برای ورودی دادن به تابع می توان به سه روش زیر عمل کرد:

```
def func1(name,age):
    print(name +" : " + str(age))

func1('ali',29)


def func2(*args):
    z = 1
    for num in args:
        z *= num
    print(z)

func2(6,7,8,10)

def func3(**kwargs):
    for key, value in kwargs.items():
        print("The value of {} is {}".format(key, value))

func3(my_name="ali", your_name="somebody")

def func4(param=[]):
    param.append("something")
    print(param)

func4() # returns ["something"]
func4() # returns ["something", "something"]
```

همانطور که مشخص است در تابع func1 دو مقدار name وage به تابع پاس داده می شوند و اگر کاربر تعداد ورودی کمتر و یا بیشتی را به تابع ارسال کند، با خطا مواجه می شود؛ چون دقیقا باید تعداد ورودی های پاس داده شده به تابع با تعداد ورودی های ست شده در هنگام تعریف تابع برابر باشد. علاوه بر اینکه تعداد آنها باید برابر باشد، جایگاه آنها نیز باید در نظر گرفته شود وگرنه با خطا مواجه می شویم و یا به خروجی اشتباهی دست پیدا می کنیم. حال برای رفع این دو مشکل با کمک گرفتن از *args وkwargs** در ورودی تابع این دو مشکل رفع می شود و می توان هر تعداد که ورودی خواستیم به تابع پاس دهیم و یا حتی یک دیکشنری را به

تابع پاس دهیم و تابع از مقادیر key و value استفاده کند. در func2 وfunc3 به ترتیب از *arg و **kwargs استفاده شده است.

همچنین می توانستیم ورودی را به صورت یک لیست به ابع پاس دهیم که در تابع func4 آورده شده است و این مشکل ایجاد می شود که هر بار که تابع را فراخوانیم تابع از لیست قبلی خود استفاده می کند و ورودی در واقع بزرگتر می شود و متغیر.

خروجی ۴ مثال بالا در زیر آورده شده است:

```
ali : 29
3360
The value of my_name is ali
The value of your_name is somebody
['something']
['something', 'something']
```