Conditional Statements

```
#!/bin/bash
#Author :Aniruddha Das
#Purpose: Conditional Statement
#Usage: ./conditional.sh

file=$1
if [ -f "$file" ]; then
        echo "file exist $file"
else
        echo "file doesn't exists"
fi

file=$1
if [[ -f $file ]]; then
        echo "file exists $file"
else
        echo "file doesn't exist"
fi

#!/bin/bash
#Author :Aniruddha Das
#Purpose: Conditional Statement
#Usage: ./conditional.sh

file=variable.sh
```

```
if [ -f $file ]; then
        echo "file exist"
fi
```

```
#!/bin/bash
echo "Hello World"
```

While Loop

```
#!/bin/bash
#Author :Aniruddha Das
#Purpose:Learning for lop
#Usage: ./for.sh

echo "Please enter the number"
read -r no
echo "Multiplication table of $no"
counter=1
while [ $counter -le 10 ]
do
        mult=` expr $no \* $counter`
        echo "$no * $counter = $mult"
        counter=` expr $counter + 1 `
done
```

IP Ping

```
#!/bin/bash
#Author :Aniruddha Das
```

```
#Purpose:Learning ip ping
#Usage: ./ipping.sh

echo -e "please enter the ip address to ping: \c"
read -r ip
until ping $ip
do
        echo "Host in $ip is down"
        sleep 1
done
echo "host in $ip is up"
```

For loop

```
#!/bin/bash
#Author :Aniruddha Das
#Purpose:Learning for lop
#Usage: ./for.sh

fruits=("apple" "banana" "cherry" "mango")
for fruit in "${fruits[@]}"; do
        echo "I like to eat $fruit"
done



fruits=("apple" "banana" "cherry" "mango")
for i in "${!fruits[@]}"; do
```

```
    echo "FRUIT ATE $i is ${fruits[$i]}"
done



fruits=("apple" "banana" "cherry" "mango")
for i in "${!fruits[@]}"; do
        if (( $i%2 != 0 )); then
                echo "I like ${fruits[$i]}"
        else
                echo "I dont like ${fruits[$i]}"
        fi

done
```

File Check

```
#!/bin/bash
 file="example.txt"
 if [ -f "$file" ]; then
 echo "File exists."
Else
 echo "File does not exist."
 fi
```

Key Notes:

1. **Spaces**: Ensure there is a space after [, and before].
2. **Quoting variables**: It's a good practice to quote variables ("$variable") to prevent issues with spaces or empty values.

3. **Comparison operators**: For numbers, use -gt (greater than), -lt (less than), -eq (equal), etc. For strings, use = for equality and != for inequality.

Count using while loop

```
#!/bin/bash


count=1
while [ $count -le 10 ]; do
   if [ $count -eq 5 ]; then
      echo "Count is 5, exiting the loop."
      break
   fi
   echo "Count is $count"
   ((count++))
done
```