در اين بخش دستوراتي را معرفي مي كنيم كه در نوشتن برنامه هاي مختلف به ما كمك مي كند.اين دستورات در اغلب زبان هاي برنامه نويسي مشترك است و تنها گرامر آن ها متفاوت است.

ساخت function file:

تا به حال تنها از توابعي استفاده مي كرديم كه قبلا براي MATLAB تعريف شده بود؛ ولي ممكن است اين توابع نتوانند نيازهاي ما را پاسخ دهند، يا بخواهيم توابعي با كاربري خاص بنويسيم.

يك تابع (function file) مانند يك M-file است با اين تفاوت كه خط اول آن به صورت زير است:

```
function [outputs]= name(inputs)
```

اين خط مشخص مي كند كه اين M-file يك تابع است. همچنين تعداد ورودي ها و خروجي ها را مشخص كرده و هر يك را در يك متغير قرار مي دهد.در صورتي كه تنها يك ورودي داشته باشيم نيازي به كلوشه ([]) نيست. name نيز نام تابع را مشخص مي كند.

بهتر است براي خواناتر شدن برنامه از عبارات توضيحي استفاده كنيم.اين عبارات بايد ورودي ها و خروجي ها را مشخص كند.همچنين مي توان نام برنامه نويس و تاريخ نوشتن آن را نيز مشخص كرد. اين خطوط با اجرا دستور help name>> به نمايش درمي آيند.
به عنوان مثال M-file زير دستور prod را شبيه سازي مي كند.

```
function p=prod۲ (x)
% function p=PROD۲ (x)
% shabih sazi farman PROD
% a:radif   b:sotoon
[a,b]=size۲(x);
p(۱,:)=x(۱,:);
for i=۲:a,
%   satr aval p dar satr haye x zarb shode
%   va dar satr aval p zakhir mishavad
   p(۱,:)=p(۱,:).*x(i,:);
end
% if x is rowvector
while a==۱;
   p=۱;
   for i=۱:b,
      p=p*x(i);
   end
   a=۰;
end
```

حلقه هاي تكرار:

اين دستورات در اغلب زبان هاي برنامه نويسي به خصوص c وجود دارند.

» حلقه For
اين حلقه اين امكان را به وجود مي آورد كه تعدادي از دستورات به تعداد دفعات از قبل تعيين شده تكرار شوند.شكل كلي آن به صورت زير است:

```
for variable = a
    statement ۱
    statement ۲
   ...
end
```

كه a يك ماتريس است.در هر بار تكرار حلقه يك ستون ماتريس a در variable قرار مي گيرد.به اين ترتيب حلقه به تعداد ستون هاي a تكرار مي شود.
اين حلقه را مي توان به صورت تو در تو استفاده كرد. مثال زير با استفاده از حلقه هاي تو در تو جدول ضرب ايجاد مي كند.

```
for i=١:٥
   for j=١:٥
      s(i,j)=i*j;
   end
end
```

» حلقه While
اين حلقه چند دستور را به تعداد دفعات نامحدود تكرار مي كند. از اين دستور هنگامي استفاده مي شود كه تعداد دفعات تكرار مشخص نباشد. شكل كلي اين دستور به صورت زير است:

```
while expression
        statements
end
```

expression يك عبارت شرطي است و تا هنگامي كه درست باشد، حلقه تكرار مي شود. (عبارات شرطي در قسمت بعد شرح داده مي شود)

عملگرهاي رابطه اي:

اين عملگرها شامل موارد زير مي باشد:

عملگرهاي رابطه اي	شرح
<	كوچكتر از
<=	كوچكتر يا مساوي
>	بزرگتر
>=	بزرگتر يا مساوي
==	مساوي با
~=	مخالف با (نامساوي)

عملگرهاي منطقي:

اين عملگرها را در جدول زير مشاهده مي كنيدك

عملگر منطقي	شرح
&	AND
|	OR
~	NOT
xor(x,y)	OR انحصاري (در صورتي كه تنها يكي (x يا y) مقدار درستي داشته باشند مقدار True برمي گرداند)

ساختارهاي تصميم:

» شرط If - Else – End
حتما با عملكرد اين دستور در زبان هاي برنامه نويسي ديگر آشنا شده ايد.شكل كلي اين دستور را در زير مي بينيد.

```
 if expression ١
    statements ١
 elseif expression ٢
    statements ٢
...
elseif expression n
    statements n
else
    statements
end
```

همان طور كه مشاهده مي كنيد در حالت كلي مي توان از يك If ،بيشمار Elseif و يك Else و End استفاده كرد.استفاده از Else و Elseif اختياري است.

اگر شرط مقابل If درست باشد دستورات شماره ١ اجرا مي شوند، در غير اين صورت شرط ٢ (مقابل Elseif) بررسي مي شود در صورتي كه درست باشد دستورات ٢ ودر غير اين صورت شرط ٣ بررسي مي شود ... در صورتي كه n شرط بررسي شد و درست نبود دستورات قسمت Else اجرا مي شود.

» شرط Switch-Case

از اين ساختار براي تصميم گيري چندگانه بر اساس مقادير مختلف يك عبارت استفاده مي شود.به طور كلي در تمام تصميم گيري ها كه بيش از ٣ انتخاب وجود داشته باشد از اين دستور استفاده مي شود.

به عنوان مثال فرض كنيد متغيري مثل x مقادير ١، ٢، ٣ را اختيار مي كند و مي خواهيد بر اساس مقادير مختلف x تصميم گيري مختلفي را انجام دهيد.اگر برابر ١ بود دستورات ١، اگر برابر ٢ بود دستورات ٢ و اگر برابر ٣ بود دستورات ٣ اجرا شوند و در صورتي كه هيچ كدام از اين ها نبود دستورات ٤ (otherwise) اجرا شوند.حالت كلي اين دستور را مشاهده مي كنيد:

```
switch switch_expr
case case_expr,
        statement, ..., statement
case {case_expr١, case_expr٢, case_expr٣,...}
        statement, ..., statement
  ...
otherwise,
        statement, ..., statement
end
```

به چند نكته در اين مورد بايد دقت كرد:
١) پس از اجراي هر يك از دستورات روند اجرا برنامه به بعد از End منتقل مي شود و ساير Case ها كنترل نمي شوند.
٢) در بالا در مورد Case دوم در صورتي كه عبارت مورد نظر با هر يك ٣ عبارت داخل كلوشه ({}) برابر باشد دستورات اجرا مي شوند.
٣) استفاده از Otherwise نيز اختياري است.

بلوك Try-Catch:

شكل كلي اين دستور به اين صورت مي باشد:

```
try
   commands
catch
    commands
end
```

عملكرد اين دستور به اين صورت است كه دستورات زير Try اجرا مي شوند؛ در صورتي كه خطايي رخ دهد كنترل برنامه به Catch منتقل شده و دستورات موجود در اين قسمت اجرا مي شود. اين خاصيت باعث مي شود از آن براي خطايابي برنامه ها استفاده شود.

توقف روند اجراي برنامه:

Break «

هنگامي كه اين دستور اجرا مي شود MATLAB به اولين دستور كه بعد از حلقه For قرار دارد مي رود.در صورتي كه اين دستور در حلقه هاي تو درتو (For يا While) به كار رود MATLAB فقط از حلقه جاري خارج مي شود.

Error «

اين دستور باعث توقف اجرا برنامه شده و مي تواند يك رشته كركتري را برگرداند.

```
error (' STATEMENT ')
```

Return «

هر گاه روند اجرا برنامه به اين دستور برسد مقدار مورد نظر را برمي گرداند (در Command window نمايش مي دهد)؛ و ادامه اجراي برنامه متوقف مي شود.

از اين دستور براي نمايش زود هنگام مقادير يعني قبل از به پايان رسيدن كامل برنامه استفاده مي شود. به اين ترتيب هرگاه جواب مورد نظر به دست آمد روند اجراي برنامه نيز متوقف مي شود و مقدار مورد نظر را برمي گرداند.