فصل ۹

توابع و فراخوانی آن‌ها

در این فصل، نحوه ایجاد توابع [۱] که در Matlab به کار می‌روند، توضیح داده می‌شود و چندین مثال از آن‌ها بیان می‌گردد. توابعی که در این فصل با آن‌ها سر و کار داریم، متفاوت است با توابعی که پیش از این معرفی شدند. این توابع را خودمان به عنوان یک برنامه جدا در هر برنامه اصلی تعریف می‌کنیم. پس، توابع هم خود نوعی script هستند. این توابع آرگومان‌هایی را به عنوان ورودی می‌گیرند و خروجی را ایجاد می‌کنند. دلایل زیر را می‌توان برای استفاده از توابع در متلب ذکر نمود:

- ایجاد و ذخیره توابع پیچیده
- جلوگیری از طولانی شدن فایل اصلی برنامه
- افزایش سرعت اجرا شدن برنامه
- صرفه‌جویی در حافظه کامپیوتر در هنگام اجرای برنامه

فایل اصلی برنامه و فایل مربوط به تابع تفاوت‌هایی هم دارند:

- فایل اصلی برنامه سطر به سطر اجرا مي‌شود اما توابع یک‌بار بطور کامل اجرا مي‌گردند.
- محیط کاري فایل اصلی برنامه همان محیط کاري متلب است اما محیط کاري هر تابعي مختص خود اوست یعني اگر متغیري در یک تابع تعریف شود تنها در آن تابع قابل دسترسي است و برعکس متغیرهاي تعریف شده در محیط کاري متلب در داخل توابع تعریف شده نیستند.(مگر اینکه بصورت عمومي تعریف شده باشند)
- توابع تنها از طریق آرگومان‌هایشان با محیط خارج در ارتباطند.
- توابع دارای ساختار خاصی هستند که باید حتما رعایت شود.

نکته: بهتر است برای نوشتن یک برنامه، ابتدا فایل اصلی برنامه نوشته شود و سپس توابعی از روی آن استخراج شود. با این کار عیب‌یابی برنامه راحت‌تر خواهد بود.

[۱] functions

۱.۹ ساختار یک تابع

در هر تابع، اولین خط مربوط به تعریف تابع است. این خط نام تابع و آرگومان‌های ورودی و خروجی آن را مشخص می‌کند. ساختار کلی تعریف تابع به صورت زیر است:
دقت شود که نام تابع باید با نام فایل ذخیره شده یکسان باشد.

function [آرگومان‌های خروجی] =نام تابع (آرگومان‌های ورودی)

شکل ۱.۹: ساختار تابع

تعداد متغیرهای ورودی (برای انتقال داده‌ها به داخل تابع) و متغیرهای خروجی (برای انتقال داده‌ها به خارج تابع) می‌توانند یک یا چند رشته عددی، ماتریس یا بردار باشند.
برای اجرای تابع می‌توان برروی کلید Run کلیک نمود و با نگه‌داشتن آن اطلاعات ورودی را وارد کرد.
و یا در پنجره دستورات نام تابع به همراه مقادیر متغیرهای ورودی را وارد کرد.
متغیرهای موجود در فایل تابع، به صورت محلی هستند، به این معنی که نه از متغیرهای فضای کاری هستند و نه برروی آن‌ها تاثیر می‌گذارند.
بهتر است در انتهای فایل برنامه مربوط به تابع end گذاشته شود. همچنین بهتر است که بین آرگومان‌ها، چه در خروجی و چه در ورودی، کاما گذاشته شود.

مثال ۱ *می‌خواهیم تابعی بنویسیم که در آن، با دادن شعاع دایره، محیط و مساحت آن را محاسبه کند.*

```
circle.m
1  function[p,s]=circle(r)
2
3  -   p=2*pi*r;
4
5  -   s=pi*r^2;
6
7  -   end
8
9
```

```
>> circle(10)

ans =

   62.8319

>> [mohit masahat]=circle(10)

mohit =

   62.8319


masahat =

   314.1593
```

شکل ۲.۹: مثال ۱

در شکل مشاهده می‌شود که آرگومان ورودی تنها r، شعاع دایره، انتخاب شده است و آرگومان‌های خروجی، p و s به ترتیب، محیط و مساحت دایره، انتخاب شده‌اند. در خط اول علاوه بر آرگومان‌ها، نام تابع نیز مشخص است که در اینجا، circle انتخاب شده است. دقت کنید که نام تابع با نام فایل مربوط به تابع یکسان است. فرمول‌های مربوط به محاسبه محیط و مساحت دایره در داخل تابع نوشته شده است.
حال در پنجره دستور، می‌خواهیم مساحت و محیط دایره‌ای به شعاع ۱۰ را بدست آوریم. مشاهده می‌شود

که اگر دستور circle(10) اجرا شود، تنها یک خروجی خواهیم داشت و آن هم مربوط به اولین خروجی تعریف شده برای تابع، یعنی محیط، می‌باشد. به همین دلیل، یک بردار تعریف می‌کنیم تا خروجی مورد نظر بدست آید. ملاحظه می‌شود که نیازی نیست، خروجی‌ها دقیقا با همان نام‌های تعریف شده در تابع، فراخوانی شوند. اگر بخواهیم تنها خروجی‌های خاصی در خروجی ظاهر شوند، می‌توان به جای خروجی‌هایی که نمی‌خواهیم علامت ~ را قرار دهیم. مثلا در این مثال فرض کنید که می‌خواهیم فقط مساحت را محاسبه کنیم، داریم: دو دستور nargin [۱] و nargout [۲] به ترتیب، تعداد متغیرهای ورودی

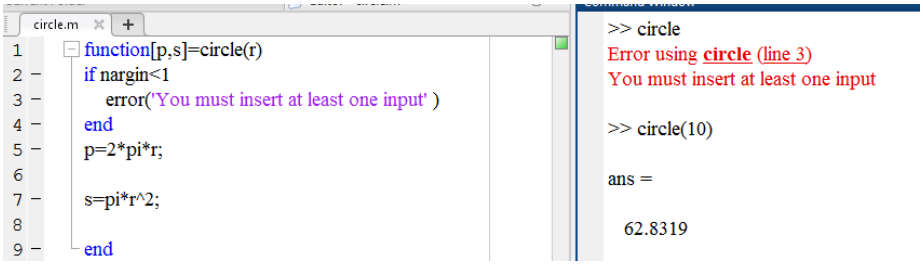
```
>> [~,masahat]=circle(10)

masahat =

   314.1593
```

شکل ۳.۹: محاسبه خروجی دلخواه

و خروجی یک تابع را می‌دهند. دقت کنید که این دو دستور حتما باید در بدنه (فضا) داخلی برنامه تابع نوشته شوند. در مثال زیر، اگر تعداد ورودی‌های تابع در فراخوانی، کمتر از یک باشد، خطا داده خواهد شد. درغیراین صورت، مقادیر محاسبه خواهند شد.

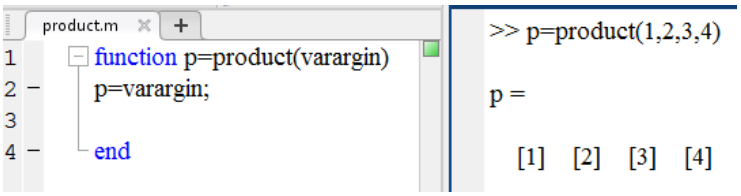


شکل ۴.۹: استفاده از دستور nargin

مثال ۲ *می‌خواهیم تابعی بنویسیم که تعداد نامشخصی عدد را در هم ضرب کند.*

در حالتی که بخواهیم تعداد نامشخصی ورودی داشته باشیم، از varargin [۳] و اگر بخواهیم تعداد نامشخصی خروجی داشته باشیم، از varargout [۴] استفاده می‌کنیم. در شکل زیر مشاهده می‌کنید که خروجی دستور varargin تنها متغیرهایی است که به عنوان ورودی در فراخوانی تابع به آن اختصاص داده شده است. پاسخ مثال ۲ در شکل زیر نوشته شده است و خروجی آن به ازای چند عدد بدست آمده است. ابتدا باید دقت شود که خروجی varargin از جنس سلول [۵] (آرایه) است. به همین دلیل در فراخوانی هر سلول باید از { استفاده نمود. در برنامه زیر، ابتدا اولین عدد ورودی در متغیر p قرار گرفته

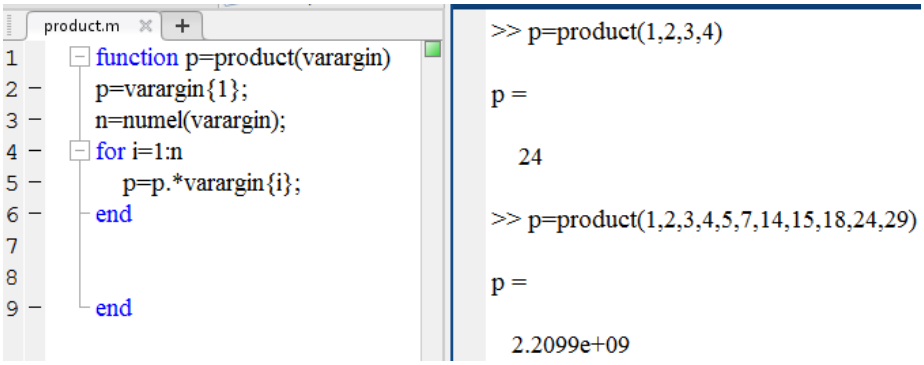
[۱] number of input argument
[۲] number of output argument
[۳] input varible
[۴] output varible
[۵] cell



شکل ۵.۹: استفاده از دستور varargin

است. سپس تعداد متغیرهای ورودی در متغیر n ذخیره شده است. در انتها در یک حلقه for تمام اعداد ورودی در هم ضرب شده‌اند.

دقت کنید که varargin همواره آخرین متغیر ورودی خواهد بود.



شکل ۶.۹: پاسخ مثال ۲