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

Shell script

var, echo, parameter expansion

```
a="abc" b=1 c=2 echo $a, $b echo $((b+c)) local a=1 echo "${name}'s
study" #replace echo "${name//abc/xyz}" #replace abc in name with
xyz #default value name:="abc"
```

Bash

function

```
f1(){ local a=1 echo $a } f1 f2(){ a=$1 b=$2 local sum=$((a+b)) echo
$sum #return } n1=1 n2=2 sum=$(f2 n1 n2)
```

Bash

stdin

```
read -p "what is your name?" name # -p i/p promt echo $name read -sp
"enter secret code:" secret IFS="," #delimiter #i/p=1,2,3 read -p
"enter numbers" n1 n2
```

Bash

command line args

```
# $1-> echo "1st arg: $1" echo "2nd arg: $2" # $# -> number of args
sum=0 while [[ $# -gt 0 ]]; do num=$1 sum=$((sum+num)) shift done
```

Bash

conditionals

```
if [ $age -ge 16 ] # -ge -> >= then echo "1" elif [ $age -eq 15] # -
eq -> == then echo "2" else echo "3" fi if [ "$str1" ]; then echo
"$str1 is not null" fi if [ -z "$str1" ]; then echo "$str1 has no
value" fi # -eq -> == # -ge -> >= # -le -> <= # -ne -> != # -lt -> <
# -gt -> > # ((a == 10)) # ((a >= 10)) # (( ((a % 2)) == 0)) # ((
((a > 2)) && ((a < 10)) )) # ||, ! # [ "$str1" == "$str2" ] # [
"$str1" != "$str2" ] # [ "$str1" > "$str2" ]
```

Bash

files

```
file1="./test_file1" if [ -e "$file1" ]; then echo "file exists" fi
# -e -> file exists # -f -> is normal file # -r -> is file readable
# -w -> is file writable # -x -> is file executable # -d -> is dir #
-L -> is symbolic link # -p -> is named pipe # -S -> is n/w socket #
-G -> is owned by group # -O -> is owned by user while read a b c;
do echo $a, $b, $c done > file1.txt #file1.txt #1 2 3 #4 5 6 while
read name empid dept do echo "$name is part of $dept department"
done < ~/employees.txt # $ cat employees.txt # Emma
Thomas:100:Marketing # Alex Jason:200:Sales
```

Bash

regex

```
read -p "enter date:" date pattern="^[0-9]{8}$" if [[ $date =~
$pattern ]]; then echo "valid" fi
```

Bash

switch case

```
case $age in [0-4]) echo "age b/w 0-8" ;; #break 5) echo "age is 5"
;; #break [6-9]|1[0-8]) echo "age b/w 6-9 pipe 10-18 i.e. 6-18" ;;
#break *) echo "everything else" ;; esac
```

Bash

ternary operators

```bash
(( age > = 18 ? (can_vote=1) : (can_vote=0) ))
```

Bash

string

```bash
#length ${#str1} #slice str="abcdefg" ${str:2} #"cdefg" ${str:2:5}
#"cdef" ${str#*ab} #"cdefg" -> everything after substr "ab"
```

Bash

loop

```bash
num=1 while [ $num -le 10 ]; do echo $num num++ done #continue,
break until [ $num -gt 10 ]; do echo $num num++ done for (( i=0; i
<= 10; i=i+1 )); do echo $i done for i in {A..Z}; do echo $i done
for i in ${arr[*]}; do echo $i done for i in ${arr[@]}; do echo $i
done
```

Bash

array

```bash
arr=(1 2 3) echo ${arr[1]} arr[3]=4 arr1=(9 8 7) arr+=arr1 #sort
array sorted_arr=$(for i in "${arr[@]}"; do echo $i; done | sort))
```

Bash

debugging

```bash
set -xv
```

Bash