

### 3. until loop:

>> It is similar to while loop.  
>> It does what while loop do, but opposite.  
>> Until loop will execute the block of code as long as the condition is 'False'.  
>> If the condition becomes 'True', it will terminate the loop.

#### -print (1-5) using while

```
i=1
while [ $i -lt 6 ]
do
echo $i
i=$((i+1))
done
```

1 < 6 ✓  
2 < 6 ✓  
3 < 6 ✓  
4 < 6 ✓  
5 < 6 ✓  
6 < 6 ✗

#### -print (1-5) using until

```
i=1
until [ $i -ge 6 ]
do
echo $i
i=$((i+1))
done
```

1 >= 6 ✗  
2 >= 6 ✗  
3 >= 6 ✗  
4 >= 6 ✗  
5 >= 6 ✗  
6 >= 6 ✓

o/p: 1 2 3 4 5

cond. is 'True',  
so, it will stop  
the execution.

### SCRIPT: WRITE A SCRIPT USING UNTIL LOOP TO PRINT EVEN NUMBERS FROM 1-20

```
until [ $i -gt 21 ]
do
if [ $((i%2)) -eq 0 ]; then
echo $i
fi
i=$((i+1))
done
```

20 / 2 = 10  
20 / 2 = 10  
0 %

### FUNCTIONS

```
details(){
read -p "Enter name:" name
read -p "Enter aadhar:" aadhar
read -p "Enter mobile number:" number
read -p "Enter PAN number:" PAN
read -p "Enter address:" address
}
```

BANK → 100 users

Name → read -p "Enter name:" name  
Aadhar → read -p "Enter aadhar:" aadhar  
mobile →  
PAN →  
Address →

1 user

user3      user4      user5  
details()   details()   details()

>> Functions allows users to group commands/block of code in a single unit & it can be reused throughout the script.

>> 2 parts are involved in working with functions:

- Defining the function (creating a function)
- Calling the function (making use of the function)

#### i. Defining the function

```
function-name(){
# statements/block of code
}
```

>> Give a Function-name and follow it with a parenthesis ().  
>> This parenthesis (), will treat it as a function,

eg:

```
add(){
num1=10
num2=4
echo $((num1+num2))
}
```

add → Function call

./script1.sh

```
function function-name {
# statements/block of code
}
```

>> Make use of a reserved keyword called as 'function'.  
>> Parenthesis() is not required in this syntax.

```
function bhavana {
num1=10
num2=4
echo $((num1-num2))
}
```

bhavana → Function call

#### ii. Calling the function

>> Making use of the function.  
>> Calling the function is done by the function's name & it is called inside the script.

>> Function block will invoke & execute, ONLY if the function call is there.

### SCRIPT: TAKE 2 NUMBERS FROM USER INPUT & CREATE ADD, SUBTRACT, MULTIPLY & DIVISION FUNCTION AND PERFORM ALL THESE TASKS.

```
read -p "Enter num1:" num1
read -p "Enter num2:" num2
```

```
function add {
echo "Addition of $num1 and $num2 is $((num1+num2))"
}
```

```
subtraction(){
echo "Diff of $num1 and $num2 is $((num1-num2))"
}
```