

# Bash-like Customizable Prompt in Korn Shell

**Finnbarr P. Murphy**

(fpm@fpmurphy.com)

Bash has built-in support for extensive PS1 prompt customization using parameterless macros. As a result many people customize their shell prompts. There is no equivalent built-in support for PS1 customization in *ksh93* but such support can easily be added using a discipline function. This post provides an example of such a PS1 discipline function.

Add the following discipline function to your `~/.kshrc` and ensure that `~/.kshrc` is included in your `~/.profile` shell startup script which, by the way, only gets executed at login if *ksh93* is your default shell.

```
function PS1.set
{
    typeset prefix remaining=${sh.value} var= n= k=
    set -A .sh.lversion ${sh.version}
    while [[ $remaining ]]
    do
        prefix=${remaining%'\*'}
        remaining=${remaining#$prefix}
        var+="$prefix"
        case ${remaining:1:1} in
            A) var+="\$(printf '%(%R)T')";;
            @) var+="\$(printf '%(%H:%M %p)T')";;
            d) var+="\$(printf '%(%a %b:%d)T')";;
            e) var+="\$('e)";;
            h) var+=$(hostname -s);;
            H) var+=$(hostname);;
            j) var+="\$(jobs | wc -l)";;
            l) var+="\$(basename \"\$(tty)\")";;
            n) var+='\n';;
            r) var+='\r';;
            s) var+="\$(basename \"\${0}\")";;
            t) var+="\$(printf '%(%H:%M:%S)T')";;
            T) var+="\$(printf '%(%I:%M:%S)T')";;
            u) var+=USER;;
            v) var+="\${.sh.lversion[2]}";;
            V) var+="\${.sh.lversion[2]} (\${.sh.lversion[1]}";;
            w) var+="\$(pwd)";;
            W) var+="\$(basename \"\$(pwd)\")";;
            '#' ) var+=!;;
            !) var+=!;;
            '$' ) if (( $(id -u) == 0 ))
                then
                    var+= '#'
                else
                    var+= '$'
                fi;;
            '\ ' ) var+= '\ ';;
            '[' | '|' ) ;;
            [0-7] ) case ${remaining:1:3} in
                [0-7][0-7][0-7]) k=4;;
                [0-7][0-7]) k=3;;
                *) k=2;;
            esac
        eval n="\${remaining:0:k}"
    done
}
```

```
        var+=$n
        remaining=${remaining:k}
        continue ;;
    "")    ;;
    *)    var+=\'\'${remaining:0:2};;
esac
remaining=${remaining:2}
done
.sh.value=$var
}
```

Dave Korn originally published the guts of this discipline function some 11 years ago. I have merely extended the code to include all the Bash PS1 prompt parameterless macros that I am aware of.

Here is a list of the currently supported parameterless macros:

```
\d : the date in "Weekday Month Date" format (e.g., "Tue May 26")
\e : an ASCII escape character (033)
\h : the hostname up to the first '.'
\H : the hostname
\j : the number of jobs currently managed by the shell
\l : the basename of the shell's terminal device name
\n : a newline
\r : a carriage return
\s : the name of the shell (basename of $0)
\t : the current time in 24-hour HH:MM:SS format
\T : the current time in 12-hour HH:MM:SS format
\@ : the current time in 12-hour am/pm format
\A : the current time in 24-hour HH:MM format
\u : the username of the current user
\v : the version of ksh93
\V : the version of ksh93 and supported options
\w : the current working directory, with $HOME abbreviated with a tilde
\W : the basename of the current working directory, with $HOME abbreviated with a tilde
\! : the history number of this command
\# : the command number of this command
\$$ : if the effective UID is 0, #, otherwise $
\nnn : the character corresponding to the octal number nnn
\\ : a backslash
\[ : begin a sequence of non-printing characters
\] : end a sequence of non-printing characters
```

Here is a screenshot showing some examples of how to customize the *ksh93* PS1 prompt using the above PS1 discipline function:

```
[fpm@centos ~]$ ksh
$ set -o vi
$ PS1="\v $ "
93u+ $ PS1="\V $ "
93u+ (AJM) $ PS1="\@ $ "
01:48 AM $ PS1="\A \u $ "
01:49 fpm $ PS1="\! \u $ "
430 fpm $ PS1="\! \u $ "
431 fpm $ PS1="\u \$ "
fpm $ PS1="\w \$ "
/home/fpm $ PS1="\h \$ "
centos $ PS1="\H \$ "
centos.local $ PS1="\e[31m \$ "
$ █
```

Note that discipline functions are a feature of later versions of *ksh93* and are NOT supported in the original Korn Shell (*ksh88*) or clones of the Korn Shell such as *pdksh*.

Enjoy!