فصل سوم

توابع

ساختار یک برنامه خیلی شبیه به ساختار یک سازمان است . یعنی، در هر سازمان ساختار سلسله مراتبی حاکم است . در بالاترین سطح سازمان، مدیریت قرار دارد .هر مدیر میتواند چند معاون داشته باشد .هر یک از معاونین نیز میتوانند چندین کارمند داشته باشند . این برنامه ها مانند شرکت های کوچک هستند که مدیر شرکت همه کارهای شرکت را انجام میدهد .ولی، در بسیاری از سازمان ها چنین وضعیتی حاکم نیست .برنامه های واقعی و کاربردی مانند سازمان های بزرگ طولانی و پیچیده هستند.

مدیر برای انجام هر وظیفه اش یکی از توابع(معاونین)خودش را صدا میزند و احتمالاً پارامترهای (پرونده های)را در اختیار او قرار داده، از او میخواهد کار را انجام داده، نتیجه را برگرداند .هر یک ازتوابع(معاونین)نیز خود توابع دیگر(کارمندان خودش)را صدا میزنند تا بخشی از کار را به آن ها محول نمایند و این روند تا انجام کار ادامه دارد .

با این تفاسیر، کاربری که از برنامه استفاده می نماید،نقش مشتری را بازی خواهد کرد که میتواند اطلاعاتی را در اختیار سازمان(برنامه)قرار داده، نتایجی را دریافت کند.

استفاده از تابع در برنامه نویسی دارای مزایای زیر است:

۱- برنامه نویسی ساخت یافته را امکانپذیر میکند.

۱- خوانایی برنامه را افزایش میدهد .همچنین تست، اشکال زدایی و خطایابی برنامه نیاز آسان تر خواهد شد . چون، برنامه ها به بخشهای کوچک تری تبدیل میشوند .لذا، خطایابی و اصلاح برنامه های کوچکتر آسانتر خواهد بود

۲- میتوان توابع مورد نیاز را در یک برنامه نوشت و از آنها در برنامه های دیگر نیز استفاده کرد . این امر، **استفاده مجدد** ۰ نام دارد .بدین ترتیب، کد نویسی کمتر خواهد شد و تولید نرم افزار سریع تر انجام میشود.

۳- توابع، امکان کار گروهی را فراهم می کنند .زیرا، پس از این که برنامه به بخشهای کوچک تر ی تقسیم شدند، هر یک از اعضای گروه وظیفه نوشتن و تست توابع مشخص را بر عهده میگیرند .بدین ترتیب، اعضای گروه به صورت هم زمان روی بخشهای مختلف برنامه کار میکنند(بدون این که منتظر همدیگر باشند .)انجام کار به صورت گروهی موجب میشود تا برنامه ها سریعتر آماده شوند.

۴- توابع امکان استفاده از کارهایی که دیگران انجام داده اند، را فراهم می کنند . یعنی، در برنامه هایتان می توانید از توابعی که دوستانتان آماده کرده اند، استفاده کنید.

۵- توابع، امکان ایجاد کتابخانه را فراهم میکنند .کتابخانه، مجموعه توابعی هستند که مورد نیازتان میباشند (مجموعه توابعی که به هم مرتبطند .)بنابراین، میتوان مجموعه توابع مرتبط به هم را دریک فایل کتابخانه (بسته)قرار داد و در برنامه ها از این فایل کتابخانه استفاده نمود.

انواع توابع

در هر زبان برنامه نویسی دو نوع تابع وجود دارد که عبارتند از:

۱- **توابع کتابخانه ای**، توابعی میباشند که همراه کامپایلر یا مفسر وجود دارند .این توابع را توابع عمومی نیز می نامند .زیرا، کاربردهای زیادی دارند .توابع کتابخانه ای را با توجه به کاربرد آن ها دسته بندی کردند و هر یک از دسته ها را در فایل ماژول خاصی قرار داده اند .تاکنون با برخی از این توابع نظیر print() ، int() آشنا شدید .در ادامه با بعضی از توابع مهم کتابخانه ای به همراه کاربرد آنها آشنا خواهید شد.

۲- **توابعی که برنامه نویس مینویسد**

همانطور که میدانید پایتون شامل ماژولهای متنوعی است .اما، با این وجود، توابع موجود در ماژول پایتون، پاسخگوی همه در خواستهای برنامه نویس نیستند .لذا، برنامه نویس باید بتواند توابعی را نوشته، از آنها استفاده کند .

برای این منظور، برنامه نویس باید دو کار زیر را انجام دهد:

۱- نوشتن تابع ۲- فراخوانی تابع

۱– نوشتن تابع

برای نوشتن تابع باید آن را تعریف کرد:

تعریف تابع

قبل از این که تابعی را بنویسید باید تابع را تعریف کنید .تعریف تابع تعیین می کند، این تابع چه ورودی های دارد، چه چیزی را برمیگرداند(خروجی تابع چیست) و چه عملی را انجام می دهد.

الگوی تابع، به صورت زیر است:

```
def نام تابع )لیست پارامترها(:
    بدنه تابع
```

نکته: توابع از لحاظ مقداری که برمیگردانند به سه نوع زیر تقسیم میشوند:

الف) **توابعی که هیچ مقداری را برنمیگردانند.**

ب) توابعی که فقط یک مقدار را برمیگردانند . برخی از این توابع عبارتند از:

۱- تابعی که بزرگ ترین مقدار بین سه عدد را برمیگرداند.

۲- تابعی که تعیین میکند عددی اول است یا خیر؟

۳- تابعی که تعیین میکند عددی کامل(تام)است یا خیر؟

۴-تابعی که حاصل ضرب دو عدد را برمیگرداند.

۵-و غیره

این توابع برای برگشت مقدار از دستور return استفاده میکنند .دستور return به صورت های زیر به کار می رود:

```
1. return مقدار;
2. return ( عبارت);
3. return;
```

ساختار اول، یک مقدار را برمی گرداند. به عنوان مثال، دستورات زیر را ببینید:

```
return False;
return 10;
```

دستور اول، مقدار False را برمی گرداند و دستور دوم، مقدار ۱۰ را برگشت خواهد داد.

اما، ساختار دوم، یک عبارت را ارزیابی کرده، نتیجه ارزیابی عبارت را برگشت خواهد داد. به عنوان مثال، دستور زیر را مشاهده کنید:

```
return (2 * i – 3)
```

این دستور ، ۲ را در i ضرب کرده، ۳ واحد از این حاصل کم می کند و برمی گرداند.

ساختار سوم، بدون این که تابع مقداری را برگرداند، از تابع برمی گردد.

ج-توابعی که چندین مقدار را برمیگردانند .این نوع توابع را در ادامه میبینید.

بررسی جزئیات الگوی تعریف یک تابع:

نام تابع، از قانون نام گذاری شناسه ها و متغیرها پیروی میکند و برای دسترسی به تابع به کار می رود.

پارامترهای تابع، اطلاعاتی هستند که در هنگام فراخوانی تابع باید به آن ارسال گردند .توابع می توانند از لحاظ تعداد پارامترهایی که میپذیرند به گروههای زیر تقسیم گردند:

۱- توابع بدون پارامتر، این توابع معمولاً برای چاپ پیغام مشخصی به کار میروند.

۲- تابع ممکن است یک پارامتر داشته باشد .در این صورت باید نام پارامتر تعیین گردد.

برخی از این توابع عبارتند از:

تابعی که عددی را دریافت کرده، تعیین میکند اول است یا خیر؟

تابعی که عددی را دریافت کرده، تعیین میکند تام است یا خیر؟

تابعی که عددی را دریافت کرده، فاکتوریل آن را برمیگرداند.

تابعی که عددی را دریافت کرده، مجموع ارقام آن را برمیگرداند.

۳-تابع ممکن است چندین پارامتر داشته باشد .در این صورت باید پارامترها با استفاده از کاما (,) ازهم جدا شوند .برخی از این توابع در زیر آمده اند:

تابعی که دو عدد را دریافت کرده، بزرگ ترین مقسوم علیه مشترک آنها ر ا برمی گرداند(این تابع دارای دو پارامتر است.)

تابعی که سه عدد را دریافت کرده، کوچک ترین عدد را برمیگرداند(این تابع سه پارامتر دارد.)

تابعی که دو عدد را دریافت کرده، اولین عدد را به توان عدد دوم رسانده و برمیگرداند.

تابعی که دو عدد را گرفته، محتویات آنها را تعویض میکند.

بدنه تابع,عملی که تابع باید انجام دهد، در بدنه تابع قرار میگیرد .**بدنه تابع**، مجموعه دستوراتی هستند که تابع باید اجرا کند.

****

۲-فراخوانی تابع :

دستوری که تابع را صدا زده و از آن استفاده میکند، **فراخوانی تابع** نام دارد . فراخوانی تابع به صورت زیر انجام میشود:

```
(لیست آرگومانها ) نام تابع
```

در هنگام نوشتن و استفاده از توابع باید به نکات زیر توجه کنید:

۱- تعداد پارامترها(در هنگام تعریف تابع)باید با تعداد آرگومان ها(در هنگاام فراخوانی) یکسان باشد(ممکن است نام آنها یکی نباشد.)

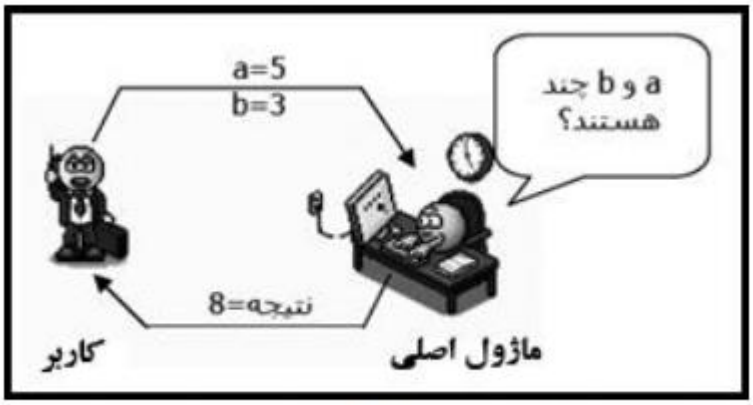
۲-در پایتون می توان تابعی را در داخل تابع دیگر تعریف کرد.

درک عملکرد تابع

برای درک عملکرد توابع، برنامه ای را بدون استفاده از توابع و سپس از طریق توابع پیاده سازی میکنیم .با همین مثال نیز چگونگی تبدیل یک برنامه معمولی به توابع را می آموزیم . فرض کنید، بخواهید برنامه ای بنویسید که دو عدد را خوانده، حاصل جمع آنها را نمایش دهد .همانطور که قبلاً دیدید، این برنامه به صورت زیر پیاده سازی میشود (روش اول:)

```
a = int (input("Enter a:")
b = int (input("Enter b:")
c = a + b
print (a, "+", b, "=", c)
```

در این برنامه، ماژول اصلی همه کاره است .یعنی، دو عدد صحیح را از کاربر(مشتری)گرفته، خودش حاصل جمع دو عدد را محاسبه کرده، چاپ میکند(در اختیار مشتری قرار میدهد .)این فرآیند در شکل زیر آمده است.



شکل ۱-۳ فرآیند اجرای برنامه.

روش دوم:

```
def addition (a, b):
  return a + b
a = int (input("Enter a:")
b = int (input("Enter b:")
c = addition (a, b)
print( a, " + ", b, " = ", c)
```

در روش دوم پیاده سازی، ماژول اصلی دو عدد a و b را از کاربر گرفته، در اختیار تابع addition (کارمند خودش)قرار میدهد تا حاصل جمع این دو عدد را حساب کند .تابع addition پس از محاسبه حاصل جمع دو عدد(مانند کارمند)نتیجه را در اختیار ماژول اصلی(مدیریت)قرار می دهد و ماژول اصلی این نتیجه را به کاربر (مشتری)میدهد.

مثال ۱-۳: برنامه ای بنویسید که تابعی بسازد که عددی دریافت کرده و آنرا به توان سه برساند(مکعب عدد)

مراحل طراحی و اجرا:

1.ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
def mokaab(x):
        return x**3
a= mokaab (int(input("enter a number:")))
print(a)
```

2ماژول را ذخیره و اجرا کرده، عدد ۴ را وارد کنید تا خروجی زیر را ببینید:

```
enter a number:4
64
```

مثال ۲-۳: **برنامه ای بنویسید که عدد n را خوانده ،سپس اعداد کامل(تام) 1تا n را نمایش دهد؟**

مراحل طراحی و اجرا:

۱. ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

تابع	متغیر	هدف
ماژول	n	عدد ورودی
اصلی	i	شمارنده از ۱ تا n
sum	s	مجموع مقسوم علیه‌ها
	i	شمارنده از ۱ تا n
	n	پارامتر ورودی
isPerfect	n	عدد به عنوان پارامتر

```
def sum(n):
  s = 0
  for i in range(1, n):
    if(n % i == 0):
      s += i
  return s
def isPerfect(n):
  return(n == sum(n))
n=int(input("Enter n:"))
for i in range(1, n + 1):
  if isPerfect(i) == True:
    print(i)
```

ماژول

را ذخیره واجرا کرده،عدد ۱۰۰۰ را وارد کنید تا خروجی زیر را ببینید:

```
Enter n:1000
6
28
496
```

مثال ۳-۳-برنامه ای که عددی را خواند ، اعداد مربعی 1 تا آن عدد را نمایش میدهد . چند عدد مربعی عبارتند از: ... 1 4 9 16 25 36 49 64

مراحل طراحی و اجرا:

1.ماژول جدیدی ایجاد کرده، دستورات آن را به صورت زیر تایپ کنید:

```
def isSquare(n):
        i = 1
        while i * i <= n:
                if (i * i == n):
                        return True
                i = i + 1
        return False
n = int(input("Enter n:"))
for i in range(1, n + 1):
        if isSquare(i) == True:
                print(i, end = '\t')
```

2.ماژول را ذخیره و اجرا کرده، عدد ۱۰۰ را وارد کنید تا خروجی زیر را ببینید:

```
Enter n:100
 1 4 9 16 25 36 49 64 81 100
```

تمرین دانشجویی: (جوابها را برای من در کانال واتساپی ارسال کنید)

۱- به کمک تعریف توابع برنامه ای بنویسید که با دریافت عدد دلخواه شعاع دایره , محیط دایره ای به آن شعاع را بدست آورد.

۲- به کمک تعریف توابع برنامه ای بنویسید که با دریافت ۳عدد دلخواه بزرگترین آنها را نمایش دهد.