Git ignore patterns

.gitignore uses globbing patterns to match against file names. You can construct your patterns using various symbols:

Pattern	Example matches	Explanation*
**/logs	logs/debug.log logs/monday/foo.bar build/logs/debug.log	You can prepend a pattern with a double asterisk to match directories anywhere in the repository.
**/logs/debug.log	logs/debug.log build/logs/debug.log *but not* logs/build/debug.log	You can also use a double asterisk to match files based on their name and the name of their parent directory.
*.log	debug.log foo.log .log logs/debug.log	An asterisk is a wildcard that matches zero or more characters.
*.log !important.log	debug.log trace.log *but not* important.log logs/important.log	Prepending an exclamation mark to a pattern negates it. If a file matches a pattern, but *also* matches a negating pattern defined later in the file, it will not be ignored.
.log !important/.log trace.*	debug.log important/trace.log *but not* important/debug.log	Patterns defined after a negating pattern will re-ignore any previously negated files.
/debug.log	debug.log *but not* logs/debug.log	Prepending a slash matches files only in the repository root.
debug.log	debug.log logs/debug.log	By default, patterns match files in any directory
debug?.log	debug0.log debugg.log *but not* debug10.log	A question mark matches exactly one character.
debug[0-9].log	debug0.log debug1.log *but not* debug10.log	Square brackets can also be used to match a single character from a specified range.

debug[01].log	debug0.log debug1.log *but not* debug2.log debug01.log	Square brackets match a single character form the specified set.
debug[!01].log	debug2.log *but not* debug0.log debug1.log debug01.log	An exclamation mark can be used to match any character except one from the specified set.
debug[a-z].log	debuga.log debugb.log *but not* debug1.log	Ranges can be numeric or alphabetic.
logs	logs logs/debug.log logs/latest/foo.bar build/logs build/logs/debug.log	If you don't append a slash, the pattern will match both files and the contents of directories with that name. In the example matches on the left, both directories and files named *logs* are ignored
logs/	logs/debug.log logs/latest/foo.bar build/logs/foo.bar build/logs/latest/debug.log	Appending a slash indicates the pattern is a directory. The entire contents of any directory in the repository matching that name – including all of its files and subdirectories – will be ignored
logs/ !logs/important.log	logs/debug.log logs/important.log	Wait a minute! Shouldn't logs/important.log be negated in the example on the left Nope! Due to a performance-related quirk in Git, you *can not* negate a file that is ignored due to a pattern matching a directory
logs/**/debug.log	logs/debug.log logs/monday/debug.log logs/monday/pm/debug.log	A double asterisk matches zero or more directories.

logs/*day/debug.log	logs/monday/debug.log logs/tuesday/debug.log *but not* logs/latest/debug.log	Wildcards can be used in directory names as well.
logs/debug.log	logs/debug.log *but not* debug.log build/logs/debug.log	Patterns specifying a file in a particular directory are relative to the repository root. (You can prepend a slash if you like, but it doesn't do anything special.)