

EXOTIC PROGRAMMING LANGUAGES

The word “exotic” is defined by the Oxford dictionary as “of a kind not ordinarily encountered”. If that is so, what is meant by exotic programming languages? These are languages that are not used for commercial purposes or even for anything “useful”. This definition can be applied a little loosely and, hence, many different types of programming languages can be classified as “exotic”. So, after some consideration, I have included three types of languages as belonging to this category. Exotic programming languages include languages that are intentionally designed to be difficult to learn and program with. Such languages are often used to analyse the power of programming languages. They are known as esoteric programming languages. Some exotic languages, known as joke languages, are created for the sake of fun. And the third type includes non – English – based programming languages. The definition clearly mentions that these languages do not serve any practical purpose. Then what’s all the fuss about exotic programming languages? To understand their importance, we need to first understand what makes a programming language a programming language.

The “Turing completeness” of programming languages

A language qualifies as a programming language in the true sense only if it is “Turing complete” [1, c. 4]. A language that is Turing complete can be used to represent any computable algorithm. Many of the so called languages are, in fact, not languages, in the true sense. Examples of tools that are not Turing complete and hence not languages in the strict sense include HTML, XML, JSON, etc. But it is often quite surprising to come across Turing complete languages in the most unexpected places. Indeed, it was surprising for me to learn that tools like Vim, sed, AWK, etc, are Turing complete. This knowledge about Turing completeness can be used to shed some light on exotic programming languages. Esoteric programming languages are tiny programming languages that are Turing complete. In layman’s terms we can say that esoteric languages are powerful enough to do any task that is achievable by heavyweight languages like C, C++, Java, etc. Some of the joke languages and most of the non – English – based languages are also Turing complete.

The importance and relevance of exotic programming languages

Now we know that the languages we will be discussing here are Turing complete and are worthy of being called a “programming language”. What about their relevance in the open source arena? In my opinion, these languages truly capture the ethos of open source program development. They neither have any commercial applications nor will they ever allow the developer to earn even a single rupee. No profit – seeking company will ever develop any of these languages. But each and every one of these languages has a small but fervent group of supporters responsible for development and maintenance. This underlines the fact that the real value of open source development is broadening the frontiers of knowledge and not just monetary benefit.

Another question that needs an answer is regarding the uses of exotic programming languages. If these languages are not widely used then why study or write

about them? For starters, programming is not always about developing applications. Programming can also be used to test the skills of a potential candidate in an interview, to create puzzles that will baffle even the most accomplished programmer, for recreational purposes, etc. So, in essence, exotic programming languages are somewhat similar to crossword puzzles or word jumble games; they offer very good exercise for the mind. Over the years, thousands of pages have been filled about the features of different programming language paradigms like object oriented programming, structured programming, functional programming and logic programming. Also, computer magazines have published volumes about different languages like C, C++, Java, Python, etc. But none of them ever considered exotic programming languages seriously. So it will be pure fun for readers to get to know these weird languages.

Esoteric programming languages

Esoteric programming languages (also known as esolangs) are Turing complete languages in which program development is done with a tiny set of tokens. These languages are created to test the patience and perseverance of potential geeks. The sheer variety of languages available is quite amazing. In this section we will discuss just five esoteric programming languages. In fact, there are more than 750 entries in the wiki page about esoteric languages. We will discuss esolangs called Ook!, Piet and Whitespace.

Ook!

Ook! [2, c. 4] is not an independent programming language. It is a derivative language of another exotic programming language. What makes Ook! interesting is the fact that there are only three keywords — Ook., Ook?, and Ook!. These three keywords are taken in pairs to form the eight keywords of another exotic programming language. The three keywords of the language Ook! are created in such a way that they are supposed to be writable and readable even by orangutans. There are some online interpreters freely available for Ook!. The size of programs tends to become even more longer with Ook!. The following is an Ook! program to print “Hello World!” on the screen.

```
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook.  
Ook. Ook. Ook. and etc.
```

The program is too large for debugging and comprehension; hence, all we can do is hope and pray that the program gives the correct output.

Piet

Piet [3, c. 4] is an even more bizarre esolang. The programs in Piet do not use any text – based keywords; instead, Piet uses coloured blocks as keywords. The final program obtained in the Piet language will look like an abstract painting. This is the reason why it’s named after the Dutch painter Piet Mondarin, who is the pioneer of geometric abstract art. The extension of a Piet program is .ppm (Portable Pixelmap Graphics), which is an open graphics file format. The most widely used IDE for Piet is the open source Piet Creator. Here’s how “Hello World!” would look like:



Whitespace

In almost every programming language, whitespace characters are ignored by the compilers and interpreters or have minor implications. But Whitespace [4, c. 4] is an exception to this rule. It is an esoteric programming language in which the whitespace characters alone are used to generate the program. Every non – whitespace character used in the program is considered as a comment and ignored by the compiler. This language can never be included in the academic curriculum because the answer script containing the correct Whitespace programs will look exactly like the one without any answer – it will be blank! The programs are generated by using spaces, tabs and line feeds alone. The five commands in the language Whitespace are described below. [Space] is used for stack manipulation. The combination [Tab][Space] is used for arithmetic operation. The combination [Tab][Tab] is used for heap access. [Linefeed] is used for flow control, and the combination [Tab][Linefeed] is used for I/O manipulation. Since Whitespace ignores all the non – whitespace characters, programs can be easily embedded inside other free form languages like C. Thus Whitespace can be used to generate *polyglot programs*, which are programs written in a valid form of multiple programming languages.

Joke languages

Joke languages are those exotic languages that do not serve any practical purpose other than the potential fun that can be had by dabbling with them. They are mostly generated from existing esoteric programming languages and not usable even in theoretical scenarios. Sometimes, joke languages are created by ingenious programmers to win certain contests. There are about hundred entries in the wiki page about joke languages with languages even named ###, ***, *COW*, *FIFTH*, etc. Many of these joke languages are not Turing complete and are not programming languages in the true sense. Joke languages include those used to generate programs that will serve just a single purpose. For example, a joke language called 99 is used only to generate the lyrics of the song “99 bottles of beer”. Joke languages also include languages like Malbolge [5, c. 4], often considered the most difficult language ever invented to write an executable program with.

This precise has discussed some of the weirdest programming languages that have ever been created by humans. On the surface, it may seem that these languages are purposeless and a waste of time. But, in reality, they are useful in many different ways. They expand our knowledge of how programming languages work, they offer challenging puzzles to budding programmers and they also represent the real spirit of open source development.

REFERENCES

1. Turing completeness – https://en.wikipedia.org/wiki/Turing_completeness.
2. Ook! – <https://esolangs.org/wiki/ook!>.
3. Piet – <https://esolangs.org/wiki/piet>.
4. Whitespace – <https://esolangs.org/wiki/whitespace>.
5. Malbolge – <https://esolangs.org/wiki/malbolge>.