·) Foreign function Interface → refers to the ability for code written in one language (the host language) to access and invoke functions written in another language (the guest language). Here the term foreign refers to the fact that the functions come from another language and environment.

↓

Depending on the language and its FFI support, we might also be able to call global named variables, automatically convert data types between the host and the guest, and have code in guest language invoke functions in the host language as callbacks.

↓

FFI access the library's binary code directly, without compiling any code, hence in interpreted languages like Ruby, it's usually not possible to use a library's compile-time features like C preprocessor macros and constants.

↓

But the FFI support in some compiled languages works by compiling down to C code, and here compile-time features can be accessed.

↓

piece of executable code that is passed as an argument to other code. The other code is expected to call back the passed piece of code at some time.

·) Benefits :