types of argument

In [ ]:

```
1.positional argument
```

In [1]:

```
def sub(a,b):
    print(a-b)
sub(200,20)
sub(20,200)
```

```
180
-180
```

In [ ]:

```
2.keyword argument
```

In [7]:

```
def wish(name,msg):
    print("hello",name,msg)
wish(name = "raviraj",msg = "how are you")
wish(msg = "good morning",name = "Raviraj")
wish("arman","good morning")
wish("arman",msg = "good morning")
wish(name = "arman","good morning")

```

```
  File "<ipython-input-7-752c2017843c>", line 7
    wish(name = "arman","good morning")
                                      ^
SyntaxError: positional argument follows keyword argument
```

In [8]:

```
def wish(name,msg):
    print("hello",name,msg)
wish(name = "raviraj",msg = "how are you")
wish(msg = "good morning",name = "Raviraj")
wish("arman","good morning")
wish("arman",msg = "good morning")
```

```
hello raviraj how are you
hello Raviraj good morning
hello arman good morning
hello arman good morning
```

In [ ]:

```
3.deafault argument
```

In [13]:

```
def wish(name= "parmar"):
    print("hello",name,"good morning")
wish("raviraj")
wish()
```

```
hello raviraj good morning
hello parmar good morning
```

In [ ]:

```
4.variable argument
```

In [20]:

```
def sum(*n):
    total = 0
    for n1 in n:
        total = total + n1
    print("total ",total)
sum(17)
sum()
sum(10,20,30,40)
```

```
total  17
total  0
total  100
```

In [37]:

```
def f1(*s):

    for S1 in s:
        print(s)
f1(10)
f1(20,30,40,50)
```

```
(10,)
(20, 30, 40, 50)
(20, 30, 40, 50)
(20, 30, 40, 50)
(20, 30, 40, 50)
```

In [2]:

```
def f1(*s,n1):
    for s1 in s:
        print(s)
f1(10)

```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-2-1a35db6ca24f> in <module>
      2     for s1 in s:
      3         print(s)
----> 4 f1(10)
      5 

TypeError: f1() missing 1 required keyword-only argument: 'n1'
```

In [5]:

```
def f1(*s,n1):
    for s1 in s:
        print(s)
f1(20,30,40,50,n1 = 10)

```

```
(20, 30, 40, 50)
(20, 30, 40, 50)
(20, 30, 40, 50)
(20, 30, 40, 50)
```

In [ ]:

```
7.variable length keyword argument
```

In [13]:

```
def display(**kwargs):
    print(kwargs)
    for k,b in kwargs.items():
        print(k,b)
display(n1 = 200,n2 = 300,n4 = 400)

```

```
{'n1': 200, 'n2': 300, 'n4': 400}
n1 200
n2 300
n4 400
```

make a function of calculator

In [17]:

```
a = int(input("Ente a :"))
b = int(input("Ente b :"))

def cal(choice):

    if choice == "+":
        print("addition = ",a+b)
    elif choice == "-":
        print("sunstraction = ",a-b)
    elif choice == "*":
        print("multiphication = ",a*b)
    elif choice == "/":
        print("division = ",a/b)
c = input("Enter sign of operation : ")
cal(c)

```

```
Ente a :10
Ente b :20
Enter sign of operation : *
multiphication =  200
```

In [23]:

```
start = int(input("Enter start range : "))
end = int(input("Enter end range : "))
n = int(input("Enter number : "))
def gap(s,e,n1):
    if e-n1 > 0:
        print("yes")
    else:
        print("no")
gap(start,end,n)
```

```
Enter start range : 10
Enter end range : 20
Enter number : 22
no
```

make function to chack given numbe is even or odd

In [25]:

```python
n = int(input("Enter number : "))
def chack(n1):
    if n1%2 == 0:
        print("even")
    else:
        print("odd")
chack(n)
```

```
Enter number : 17
odd
```

In [ ]:

```python

```