

Quotation Marks in Shell Scripts

Group 14

UM-SJTU Joint Institute

Baole Fang
Caiwei Chen
Yizi Liu
Ying Huai

September 23, 2021

Introduction

Cases:

- echo Today is (\$date)

```
Today is Thu Sep 23 07:56:04 CST 2021
```

- echo "This is a quoted sentence"

```
This is a quoted sentence
```

- name=hello world

```
Command 'world' not found
```

```
name="hello world"
```

```
echo $name
```

```
hello world
```

Problem

How to disable the meaning of a special character or the white space?

Single Quote

Enclosing characters in single quotes preserves the literal value of each character within the quotes. A single quote may not occur between single quotes, even when preceded by a backslash.

Cases:

- `echo '$name'`

```
$name
```

- `echo `date``

```
date
```

Double Quote

Enclosing characters in double quotes preserves the literal value of all characters within the quotes, with the exception of \$, `, \.

The characters \$ and ` retain their special meaning within double quotes.

- Shell parameter expansion: $\${parameter}$
 - Case: `string=1234567 echo "${string}"`
`1234567`
- Command substitution: $\$(command)$ or ``command``
 - Case: `echo "`date`"`
`Thu Sep 23 11:47:31 CST 2021`
- Arithmetic expansion: $\$((expression))$
 - Case: `echo "$((12345 + 6789))"`
`19134`

Appending special characters like \$, `, ', " and \ to the backslash will retain their literal values.

- Hybrid output:

echo "Command \\$SHELL will give the result \$SHELL"

```
Command $SHELL will give the result /bin/bash
```

Example 1

Shell

```
$ test=10  
$ echo "$test"  
$ echo '$test'  
  
10  
$test
```

Example 2

Shell

```
$ array=(10)
$ echo "$array[0]"
$ echo '$array[0] '

10
$array[0]
```

Example 3

Shell

```
$ a=10  
$ echo "'$a'"  
$ echo '"$a"'
```



```
'10'  
"$a"
```


Example 4

Shell

```
$ printf "a\nb"  
$ printf "a\\nb"  
$ printf 'a\nb'  
$ printf 'a\\nb'
```

a

b

a

b

a

b

a\nb

Thank you!