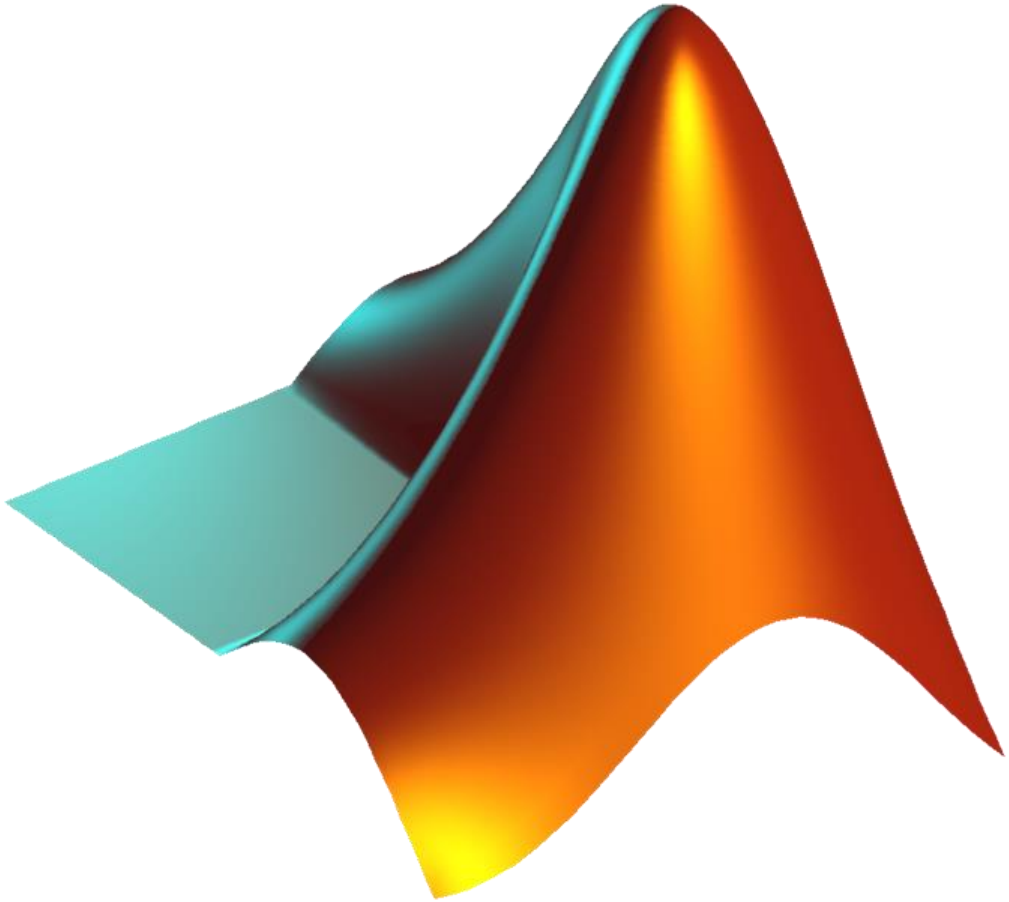
MATLAB COURSE



مدرس دوره : احمد خیراندیش

Session: 4

دانشگاه شاهد

Script Files

دلایل استفاده از فایل های اسکریپت :

1. دستوراتی را که در پنجره ی Command مورد استفاده قرار می گیرند را نمی توان ذخیره کرد و آن را اجرا نمود .
2. پنجره ی Command یک پنجره ی فعل و انفعالی بوده که قابل ویرایش نمی باشد .
3. باید بتوانیم چندین فایل کد را در صورت نیاز در دیگری مورد استفاده قرار دهیم .
4. بتوان درصورت لزوم دستورات را تغییر و یا اصلاح نمود .
5. به تعداد دلخواه آن مجموعه دستورات را در هر زمانی و به تعداد دلخواه اجرا کرد .

در این روش خروجی دستورات لیست شده و هنگام اجرای فایل ، نمایش داده می شود .

Script Files

انواع ورودی برای یک فایل اسکریپت:

۱ . متغیر تعریف شده باشد و مقدار آن در فایل های اسکریپت وجود دارد .

۲ . متغیر تعریف شده باشد و مقدار آن در پنجره ی Command تعیین شود .

۳ . متغیر در فایل های اسکریپت تعریف شده باشد اما هنگامی که این فایل ها اجرا می شود ،مقدار مشخصی از طریق پنجره ی Command وارد شود .

نکته ی مهم !

برای ذخیره سازی فایل های اسکریپت از نام های استفاده شده برای متغیرها،متغیرهای پیش فرض ،دستورات و توابع Matlab نمی توان استفاده کرد .

دستورات خروجی

دستور disp

کاربرد این دستور به دو صورت است :

1. از این دستور می توان برای نشان دادن المان های یک متغیر استفاده کرد .
2. برای نمایش یک عبارت متنی نیز می توان استفاده کرد.

disp (نام یک متغیر)

disp(رشته ی کارکتری)

```
A=[1 3 ; 6 9];
disp(A)
```

```
disp('Hello Class!')
```

برخی مفاهیم ضروری متلب:

دستور ورودی

دستور input

برای انجام دادن اکثر کارها، اسکریپت باید از کاربر سوال هایی را بپرسد و بر اساس ورودی کاربر واکنش نشان دهد. در غیر این صورت، اسکریپت یا باید مانند معمول عمل کند و یا از دیگر منابع اطلاعات را به دست آورد. دریافت ورودی از کاربر، این امکان را به وجود می آورد که اسکریپت به طور متفاوتی کار کند.

تابع () inputاز کاربر یک ورودی را درخواست می کند. در داخل این تابع یک پیام قرار می دهیم تا به هنگام وارد کردن ورودی توسط کاربر نمایش داده شود. اگر بخواهید که کاربر بتواند فقط متن(نه عدد) را وارد کند، باید از آرگومان 's' استفاده کنید. هنگامی که کاربر یک نام را تایپ می کند و کلید Enterرا می فشارد، مقدار مورد نظر در متغیر مورد نظر قرار می گیرد.

دستور input

```
x=input('Enter your value? ');
disp(x)
```

```
YourName=input('What is your name? ' , 's' );
disp(YourName)
```

دستورات خروجی

دستور fprintf

این دستور خروجی را چه متن و چه داده را بر روی صفحه نمایشگر نمایش می دهد و یا آن را در یک فایل ذخیره می کند ،با استفاده از این دستور می توان خروجی را قالب بندی نیز کرد .

```
a = [1.02 3.04 5.06];
fprintf('%d\n',round(a));
```

% \n → به خط بعدی می برد

دستورات خروجی

دستور fprintf

این دستور خروجی را چه متن و چه داده را بر روی صفحه نمایشگر نمایش می دهد و یا آن را در یک فایل ذخیره می کند ،با استفاده از این دستور می توان خروجی را قالب بندی نیز کرد .

```
a = [1.02 3.04 5.06];
fprintf('%d\n',round(a));
```

% \n → به خط بعدی می برد

% %d → بر پایه 10

عملگرهای رابطه ای و منطقی

عملگر رابطه ی: مقایسه دو عدد توسط عملگر مربوطه و اینکه جمله ی مورد نظر **True** یا**False** است .

عملگر منطقی : **True** یا **False** بودن شرایط را بررسی می کند و نتیجه ی چاپ شده در خروجی براساس عملگر استفاده شده ۱ یا ۰ است.

True→1

False→0

عملگرهای رابطه ای

عملگر رابطه ای	توضیح
<	کوچک تر
<=	کوچک تر یا مساوی
>	بزرگ تر
>=	بزرگتر یا مساوی
==	مساوی
~=	نامساوی

عملگرهای رابطه ای

از عملگرهای رابطه ای می توان برای مقایسه ی دو آرایه با اندازه ی یکسان یا مقایسه ی یک آرایه با یک مقدار عددی استفاده کرد .

```
A= 1:9
B= 9-A
X= A > 5
Y= (A==B)
Z=(A~=B)
```

برخی مفاهیم ضروری متلب:

عملگرهای منطقی

عملگر منطقی	توضیح	نام
& **A & B**	روی دو عملوند A وB عمل کرده و اگر True باشد نتیجه ی ۱ و در غیراینصورت ۰ را برمی گرداند.	AND
| **A | B**	روی دو عملوند A وB عمل کرده و اگر یکی یا هردو True باشد نتیجه ی ۱ و در غیراینصورت ۰ را برمی گرداند.	OR
~ **~A**	روی یک عملوند A عمل کرده و معکوس عملوند را به مامی دهد .نتیجه ی ۱ اگر True و ۰ اگرFalse باشد .	NOT

عملگرهای منطقی

```
A= 1:9
B= 9-A
X= ~ (A>4)
Y= (A>2) & (A<6)
Z=(A<2) | (A>7)
```

```
A= 1:9
B= A *0.3
X=(B>2)
Y= (B>2) & (A<6)
```

برخی مفاهیم ضروری متلب:

اولویت استفاده از عملگرها

عملگر	اولویت
پرانتز ← اگر پرانتز های تو در تو بودند ،پرانتز داخلی اولویت دارد .	1
توان	2
عملگر منطقی NOT (~)	3
ضرب ، تقسیم	4
جمع ، تفریق	5
عملگرهای رابطه ای	6
عملگر منطقی AND (&)	7
عملگر منطقی OR (|)	8

توابع داخلی منطقی

تابع	توضیحات
and(A,B)	اگر یکی از Aیا B صفر باشد ،۰ برمی گرداند در غیر اینصورت ۱ برمی گرداند.
or(A,B)	اگر هردو A و B صفر باشد ،صفر برمی گرداند در غیر اینصورت ۱ برمی گرداند.
not(A)	هم ارز با ~A
xor(A,B)	یای انحصاری - اگر یکی از عملوند ها True و دیگری False باشد ،True را برمی گرداند .
all(A)	اگر همه ی عناصر در بردار A ، True (غیر صفر)باشند ۱ را باز می گرداند و اگر یکی یا بیشتر عناصر False(صفر) باشد ۰ را برمی گرداند . اگر Aیک ماتریس باشد ، ستون های A مانند یک بردار عمل می کنند و برداری شامل 1 و ۰ باز می گرداند .

برخی مفاهیم ضروری متلب:

توابع داخلی منطقی

any(A)	اگر هر عنصر در بردار A،True(غیر صفر) باشد ۱ را باز می گرداند و اگر همه ی عناصر False (صفر) باشد ، ۰ را باز می گرداند . اگر A یک ماتریس باشد این اتفاق برای بردار ستون های آن رخ می دهد.
find(A)	اگر A یک بردار باشد ،اندیس عناصر غیرصفر را باز می گرداند.
find(A>d)	اگر A یک بردار باشد ، آدرس عناصری که از d بزرگتر هستند را باز می گرداند .
find(A>d)	اگر A یک بردار باشد ، آدرس عناصری که از d کوچکتر هستند را باز می گرداند .
find(A...d)	از هر عملگر رابطه ای می توان استفاده کرد.

حلقه ی for

حلقه ی for این امکان را به گروهی از دستورات می دهد تا به صورت ثابت و به تعداد دفعات از پیش تعیین شده ای تکرار شوند .
دستوراتی که بین for و end قرار می گیرند ،یکبار برای هر ستون آرایه اجرا می شوند .

```
for value=start:counter:finish
  [do something]
end
```

for نام متغیر دلخواه بهمراه مقدار دهی دلخواه به آن

(دستورات دلخواه)

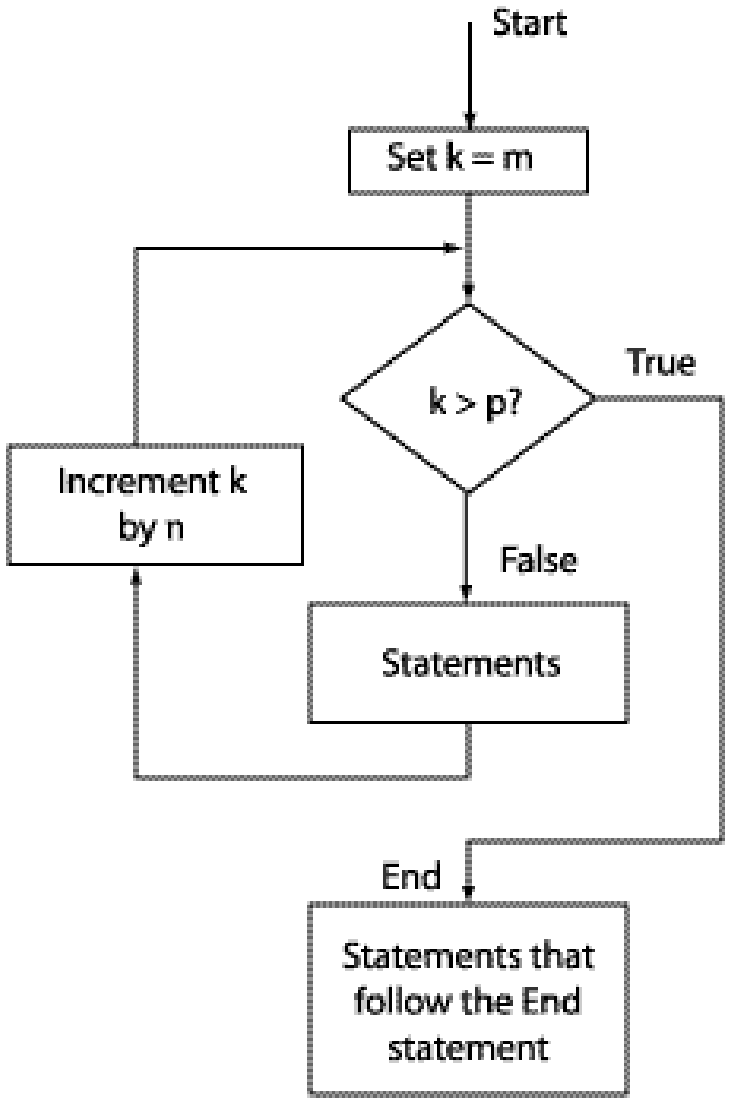
end

مفاهیم اولیه برنامه نویسی:

حلقه ی for

```
for  n=1 : 7
   x=sin(n * pi/10)
end
disp(x)
```

```
i=1;
for  x=rand(4,5)
     y(i)=sum(x);
      i=i+1;
end
disp(y)
```



Flowchart of the for loop

حلقه ی for

حلقه ی for را می توان به صورت حلقه های تو در تو نیز مورد استفاده قرار داد.

ایجاد ماتریس دو بعدی :

```
for n= 1 : 5
    for  m=5 : -1 : 1
        A(n,m)=n^2 + m^2 ;
     end
end
```

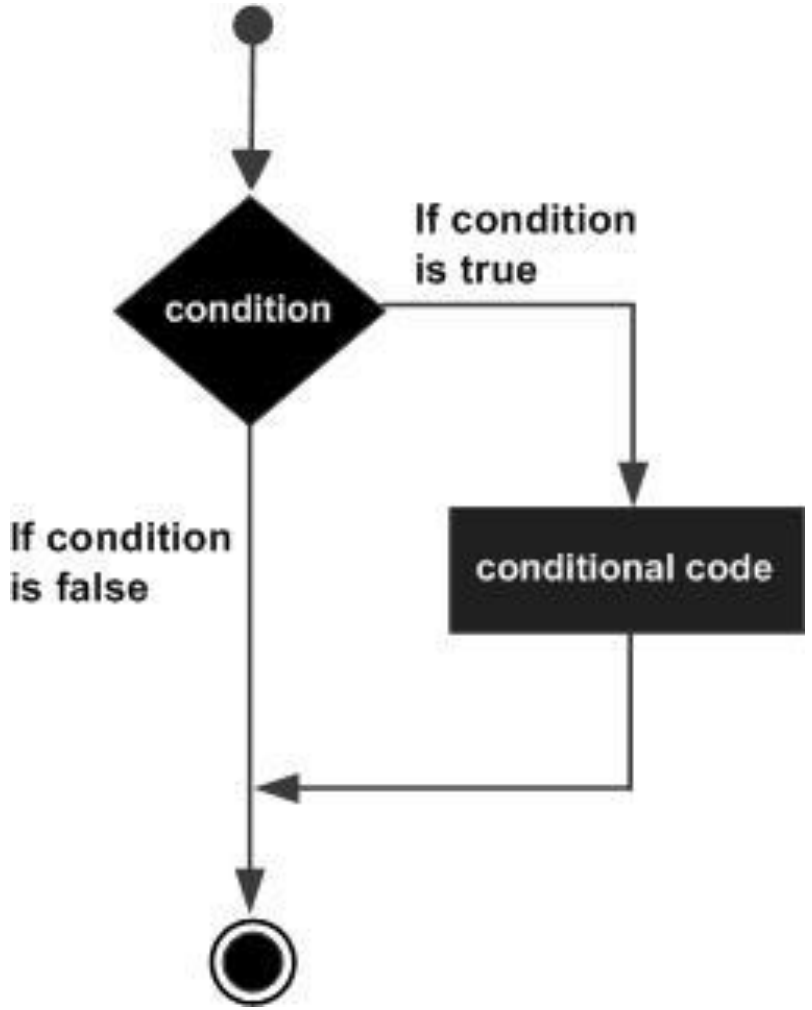
جمله شرطی IF

جمله ی شرطی دستوری است که به نرم افزار متلب این امکان را می دهد تا گروهی از دستورات که توضیح شرطی را دنبال می کنند ، اجرا کرده و نیز جمله شرطی می تواند اجرای دستورات متلب را کنترل کند .
اگر عبارت منطقی true باشد جملات بین if وend اجرا می شوند ، اگر شرط منطقی false باشد آنگاه دستورات اجرا نشده و به بعد از end منتقل خواهد شد .

```
if   عبارت منطقی یا رابطه ای

(دستورات دلخواه)

end
```

جمله شرطی if

```
a=input('Enter Value of a :');
if  a<5
    x=a*2;
    disp(x)
end
if  a>=5
    x=a*3;
    disp(x)
end
```



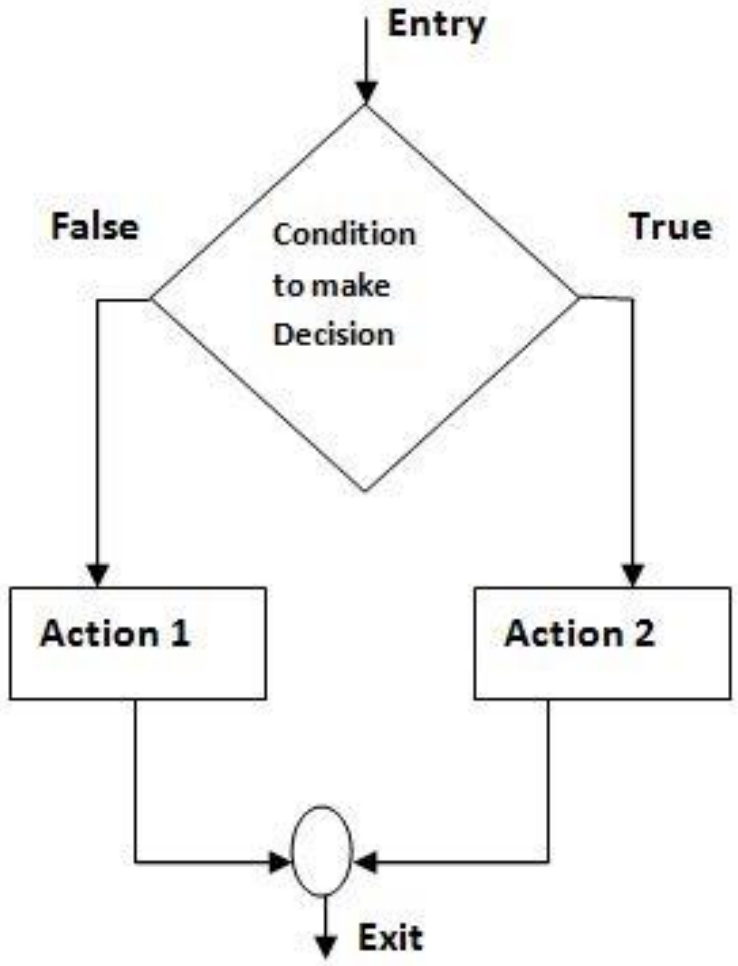
جمله شرطی if بهمراه بند else

بند else به شما اجازه می دهد که اگر عبارت منطقی درست باشد یک دسته از جملات بعد از دستور if اجرا شود و اگر عبارت منطقی نادرست باشد ، عبارات بعد از else اجرا شوند .

```
if   عبارت منطقی

(دستورات دلخواه)

else

(دستورات دلخواه)

end
```

جمله شرطی if بهمراه بند else

```
a=input('Enter Value of a :');
if  a<5
    x=a*2;
    disp(x)
else
x=a*3;
    disp(x)
end
```



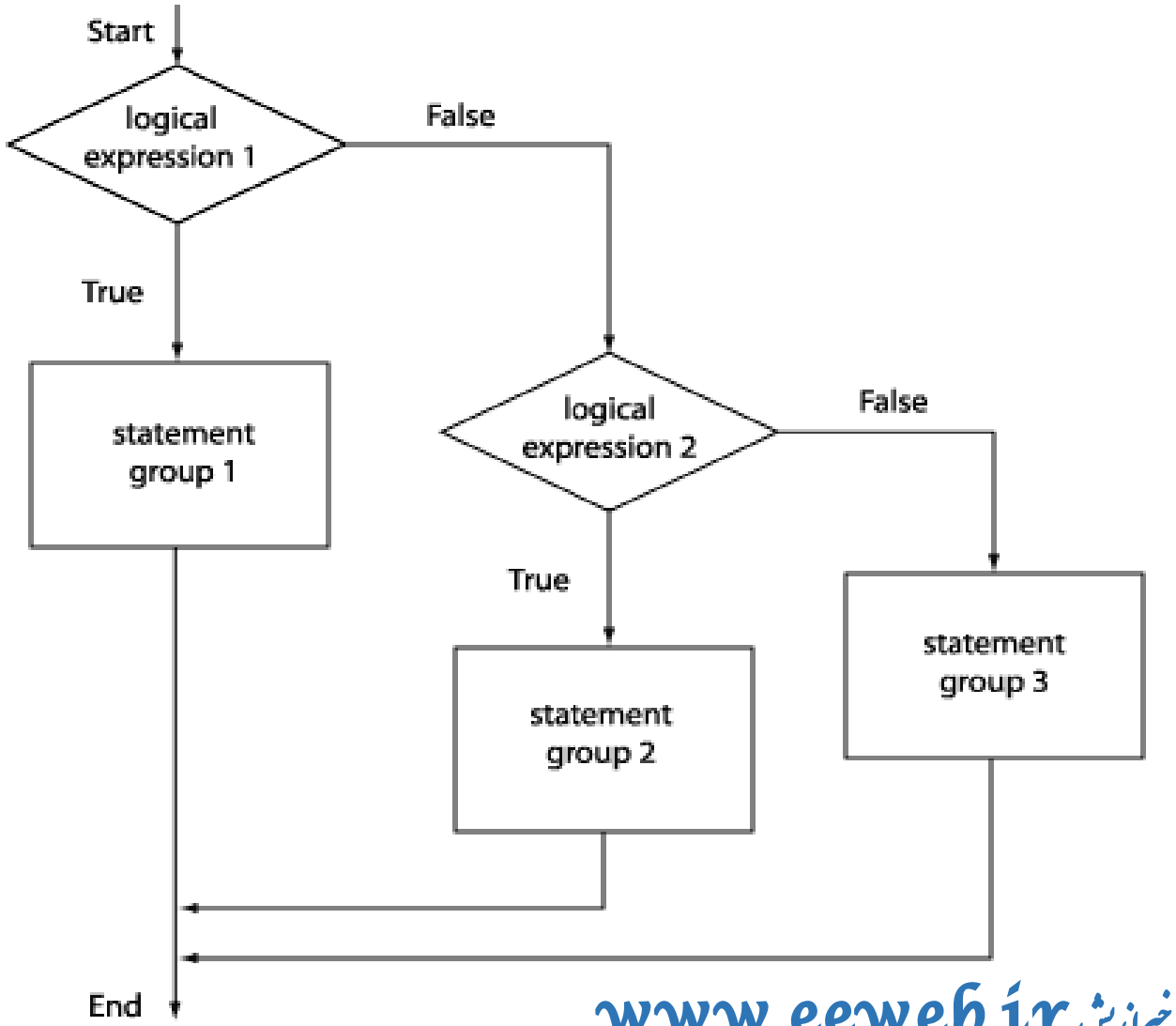
جمله شرطیif بهمراه بند ifelse

هنگامی که از چندین عبارت if-else به صورت تو در تو استفاده می کنیم ، ممکن است در مشخص کردن اینکه کدام جمله صحیح است دچار اشتباه شویم ،در این حالت از بند elseifاستفاده می کنیم .

```
if   عبارت منطقی

(دستورات دلخواه)

elseif عبارت منطقی

(دستورات دلخواه)

end
```

جمله شرطی if بهمراه بند ifelse

```
a=input('Enter Value of a :');
if  a<5
    x=a*2;
    disp(x)
elseif  a>5
    x=a*3;
    disp(x)
else
     a=5;
    disp(a)
end
```



حلقه ی while

می توان گفت این حلقه ترکیبی از حلقه ی for و جمله ی شرطی if است .شکل کلی آن بصورت زیر است :

```
while (condition)
  [perform code]
end
```

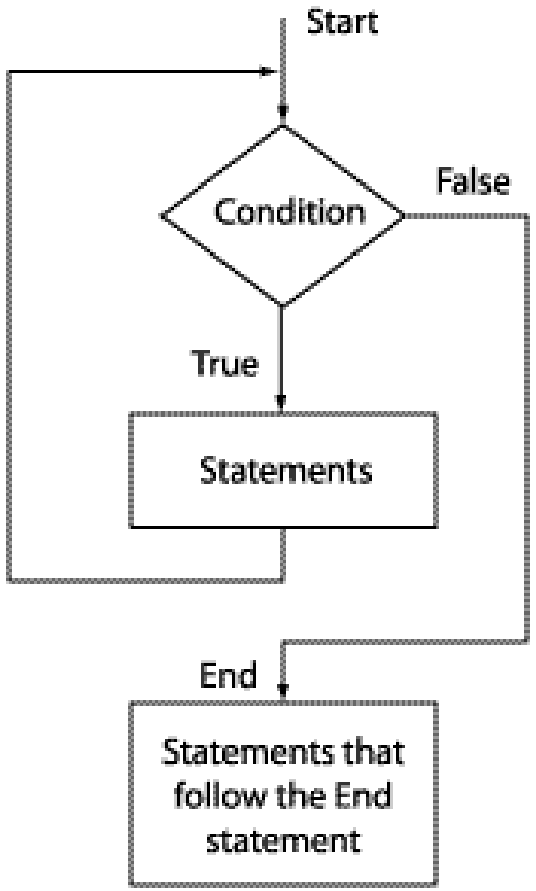
while عبارت رابطه ای

این مجموعه از جملات تا زمانی که عبارت رابطه ای درست باشد به صورت نامحدود اجرا خواهد شد .

end

حلقه ی while

```
a=input('Enter Value of a :');
while a<10
    a=a+ 1;
    disp(a)
end
```



Flowchart of the while loop