function-load

- [Examples](#)

Load a library to Valkey.

The command receives a single mandatory parameter which is the source code that implements the library. The library payload must start with Shebang statement that provides a metadata about the library (like the engine to use and the library name). Shebang format: `#!<engine name> name=<library name>`. Currently engine name must be `lua`.

For the Lua engine, the implementation should declare one or more entry points to the library with the `server.register_function()` API. Once loaded, you can call the functions in the library with the `FCALL` (or `FCALL_RO` when applicable) command.

When attempting to load a library with a name that already exists, the Valkey server returns an error. The `REPLACE` modifier changes this behavior and overwrites the existing library with the new contents.

The command will return an error in the following circumstances:

- An invalid *engine-name* was provided.
- The library's name already exists without the `REPLACE` modifier.
- A function in the library is created with a name that already exists in another library (even when `REPLACE` is specified).
- The engine failed in creating the library's functions (due to a compilation error, for example).
- No functions were declared by the library.

For more information please refer to Introduction to Valkey Functions.

Examples

The following example will create a library named `mylib` with a single function, `myfunc`, that returns the first argument it gets.

```
127.0.0.1:6379> FUNCTION LOAD "#!lua name=mylib \n server.register_function('myfunc',
function(keys, args) return args[1] end)"
mylib
127.0.0.1:6379> FCALL myfunc 0 hello
"hello"
```

□□□ #1
□ □□□ □□□ 2026-10-31 15:13:42 UTC
□ □□□ □□□ 2026-10-31 15:13:42 UTC