(almost) Every Programming Language in < 4 minutes

C: The FSF Special

How it works:	The compiler converts code directly into machine code for the type of CPU you're using.
What you need:	Nothing except a CPU you can compile for.
What's Good:	It's really fast, great for networking, and runs anywhere.
What's Not:	Unwieldy for large projects; impractical for scripting.

C++: Access Friends' Privates

How it works:	Compiles natively to machine code.
What you need:	No requirements, runs on any compatible CPU
What's Good:	Fast large projects where you need to render on-the-fly.
What's Not:	It's hard to read.

C#: The Language of MS Paint

How it works:	Your code either gets compiled into a not-quite-machine code instruction set that's portable, or it can be compiled natively for Windows.
What you need:	Windows, with mediocre support for Mac and Linux.
What's Good:	Servers and database applications on Windows.
What's Not:	FOSS implementations, use on Linux.

Java: Valgrind's Archenemy

How it works: Your code is transformed into 'bytecode' that's ran by the Java Virtual Machine (JVM).

What you need: An OS and the JVM.

What's Good: Reflection heavy applications, high readability, dynamic code loading.

What's Not: It's not very fast.

Kotlin: Java but with more features

How it works:	Runs on the JVM, compiles to JavaScript, or compiles to machine code.
What you need:	Varies (listen to Jack)
What's Good:	JVM-based scripts. It's exremely portable.
What's Not:	It's tough to understand big programs.

Go: Yes, That's its Logo

How it works:	Like C, but it cleans up after you with garbage collection.
What you need:	Google compiler toolchain, or compile to bare metal.
What's Good:	It's a little like C.
What's Not:	Pretty much everything else.

Lua: Modding and what else?

How it works:	It's interpreted, just like JavaScript!
What you need:	The Lua Interpreter.
What's Good:	Scripting for video games; highly controllable libraries.
What's Not:	Trying to write applications kinda sucks.

JavaScript: More Libraries than Illinois

How it works:	Interpreted on-the-fly by your browser sandbox/ Node server.
What you need:	A web browser.
What's Good:	Scripting for web apps and pages.
What's Not:	Using it for anything that's not a web page.

$

Unix Shell: Can you hear the C?

How it works:

What you need:

What's Good:

What's Not:

Works Cited

Monty, Julien, and Konpa. “Devicon 2.0.” *DEVICON | All Programming Languages and Development Tools Related Icons Font*, GitHub, konpa.github.io/devicon/. Icons for programming languages and platforms used with permission under the MIT License