

String Manipulation ????

????

```
my_string="abhishek"  
echo "length is ${#my_string}"
```

Using expr

```
str="my string"  
length=$(expr length "$str")  
echo "Length of my string is $length"
```

Using awk

```
echo "my string" | awk '{print length}'
```

Using wc

```
str="my string"  
length=$(echo -n "my string" | wc -m)  
echo "Length of my string is $length"
```

????

Using wildcards

```
#!/bin/bash  
  
STR='GNU/Linux is an operating system'  
SUB='Linux'  
if [[ "$STR" == *"$SUB"* ]]; then  
    echo "It's there."  
fi
```

Using case

```
#!/bin/bash
```

```
STR='GNU/Linux is an operating system'
```

```
SUB='Linux'
```

```
case $STR in
```

```
  *"$SUB"*)
```

```
    echo -n "It's there."
```

```
    ;;
```

```
esac
```

Using Regex

```
#!/bin/bash
```

```
STR='GNU/Linux is an operating system'
```

```
SUB='Linux'
```

```
if [[ "$STR" =~ .*"$SUB".* ]]; then
```

```
  echo "It's there."
```

```
fi
```

Using Grep

```
#!/bin/bash
```

```
STR='GNU/Linux is an operating system'
```

```
SUB='Linux'
```

```
if grep -q "$SUB" <<< "$STR"; then
```

```
  echo "It's there"
```

```
fi
```

????

```
str1="hand"
```

```
str2="book"
```

```
str3=$str1$str2
```

????

```
foss="Fedora is a free operating system"
echo ${foss:0:6}
```

```
_admin_ip="202.54.1.33|MUM_VPN_GATEWAY 23.1.2.3|DEL_VPN_GATEWAY 13.1.2.3|SG_VPN_GATEWAY"
for e in $_admin_ip
do
    ufw allow from "${e%*|*}" to any port 22 proto tcp comment 'Open SSH port for ${e##*|}'
done
```

????

```
foss="Fedora is a free operating system"
echo ${foss/Fedora/Ubuntu}
```

Summary: String Manipulation and Expanding Variables

<code>\${parameter:-defaultValue}</code>	Get default shell variables value
<code>\${parameter:=defaultValue}</code>	Set default shell variables value
<code>\${parameter:? "Error Message"}</code>	Display an error message if parameter is not set
<code>\${#var}</code>	Find the length of the string
<code>\${var%pattern}</code>	Remove from shortest rear (end) pattern
<code>\${var%%pattern}</code>	Remove from longest rear (end) pattern
<code>\${var:num1:num2}</code>	Substring
<code>\${var#pattern}</code>	Remove from shortest front pattern
<code>\${var##pattern}</code>	Remove from longest front pattern
<code>\${var/pattern/string}</code>	Find and replace (only replace first occurrence)
<code>\${var//pattern/string}</code> <code>echo "\${PATH//:/\$'\n'}"</code>	Find and replace all occurrences
<code>\${!prefix*}</code>	Expands to the names of variables whose names begin with prefix.
<code>\${var,}</code> <code>\${var,pattern}</code>	Convert first character to lowercase.
<code>\${var,,}</code> <code>\${var,,pattern}</code>	Convert all characters to lowercase.
<code>\${var^}</code> <code>\${var^pattern}</code>	Convert first character to uppercase.

```
{var^}  
{var^pattern}
```

Convert all character to uppercase.

Cheatsheet: String Manipulation

[bash_string.jpeg](#) bash_string.jpeg unknown

String To Integer

```
# ${string}  
sum=$((3+6))  
echo $sum  
  
a=11  
b=3  
c=$((a+b))  
echo $c
```

Alternate method: `expr`

“ Note that it is not a “native” Bash procedure, as you need to have `coreutils` installed (by default on Ubuntu) as a separate package.

```
a=5; b=3; c=2; d=1  
expr $a + $b \* $c - $d
```

????????? .XXX

```
# Operator "#" means "delete from the left, to the first case of what follows."  
x="This is my test string."  
echo ${x#* }
```

is my test string.

```
# Operator "##" means "delete from the left, to the last case of what follows."  
x="This is my test string."  
echo ${x##* }
```

string.

Operator "%" means "delete from the right, to the first case of what follows."

```
x="This is my test string."
```

```
echo ${x% *}
```

This is my test

Operator "%%" means "delete from the right, to the last case of what follows."

```
x="This is my test string."
```

```
$ echo ${x%% *}
```

This

Revision #15

Created 26 August 2020 11:52:57 by Admin

Updated 14 February 2023 15:44:56 by Admin