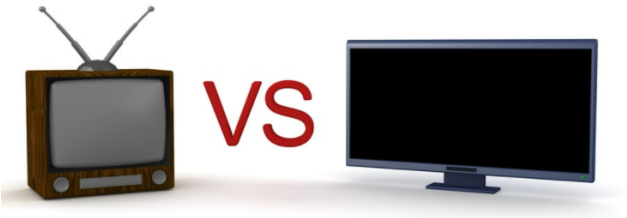
functions (part-1)

What are Functions?

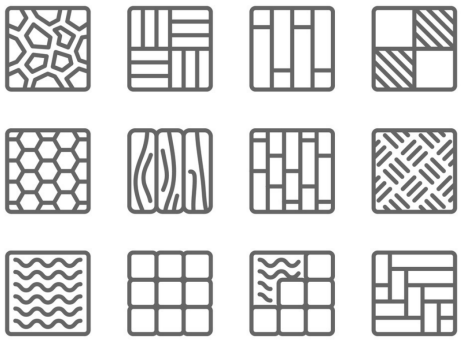
> ***Technical Term:-*** *A block of code designed to perform a particular task; they are very useful in making code simplified and manageable.*

- **Example 1:** Take an example of **products on amazon** and how they are stored, and how much amount of code is there to manage that thing.



Pattern

What do you think when you hear the word Pattern
The pattern is something or a series of something is repeated in sequence.



The pattern you have seen before in colors, shapes, music, and even Maths.

```
"hand-wash"
"eat-dinner"
"hand-wash"

"hand-wash"
```

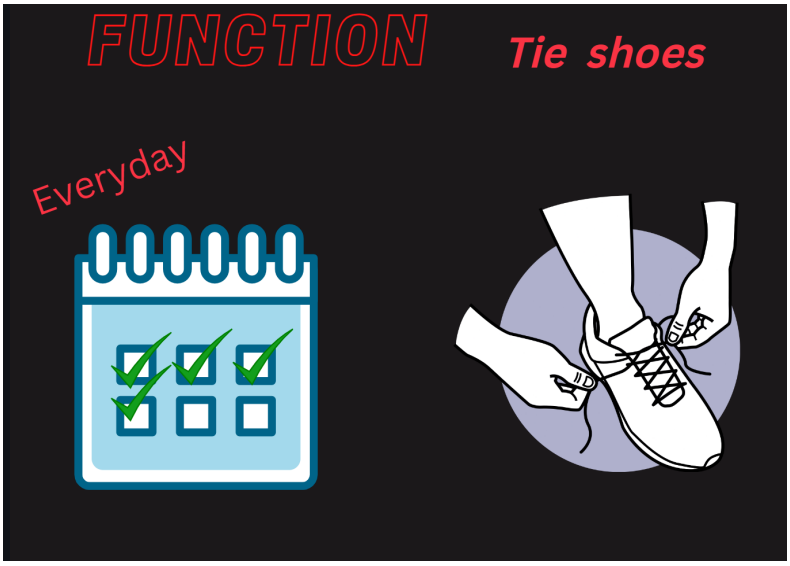
```
"eat-dinner"
"hand-wash"

"hand-wash"
"eat-dinner"
"hand-wash"

"hand-wash"
"eat-dinner"
"hand-wash"

"hand-wash"
"eat-dinner"
"hand-wash"


def getDinner():
  "hand-wash"
  "eat-dinner"
  "hand-wash"

# here we are repeating the activities like hand-wash,eat-dinner, hand-wash multiple times
# so we have created this function which we call again and again and this will reduce the repetitions of activity
```

Function

The function is a way by which you can control your code.

You perform a function in your day-to-day life.

Suppose whenever you go to Mall or School. You tie your shoes. Tying shoes is a daily activity you every day do.



In Tying shoes, there are certain steps, such as

1- Gather Laces

2- Knot

3- Loop

4- Swoop

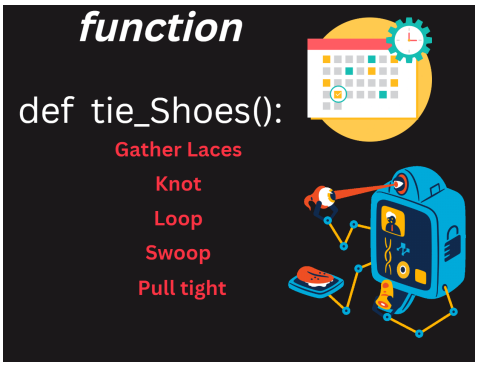
5- Pull Tight

And you follow these steps on a daily basis

Now, suppose the same activity you assigned to the computer or a system to perform

So what you will do you repeat the same steps every day on the computer or a machine.

Here, for your help, there is an effective tool known as a function, which will follow the steps whenever you call it.




```
def product() # will have the Product page code.
def payment() # will have the Payments page code.
def cart()   # will have the cart page code.
```

Syntax of a function declaration

```
#def declaration syntax.

def  name_of_function():
```

- Take an example of printing a name.

```
/*
1. Print your name using a function
2. First, show them how you can print without using function
3. Then declare a def and put the code inside the function.
4. Either you can take the TV channel and button pressing on
   remote example or a friend taking your stuff and now you
   have to call him to get the stuff.
*/

def channel_1():
    x="Masai";
    print(x)


#To get this thing to work you have to call the function
channel_1(); # Output will be "Masai"
```

- What will happen if i call the function multiple times.

```
def channel_1():
     x="Masai";
     print(x);

channel_1()
channel_1()
```

```
channel_1()
channel_1()

#This will print "Masai" 4 times as I have called the def 4 times.
```

- Dry run this code for better understanding.
- Take two example of adding two numbers and subtracting 2 numbers without using functions.

```
#Add without using the function
a=10;
b=15;
sum=a+b;
print(sum); # 25

#Subtract without using the function
x=10;
y=5;
difference=x-y;
print(difference);  # 5
```

- Convert the **above** example in **functions** by just putting the above code in two separate functions, **first show the output without calling them then, call them one by one.**

```
#Add using the function
def superman():
   a=10;
   b=15;
   sum=a+b;
   print(sum);


#Subtract using the function
def batman():
  x=10;
  y=5;
  difference=x-y;
  print(difference);  # 5


superman(); # 25
batman();  # 5
```

- **Activity Time:-** Create 4 functions to do **Addition, Subtraction, Multiplication, Division.**

```
#Add using the function
def superman():
    a=10;
    b=15;
   sum=a+b;
   print(sum);


#Subtract using the function
def batman():
    x=10;
    y=5;
```

```
  difference=x-y;
  print(difference);


#Multiplication using the function
def spiderman():
  x=5;
  y=2;
  mult=x*y;
  print(mult);


#Division using the function
def ironman():
  x=10;
  y=5;
  div=x/y;
  print(div);


superman(); # 25
batman(); # 5
spiderman(); # 10
ironman(); # 2
```

- Calling functions inside a loop.

```
def superman():
   a=10;
   b=15;
   sum=a+b;
   print(sum);


for i in range(5):
     superman()

/*

25
25
25
25
25

*/
```

Parameters

```
def superman(a,b):
  sum_val=a+b
  print(sum_val)


x=4;
y=5;
superman(x,y); # 9
A=5;
```

```
ar B=10
superman(A,B); # 15
```

- dry run the above code for proper visualization.

```
#Addition using a def and return statement

def superman(a,b):
  sum=a+b;
  return sum;


  bucket=superman(10,15);
  print(bucket); # 25
```

IMPORTANCE OF RETURN BY TAKING AN EXAMPLE

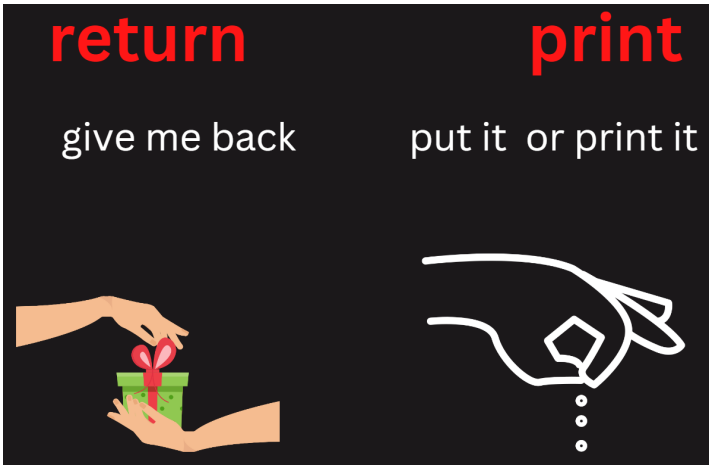
> **Step 1: Superman** plans to **add** two numbers and send the answer to Batman.

> **Step 2: Batman** will take the answer from **Superman, square** it and send it to Aqua man.

> **Step 3: Aqua man** will take the answer from **Batman** and divide it by 10.

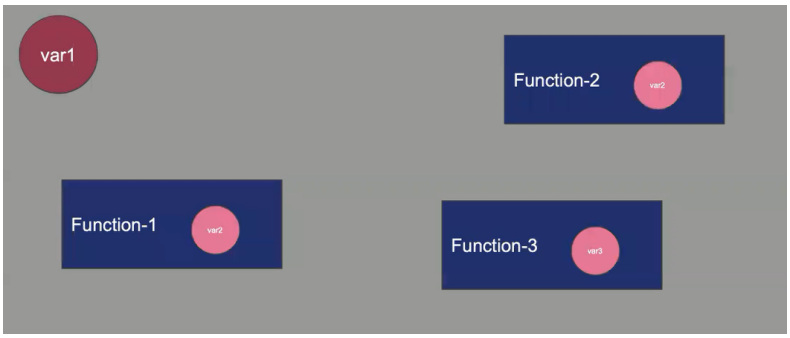
```
def superman(a,b):
  sum=a+b;
  return sum;


def batman(x):
   square=x*x;
  return square


def aquaman(y):
  div=y/10;
  return div;


step1=superman(10,15);

step2=batman(step1);

step3=aquaman(step2);

print(step3);  #62.5
```

- **Return** vs. **print()**



Global Scope and Local Scope.

> Give an example of a Personal Phone and a Public Phone (PCO).



> **Problem 1:** Create a function to check if a number is prime or not.

```
def check_prime(num):
  count=0;
  for i in range(num):
    if(num%i==0):
      count++;

    if(count==2):
      return true;
   else:
      return false;



 x=check_prime(13)
if x==true:
  print("Prime Number");
else:
  print("Not a Prime Number");
```

- Dry run the above code for better understanding.

> **Problem 2:** Use the Above code to print Primes from 2 to a given limit.

```
#using the above check_prime() function I am checking the prime numbers
#in a given limit.

for i=lower_limit in range(upper_limit):
   x=check_prime(i);
  if(x==true):
    print(i,"is a Prime Number");
  else:
  print(i, "is Not a Prime Number");
```

- Dry run the above code for better understanding.

> **Problem 3:** Write a function to check if the char is a small case or not (Write the code and do the dry run).