

فصل هفتم:
ایجاد توابع در متلب
Functions

دانشگاه کاشان
سید مهدی وحیدی پور

MATLAB®

کلاس آموزشی

با تشکر از عباس طبیبی-دانشگاه صنعتی اصفهان

فصل هفتم: ایجاد توابع در متلب

۷-۱- مزایای استفاده از توابع به جای فایل‌های اسکریپت

1. سرعت بالاتر

2. صرفه‌جویی در حافظه کامپیوتر

3. توسعه توانایی‌های متلب

توابع برخلاف فایل‌های اسکریپت در هنگام اجرا یکبار کامپایل شده و اجرا می‌شوند. در حالیکه فایل‌های اسکریپت سطر به سطر کامپایل و اجرا می‌گردند. این امر باعث افزایش سرعت اجرای توابع در مقایسه با فایل‌های اسکریپت می‌شود.

متغیرهای تعریف شده در توابع پس از پایان اجرای آن از حافظه پاک می‌شوند و بطور کلی فضای کاری توابع مستقل از فضای کاری متلب است. خصوصاً در مواقعی که برنامه با ماتریسهای بزرگ (مانند تصاویر) کار می‌کند بهتر است از توابع استفاده شود

فصل هفتم: ایجاد توابع در متلب

۷-۱- مزایای استفاده از توابع به جای فایل‌های اسکریپت-ادامه-

اکثر دستورات اصلی متلب و جعبه‌ابزارهای آن با استفاده از توابع نوشته شده است. به بیان دیگر به راحتی می‌توان قابلیت‌هایی که در حال حاضر در متلب وجود ندارد را با نوشتن یک مجموعه از توابع به آن افزود. همین امر باعث شده است که در دهه گذشته قابلیت‌های متلب در رشته‌های مختلف علمی و فنی با سرعت چشمگیری توسعه یابد.

نکته: بهتر است در هنگام نوشتن یک برنامه آنرا بصورت اسکریپت بنویسیم تا اشکالزدایی آن آسانتر باشد اما پس از کامل شده برنامه آنرا به فانکشن تبدیل کنیم تا سرعت و کیفیت آن افزایش یابد.

فصل هفتم: ایجاد توابع در متلب

۷-۲- تفاوت‌های توابع و فایل‌های متنی

1. فایل‌های متنی سطر به سطر ترجمه و اجرا می‌شوند اما توابع یکبار بطور کامل ترجمه و سپس اجرا می‌گردند.
2. محیط کاری فایل‌های متنی همان محیط کاری متلب است اما محیط کاری هر تابعی مختص خود اوست یعنی اگر تغییری در یک تابع تعریف شود تنها در آن تابع قابل دسترسی است و برعکس متغیرهای تعریف شده در محیط کاری متلب در داخل توابع تعریف شده نیستند. (مگر اینکه بصورت عمومی تعریف شده باشند)
3. توابع تنها از طریق آرگومان‌هایشان با محیط خارج در ارتباطند

فصل هفتم: ایجاد توابع در متلب

۳-۷- نحوه ایجاد توابع

تنها تفاوت ظاهری یک تابع و یک فایل متنی آن است که سطر اول یک تابع با کلمه کلیدی **function** شروع می‌شود که شکل کلی آن بصورت زیر است:

```
function [argout1 , argout2, ... ] =  
    funcname(argin1,argin2,...)
```

% معرفی فانکشن در یک سطر

% راهنمای استفاده

% از این فانکشن

% نویسنده فانکشن ، نسخه و سال ساخت

بدنه تابع

...

فصل هفتم: ایجاد توابع در متلب

۳-۷- نحوه ایجاد توابع - ادامه -

نکات:

1. تابع ممکن است هیچ آرگومان ورودی یا خروجی نداشته باشد.
2. اولین سطر بعد از اعلان تابع، یک جمله توضیحی است که در هنگام استفاده از دستور **lookfor** در متلب مورد جستجو قرار می‌گیرد
3. تمامی سطرهای توضیحی تا نخستین سطر غیر توضیحی در هنگام استفاده از دستور **help** نمایش داده می‌شود.

نکته: بهتر است هنگام نوشتن یک تابع حتما یکی دو سطر در مورد نحوه استفاده از آن و عملکرد آن توضیح داده شود تا کاربر بتواند با استفاده از دستور **help** متلب با روش استفاده از آن تابع و قابلیت‌های آن آشنا شود.

فصل هفتم: ایجاد توابع در متلب

۷-۴- فرمانهای error و return

با استفاده از این دو دستور می‌توان اجرای یک تابع را پیش از رسیدن به انتهای آن متوقف کرد. تفاوت دستور error با دستور return آن است که دستور error می‌تواند یک پیغام خطا نیز بمنظور آگاهسازی کاربر نمایش دهد.

مثال:

```
s= input('Please enter a scalar value= ');  
if length(s) > 1  
    error('Error! Your input isn"t a scalar!');  
end  
a= linspace( 0 , abs(s) , 100);
```

فصل هفتم: ایجاد توابع در متلب

۷-۵- تعیین تعداد آرگومانهای بکار رفته در حین اجرا

در متلب می‌توان توابع را با تعداد آرگومان کمتر از تعداد آرگومان موجود در تعریف تابع نیز فراخوانی کرد. مثلاً تابع size در متلب با دو آرگومان نوشته شده است اما با یک آرگومان نیز قابل اجراست که البته مقدار بازگشتی به تعداد آرگومانهای مورد استفاده بستگی خواهد داشت.

در صورتیکه بخواهیم از تعداد آرگومانها در حین اجرا مطلع شویم باید از توابع nargin و nargout به ترتیب برای تعداد آرگومانهای ورودی و تعداد آرگومانهای خروجی استفاده کنیم.

همچنین توابع nargchk و nargoutchk تعداد آرگومانهای ورودی و خروجی را چک می‌کنند و در صورتیکه با تعداد درخواست شده برابر نباشند پیام خطای مناسب را نشان می‌دهند.

فصل هفتم: ایجاد توابع در متلب

۶-۷- نکاتی در مورد توابع

- در یک فایل می‌توان بیش از یک تابع تعریف کرد. در اینصورت تمامی این توابع می‌توانند یکدیگر را فراخوانی کنند اما تنها نخستین تابع از خارج از این فایل قابل فراخوانی است.
- نام فایل با نام نخستین تابع آن باید یکسان باشد. در غیر اینصورت بمنظور اجرای تابع باید از نام فایل به جای نام تابع استفاده گردد که البته کار درستی نیست.

فصل هفتم: ایجاد توابع در متلب

مثال ۷-۱- تابعی بنویسید که یک بردار(آرایه سطری یا ستونی) را از کاربر بگیرد و مراحل زیر را انجام دهد:

- تعداد آرگومان ورودی و خروجی که توسط کاربر وارد شده است را چک کند و در صورتیکه تعداد آرگومان ورودی بیشتر یا کمتر از یک و تعداد آرگومان خروجی بیشتر از یک باشد، پیام خطا نمایش داده از تابع خارج شود.
- ابعاد آرگومان ورودی را چک کند و در صورتیکه آرایه‌ای غیر سطری یا غیر ستونی باشد(یعنی در صورتیکه به جای بردار، ماتریس باشد)، با پیام خطا از تابع خارج شود.
- عبارت زیر را بر روی مقادیر ورودی اعمال نموده به عنوان خروجی بازگرداند.

$$y=2\exp(4x^2)+3\sin(2\pi x)+10$$

- تعداد آرگومان خروجی را چک کند و در صورتیکه برابر با صفر باشد، نمودار تغییرات y در مقابل x را رسم کند.(راهنمایی: برای رسم نمودار از تابع $\text{plot}(x,y)$ استفاده کنید.

فصل هفتم: ایجاد توابع در متلب

تکلیف ۷-۱- تابعی بنویسید که یک عبارت ریاضی دلخواه را از کاربر (به صورت یک رشته کاراکتری) به عنوان آرگومان اول و یک آرایه را به عنوان آرگومان دوم بگیرد و :

- چک کند که تعداد آرگومان ورودی دقیقا دو عدد باشد (با استفاده از تابع `nargchk`)
- چک کند که تعداد آرگومان خروجی دقیقا یک عدد باشد. (با استفاده از تابع `nargoutchk`)
- چک کند که آرگومان اول حتما یک رشته کاراکتری باشد و آرگومان دوم حتما یک متغیر عددی. (از توابع `isstr` و `isnumeric` استفاده کنید)
- با استفاده از تابع `eval` عبارات ریاضی وارد شده توسط کاربر را بر روی تمامی عناصر آرایه ورودی اعمال نموده، باز گرداند.