There are two mechanisms for substituting the result of an operation into a command:

```
$ ls -l ‘which --skip-alias emacs‘
$ ls -l $(which --skip-alias emacs)
```

The second form permits nesting, while the first form does not. Note that the first form has “backticks” (‘) not apostrophes.

Arithmetic expressions may be evaluated in two different ways, using the **expr** utility, or the **$((..))** syntax:

For **x=3**:

Arithmetic Expression Evaluation Forms

Expression	Gives
echo $x + 1	3+1
echo $(expr $x + 1)	4
echo $((x+1))	4
echo $(($x + 1))	4
echo $(expr $x+1)	3+1

The **$((..))** syntax is more modern and preferred; **expr** is less efficient, as it invokes an external program and is trickier to use.

Note that **$var**, **$(cmd)**, **‘cmd‘**, and **$((...))** all expand inside double quotes.

Mark as completed