

فصل ششم :  
تصمیم‌گیری و کنترل روند،  
استفاده از حلقه‌ها و  
دستورات شرطی در متلب

دانشگاه کاشان  
سید مهدی وحیدی پور

MATLAB®

کلاس آموزشی

با تشکر از عباس طبیبی-دانشگاه صنعتی اصفهان

## فصل ششم: تصمیم‌گیری و کنترل روند

در این فصل در مورد جملات شرطی و انواع حلقه‌های تکرار صحبت خواهیم کرد.

۶-۱- حلقه for:

شکل کلی حلقه for در متلب بصورت زیر است:

```
for x = آرایه
```

```
    دستورات
```

```
end
```

در اینصورت حلقه فوق به تعداد ستونهای آرایه مشخص شده تکرار خواهد شد و در هر تکرار یکی از ستونهای این آرایه در متغیر X قرار گرفته و در بدنه حلقه قابل استفاده است. در صورتیکه آرایه یک بردار باشد، هر بار یک عنصر از آن در متغیر X قرار خواهد گرفت.

تذکر: با توجه به تواناییهای ماتریسی متلب از کاربرد حلقه‌ها در متلب تا حد ممکن باید پرهیز گردد زیرا اینکار باعث کند شدن شدید برنامه می‌شود و نیاز به کد نویسی بسیار بیشتری دارد.

## فصل ششم: تصمیم‌گیری و کنترل روند

۶-۱- حلقه for-ادامه-

مثال:

```
for n=1:10
    x(n) = sin(n * pi / 10);
end;
-----
for k=[1,2,3,7]
    x(k) = k+1;
end;
>>x
x=
    2    3    4    0    0    0    0    8
```

## فصل ششم: تصمیم‌گیری و کنترل روند

۶-۲- حلقه while :

در مواردی که بخواهیم یک یا چند دستور تا برقراری شرط خاصی تکرار گردند از این حلقه استفاده می‌کنیم. شکل کلی حلقه while بصورت زیر است:

```
while شرط
    دستورات
end
```

حلقه فوق تا زمانی که شرط ذکر شده برقرار باشد تکرار خواهد شد.

## فصل ششم: تصمیم‌گیری و کنترل روند

۲-۶- حلقه while – ادامه-

مثال:

```
t=1;
while t ~= -1
    t = input( ' Enter a number to continue or -1 to exit from
              this block: ');
    ...
end
```

## فصل ششم: تصمیم‌گیری و کنترل روند

۲-۶- حلقه while – ادامه-

مثال:

در این مثال بالاترین دقت محاسبات نقطه اعشار در متلب محاسبه می‌شود

```
Epsilon=1;
while ( 1 + Epsilon ) > 1
    Epsilon = Epsilon / 2;
end;
disp('This is the smallest floating point number in matlab: ');
disp ( Epsilon);
```

نکته : متغیر ویژه eps در متلب حاوی کوچکترین عددی است که اگر با یک جمع شود مقدار حاصل از یک بزرگتر خواهد بود. که این عدد دوبرابر Epsilon بدست آمده از روش فوق است.

## فصل ششم: تصمیم‌گیری و کنترل روند

### ۳-۶- ساختار if-else-end

هرگاه بخواهیم یک یا چند جمله در صورت برقرار بودن شرط خاصی (یکبار) اجرا شود، از بلوک `if` استفاده می‌کنیم. شکل کلی استفاده از این دستور بصورت زیر است:

```
if شرط ۱
    دستورات
elseif شرط ۲
    دستورات
elseif ...
    ...
else
    دستورات
end;
```

## فصل ششم: تصمیم‌گیری و کنترل روند

### ۳-۶- ساختار if-else-end - ادامه -

مثال:

```
Epsilon = 1;
while 1 > 0
    Epsilon = Epsilon / 2;
    if Epsilon + 1 == 1
        break;
    end
end
```

□ نکته: با دستور `break` می‌توان یک حلقه `while` یا `for` را شکست. در اینصورت اجرای برنامه از نخستین دستور بعد از حلقه ادامه خواهد یافت.

## فصل ششم: تصمیم‌گیری و کنترل روند

تکلیف ۱-۶: برنامه‌ای بنویسید که نمرات چند دانشجو را به صورت یک بردار بگیرد و عملیات زیر را انجام دهد:

- در صورتیکه ورودی کاربر بردار نباشد (ماتریس یا اسکالر باشد) پیام خطا دهد. (راهنمایی برای دادن پیام خطا می‌توانید از تابع `error` به جای `disp` استفاده کنید)
- با استفاده از حلقه `for` و دستورات شرطی `if-else-end` تک تک نمرات را چک کند و به صورت زیر آنها را تغییر دهد:
  - نمرات کمتر از ۵ را به ۹ تغییر دهد
  - نمرات بین ۵ و ۸ را به ۹٫۵ تغییر دهد.
  - نمرات بین ۸ و ۱۰ را به ۱۰ تغییر دهد.
  - نمرات بین ۱۰ و ۱۵ را ۱ نمره افزایش دهد
  - نمرات بیشتر از ۱۵ و کمتر از ۲۰ را ۰٫۵ نمره افزایش دهد.

تکلیف ۲-۶: برنامه دیگری بنویسید که همان کارهای برنامه ۱-۶ را بدون استفاده از حلقه انجام دهد.