

## Eliza

ELIZA: Is something troubling you ?  
YOU: whats up  
ELIZA: Please go on.  
YOU: how are you  
ELIZA: What if I were ?  
YOU: that's great to hear  
ELIZA: What does that suggest to you ?  
YOU: im a loner  
ELIZA: Do you feel strongly about discussing such things ?  
YOU: yes  
ELIZA: I see.

For a comfortable terminal-like interface see the [ELIZA-Terminal](#).

For a modern interface (featuring speech I/O) see [E.L.I.Z.A. Talking](#).

ELIZA is a natural language conversation program described by Joseph Weizenbaum in January 1966 [\[1\]](#).

It features the dialog between a human user and a computer program representing a mock Rogerian psychotherapist.

The original program was implemented on the IBM 7094 of the Project MAC time-sharing system at MIT and was written in MAD-SLIP.

This is how Joseph Weizenbaum discussed his choice for a conversation model as it would be found in psychotherapist's session:

At this writing, the only serious ELIZA scripts which exist are some which cause ELIZA to respond roughly as would certain psychotherapists (Rogerians). ELIZA performs best when its human correspondent is initially instructed to "talk" to it, via the typewriter of course, just as one would to a psychiatrist. This mode of conversation was chosen because the psychiatric interview is one of the few examples of categorized dyadic natural language communication in which one of the participating pair is free to assume the pose of knowing almost nothing of the real world. If, for example, one were to tell a psychiatrist "I went for a long boat ride" and he responded "Tell me about boats", one would not assume that he knew nothing about boats, but that he had some purpose in so directing the subsequent conversation. It is important to note that this assumption is one made by the speaker.

Whether it is realistic or not is an altogether separate question. In any case, it has a crucial

psychological utility in that it serves the speaker to maintain his sense of being heard and understood. The speaker further defends his impression (which even in real life may be illusory) by attributing to his conversational partner all sorts of background knowledge, insights and reasoning ability. But again, these are the *speaker's* contribution to the conversation.

For a conversation example given by Joseph Weizenbaum in his article see the [Eliza Test](#) page.



## About elizabot.js

"elizabot.js" is an object oriented JavaScript library for [multiple] instances of the Eliza program.

Synopsis:

```
new ElizaBot( <random-choice-disable-flag> )  
ElizaBot.prototype.transform( <inputstring> )  
ElizaBot.prototype.getInitial()  
ElizaBot.prototype.getFinal()  
ElizaBot.prototype.reset()
```

Usage:

```
var eliza = new ElizaBot();  
var initial = eliza.getInitial();  
var reply = eliza.transform(inputstring);  
if (eliza.quit) {  
    // last user input was a quit phrase  
}  
  
// method `transform()` returns a final phrase in case of a quit phrase
```

```
// but you can also get a final phrase with:
var final = eliza.getFinal();

// other methods: reset memory and internal state
eliza.reset();

// to set the internal memory size override property `memSize':
eliza.memSize = 100; // (default: 20)

// to reproduce the example conversation given by J. Weizenbaum
// initialize with the optional random-choice-disable flag
var originalEliza = new ElizaBot(true);
```

As ElizaBot is a totally self-contained object and instances use their own internal memory it's possible to have multiple instances of the ElizaBot object talking to each other.

ElizaBot is also a general chatbot engine that can be supplied with any rule set.

Data Structures (cf: "elizadata.js"):

```
elizaInitials ..... Array of initial phrases
elizaFinals ..... Array of final phrases
elizaQuits ..... Array of quit phrases
elizaPres ..... Array of alternating name value pairs for preprocessing
elizaPosts ..... Array of alternating name value pairs for postprocessing
elizaSynons ..... Object of words and their synonyms (as array)
elizaPostTransforms ... regexp/replacement pairs to be performed as final cleanings
elizaKeywords ..... Array of keywords with decompositions and reassemblies
```

elizaKeywords elements are of:

```
[ "<key>", <rank>,
  [
    [ // first decomposition pattern
      "<decomp>",
      ["<reasmb>", "<reasmb>", "<reasmb>"]
    ],
    [ // second decomposition pattern
      "<decomp>",
      ["<reasmb>", "<reasmb>", "<reasmb>"]
    ]
  ]
]
```

keywords with higher rank take precedence.

decompositions are matched in definition order.

reassemblies are chosen by random or cycled through if the no-random flag is set.

the special keyword "xnone" holds the rules for default phrases (no match).

decomposition and reassembly syntax follow the "canonical" form:

decomposition:

```
* ... any words (incl. none)
$ ... (in first position) reassemble for memory only
@synon ... substitute entry with synonym expression
```

reassembly:

```
(n) ... insert param of position n (first is "1")
      positions are any matchings of "*" or "@synon"
```

pres, posts, synonyms, keywords, decompositions all in lower case.  
all definitions are optional but at least elizaKeywords should be supplied.  
if no keywords are found `transform()` returns the default string:  
"I am at a loss for words."

Note: "elizaPostTransforms" is not a standard ELIZA feature and was included to provide a smoothing mechanism for any productions of a bot-to-bot conversation.

Data representations and syntax follow the "canonical" form.

(Since all data is transformed to and matched as regular expressions, you could also use regexps in the keywords and decompositions. Do not use any "\*" expressions in decomposition patterns as "\*" will be transformed to a regexp-pattern itself.)

The structure of "elizaKeywords" follows the internal data model as described by J. Weizenbaum in his article [\[1\]](#).

"elizabot.js" by © Norbert Landsteiner 2005;  
mass:werk – media environments <<http://www.masswerk.at>>.

#### Distribution:

"elizabot.js" is free software and provided "as is". It is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

#### Compatibility:

"elizabot.js" should be compatible to most browsers with support of RegExp (that is all standard browsers).

(Version 1.1 doesn't require the support of lambda functions in RegExps any more.)

**Download:** [elizabot.zip](#) (ZIP file, 15 KB incl. documentation)

#### References:

- [1] Weizenbaum, Joseph "ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine"  
in: Communications of the ACM; Volume 9 , Issue 1 (January 1966): p 36-45.

> [top of page](#)

N. Landsteiner 2005; <<http://www.masswerk.at>>