

---

**Scientific Computing 1**  
**Handout 1**  
**October 14, 2014**

---

**Special Characters and their Meanings in `bash`**

*	serves as a placeholder for arbitrarily many characters
?	a placeholder for a single character
/	directory separator
\	escape character for quoting special characters and to mark linebreaks
~	abbreviation for your home directory
	the pipe operator: connects two simple commands to a new one by redirecting the output of the one on the left to the other one on the right. <code>  </code> represents a logic OR.
<	fetches the input for a command (on the left) from a file or device (on the right)
>	redirects the output of a command (on the left) to a file or device (on the right)
2>	same as above for the error output only, can be used to redirect the standard error messages to standard output so it is recognized by the <code>&gt;</code> and <code> </code> as well via <code>2&gt;&amp;1</code>
1>	same as above for the standard output without the errors
>>	as <code>&gt;</code> but appends the output instead of overwriting the file
\$	used in command substitution and for referring to shell and environment variables
&	a single <code>&amp;</code> after a command name sends the execution to the background. Double <code>&amp;&amp;</code> stand for the logic AND.
`	accent grave is used for command substitution
'	single quotes removes the special meaning of all special characters enclosed by them.
"	double quotes act the same as single quotes with the exception of the <code>\$</code> , <code>`</code> , <code>\</code> (and sometimes <code>!</code> ) characters keeping their special properties.
blank	the simple blank is used to separate words and thus needs to be escaped when , e.g., a file name contains it.
#	comment character; everything following this character on the same line will be dropped