OSC

Shell Scripting

Agenda

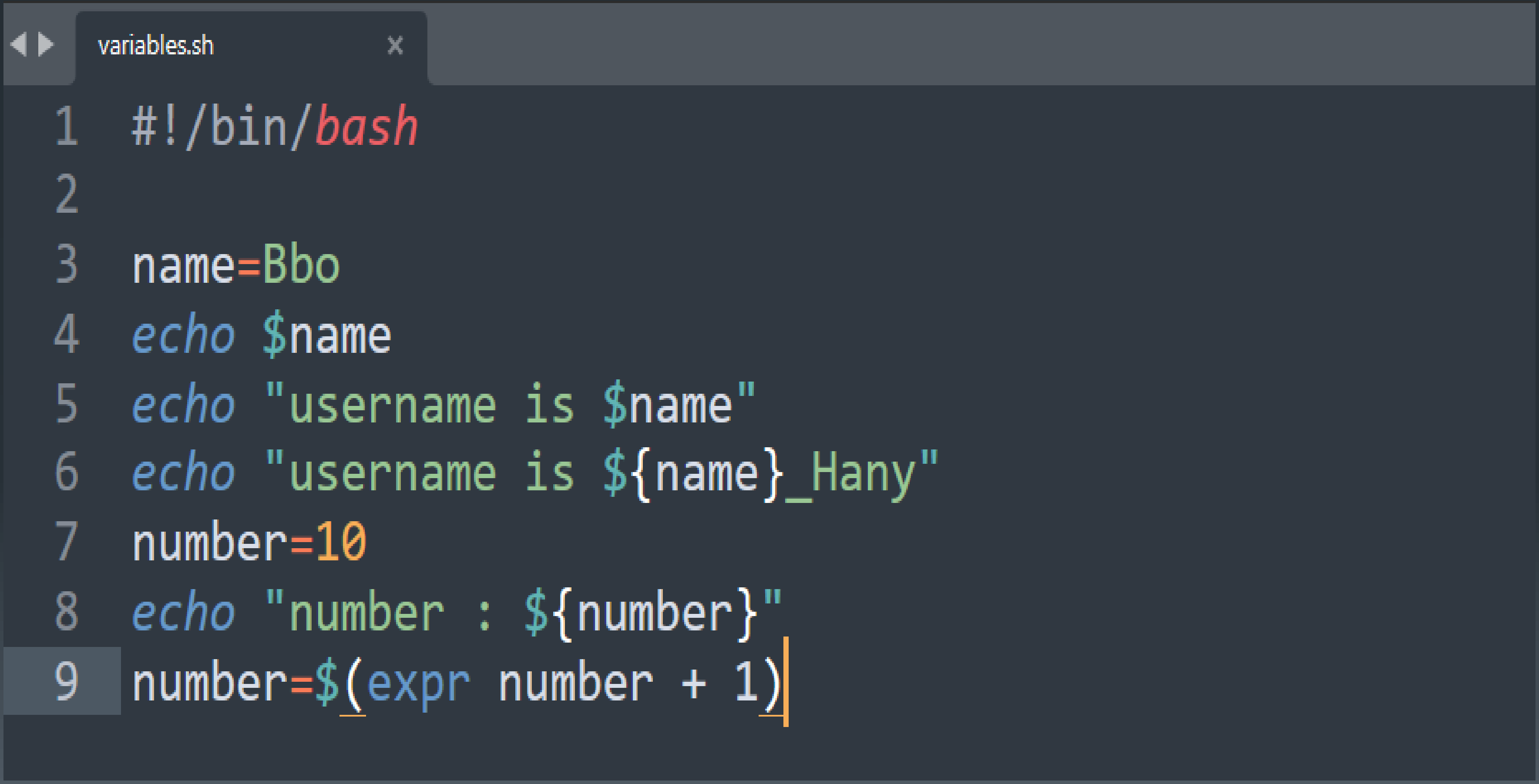
- What are shell scripts
- Variables
- Conditions
- Case
- For loop
- While loop
- File
- Control
- Functions
- Demo Projects

What are shell scripts

A shell script is a text file that contains a sequence of commands.

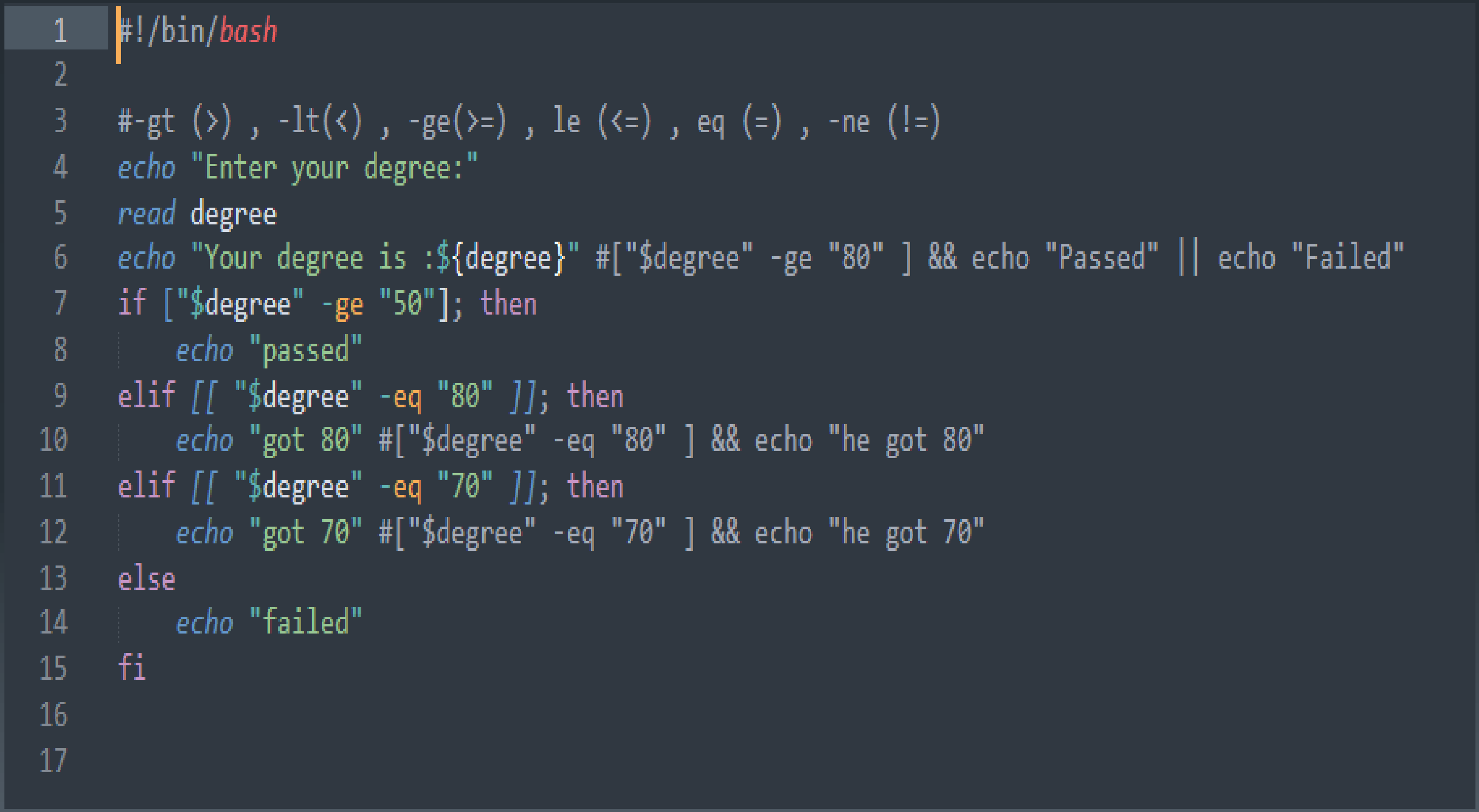
It is called a shell script because it combines a sequence of commands, that would otherwise have to be typed into the keyboard one at a time, into a single script

Variables



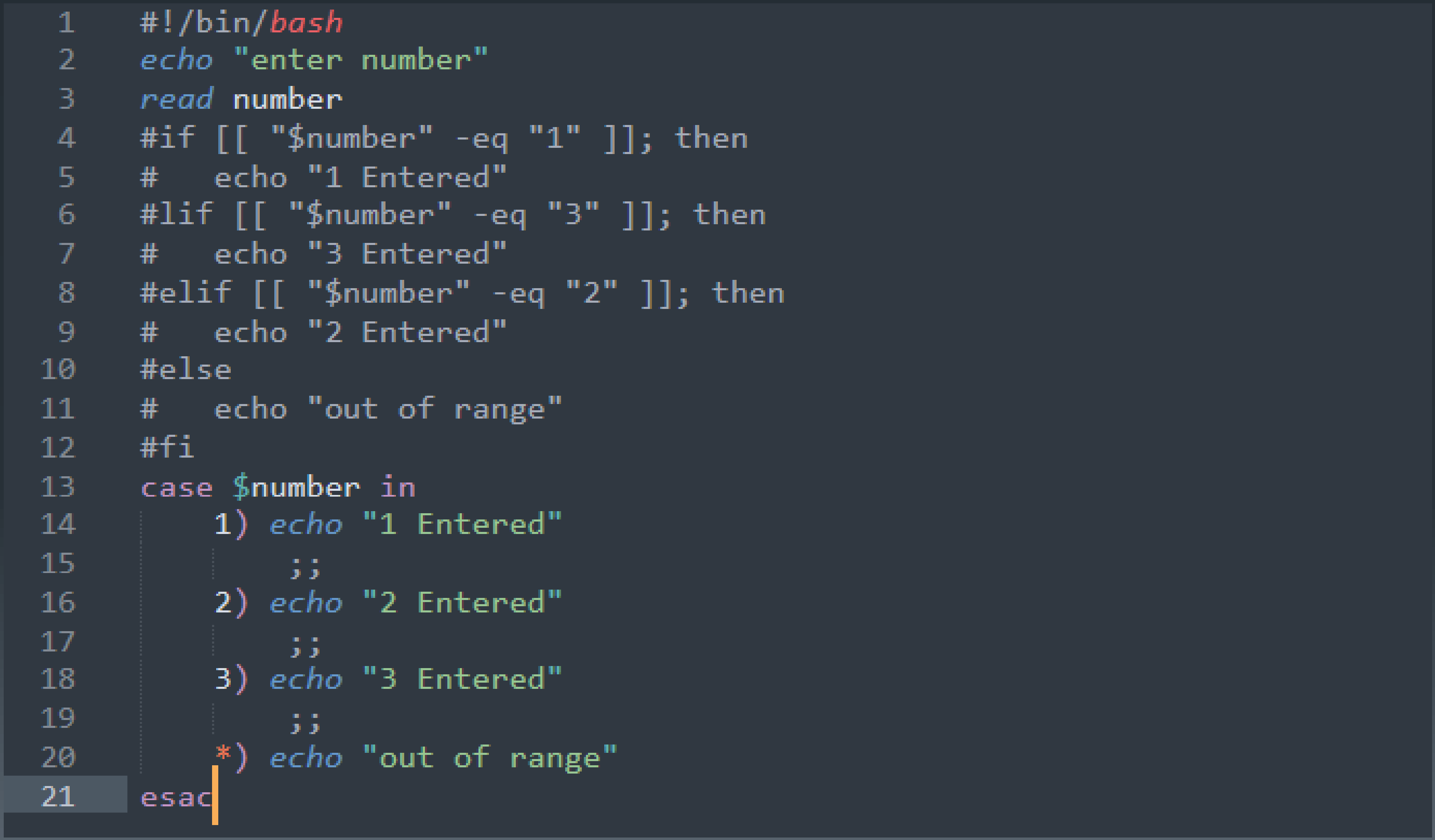
```bash
#!/bin/bash

name=Bbo
echo $name
echo "username is $name"
echo "username is ${name}_Hany"
number=10
echo "number : ${number}"
number=$(expr number + 1)
```

Conditions



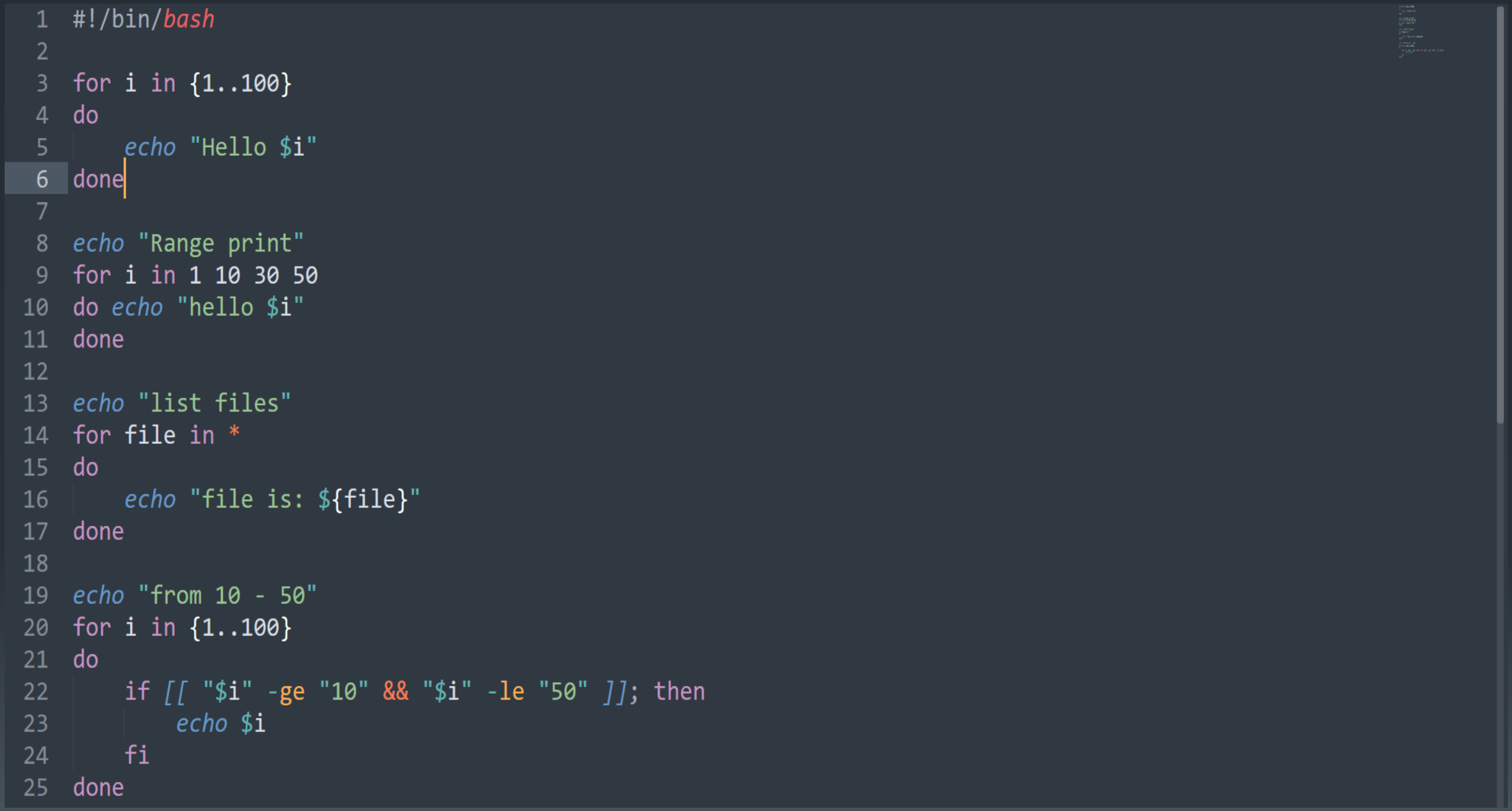
```bash
#!/bin/bash

#-gt (>) , -lt(<) , -ge(>=) , le (<=) , eq (=) , -ne (!=)
echo "Enter your degree:"
read degree
echo "Your degree is :${degree}" #["$degree" -ge "80" ] && echo "Passed" || echo "Failed"
if ["$degree" -ge "50"]; then
    echo "passed"
elif [[ "$degree" -eq "80" ]]; then
    echo "got 80" #["$degree" -eq "80" ] && echo "he got 80"
elif [[ "$degree" -eq "70" ]]; then
    echo "got 70" #["$degree" -eq "70" ] && echo "he got 70"
else
    echo "failed"
fi


```

Case



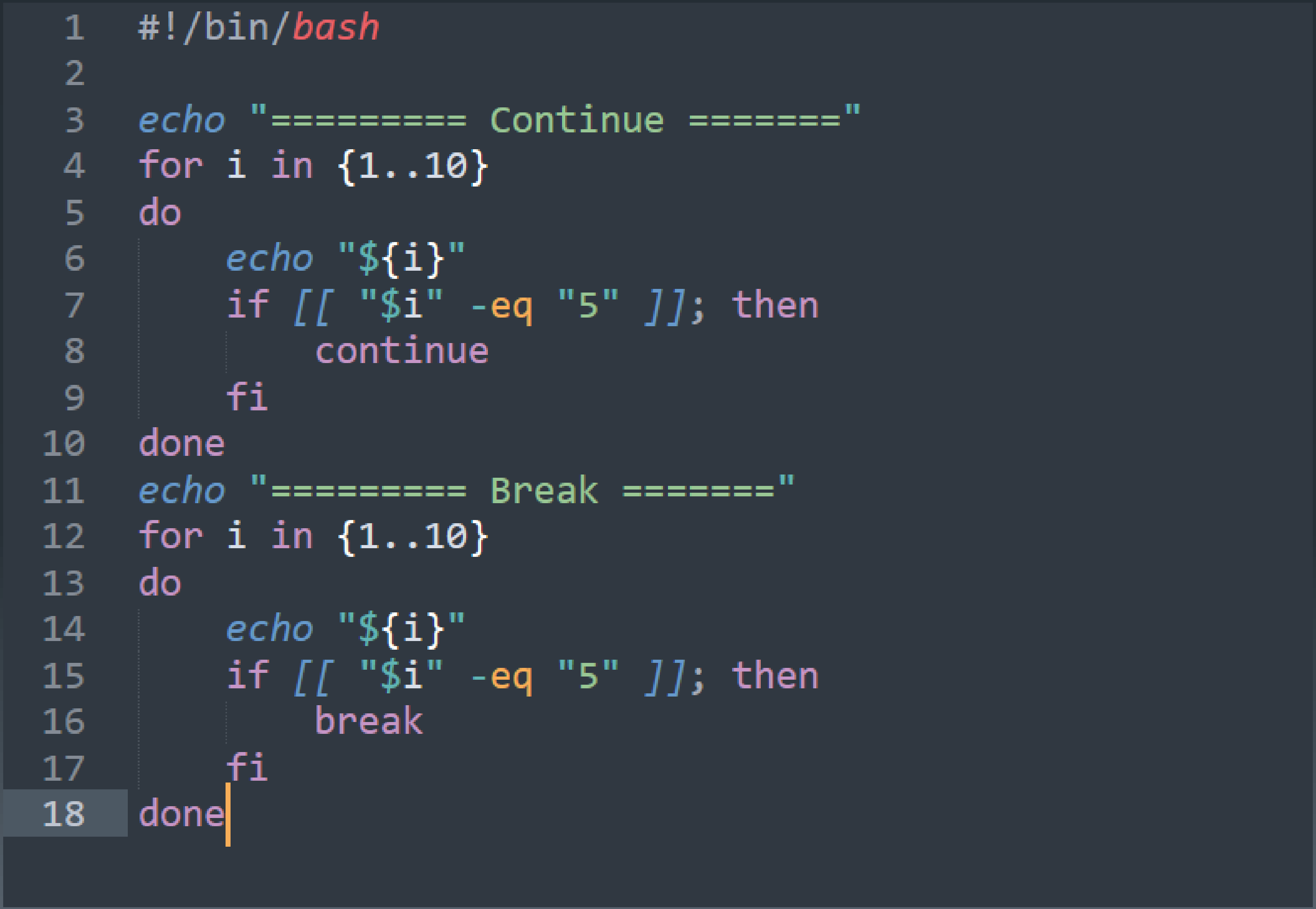
```
#!/bin/bash
echo "enter number"
read number
#if [[ "$number" -eq "1" ]]; then
#   echo "1 Entered"
#lif [[ "$number" -eq "3" ]]; then
#   echo "3 Entered"
#elif [[ "$number" -eq "2" ]]; then
#   echo "2 Entered"
#else
#   echo "out of range"
#fi
case $number in
    1) echo "1 Entered"
        ;;
    2) echo "2 Entered"
        ;;
    3) echo "3 Entered"
        ;;
    *) echo "out of range"
esac
```

For Loop



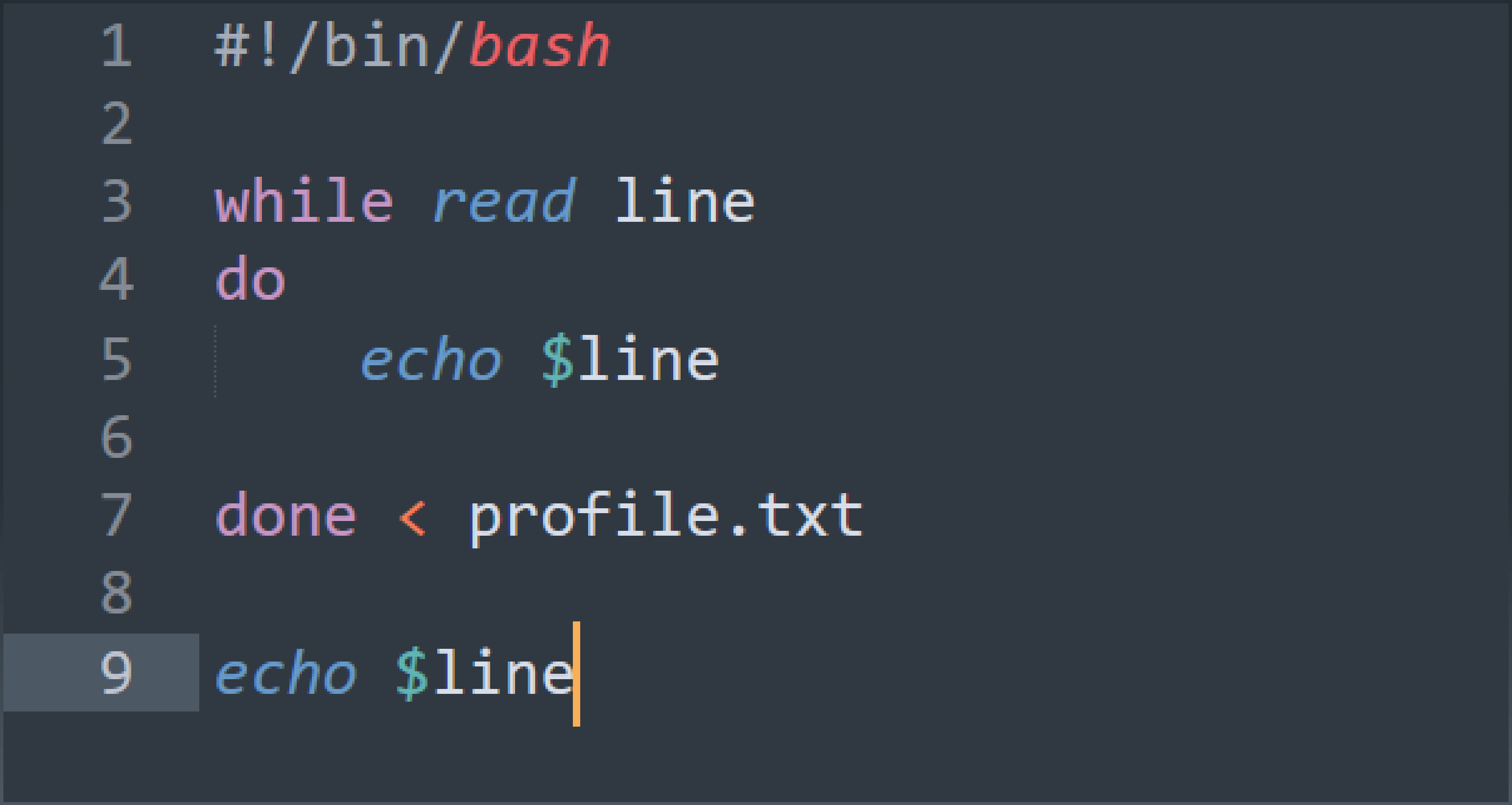
```bash
#!/bin/bash

for i in {1..100}
do
    echo "Hello $i"
done

echo "Range print"
for i in 1 10 30 50
do echo "hello $i"
done

echo "list files"
for file in *
do
    echo "file is: ${file}"
done

echo "from 10 - 50"
for i in {1..100}
do
    if [[ "$i" -ge "10" && "$i" -le "50" ]]; then
        echo $i
    fi
done
```

Control



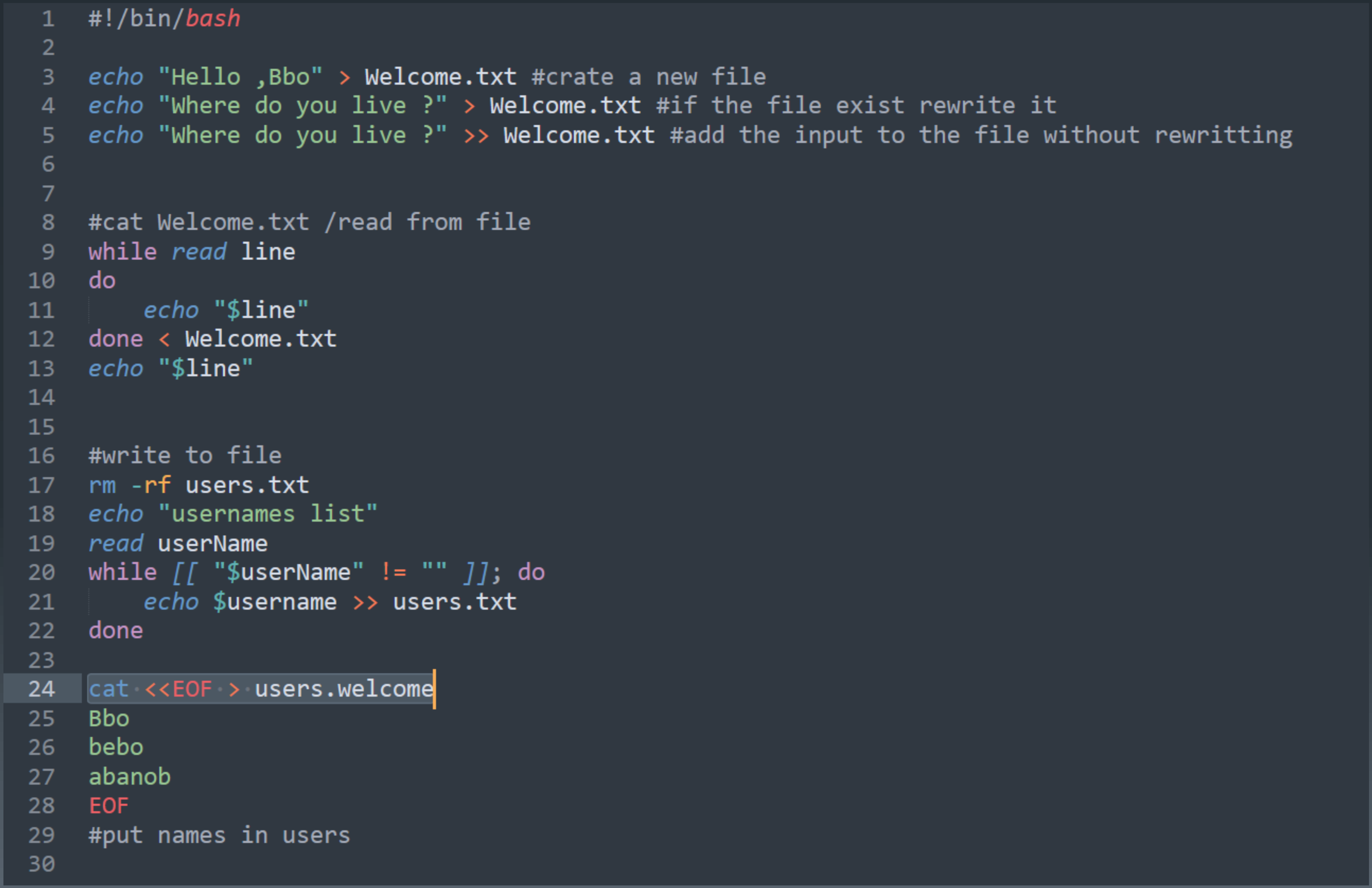
```bash
#!/bin/bash

echo "========== Continue ======="
for i in {1..10}
do
    echo "${i}"
    if [[ "$i" -eq "5" ]]; then
        continue
    fi
done
echo "========== Break ======="
for i in {1..10}
do
    echo "${i}"
    if [[ "$i" -eq "5" ]]; then
        break
    fi
done
```

While loop



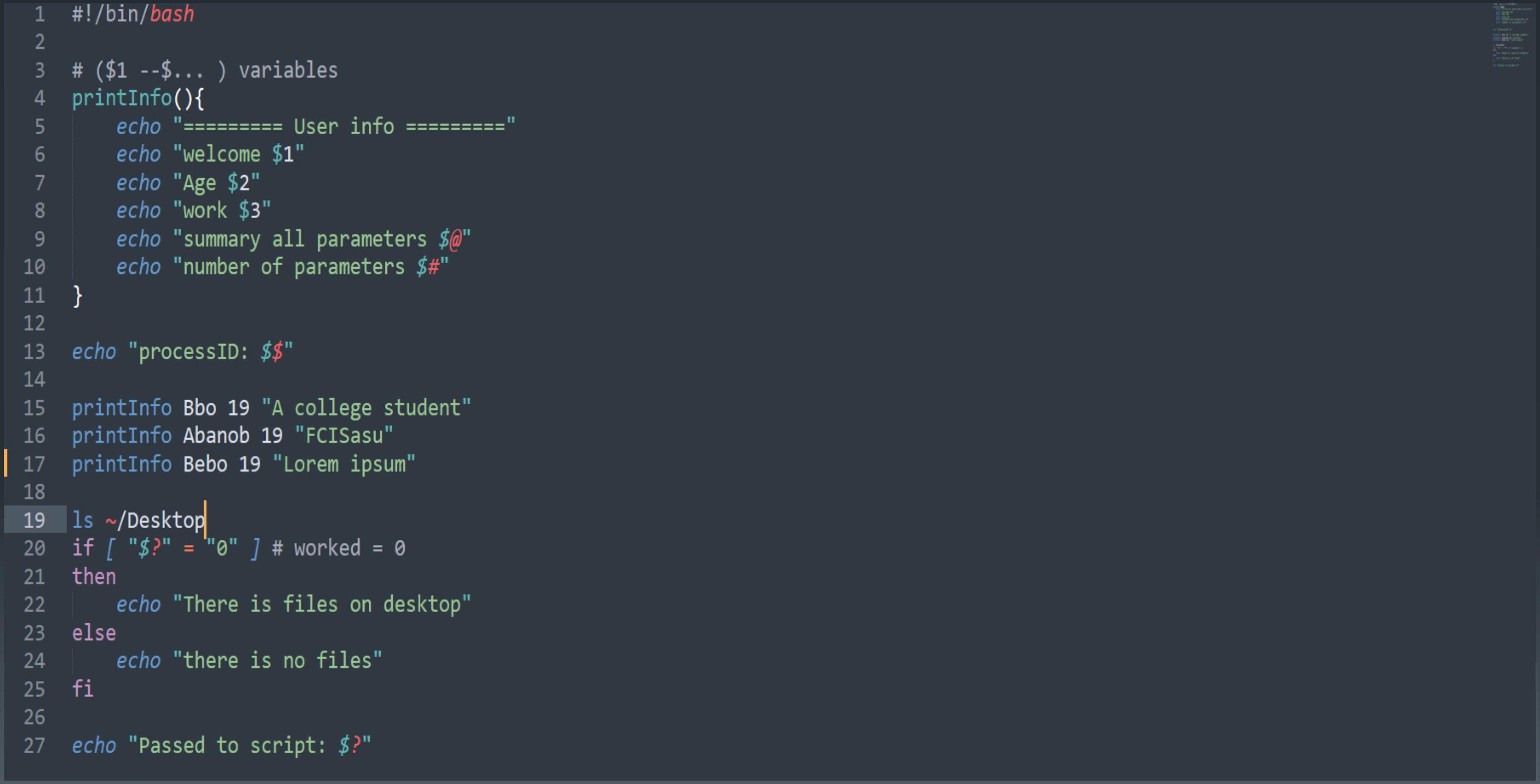
```bash
#!/bin/bash

while read line
do
    echo $line

done < profile.txt

echo $line
```

Files



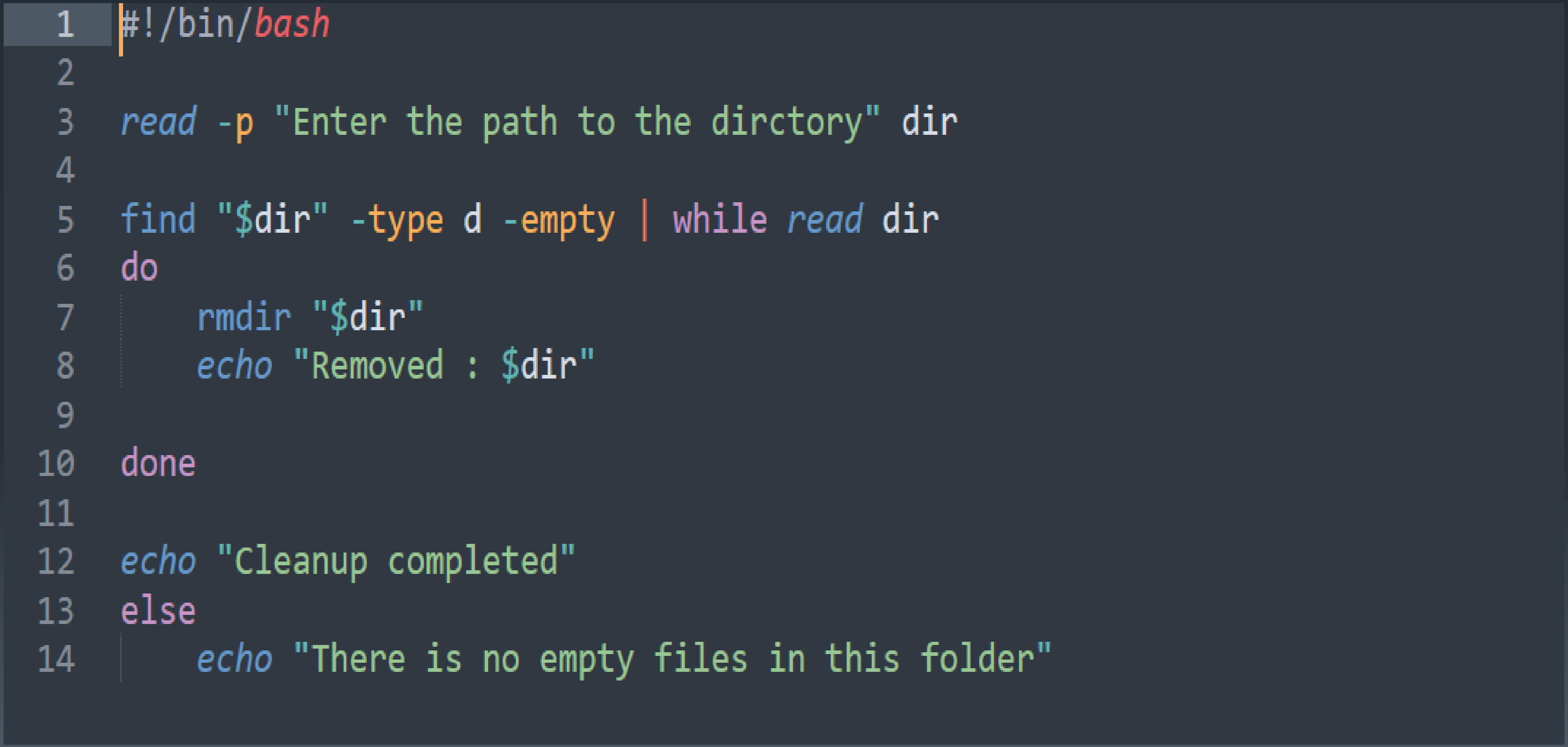
```
#!/bin/bash

echo "Hello ,Bbo" > Welcome.txt #crate a new file
echo "Where do you live ?" > Welcome.txt #if the file exist rewrite it
echo "Where do you live ?" >> Welcome.txt #add the input to the file without rewritting


#cat Welcome.txt /read from file
while read line
do
    echo "$line"
done < Welcome.txt
echo "$line"


#write to file
rm -rf users.txt
echo "usernames list"
read userName
while [[ "$userName" != "" ]]; do
    echo $username >> users.txt
done

cat <<EOF > users.welcome
Bbo
bebo
abanob
EOF
#put names in users

```

Function



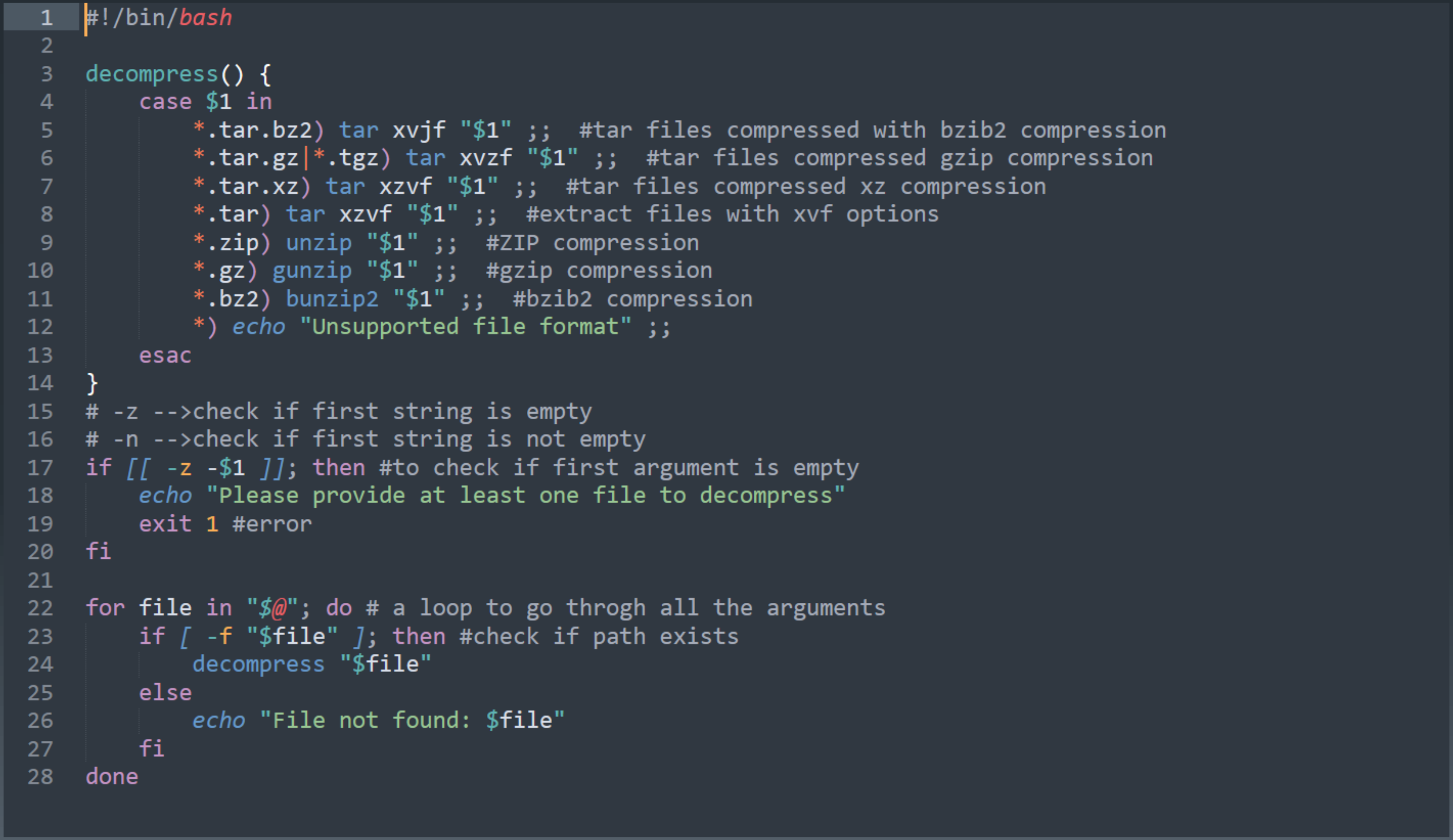
```bash
#!/bin/bash

# ($1 --$... ) variables
printInfo(){
    echo "========= User info ========="
    echo "welcome $1"
    echo "Age $2"
    echo "work $3"
    echo "summary all parameters $@"
    echo "number of parameters $#"
}

echo "processID: $$"

printInfo Bbo 19 "A college student"
printInfo Abanob 19 "FCISasu"
printInfo Bebo 19 "Lorem ipsum"

ls ~/Desktop
if [ "$?" = "0" ] # worked = 0
then
    echo "There is files on desktop"
else
    echo "there is no files"
fi

echo "Passed to script: $?"
```

Folder Clean up script



```bash
#!/bin/bash

read -p "Enter the path to the dirctory" dir

find "$dir" -type d -empty | while read dir
do
    rmdir "$dir"
    echo "Removed : $dir"

done

echo "Cleanup completed"
else
    echo "There is no empty files in this folder"
```

Decompress Script



```bash
#!/bin/bash

decompress() {
    case $1 in
        *.tar.bz2) tar xvjf "$1" ;;  #tar files compressed with bzib2 compression
        *.tar.gz|*.tgz) tar xvzf "$1" ;;  #tar files compressed gzip compression
        *.tar.xz) tar xzvf "$1" ;;  #tar files compressed xz compression
        *.tar) tar xzvf "$1" ;;  #extract files with xvf options
        *.zip) unzip "$1" ;;  #ZIP compression
        *.gz) gunzip "$1" ;;  #gzip compression
        *.bz2) bunzip2 "$1" ;;  #bzib2 compression
        *) echo "Unsupported file format" ;;
    esac
}
# -z -->check if first string is empty
# -n -->check if first string is not empty
if [[ -z -$1 ]]; then #to check if first argument is empty
    echo "Please provide at least one file to decompress"
    exit 1 #error
fi

for file in "$@"; do # a loop to go throgh all the arguments
    if [ -f "$file" ]; then #check if path exists
        decompress "$file"
    else
        echo "File not found: $file"
    fi
done
```