

## حلقه های شرطی :

if ..... if (1)  
 else if .....  
 else .....  
 end

شرط نیازی به ( ) ندارد ولی می توان شرط را در داخل ( ) نوشت.

در انتهای if باید end را قرار دهیم . حتی اگر یک دستور داشته باشیم.

شرط های رایج ، شرط ها باید معادل با یک متغیر boolean باشند که در صورت درستی شرط مقدار 1 و در

غیر این صورت مقدار صفر بدهند .      &&    ||    <    <=    >=    >    !=    ==

while ..... while (2)      switch i      switch (3)  
 .....  
 end      case v1  
 .....  
 case v2  
 .....  
 otherwise  
 .....  
 end

for i = I ..... for (4)  
 .....  
 end

I بردار سطری شامل اندیس های مورد نظر برای متغیر i می باشد : for i = 0:0.1:1

اگر در داخل for تغییراتی روی i ایجاد کنیم بعد از بازگشت به خود for ، i برابر با اندیس بعدی می شود

به عبارتی با تغییر دادن i در داخل for نمی توانیم متغیر به تغییر در روند for شویم



**return**، هر وقت برنامه به **return** برسد از حلقه ای که در آن قرار دارد خارج می شود.

## برنامه نویسی در **Matlab** :

محیط برنامه نویسی در **matlab**، **rm-file** نام دارد، **file/new/m-file**

**m-file** شبیه به یک فایل **text** که دکمه های اجرای ... دارد.

در نوشته ی بالا سمت راست، یک مربع رنگی وجود دارد که وضعیت برنامه را به لحاظ خطاهای مربوط به متن مشخص می کند

(1) اگر مربع قرمز باشد، خطایی که در متن برنامه وجود دارد مانع اجرای برنامه می شود همانند قرار دادن

**end** در انتهای حلقه های شرطی. در این صورت، در سایر مربع، مکان خطاهای به صورت نواری نمایش داده می شوند.

(2) اگر مربع نارنجی باشد، خطاهایی که در متن برنامه وجود دارند مانع اجرای برنامه نمی شوند ولی رفع کردن آنها

باعث افزایش کیفیت و حتی سرعت اجرای برنامه می شود. مانند '؛' گذاشتن جلوی یک دستور. اگر هدف

نمایش دادن یک متغیر باشد از دستور **display** استفاده می کنیم ولی فرض **matlab** بر این است که جلوی

همه دستورات '؛' باشد.

(3) اگر مربع سبز باشد، خطایی در متن برنامه وجود ندارد.



## انواع m-file

(1) **script** : به ازای یک ورودی مشخص یک بار اجرا می شوند

(2) **function** : به ازای هر ورودی، خروجی متناظر را مشخص می کنند.

## نوشتن و اجرای script

(1) باز کردن m-file

(2) نوشتن دستورات

(3) ذخیره سازی و اجرا : لازم است از منوی debug در محیط mfile گزینه‌ی save & run

انتخاب شود. ابتدا محیط ذخیره سازی پاز می شود ( اسم دستورات را نمی توان به عنوان نام فایل انتخاب کرد)

سپس برنامه اجرا شده و خروجی ها در com.win نمایش داده می شوند.

( **cur.dir.** ) نرم افزار matlab پوشه ای با نام current directory دارد که در بالای com.win

مشخص و قابل تغییر است. m-file ها باید در cur.dir ذخیره شوند. اگر بخواهیم در پوشه ی دیگری

ذخیره کنیم باید ابتدا cur.dir را تغییر دهیم.

بهتر است از حافظه ی جانبی برای خواندن فایل ها ( cur.dir ) استفاده نکنیم و نتیجه ی نهایی را به آنها ( انتقال دهیم.



% comment : plot توضیح برنامه

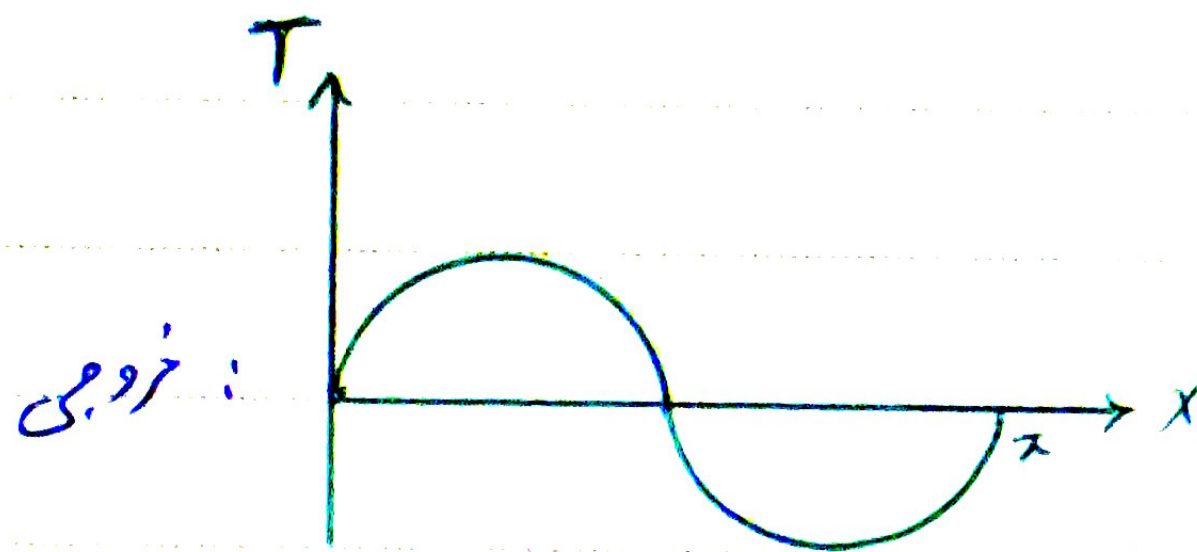
$x = \text{linspace}(0, \pi);$

$y = \sin(x);$

$z = \cos(x);$

$T = y * z;$

$\text{plot}(x, T);$



مثال:

نوشتن و اجرای function:

(1) باز کردن m-file

(2) نوشتن دستورات: خط اول function به صورت مقابل است:  $\text{function}[out1, \dots, outn] = f_{name}[in1, \dots, inm]$

در انتهای این دستور '؛' قرار نمی دهیم. در غیر این صورت خطا خواهیم داشت. در ادامه بقیه دستورات نوشته می شوند.

(3) ذخیره سازی و مقدار دهی به ورودی ها: برای ذخیره سازی از روشی رو برو استفاده می کنیم. file / save as

نام file باید با نام تابع یکی باشد، در غیر این صورت نمی توان از تابع استفاده کرد.

برای مقدار دهی داریم:  $[out1, \dots, outn] = f_{name}(in1, \dots, inm | \text{values})$

اگر تابع را استفاده کنیم می برای آن خروجی بگیریم، اولین خروجی را به صورت ans نمایش می دهد

حتی اگر '؛' نداشته باشیم