

جدول ۱-۱: توابع سازنده ماتریس ها

ماتریس همانی ایجاد می کند	eye	1
ماتریس صفرها را ایجاد می کند	zeroes	2
ماتریس یک ها را ایجاد می کند	ones	3
ماتریس قطری ایجاد می کند	diag	4
ماتریس بالا مثلثی ایجاد می کند	triu	5
ماتریس پایین مثلثی ایجاد می کند	tril	6
ماتریس تصادفی ایجاد می کند	rand	7
ماتریس هیلبرت ایجاد می کند	hilb	8
ماتریس جادویی ایجاد می کند	magic	9
لیستی از ماتریس های معروف ایجاد می کند	gallery	10

جدول ۲-۲: عملگرهای ماتریسی

جمع	+	1
تفریق	-	2
ضرب اسکالر و آرایه	*	3
ضرب جزء به جزء آرایه ها	.*	4
تقسیم خارج قسمت	/	5
تقسیم بک اسلش (چپ یا معکوس)	\	6
تقسیم جزء به جزء آرایه ها	./	7
تقسیم بک اسلش جزء به جزء آرایه ها	.\	8
توان	^	9
توان جزء به جزء آرایه ها	.^	10
دو نقطه؛ بردارهای با مولفه های متساوی الفاصله ایجاد می نماید، برد اعضا آرایه ها را نشان می دهد	:	11
عمالگر تخصیص یا انتساب	=	12
پرانتز (حق تقدم در عملیات متلب، آرگومان های ورودی در توابع و اندیس های آرایه ها را در بر می گیرد	()	13
براکت؛ برای تشکیل آرایه ها آرگومان های خروجی در توابع را در بر می گیرد	[]	14
کاما، اندیس های آرایه و آرگومانهای تابع را جدا می کند، دستورات را در یک خط از هم جدا می کند	,	15
سیمکالون؛ توقف نمایش تمام سطرها در آرایه	;	16
کوئیشن؛ ترانهاده ماتریس ایجاد رشته، ترانس پوز کردن	,	17
سه نقطه ادامه خط	...	18
درصد، نوشتن یک شرح، تعیین فرمت خروجی	%	19

جدول ۲-۳: عملگرهای رابطه ای منطقی

کوچکتر از	<	1
بزرگتر از	>	2
کوچکتر یا مساوی	<=	3
بزرگتر مساوی	>=	4
مساوی	==	5
نامساوی	~=	6
AND (و) منطقی	&	7
OR (یا) منطقی		8
NOT (نه) منطقی	~	9

جدول ۱-۳: دستورهایی کنترلی

پاک کردن پنجره command	clc	1
حذف تمام متغیرها در حافظه	clear	2
حذف متغیرهای x، y، z از حافظه	clear xyz	3
بستن یک فایل	fclose	4
باز کردن یک فایل	fopen	5
اعلام متغیرهای جهانی (عام)	global	6
نمایش help (راهنما) مربوط به توابع متلب	help	7
جستجوی کلمه مشخص در help متلب	lookfor	8
نمایش متغیرهای فعلی موجود در حافظه	who	9
نمایش اطلاعات در متغیرهای در حافظه	whos	10

جدول ۴-۲: نوع فراخوانی داده در Matlab

دستورهای مربوط به فراخوانی دادها	نوع فایل	
Aa=xlsread ('e: / Nam File.xlsx'); برای فراخوانی داده های excel حتما باید از دستور xlsread بعد از عنوان فایلی (در اینجا Aa) که قرار است فایل مورد نظر در آن قرار گیرد آورده شود. (لازم به ذکر است که حتما کوتیت ('دستور فراخوانی') در دو طرف جمله فراخوانی شده آورده شود	Excel	1
load('E:\IN\data\month\shamsi.mat') این دستور به این شرح است که می آید لود می کند فایل مورد نظر را از درایو e سپس در فایل IN\data\month\ فایل به اسم شمسی و با پسوند mat یا متلب را فراخوانی	Matlab	2

می‌کند.		
با گرفتن فایل مورد نظر در جعبه Workspace سپس گزینه Matrix و در نهایت روی Input کلیک می‌کنیم تا فایل مورد نظر انتقال داده شود.	txt	3

جدول ۲-۹:

کاربرد	دستور find	
در این دستور عدهای بزرگتر و مساوی X را می‌دهد و شماره ردیفهای که اعداد آنها دارای ارزشی بالای X هستند در فایل m و شماره ستون های که ارزشی بالای X دارند در فایل n ذخیره می‌کند. (نکته در این جا به جای X اعداد مورد نظر می‌باشد. برای مثال می‌توان به جای X عدد ۲ یا ۴ یا هر عدد دیگر را قرار دهیم؛ حالا اگر به جای آن ۲ قرار دهیم شکل دستوری آن به شکل زیر می‌باشد...): [m n]=find(P(:,:)>=2);	[m n]=find(Nam File(:,:)>=x);	1
[m n]= find (a (:,:) >=mean (a (:))) [m n]=(find(a(:,:)>=prctile(a(:,:),90)))	نمونه مشابه دستور find	
f=find(shamsi(:,5)==1380 & (shamsi(:,6)==2 & (shamsi(:,7)==23)));		
>> nam=find((ah(:,1)>=91)&(ah(:,1)<=99) (ah(:,1)==29) (ah(:,1)==17));		
>> nam=find((ah(:,1)>=91)&(ah(:,1)<=99)... (ah(:,1)==29) (ah(:,1)==17));		
این دستور (a) بیانگر این است که از nam file از همه ردیفها و ستون اول عدهای بزرگتر مساوی با ۱۳۴۰ و کوچکتر مساوی با ۱۳۵۰ را استخراج می‌کند. بع عبارت دیگر کاربرد این دستور این است که در دوبازه مورد نظر اعداد خواسته شده را می‌دهد که در این دستور بازه مورد نظر ۱۳۴۰ تا ۱۳۵۰ می‌باشد یعنی ۱۳۴۱، ۱۳۴۲.....۱۳۴۹ را می‌دهد. (در این دستور "&" به معنی "و" می‌باشد)	a=find((nam file (:,1)>=1340) & (nam file (:,1)<=1350));	
دستور اول: با این دستور دو فایل m و n در یک فایل قرار می‌گیرند. شرط استفاده از این دستور این است که size (اندازه) دوفایل m و n مساوی باشد. به عبارتی R=C یعنی ردیفها و ستونها یکسانی داشته باشند. دستور دوم: در این دستور می‌آید در ستون دوم فایل m فایل n را قرار می‌دهد. شرط دستور اول برای دستور دوم صادق می‌باشد.	نام=[m,n]; دستور اول or m(:,2)=n; دستور دوم	2
این دستور برای ترانسپوز کردن می‌باشد. در واقع این دستور جای سطر و ستون را عوض میکند. به عبارت دیگر جای سطرها را به جای ستون ها و جای ستون های را به جای سطرها می‌دهد.	Nam = a';	3
B=reshape(fail nam,1,r*c); فرض کنید ماتریسی داشته باشیم که بخواهیم اندازه (ابعاد) آن را تغییر دهیم، برای این منظور، می‌توانیم از دستور reshape در متلب، استفاده کنیم. البته باید دقت داشته باشیم که تعداد عناصر ماتریس اولیه و تعداد عناصر ماتریسی که ساخته می‌شود، باید برابر باشد. دستور B دارد فایل مورد نظر را در یک ردیف پشت سر هم مرتب می‌کند C = reshape(a,3,4);	دستور ردیفی ماتریس	

<p>$a=3*4$; این دستور (C) این را بیان می کند که می آید فایل a را در سه ردیف ۴ تای مرتب می کند. از آنجای که اندازه ماتریس a، ۳ در ۴ یعنی ۱۲ می باشد. بنابراین اگر بخواهیم فایل را در دو ردیف مرتب کنیم باید به جای عدد ۳ عدد ۲ قرار گیرد و به جای عدد ۴ عدد ۶ قرار گیرد آنگاه فایل مورد نظر را در دو ریف ۶ تای مرتب می کند.</p> <p>$B = \text{reshape}(a,x,[])$; در این دستور با قرار دادن اعداد به جای x فایل مورد نظر را در ردیف های مورد نظر مرتب می کند. برای مثال اگر اندازه فایل a برابر با ۱۲ باشد و به جای x عدد ۳ گذاریم آنگاه فایل مورد نظر را در ۳ ردیف ۴ تای مرتب می کند.</p>	
<p>برای این اکار ابتدا فایل مورد نظر را ترانسپوز می کنیم سپس تما مراحل بالا که برای ردیفی کردن شرح داده شد برای ستونی کردن هم انجام می دهیم؛ این بار فایل مورد نظر را به شکل ستونی زیر هم مرتب می کند.</p>	دستور ستونی کردن
<p>این دستور از ماتریس a ستون x را حذف می کند. در اینجا x می تواند هر عددی یا ستون مورد نظری باشد که بخواهیم حذف کنیم.</p>	$\begin{bmatrix} \\ \\ \end{bmatrix}$ $a(:,x)=[]$
<p>در این دستور ماتریسی درست می کند که ابتدا آن با عدد a و انتهای آن عدد b می باشد. حرف C یعنی اینکه به C قسمت تقسیم می کند و فاصله آن از طریق b/c (تقسیم بر C) بدست می دهد. برای مثال اگر به جای a عدد ۱، به جای b عدد ۴۸ و به جای C ۱۲ قرار دهیم؛ ماتریسی تشکیل می دهد که ابتدای آن با عدد ۱ و انتهای آن با ۴۸ و به ۱۲ قسمت تقسی و فاصله آنها ۴۸ تقسیم بر ۱۲ (یعنی ۴) می باشد.</p>	$A = \text{linspace}(a,b,c)$
<p>روش دیگر برای مرتب کردن ماتریس، یا به عبارتی همه سطرهای یک ماتریس را در یک ستون قرار می دهد. برای مثال اگر ماتریس A اندازه آن ۲ در ۳ باشد. ابتدا باید سطر و ستون را در هم ضرب کنیم که برای این کار یا ۲ در ۳ ضرب شود تا جواب ۶ حاصل می شود. دستور ZZ دارد اندازه ضرب سطر در ستون را نشان می دهد بنابراین چه سطر و ستون را در هم ضرب کنیم چه از دستور $zz=\text{numel}(A)$ استفاده کنیم جواب یکی حاصل می شود.</p>	$A=(2*3)$; $zz=\text{numel}(A)$; $\text{moratab}=A(1:zz)$; یا $\text{Moratab}=(1:6)$;
<p>این دستور دارد این را بیان می کند که بیابید از تاریخ ها فایل آدرس ها را از فولدر ader2 ردیف ۱ تا ۱۳۲ و ستون اول آن را از فایل همه ستون تاریخ ها استخراج کند</p>	$\text{Month}=\text{tarikh}(\text{ader2}(1:132, 1, :))$;

	دستوراتی برای اندازه ماتریس
<p>در متلب، با استفاده از دستور length، می توانیم طول یک بردار (تعداد کل عناصر بردار) را محاسبه کنیم. به مثال زیر توجه کنید:</p> <p>برای مثال اگر ماتریس $a=\text{zeros}(7,3)$ داشته باشیم با اجرای دستور BB جواب عدد ۷ را می دهد.</p> <p>البته دستور length، برای ماتریس ها نیز می تواند مورد استفاده قرار بگیرد. اگر دستور length را برای یک ماتریس به کار ببریم، این دستور، تعداد ردیف ها و تعداد ستون های ماتریس را محاسبه می کند و هر کدام از این دو عدد که بزرگتر باشد را در خروجی نمایش خواهد داد. به مثال زیر توجه کنید :</p>	$BB=\text{length}(a)$

<p>مثال: $A=[1\ 2\ 3;4\ 5\ 6]$ $B=length(A)$ که با اجرای دستور B عدد ۳ حاصل می شود</p>		
<p>فرض کنید که ماتریسی داریم که می خواهیم بدانیم اندازه آن چقدر است، برای این منظور، دستور $size$ در متلب، مورد استفاده قرار می گیرد. به مثال زیر توجه کنید:</p> <p>$A=[1\ 2\ 3;4\ 5\ 6]$ $B=size(A)$</p> <p>حالا اگر دستور را اجرا کنیم عدد ۲ و ۳ را می دهد یعنی اندازه ردیف ها ۲ و ستون ها ۳ می باشد. مشاهده می کنید که دستور $B=size(A)$، تعداد ردیف ها و تعداد ستون های ماتریس A را محاسبه کرده است و به ترتیب، عدد مربوط به آنها را در متغیر B ذخیره کرده است.</p>	<p>$B=size(a)$</p>	
<p>$B=length(a(:,1))$ $B=size(a(:,1))$ $m=size(a,1)$</p>	<p>دستورهای مشابه (برای ستون ها)</p>	
<p>$B=length(a(1,:))$ $B=size(a(1,:))$ $m=size(a,2)$</p>	<p>دستورهای مشابه (برای ردیف ها)</p>	
<p>این دستور همانند دستور $size$ می باشد با این تفاوت که در دستور $size$ دو عدد را می دهد که اولی ردیف و دومی اشاره به ستون ها می باشد اما در با اجرای دستور $minfo$ جوابی که حاصل می شود به این شکل زیر است:</p> <p>$n\ rows\ m\ cols: regular\ MATLAB\ matrix$</p> <p>به طوری که در دستور بالا به جای n ردیف ها و به جای m ستون ها را می دهد.</p>	<p>$minfo(a)$</p>	
<p>در متلب، با استفاده از دستور $numel$، می توانیم تعداد کل عناصر یک ماتریس را محاسبه کنیم. به مثال زیر توجه کنید:</p> <p>$A=[1\ 2\ 3;7\ 8\ 9]$ $B=numel(A)$</p> <p>که با اجرای دستور B عدد ۶ حاصل می شود</p> <p>یعنی تعداد کل عناصر ماتریس A را اگر بشماریم بع عدد ۶ می رسیم (لازم به ذکر است که این دستور در حین اسکریپت نویسی کاربرد فراوانی دارد مخصوصا دستورات if یا شرطی)</p>	<p>$numel$</p>	
<p>در متلب، از دستور $triu$، برای ساخت ماتریس بالامثلثی و از دستور $tril$، برای ساخت ماتریس پائین مثلثی استفاده می شود.</p> <p>یک ماتریس بالامثلثی، ماتریسی می باشد که عناصری از آن که زیر قطر اصلی قرار دارند، برابر با ۰ باشند. چنانچه از دستور $triu$ برای یک ماتریس استفاده کنیم، آنگاه دستور $triu$، آن ماتریس را به یک ماتریس بالا مثلثی تبدیل خواهد کرد. به مثال زیر توجه کنید:</p>	<p>$triu\ and\ tril$</p>	
<p>دستور horzcat برای الحاق سطری دو ماتریس:</p> <p>چنانچه بخواهیم در متلب، دو ماتریس را به صورت سطری، به یکدیگر متصل (الحاق) کنیم، باید از دستور $horzcat$ استفاده کنیم. باید دقت شود که تعداد ردیف های دو ماتریس با هم برابر باشد. به مثال زیر توجه کنید:</p> <p>$A=[1\ 2;3\ 4]$ $B=[5\ 6;7\ 8]$ $C=horzcat(A,B)$ $C=$</p> <p>1 2 5 6</p>	<p>الحاق سطری و الحاق ستونی دو ماتریس با دستور $horzcat$ و دستور $vertcat$</p>	

<p style="text-align: center;">3 4 7 8</p> <p>دستور vertcat برای الحاق ستونی دو ماتریس :</p> <p>از دستور vertcat در متلب، برای اتصال (الحاق) دو ماتریس به صورت ستونی، استفاده می شود. باید دقت داشته باشیم که تعداد ستون های دو ماتریس، برابر باشد تا الحاق دو ماتریس امکان پذیر باشد. به مثال زیر توجه کنید:</p> <pre>A=[1 2;3 4] B=[5 6;7 8] C= vertcat (A,B) C= 1 2 3 4 5 6 7 8</pre>		
<p>گاهی اوقات نیاز داریم که چک کنیم آیا عناصر یک ماتریس، عضو ماتریسی دیگر نیز هستند یا نه. برای این منظور، از دستور ismember در متلب، استفاده می شود. با دستور ismember(A,B) می توانیم چک کنیم که آیا عناصر ماتریس B، عضو ماتریس A نیز می باشند یا نه. دستور ismember، ماتریسی را با اندازه ماتریس B برمی گرداند که تنها حاوی مقادیر ۱ و ۰ می باشد. هر عنصری از ماتریس B که در ماتریس A عضو بوده است، در محل آن عنصر، مقدار یک (۱) نمایش داده شده است و چنانچه آن عنصر عضو ماتریس A نباشد، در محل آن عنصر، مقدار صفر (۰) نمایش داده شده است. به مثال زیر توجه کنید :</p> <p style="text-align: right;">مثال:</p> <pre>A=[1 2 3;4 5 6;7 8 9] B=[1 2;3 4] C=[1 2;3 10] D=ismember(B,A) E=ismember(C,A) D= 1 1 1 1 E= 1 1 1 0</pre>	ismember	
<p>گاهی نیاز داریم که بدانیم چه عناصری از یک ماتریس برابر صفر هستند یا خیر و همچنین ممکن است بخواهیم تعداد عناصر صفر یا غیرصفر ماتریس و موقعیت آنها را بدانیم. در ادامه دستوراتی را برای این منظور معرفی می کنیم.</p> <p>با دستور nnz در متلب، می توانیم تعداد عناصر غیرصفر یک ماتریس را مشخص کنیم. به مثال زیر توجه کنید :</p> <p style="text-align: right;">مثال:</p> <pre>A=[0 1 2;3 0 4] B=nnz(A)</pre>	nnz	

<p>با اجرای دستور بالا عدد ۴ حاصل می شود یعنی ۴ عدد و جود دارد که مقدار آنها صفر نیست.</p> <p>نکته:</p> <p>می دانیم با دستور numel در متلب، می توانیم تعداد کل عناصر یک ماتریس را محاسبه کنیم، بنابراین با استفاده از تفاضل نتیجه دو دستور numel و nnz ، می توانیم تعداد عناصری از ماتریس را که برابر صفر می باشند، تعیین کنیم .به مثال زیر توجه کنید :</p> <pre>A=[0 1 2;3 0 4] B=numel(A) C=nnz(A) D=B-C</pre> <pre>B= 6 C= 4 D 2</pre> <p>مشاهده می کنید که B ، برابر تعداد کل عناصر ماتریس A و C برابر تعداد عناصر غیرصفر ماتریس A و D برابر تعداد عناصر برابر صفر از ماتریس A می باشد.</p>		
<p>گاهی نیاز داریم که خود مقدار عناصر غیرصفر یک ماتریس را بدانیم. دستور nonzeros مقدار عناصر غیرصفر ماتریس را، به صورت یک بردار، در خروجی نمایش می دهد. به مثال زیر توجه کنید :</p> <p>مثال:</p> <pre>A=[0 1 2;3 0 4] B=nonzeros(A) B= 3 1 2 4</pre>	nonzeros	
<p>شاید قبلا با دستور pause آشنا شده باشید و بدانید که مثلا دستور (pause ۱۵)، در میان کدهای یک برنامه، باعث می شود که نرم افزار متلب، ۱۵ ثانیه توقف کند و سپس اجرای سایر دستورات را ادامه بدهد. اما در برخی مواقع، ممکن است کاربر بخواهد در میانه برنامه، توقفی وجود داشته باشد و در عین حال بتواند با دستوراتی، بعضی از متغیرها و یا مواردی دیگر از برنامه را چک کند و در صورت نیاز، آنها را با دستوراتی که در صفحه Command می نویسد، تغییر دهد .برای این منظور، باید دستور keyboard در میانه برنامه نوشته شود. زمانی که نرم افزار متلب، به دستور keyboard برسد، به سراغ دستورات بعدی نخواهد رفت و متوقف خواهد شد،</p>	keyboard	

<p>سپس در پنجره Command ، علامت <<K نمایش داده خواهد شد. در این زمان، شما می توانید دستورات خود را در پنجره Command نوشته و اجرا کنید. برای آنکه به متلب اعلام کنید که اجرای برنامه را ادامه بدهد، باید در پنجره Command، کلمه return را تایپ کرده و کلید enter از کیبورد را فشار بدهید. به مثال زیر توجه کنید :</p> <p>مثال:</p> <pre>A=2 B=3 C=A+B keyboard B=C^2</pre> <p>نتیجه:</p> <p>متلب دو دستور اول را اجرا می کند و سپس اجرای دستورات متوقف شده و در پنجره Command ، علامت <<K نمایش داده می شود. در این زمان، هر دستوری را می توانید اجرا کنید. مثلا می توانید مقدار جدیدی برای متغیر C تعریف کنید (مثل C=10). برای ادامه اجرای برنامه، در پنجره Command کلمه return را تایپ کرده و کلید enter از کیبورد را فشار بدهید .</p>	
<p>در صورتی که در دستورات save و load ، هیچ آدرسی را مشخص نکنیم، save و load کردن فایل ها، در فولدر مشخص شده در پنجره Current Folder ، صورت می گیرد و فایل ها در آن محل، ذخیره و یا بازخوانی می شوند. روش ساده برای تغییر این فولدر، کلیک بر روی بالای پنجره و تغییر مسیر به صورت دستی می باشد، اما ممکن است در برنامه ای مجبور شویم که فایل هایی را در فولدرهای مختلف save و یا load کنیم، بنابراین باید این عمل را با دستوراتی در وسط برنامه اجرا کنیم. دستور لازم برای تغییر فولدر مربوط به Current Folder ، دستور cd می باشد. به مثال زیر توجه کنید :</p> <p>مثال:</p> <p>فرض کنید بخواهیم آدرس فولدر مربوط به Current Folder را به e:\Project تغییر دهیم، می نویسیم :</p> <pre>cd e:\Project</pre> <p>نتیجه:</p> <p>چنانچه به آدرس بالای پنجره Current Folder نگاه کنید، مشاهده خواهید کرد که آدرس آن به آدرس e:\Project تغییر یافته است .</p>	cd
<p>اگر با نرم افزار متلب کار کرده باشید، حتما تاکنون صدای ((بیپ)) را در هنگام وقوع یک خطا شنیده اید. در واقع این صدا در متلب، به عنوان یک هشدار در نظر گرفته شده است.</p>	beep

<p>نرم افزار متلب این امکان را به وجود آورده است که بتوانیم این صدا را در هر مرحله از اجرای برنامه خود که می خواهیم، به صدا درآوریم. دستور beep در متلب، برای پخش این صدا به کار می رود. برای به کار بردن دستور beep ، تنها کافی است که آن را در خطی از کدهای برنامه بنویسید و زمانی که متلب به دستور beep برسد، صدای ((بیپ)) شنیده خواهد شد. ممکن است بخواهیم اگر نتایج قسمت های اول برنامه، به صورت مقادیری غیر از مقادیر مورد نظرمان بود، این صدا شنیده شود تا از ادامه اجرای برنامه جلوگیری کنیم. حتی این صدا می تواند برای اعلام پایان یافتن اجرای کدهای برنامه نیز به کار رود و تنها کافی است که دستور beep را در آخرین خط از کدهای برنامه خود بنویسیم. کدهای مثال زیر را اجرا کنید تا یک بار دیگر این صدای آشنا را بشنوید:</p>	
---	--

<p>ممکن است که بخواهید برنامه نوشته شده خود را در اختیار فردی قرار دهید تا از آن استفاده کند، اما چون زحمت زیادی برای کدنویسی آن کشیده اید، بنابراین دوست داشته باشید که وی نحوه نوشته شدن کدها را نبیند، یا به عبارتی سو استفاده از برنامه نشود در این مواقع می توانید از دستور pcode در متلب استفاده کنید. دستور pcode در متلب، این امکان را به ما می دهد که از برنامه ای که در یک m-file با پسوند m نوشته اید، فایلی دیگر با پسوند p تولید کنیم، که این فایل اگرچه در متلب قابل اجرا شدن می باشد، اما کسی نمی تواند کدهای نوشته شده در آن را ببیند. مراقب باشید که خود شما هم نمی توانید کدهای درون فایل با پسوند p را ببینید، بنابراین همیشه فایل اصلی را که دارای پسوند m می باشد را نزد خود نگهدارید. نام m-file را به دلخواه، برای مثال data انتخاب می کنیم. حال m-file را می بندیم و چک می کنیم که آدرسی که در پنجره Current Folder نوشته شده است، حتما آدرس فولدر حاوی m-file باشد. سپس کد زیر را در Command نوشته و اجرا می کنیم :</p> <p>pcode data</p> <p>با اجرای کد بالا، یک فایل با نام data.p ساخته می شود. این فایل، توسط متلب، قابل باز شدن نیست و کدهای آن را نمی توان دید، اما می توانید آن را با متلب اجرا کنید، به این صورت که در پنجره Command، اسم data را نوشته و سپس کلید enter را فشار می دهیم تا برنامه درون فایل data.p اجرا شود . دقت کنید که قبل از آن، فایل data.m را بردارید تا مطمئن شوید که فایل data.p اجرا شده است.</p>	<p>pcode (name program) رمز گذاری برای برنامه متلب</p>
<p>گاهی اوقات، در نتیجه یک سری محاسبات، ممکن است که تعدادی از عناصر ماتریس، به صورت NaN (یک عدد نیست) در آیند. اما وجود NaN ، می تواند باعث شود سایر محاسبات مورد نظر خود را نتوانیم بر روی ماتریس انجام دهیم. در اینگونه مواقع، نیاز داریم که NaN ها را در ماتریس یافته و به جای آنها، مقدار صفر قرار بدهیم .</p> <p>برای آشنایی با نحوه تبدیل NaN های یک ماتریس، به مقدار صفر، به مثال زیر توجه کنید :</p>	<p>NaN</p>

<pre>A=[NaN 2;3 NaN] A(isnan(A))=0 A = 0 2 3 0</pre>	
<p>در متلب، معمولا مقدار متغیرها در همان ابتدای برنامه و با علامت تساوی، تعیین می شود، اما روش دیگری نیز برای اختصاص مقدار به متغیرها وجود دارد و می توان تعیین مقدار به متغیرها را در بین اجرای کدهای برنامه نیز انجام داد. دستور input در متلب، برای این منظور به کار می رود. زمانی که متلب به دستور input برسد، منتظر خواهد ماند تا کاربر مقدار مورد نظرش برای آن متغیر را در پنجره Command وارد کند و سپس زمانی که کاربر کلید enter از کیبورد را فشار دهد، متلب مقدار وارد شده توسط کاربر را به متغیر اختصاص خواهد داد و سپس بقیه کدهای برنامه را اجرا خواهد نمود.</p>	input
<p>برای تعیین زمان اجرای برنامه و یا بخشی از برنامه، از دستورهایی tic و toc استفاده می شود. همان طور که مشخص است، نام این دو دستور از ((تیک تاک ((ساعت گرفته شده است. این دستورات، یکی از مهمترین دستورات متلب می باشند. قبل از هر چیز، اهمیت این دستورات را برای شما شرح می دهیم :</p> <p>معمولا در پروژه های پیچیده، زمان اجرای برنامه، بسیار طولانی می شود. در چنین مواقعی، برنامه نویس ممکن است که کنجکاو باشد که یکی از موارد زیر را بداند :</p> <p>1- کدام بخش برنامه است که بیشترین زمان را به خود اختصاص داده است. با دانستن این موضوع، برنامه نویس سعی می کند که کدهای آن بخش از برنامه را بهینه سازی کند تا در زمان صرفه جویی شود .</p> <p>2- ممکن است بخواهیم بدانیم که زمان اجرای برنامه چقدر است، اما نمی خواهیم که مجبور باشیم همین طور بنشینیم و ببینیم چه زمانی برنامه پایان می یابد. مثلا شب قبل از خواب، برنامه را اجرا می کنید و صبح که بیدار شوید، می توانید در پنجره Command ، میزان زمان اجرای برنامه را ببینید .</p> <p>3- در بسیاری از برنامه ها، حلقه های for هستند که زمان زیادی می گیرند. فرض کنید حلقه ای داریم که پارامتر آن از ۱ تا ۱۰۰۰ می باشد. با قرار دادن دستور در ابتدا و انتهای حلقه (ابتدا و انتهای درون حلقه) می توانیم زمان یک بار اجرا را محاسبه کنیم و سپس آن را ضرب در ۱۰۰۰ کنیم، چنانچه زمان بیش از حد زیاد باشد، می توانیم همین حالا آن را متوقف کنیم (با گرفتن کلید Ctrl و فشار دادن کلید C از کیبورد)، به جای آن که پس از اجرای نیمی از برنامه، به این نتیجه برسیم .</p> <p>چنانچه بخواهید زمان اجرای کل برنامه را متوجه شوید، تنها کافی است که دستور tic در اولین خط از برنامه و دستور toc در آخرین خط از برنامه قرار گیرد.</p>	tic toc
<p>چنانچه بخواهید مرورگر اینترنت متلب باز شده و صفحه خاصی از اینترنت را نمایش بدهد، باید آدرس آن صفحه اینترنتی را پس از web بنویسید. به مثال زیر توجه کنید :</p>	web

مثال:

[web http://www.kelidestan.com](http://www.kelidestan.com)

اشتباهات رایج در متلب

در اینجا اغلب اشتباهات که در هنگام استفاده از Matlab رخ می دهد بیان شده است. استفاده از ^، / و * به جای ^، /، و *؛ در زمانی که سعی دارید بردارها یا ماتریس ها را به صورت عنصر به عنصر ضرب و تقسیم یا به توان برسانید، باید نقطه را قبل از این سه علامت قرار دهید. گاهی اوقات نیز ممکن است نقطه در زمانی که نیاز نیست استفاده شود. مثلاً برای ورودی و دستورات نمادین نیازی به نقطه است. عدم تطابق پرانتزها، کروشه و دیگر علامات. MATLAB معمولاً به طور دقیق به این اشتباهات اشاره می کند، اگرچه ممکن است تفسیر این پیغام خطا سخت باشد، مخصوصاً اگر علامت؛ در پایان یک رشته حذف شود. ایجاد خطوط ورودی و خروجی بسیار طولانی. اگر چه در این موضوع در صحت پردازش MATLAB تاثیری ندارد، اما می تواند خروجی چاپ شده را سخت و غیر قابل خواندن کند. با توجه به نوع قالب نمایش خروجی، اگر خروجی بیشتر از ۷۰ کاراکتر باشد در خط بعد نمایش داده شده می شود. ممکن است این کار باعث پیچیدگی شود. برای خطوط ورودی می توان ادامه ورودی را با تایپ ... و زدن دکمه enter در خط بعد وارد کرد. بهتر است با استفاده از چندین متغیر اختیاری دستورات طولانی به دستورات کوچکتر تقسیم شود. بعضی اوقات خطوط خروجی طولانی و غیر قابل فهم می شود، مخصوصاً اگر از دستوره های نمادین استفاده شده باشد. در این حالت می توان با استفاده از دستور pretty خروجی را در خطوط کوتاهتر و خواناتر نمایش داد.

با تشکر

دوستکامیان بیست پنجم فروردین ۱۳۹۴