In the context of software libraries, bindings are wrapper libraries that bridge two programming languages, so that a library written for one language can be used in another language.

↓

Many software libraries are written in system programming languages such as C or C++. To use such libraries from another language, usually of higher level, a binding to the library must be created in that language, possibly requiring recompiling the language's code, depending on the amount of modification needed. However most languages offer a foreign function interface, such as Python's and OCaml's ctypes, and Embeddable Common Lisp's cffi and uffi.

↓

This can be thought of as that calling function A using Python triggers the corresponding function in C/C++? Can be thought of as a mapping system between the two languages for the same language. In other words, binding can be understood as a mechanism of using the functionality of a framework from another language, then the one it provides support for or is currently written in.