Showing all hidden files in the current directory

Let's say that you want to see only hidden (dot) files in a directory to edit a file you forget the name of or remove obsolete files. ls -a shows all files, including normally hidden ones, but that is often too noisy, and ls -a .* doesn’t do what you think it will.

Use ls -d along with whatever other criteria you have.

```
ls -d .*
ls -d .b*
ls -d .[!.]*
```

Or construct your wildcard in such a way that . and .. don’t match.

```
$ grep -l 'PATH' ~/.[!.]*
/home/jp/.bash_history
/home/jp/.bash_profile
```

Due to the way the bash shell handles file wildcards, the sequence .* does not behave as you might expect or desire. The way filename expansion or globbing works is that any string containing the characters *, ?, or [is treated as a pattern, and replaced by an alphabetically sorted list of file names matching the pattern. * matches any string, including the null string, while ? matches any single character. Characters enclosed in [ ] specify a list or range of characters, any of which will match. There are also various extended pattern-matching operators to achieve the same.