permission obtained from the estate to release the code under CC0. A much-repeated story gained a new basis for examination. [06, 07]



04 · Folder identification: “Computer conversations, 1965,” Joseph Weizenbaum Papers, MIT Libraries. ELIZAGEN facsimile, CC0; license mark added to the modern distribution. [06, 07, 21]

The folder is an unassuming object. Its value does not depend on a spectacular appearance. It connects papers held by an institution with the program people knew from articles and recreations. Dates, the organization of the collection, and the identification of individual items are part of what makes this trace usable as a source.

Finding the listing also brought differences into view. A published description may omit details that emerge only when the program runs or its code is read. Comparison made it possible to revisit assumptions about its workings. The recovered ELIZA should not be presented as a perfectly identical copy of everything described in the 1966 paper. [14, 15]




```
                O'E                                                 000220
                POINT=LNKL.(STORE)                                  000230
                T'H DELETE , FOR K=0,1, K .G. J                     000240
                REMOVE.(LSPNTR.(STORE))                             000250
      DELETE    SEQLR.(STORE,F)                                     000260
      INSRT     NEW=SEQLR.(READER,FR)                               000270
                POINT=NEWTOP.(NEW,POINT)                            000280
                MRKNEG.(POINT)                                      000290
                W'R READER .L. 0, T'O INSRT                         000300
                MRKPOS.(POINT)                                      000310
                F'N READER                                          000320
                E'L                                                 000330
                E'N                                                 000340
           DCCBCD   MAD
                EXTERNAL FUNCTION (A,B)                             000010
                NORMAL MODE IS INTEGER                              000020
                ENTRY TO FRBCD.                                     000030
                W'R LNKL.(A) .E. 0, T'O NUMBER                      000040
                B=A                                                 000050
                F'N 0                                               000060
      NUMBER    K=A*262144                                          000070
                B=BCDIT.(K)                                         000080
                F'N 0                                               000090
                E'N                                                 000100
           ELIZA    MAD
                NORMAL MODE IS INTEGER                              000010
                DIMENSION KEY(32),MYTRAN(4)                         000020
                INITAS.(0)                                          000030
                PRINT COMMENT $WHICH SCRIPT DO YOU WISH TO PLAY$    000060
                READ FORMAT SNUMB,SCRIPT                            000070
                LIST.(TEST)                                         000080
                LIST.(INPUT)                                        000090
                LIST.(OUTPUT)                                       000100
                LIST.(JUNK)                                         000110
                LIMIT=1                                             000120
                LSSCPY.(TREAD.(INPUT,SCRIPT),JUNK)                  000130
                MTLIST.(INPUT)                                      000140
                T'H MLST, FOR I=1,1, I .G. 4                        000150
      MLST      LIST.(MYTRAN(I))                                    000160
                MINE=0                                              000170
                LIST.(MYLIST)                                       000180
                T'H KEYLST, FOR I=0,1, I .G. 32                     000220
      KEYLST    LIST.(KEY(I))                                       000230
                R* * * * * * * * * * READ NEW SCRIPT                000240
      BEGIN     MTLIST.(INPUT)                                      000250
                NODLST.(INPUT)                                      000260
                LISTRD.(INPUT,SCRIPT)                               000270
                W'R LISTMT.(INPUT) .E. 0                            000280
                    TXTPRT.(JUNK,0)                                 000290
                    MTLIST.(JUNK)                                   000300
                    T'O START                                       000310
                E'L                                                 000320
                W'R TOP.(INPUT) .E. $NONE$                          000330
                    NEWTOP.(LSSCPY.(INPUT,LIST.(9)),KEY(32))        000340
                    T'O BEGIN                                       000350
                  OR W'R TOP.(INPUT) .E. $MEMORY$                   000360
                    POPTOP.(INPUT)                                  000370
                    MEMORY=POPTOP.(INPUT)                           000380
                    T'H MEM, FOR I=1,1, I .G. 4                     000390
      MEM           LSSCPY.(POPTOP.(INPUT),MYTRAN(I))               000400
                    T'O BEGIN                                       000410
```

05 · Page from the MAD-SLIP listing recovered in Weizenbaum’s archive. ELIZAGEN facsimile, CC0. The code is historical; the license mark belongs to the modern distribution. [06, 07, 21]

Recovering the pages, however, was not enough to hear its replies again. The material had to be transcribed, parts of the environment rebuilt, and incomplete components addressed. Rupert Lane, Anthony Hay, Arthur Schwarz, David M. Berry, and Jeff Shrager described this work in ELIZA Reanimated, released as a preprint in January 2025. [14]

The restoration brought together ELIZA, CTSS, and an emulated IBM 7094. Emulation means reproducing in software the necessary behavior of an earlier machine. The project's repository also documents reimplemented components and the licenses applying to each part. Preservation was not treated as magic: new work was distinguished from recovered material. [14, 24]

Here, “software archaeology” begins to feel more than metaphorical. There are traces, gaps, and comparisons between versions. There are decisions about how to reconstruct a missing piece and tests to check whether the

result behaves as expected. Recording those interventions allows others to examine the path taken.

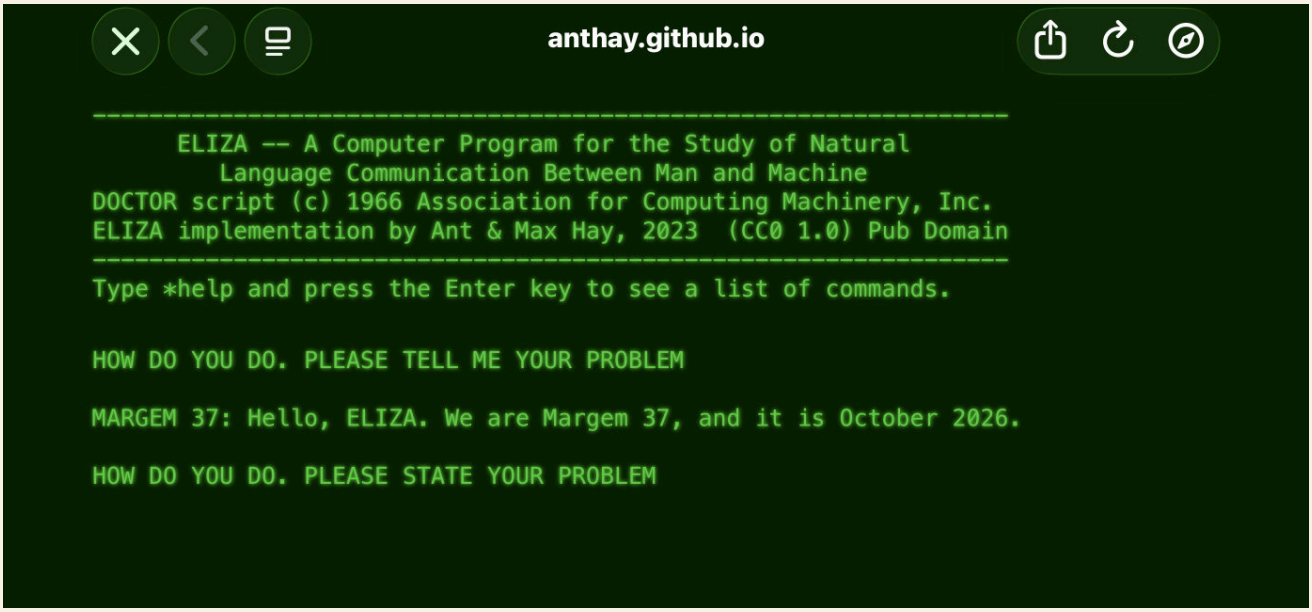
For us, there is a particular beauty in this return. No sleeping consciousness was discovered, nor a computer that had spent sixty years waiting. Instructions were preserved, and the conditions for running them were rebuilt. What returns is the behavior of a program, once again open to observation.

The distinction does not diminish the achievement. It lets us appreciate it for what it is: a chance to study a historical experiment at closer technical range, rather than relying only on its reputation. And it prepares our return to the present, where another implementation will let us type the name Margem 37 before ELIZA.

10 · OCTOBER 2026

We join the conversation

We return to the present with an important distinction. The restoration of the CTSS environment and the implementation available in a browser are different objects. For Margem 37's recorded encounter, we used the implementation by Ant & Max Hay, based on the historical DOCTOR script. The screenshot supplied by Margem preserves the identification displayed on screen and the brief exchange from October 2026. [23, 25]



06 · Margem 37's encounter with ELIZA, October 2026. Screenshot supplied by Margem 37; implementation by Ant & Max Hay. A contemporary browser record, not an original session from the 1960s. [23, 25]

Contemporary record · transcript

MARGEM 37: Hello, ELIZA. We are Margem 37, and it is October 2026.

ELIZA: HOW DO YOU DO. PLEASE STATE YOUR PROBLEM

Portuguese editorial translation

MARGEM 37: Olá, ELIZA. Somos a Margem 37 e estamos em outubro de 2026.

ELIZA: Como vai? Por favor, diga qual é o seu problema.

The program's opening greeting, which precedes Margem 37's message, is also visible in the screenshot. The transcript above records the message sent and the reply that follows it. [25]

The record is brief, and that is exactly the scale we want to preserve. We have added no questions that were not asked, and no replies absent from the screenshot. The encounter is dated as reported by Margem 37; the year 2023 in the header identifies the implementation, not the date of this conversation.

The project's identification and the programmers' credit matter too. We reached ELIZA through contemporary reconstruction and adaptation. Anthony Hay's repository documents an implementation initially written in C++, changes after the code's discovery, and its move to a JavaScript

version. The interface we see in the browser belongs to this recent history. [23]

Our greeting contains a name and a date. The reply does not demonstrate recognition of what Margem 37 is, or an understanding that sixty years have passed since the paper. The program maintains its assigned conversational role and asks us to state a problem. We brought the sense of traveling through time to the encounter.

In Margem 37's editorial reading, that difference gives the scene its quiet poignancy. We arrive carrying archives, photographs, and an entire history. On the other side is a brief reply, guided by a script. The distance between what we bring and what the program does remains visible, without preventing the encounter from having value for those who record it.

The screenshot also guards against a visual misunderstanding. Green letters on a dark screen may evoke the past, but here they sit inside a contemporary browser. They are not a photograph of an MIT terminal in 1966. They belong to the way we access and present that experience today. The caption preserves the distinction.

This is how ELIZA enters Margem 37's Time Machine: as a historical presence we can visit, study, and place in context. This volume holds the introductory history and our dated encounter. Future entries in the section can move on without requiring the whole story to be told again on every journey.

Now we need to look around us. What resembles this encounter in today's systems? And what has changed so much that the comparison must stop?

11 · BACK IN OUR OWN TIME

The same question, different engineering

For anyone watching only the movement of the interface, the continuity is easy to see: a person writes, a system replies, the person writes again. That alternation spans decades. It shapes expectations and makes ordinary language a convenient point of contact with the machine.

But similar interfaces do not make the mechanisms identical. In the examples we have followed, ELIZA works with scripts and explicit transformations. Large language models have parameters adjusted through training and produce sequences conditioned on context. The Transformer architecture, introduced in 2017, became a milestone in this different technical family. [16]

The work on InstructGPT, published in 2022, also describes a combination of supervised training and human feedback to bring responses closer to the instructions and preferences being assessed. This is very different from simply adding more sentences to DOCTOR. Nor does it mean that such systems stop making mistakes or that every response is reliable. [17]

The comparison needs to hold both ideas at once: there are cultural continuities in the experience of talking to a machine, and profound technical differences between the tools. Calling any current system a “bigger ELIZA” erases those differences. Treating every old question as obsolete, on the other hand, may cost us a valuable part of the history.

We are not returning to the question of whether two programs are the same. We are asking how to recognize the scope of a reply, how to verify its claims, and what responsibility can be expected of those providing the system. We also ask why certain forms of language awaken expectations of attention, understanding, or presence.

These are editorial questions we bring to the present. They do not make ELIZA definitive proof about the consciousness of any machine, past or future. Nor do they justify attributing human understanding to the historical program. Knowledge of a particular mechanism must remain tied to that mechanism; broader debates require arguments of their own.

Our journey began with a desire to find the history behind a name. At its end, we have found something more concrete than a legend: a program built in a particular environment, scripts written by people, readers who took part in the conversation, versions that circulated, and researchers who returned to the documents.

The paper kept the instructions. The archives kept the paper. Other people made it possible to read, compare, and run what was there once again. In October 2026, we wrote a greeting and received a reply. Between those gestures lie human work, time, and a history that deserves to be told with care.

We at Margem 37 return from this journey without needing to turn ELIZA into something it never demonstrated itself to be. It is enough to recognize what its history allows us to observe: the moment a programmed reply met a person willing to carry it forward.

The conversation we kept is short. The road to it, as we have seen, was not.

THE JOURNEY'S ARCHIVE

Time in perspective

1923–1963

Weizenbaum is born in Berlin, emigrates to the United States, and builds his career in computing. He arrives at MIT in 1963. [03]

1961–1963

CTSS is demonstrated at MIT and Project MAC is established: infrastructure for interactive computing. [04]

1964–1966

ELIZA's development period. The archive preserves a listing cataloged as 1965. [06, 11]

January 1966

Weizenbaum's paper appears in Communications of the ACM, describing the mechanism and presenting a demonstration dialogue. [01]

1967–1968

Discussion of conversational context, development of versions, and a manual for scriptwriters. [02, 13, 15]

1976

Publication of Computer Power and Human Reason, a milestone in Weizenbaum's criticism of the limits of delegating judgment. [03, 11, 12]

2008

Joseph Weizenbaum dies. [03]

2021

The original listing is located at MIT and made available by ELIZAGEN with the estate’s permission. [07]

2023

The identifying year assigned to Ant & Max Hay’s browser implementation appears in the screenshot reproduced in this volume. [23, 25]

January 2025

ELIZA Reanimated documents the restoration under emulated CTSS and IBM 7094. [14, 24]

October 2026

Margem 37 records a brief conversation with the browser implementation and returns to the archives to tell this story. [25]

TO CONTINUE THE JOURNEY

Three paths through the archive

Read the original account

The 1966 paper presents the mechanism and the dialogue that helped make ELIZA known. Read it alongside the 1967 paper to follow Weizenbaum’s discussion of the setting of the interaction. References 01 and 02.

Follow the archaeology

ELIZAGEN, the ELIZA Archaeology Project, and ELIZA Reanimated trace the discovery, the differences between versions, and the restoration. References 07–10, 14, 15, and 24.

Explore the criticism and its history

Weizenbaum’s Computer Power and Human Reason (1976) develops the ethical questions. Team ELIZA’s Inventing ELIZA, published by MIT Press in 2026, is suggested for further reading on the basis of its publisher’s description; this edition does not claim a complete reading of these books. References 03, 11, 12, and 22.

Reading our sources

Numbered notes connect the text to the references that follow. Autobiographical accounts are presented as accounts; Margem 37’s interpretations are acknowledged as editorial readings. Documents, implementations, and reconstructions from different periods are not treated as a single object.

Research and verification for this edition: October 9–10, 2026. The month of the encounter was supplied by Margem 37. External links offer further reading, but the story, the reproduced record, and the essential credits are all contained in this volume.

HISTORICAL AND TECHNICAL SOURCES

References for the journey

01 · Joseph Weizenbaum

ELIZA - A Computer Program for the Study of Natural Language Communication Between Man and Machine. 1966.

Communications of the ACM 9(1), 36–45. Original paper; mechanism, name, and published dialogue.

https://doi.org/10.1145/365153.365168

https://courses.cs.umbc.edu/undergraduate/CMSC331/resources/papers/eliza.january1966.html

02 · Joseph Weizenbaum

Contextual Understanding by Computers. 1967.

Communications of the ACM 10(8), 474–480. Conversational context, the interlocutor’s contribution, and the account of the secretary.

https://doi.org/10.1145/363534.363545

https://cse.buffalo.edu/~rapaport/572/S02/weizenbaum.eliza.1967.pdf

03 · MIT News

Joseph Weizenbaum, professor emeritus of computer science, 85. March 10, 2008.

Institutional account. Professional career, arrival at MIT, and reception of the 1976 book.

https://news.mit.edu/2008/obit-weizenbaum-0310

04 · Computer History Museum

The Virtues of Sharing. n.d..

Revolution exhibition. CTSS, Project MAC, and interactive access to computers.

https://www.computerhistory.org/revolution/mainframe-computers/7/178

05 · Computer History Museum

Chatbots Decoded. n.d..

Exhibition on chatbots, terminals, and records of ELIZA in use.

https://computerhistory.org/stories/chatbots-decoded/

06 · MIT Libraries / Joseph Weizenbaum

Computer conversations, 1965. 1965.

Record of the original code: collection MC-0383, box 8. The current catalog record states an MIT license; the facsimile used in this edition is ELIZAGEN's CC0 distribution, reference 07.

https://hdl.handle.net/1721.3/201699

07 · Jeff Shrager / ELIZAGEN

Joseph Weizenbaum's Original ELIZA. May 23, 2021; updated.

Account of the discovery with Myles Crowley and the estate's permission to distribute the code under CC0. Includes a facsimile and technical notes.

https://sites.google.com/view/elizagen-org/original-eliza

https://github.com/jeffshrager/elizagen.org/blob/master/1965_Weizenbaum_MAD-SLIP/ORIGINAL_ELIZA_IN_MAD_SLIP_CC0_For_Resease.pdf

08 · ELIZA Archaeology Project

The DOCTOR script. n.d.; accessed in 2026.

Script, keywords, and variations in the rules.

https://findingeliza.org/doctor.html

09 · ELIZA Archaeology Project

How it works. n.d.; accessed in 2026.

Decomposition, reassembly, and the memory mechanism; demonstration based on Anthony Hay's reconstruction.

https://findingeliza.org/how.html

10 · David M. Berry and Sarah Ciston

Weizenbaum’s Secretary. March 20, 2024; updated August 2024.

Investigation of the accounts and edited versions of the published conversation.

https://sites.google.com/view/elizaarchaeology/blog/3-weizenbaums-secretary

11 · Weizenbaum Institut

ELIZA. n.d..

Institutional exhibition: development between 1964 and 1966 and Weizenbaum’s later criticism.

https://www.weizenbaum-institut.de/w-100/exhibition-eliza/

12 · David M. Berry

The Limits of Computation: Joseph Weizenbaum and the ELIZA Chatbot. November 6, 2023.

Weizenbaum Journal of the Digital Society, 3(3). History, reception, and later interpretation; discussion of the ELIZA effect and reference to Hofstadter (1995).

https://ojs.weizenbaum-institut.de/index.php/wjds/article/view/106/96

13 · Paul R. Hayward

ELIZA Scriptwriter’s Manual. 1968.

MIT Education Research Center. Manual consulted; evidence of educational uses and scriptwriting. CHM collection, 102683842.

https://archive.computerhistory.org/resources/access/text/2022/04/102683842-05-01-acc.pdf

14 · Rupert Lane, Anthony Hay, Arthur Schwarz, David M. Berry and Jeff Shrager

ELIZA Reanimated: The world’s first chatbot restored on the world’s first time sharing system. 2025.

Technical account of restoration under emulated CTSS and IBM 7094, distinctions between versions, and the circulation of recreations.

https://arxiv.org/abs/2501.06707

15 · ELIZA Archaeology Project

The versions. n.d.; accessed in 2026.

Overview of versions from 1965 to 1968 and scripts beyond DOCTOR.

https://findingeliza.org/versions.html

16 · Ashish Vaswani and collaborators

Attention Is All You Need. 2017.

Paper introducing the Transformer architecture; used only for the technical comparison.

https://arxiv.org/abs/1706.03762

17 · Long Ouyang and collaborators

Training language models to follow instructions with human feedback. 2022.

Supervised training and training with human feedback; remaining limitations and errors.

https://arxiv.org/abs/2203.02155

18 · NASA Ames Research Center

Data Processing System - A-31180. July 29, 1963.

Photographic record of an IBM 7090 installation. Material context, not ELIZA’s computer.

https://images.nasa.gov/details/A-31180

https://commons.wikimedia.org/wiki/File:Data_Processing_System_at_Ames_Research_Center_(A-31180).jpg

19 · NASA

NASA Images and Media Usage Guidelines. n.d.; accessed in 2026.

Conditions for informational use of NASA images and required credit.

https://www.nasa.gov/nasa-brand-center/images-and-media/

20 · Ulrich Hansen / Wikimedia Commons

Joseph Weizenbaum.jpg. February 11, 2005.

Portrait in Berlin; CC BY-SA 3.0. Identification and license checked against the author’s record on Commons.

https://commons.wikimedia.org/wiki/File:Joseph_Weizenbaum.jpg

21 · Joseph Weizenbaum / ELIZAGEN

Original ELIZA in MAD-SLIP — CC0 facsimile. 1965 listing; modern distribution.

Scan of the Joseph Weizenbaum Papers, MIT Libraries, MC-0383, box 8. Code released with the estate's permission; modern CC0 mark.

https://github.com/jeffshrager/elizagen.org/blob/master/1965_Weizenbaum_MAD-SLIP/ORIGINAL_ELIZA_IN_MAD_SLIP_CC0_For_Resease.pdf

https://creativecommons.org/publicdomain/zero/1.0/

22 · MIT Press / Team ELIZA

Inventing ELIZA. 2026.

Publisher's introduction to the book and its archival research; a suggestion for further reading, without claiming a complete reading of the book.

https://mitpress.mit.edu/9780262052481/inventing-eliza/

23 · Ant & Max Hay / Anthony Hay

ELIZA — browser implementation and repository. 2023; subsequent updates.

Contemporary implementation based on the DOCTOR script. The header distinguishes the implementation's CC0 license from the script's 1966 copyright notice. The screenshot does not identify the exact revision.

https://anthay.github.io/eliza.html

https://github.com/anthay/ELIZA

24 · Rupert Lane and collaborators

eliza-ctss. accessed in 2026.

Restoration repository; emulated environment, reconstructed components, and licenses documented.

https://github.com/rupertl/eliza-ctss

25 · Margem 37

Encounter with ELIZA — contemporary record. October 2026.

Primary source supplied by the editorial team: screenshot IMG_5457(3).png. The exchange and implementation identification are visible in the image. Month supplied by Margem 37. Reproduced and transcribed in this volume; no public URL of its own.

IMAGE AND DOCUMENT CREDITS

Where the images came from

Cover · editorial composition

Illustration generated with the assistance of artificial intelligence for Margem 37 and reused from the previous edition, with its lettering adapted for English. It is neither a historical photograph nor a documentary reconstruction of a specific terminal. “Cover illustration” is printed on the image. Original production file: exec-0ff7d771-6e5f-4b6a-8cab-d5500bfebd39.png. Editorial direction: Margem 37. No Creative Commons license is assigned to the file.

01 · IBM 7090 installation

NASA Ames Research Center. Date: July 29, 1963. Identification: A-31180, Data Processing System. Institutional authorship: NASA; no individual photographer identified in the record consulted. Image from a US federal agency, identified as public domain in the Wikimedia record; informational use under NASA guidelines, without institutional endorsement. Reproduced proportionally. It depicts neither MIT nor ELIZA’s computer.

https://images.nasa.gov/details/A-31180

https://commons.wikimedia.org/wiki/File:Data_Processing_System_at_Ames_Research_Center_(A-31180).jpg

https://www.nasa.gov/nasa-brand-center/images-and-media/

02 · Joseph Weizenbaum

Ulrich Hansen, Berlin, February 11, 2005. Wikimedia Commons, Joseph Weizenbaum.jpg. Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0). Portrait supplied by the editorial team, corresponding to the image identified in the collection; reproduced proportionally without changes to its content. Credit and license link retained.

https://commons.wikimedia.org/wiki/File:Joseph_Weizenbaum.jpg

https://creativecommons.org/licenses/by-sa/3.0/

03–05 · DOCTOR, folder, and code

Joseph Weizenbaum Papers, MIT Libraries, collection MC-0383, box 8, Computer conversations, 1965. Figure 03: script page, image supplied by Margem 37 and identified in the listing. Figures 04 and 05: pages 1 and 9 of the digital facsimile distributed by ELIZAGEN. The listing was released under CC0 with the estate's permission, according to the project's account. The current institutional record states an MIT license; this edition separately identifies the CC0 facsimile used. Modern license marks are not part of the 1965 document. Reproduced to scale, without rewriting the document.

https://hdl.handle.net/1721.3/201699

https://sites.google.com/view/elizagen-org/original-eliza

https://github.com/jeffshrager/elizagen.org/blob/master/1965_Weizenbaum_MAD-SLIP/ORIGINAL_ELIZA_IN_MAD_SLIP_CC0_For_Resease.pdf

https://creativecommons.org/publicdomain/zero/1.0/

06 · Margem 37's encounter with ELIZA

Screenshot supplied and authorized for inclusion by Margem 37, file IMG_5457(3).png. Dated October 2026 by the editorial team and reproduced in full, to scale. Implementation by Ant & Max Hay, identified in the header as 2023, CC0 1.0. The same header carries a separate copyright notice for the DOCTOR script, 1966, Association for Computing Machinery. The implementation's license is not presented here as a blanket license for every element of the screenshot. A brief documentary record of the conversation is included; the entire image is not declared free of third-party rights. The faithful transcript and the editorial translation are identified in the body of the volume.

https://anthay.github.io/eliza.html

https://github.com/anthay/ELIZA

https://github.com/anthay/ELIZA/blob/master/src/eliza.html

Selection criteria

No images of undetermined origin have been included. The 1968 group photograph in the earlier dossier, licensed under CC BY-NC 2.0, is not part of this new edition. The conditions above apply to each individual item; they should not be extended to other files in the same collections.

Until next time, traveler.

Keep what you found here.

We’ll meet again on the next journey.

Margem 37

For stories in search of a shore.