return

`return` returns program control to the calling module. Execution resumes at the expression following the called module's invocation.

If called from within a function, the `return` statement immediately ends execution of the current function, and returns its argument as the value of the function call. `return` also ends the execution of an `eval` statement or script file.

If called from the global scope, then execution of the current script file is ended. If the current script file was `include`d or `require`d, then control is passed back to the calling file. Furthermore, if the current script file was `include`d, then the value given to `return` will be returned as the value of the `include` call. If `return` is called from within the main script file, then script execution ends. If the current script file was named by the auto_prepend_file or auto_append_file configuration options in ENTITY, then that script file's execution is ended.

For more information, see Returning values.

> [!NOTE] Note that since `return` is a language construct and not a function, the parentheses surrounding its argument are not required and their use is discouraged.

> [!NOTE] If no parameter is supplied, then the parentheses must be omitted and ENTITY will be returned. Calling `return` with parentheses but with no arguments will result in a parse error.

As of PHP 7.1.0, return statements without an argument in functions which declare a return type trigger `E_COMPILE_ERROR`, unless the return type is `void`, in which case return statements with an argument trigger that error.

□□□ #2
□ □□□ □□□ 2026-10-31 17:39:08 UTC
□ □□□ □□□ 2026-10-31 17:39:09 UTC