

Bash Shell Read

Bash Shell provides `read` command that enables a developer to read input from the user. You can also use `read` command to read from the file as well.

Syntax of Bash Shell Read Command

```
read [options] variable
```

Here, the entered data is stored in a variable.

Command line options of Bash Shell Read Command

Options	Description
-p "Message"	It is used to display message and shows prompt just after that message. You can see its usage here- How to read input from user in Bash Shell?
-s	This option prevents the entered character from being displayed on the screen. Used for inputting password from the user. Example of this option can be seen here- How to prompt for password in Bash Shell?
-t NumberOfSeconds	Here, we mention the number of seconds for the read command to wait for input. Example <code>read -t 5 firstnumber</code>
-n NumberOfCharacters	It limits the user from entering specified length characters. Example <code>read -n 2 choice</code>

How to limit user from entering characters of specified length in Bash Shell?

Use `-n` option and specify the number of characters that you want to read from the user. Below example will show its usage-

```
#Bash Shell script to check the usage of -n option
read -n 1 -p "Do you want to continue(Y/N): " choice
case $choice in
Y | y)
    echo
    echo "Go on";;
N | n)
    echo
    echo "Goodbye"
    exit;;
esac
```

Output of the above program

```
Do you want to continue(Y/N): Y
Go on
```

Explanation of the above code-

```
read -n 1 -p "Do you want to continue(Y/N): " choice
```

We want user to enter Y or N and does not want user to enter complete Yes or No. To achieve this, we have specified 1 to `-n` option.

How to set the timer for inputting characters in Bash Shell?

You might face a situation in which your script is waiting for the user to enter data. And you want your script to go one then in that case, use `-t` option to mention the number of seconds for the read command to wait for input.

```
#Bash Shell script to check the usage of -t option
read -t 5 -p "Enter your name: " name
if [ $name ];
then
    echo "Hello $name, How are you?"
else
    echo "You are too slow!"
fi
```

Output of the above program

```
Enter your name: Govind
Hello Govind, How are you?
```

```
Enter your name:
You are too slow!
```

Explanation of the above code-

```
read -t 5 -p "Enter your name: " name
```

Here, we want our script to wait for 5 seconds. If the user does not enter his name within 5 seconds then script execution will go to the next line.