Function A Function is a Self Contained Block of code which is used to perform Particular Task

Block of Code

```
a=10
b=20
print(a+b)
```

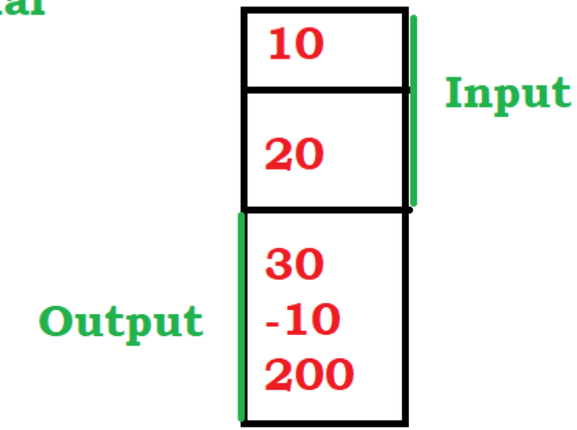
Add( )

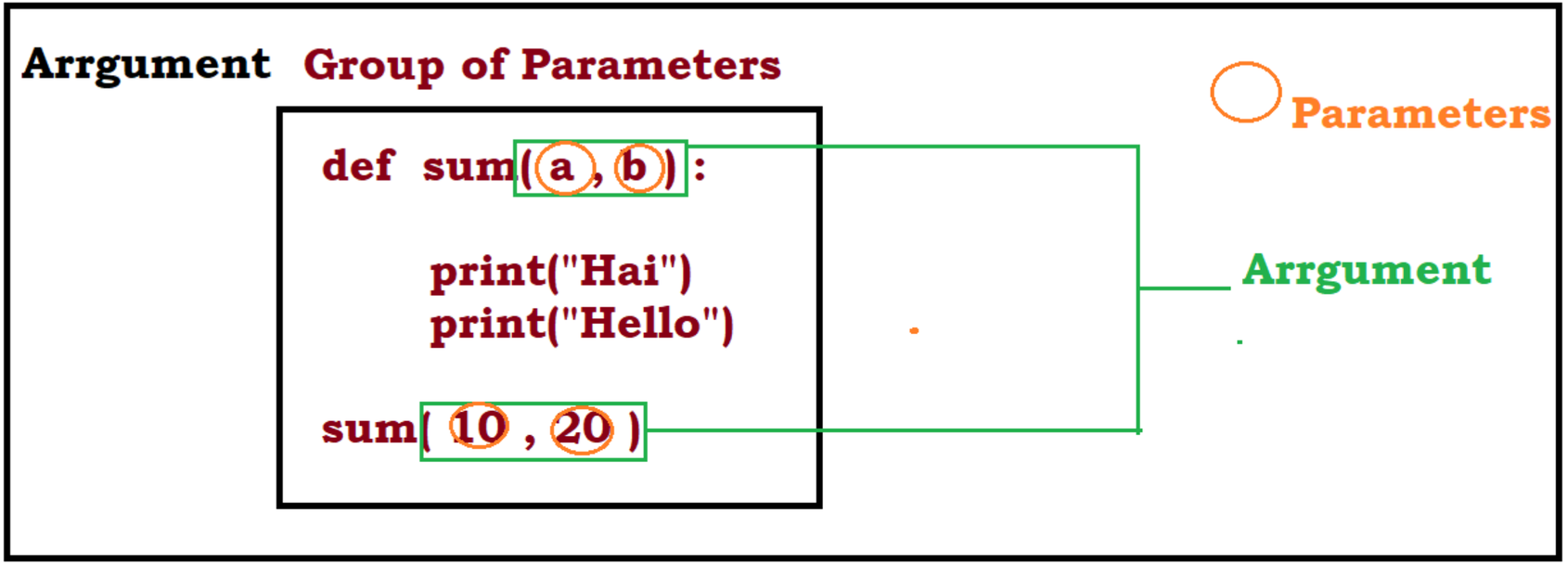
```
a=10
b=20
print(a+b)
```

```
def aa(x,y):
    print(a+b)
    print(a-b)
    print(a*b)
a=int(input())#10
b=int(input())#20
aa(a,b)
```

Formal

Actual

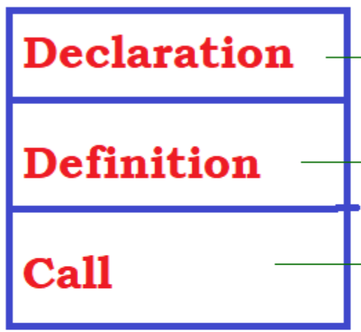



Arrgument
Group of Parameters
Parameters
def sum(a , b) :
print("Hai")
print("Hello")
sum(10 , 20)
Arrgument

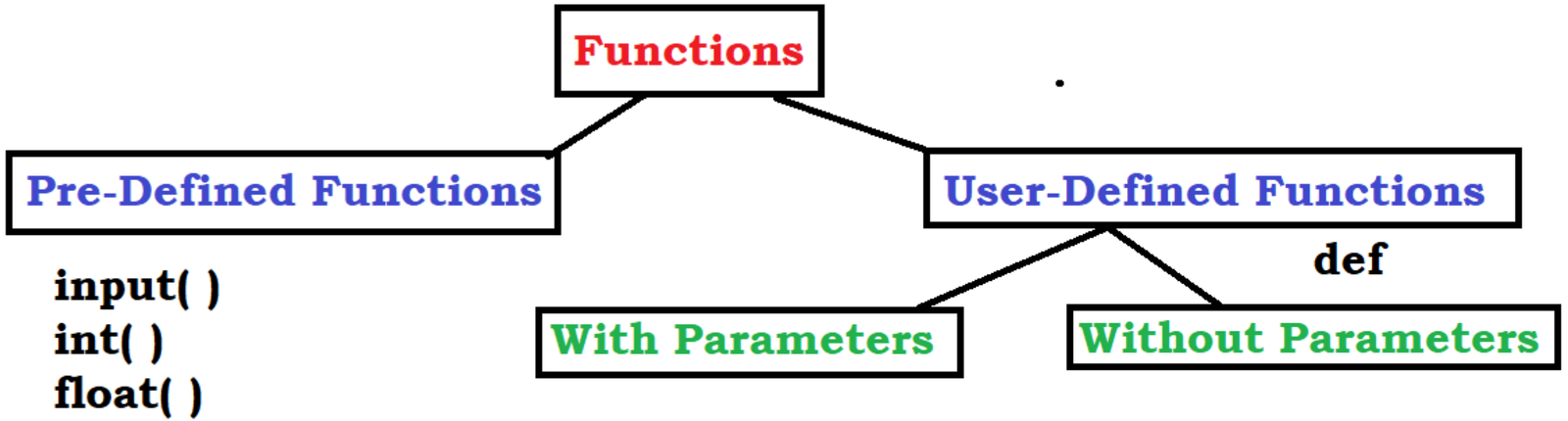
Creation of Function

We creates a Function by using def Keyword

def = Definition

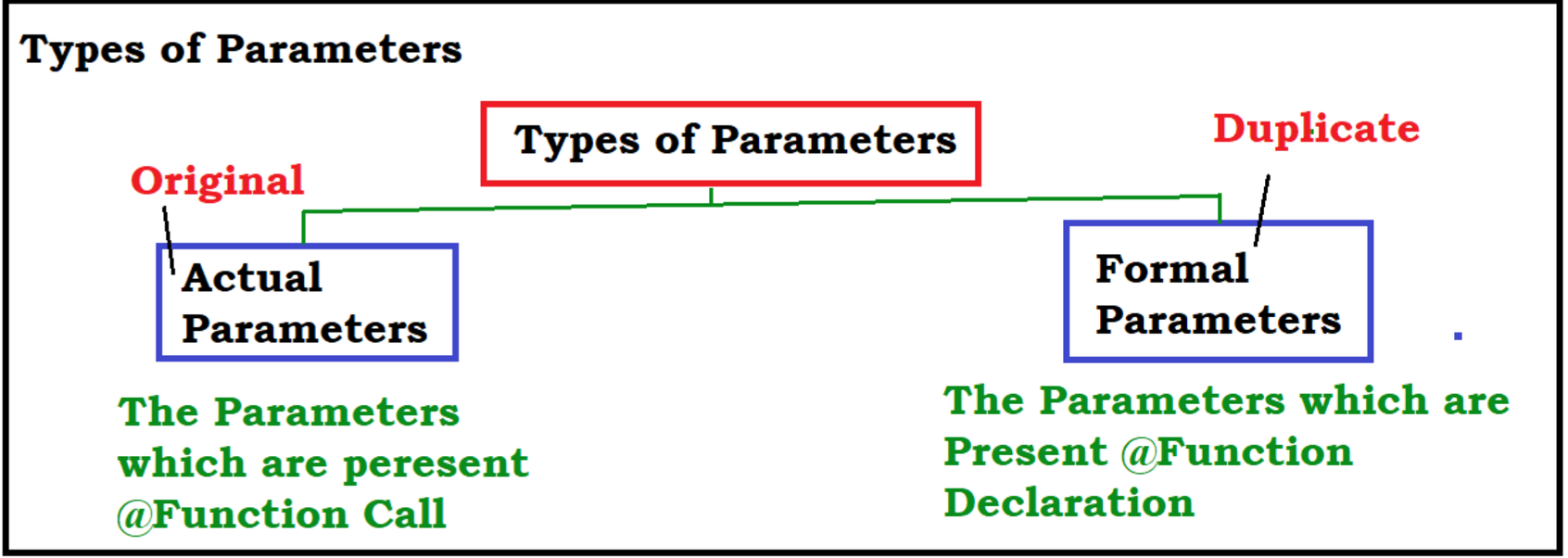
Block Diagram	Syntax	Example
	def Fun-Name(PL) : #Stmt-Block Fun-Name(PL)	def MyBlock( ): print("Hello") MyBlock( )

Types of Functions **Bascially we have two Types of Functions**



Fnction with Parameter

Parameter - A Parameter is a Values which are Pass to Function for perform a Specific Task.


Types of Parameters
Types of Parameters
Original
Duplicate
Actual Parameters
Formal Parameters
The Parameters which are peresent @Function Call
The Parameters which are Present @Function Declaration

Lambda

Use lambda Keyword

A Lambda is a Small Anonymous Function

The Lambda can take any Number of Arrgumants but it Evaluates in a Single Expression

Syntax

```
lambda Arrguments : Expression
```

```
z=lambda a,b : a-b+a
print(z(10,20))#0
```

```
z=lambda a,b : a+b+a
print(z("C","S"))#CSC
```

```
z=lambda a,b,c : a-b+a
print(z(10,20))
```

```
TypeError: <lambda>()
missing 1 required
positional argument: 'c'
```