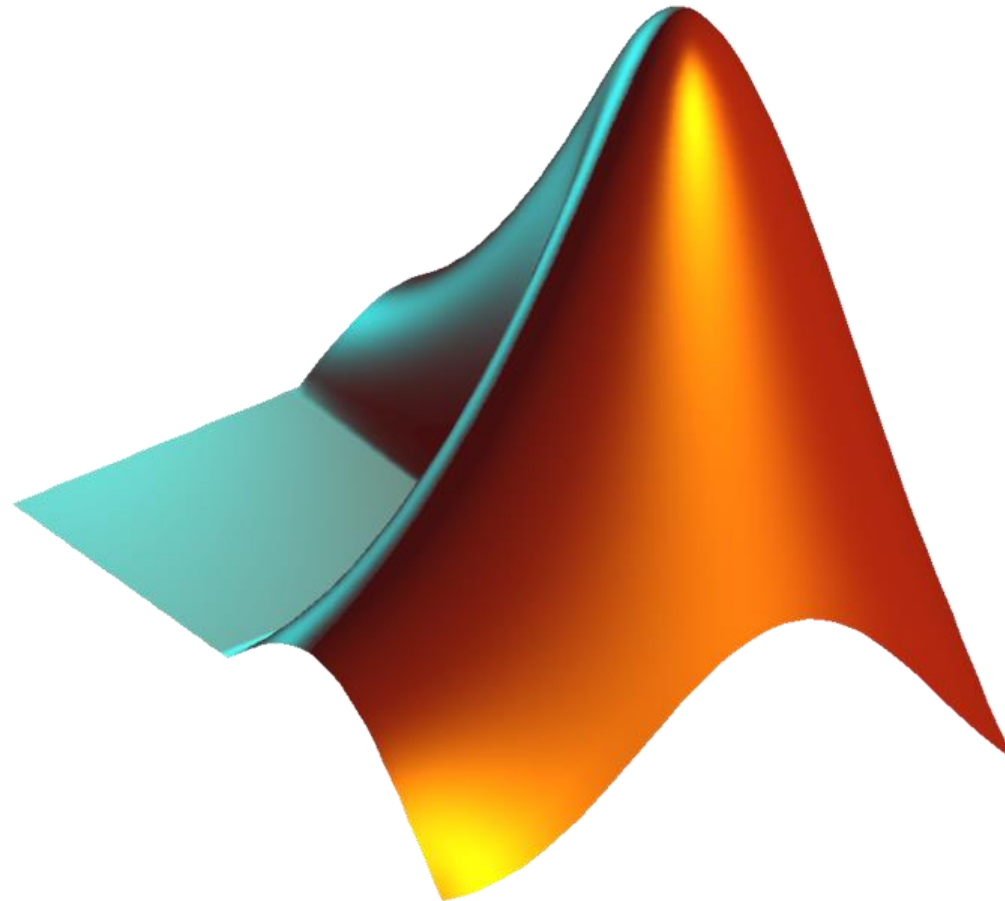




MATLAB COURSE



مدرس دوره : احمد خیراندیش

Session: **5**

دستور `continue`

```
a = 0;  
%while loop execution  
while a < 7  
    if a == 4  
        % skip the iteration  
        a = a + 1;  
        continue;  
    end  
    disp(a);  
    a = a + 1;  
end
```

زمانی که متلب در داخل حلقه `for` یا `while` با دستور `continue` مواجه می شود ، بدون اجرای دستورات بین `continue` و `end` ، بلافاصله به دستور `end` حلقه می رود به عبارتی `continue` باعث می شود تا کنترل برنامه به شرط حلقه داده شود تا مقدار بعدی در حلقه ، مورد آزمایش قرار گیرد .

دستور continue

```
for i = 1 : 10
```

```
% Disregard a few elements
```

```
if (i == 3 || i == 7 || i==9)
```

```
    continue
```

```
end
```

```
%Remaining values
```

```
    disp(i)
```

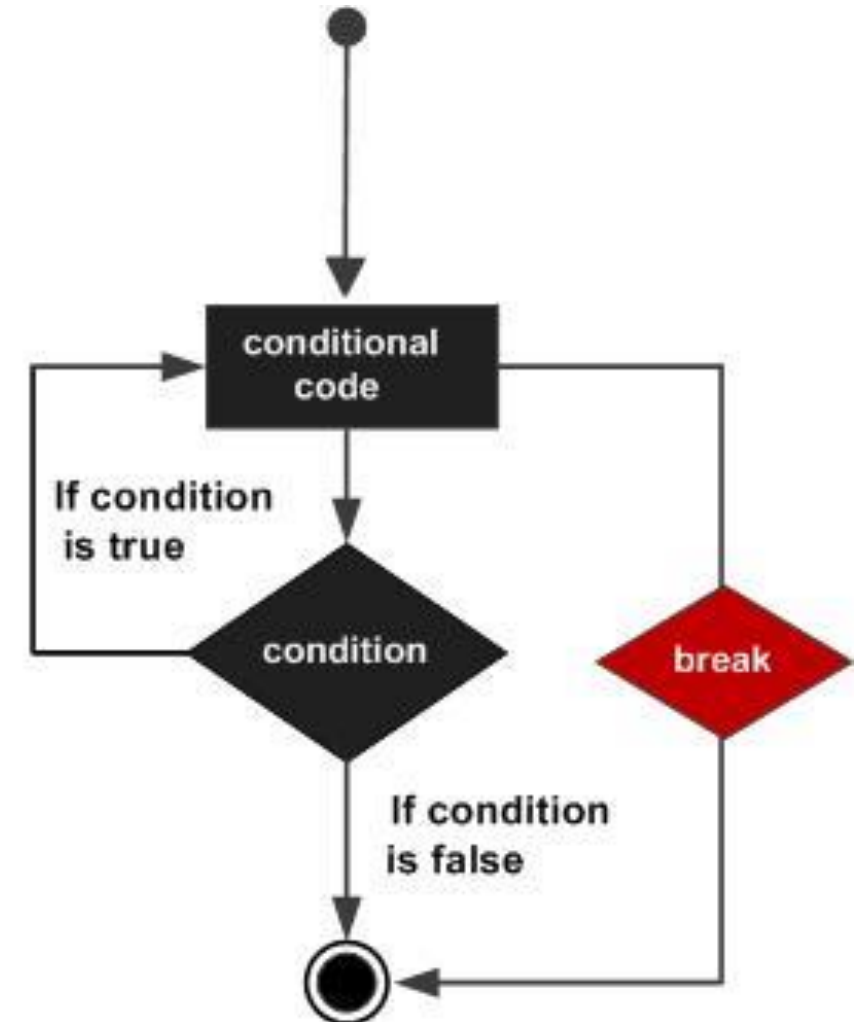
```
end
```

دستورات قطع حلقه

فرمان	توضیح
break	این دستور اجرای برنامه را در متلب برای حلقه های while و for خاتمه می دهد . در حلقه های تو در تو باتوجه به محل قرار گیری دستور break ،این دستور به اجرای داخلی ترین حلقه خاتمه می دهد .
return	این دستور عموماً در توابع متلب مورد استفاده قرار می گیرد . توابع معمولاً هنگامی که آخرین جمله اجرا می شود ، باز می گردند . برای بازگشت سریع و قبل از موعد مقرر از این دستور استفاده می شود . (در جلسات بعد با این دستور کار خواهیم کرد)
error('text')	این دستور جهت خاتمه ی اجرا و نمایش پیام هایی که در متن وجود دارند بر روی صفحه نمایشگر استفاده می شود .

دستور break

```
a = 10;  
while (a < 20 )  
    disp(a);  
    a = a+1;  
    if( a > 15)  
        break;  
    end  
end
```



دستور break

```
y = [-2 -4 0 -4 3 7];  
for i = 1 : length(y)  
    % Test for a greater-than-zero value  
    if y(i) > 0  
        % Terminate loop execution  
        break  
    end  
    % If it does not meet your condition, you can follow with your code  
    z = y(i) + 100;  
    disp(z)  
end
```

دستور error

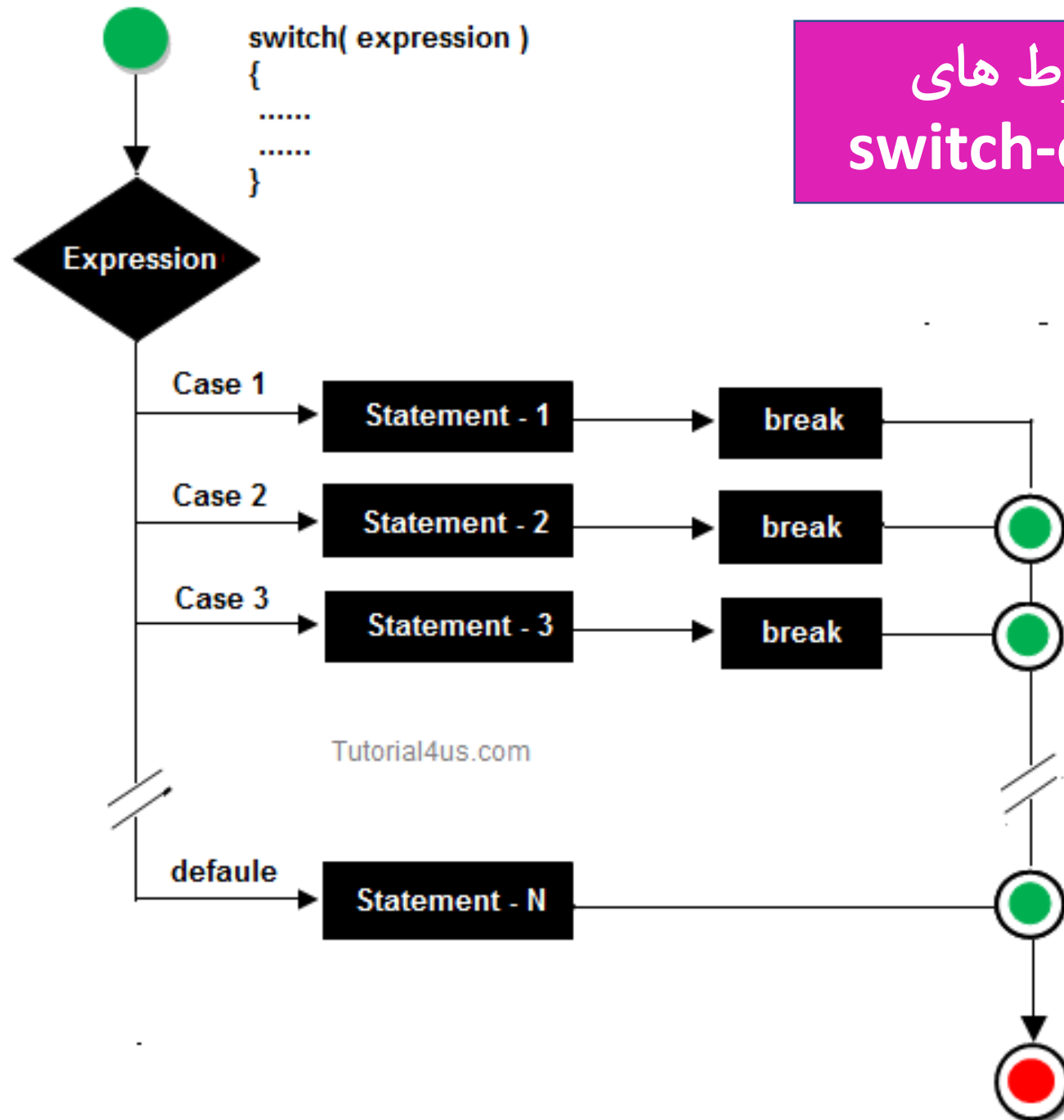
```
a = 0;  
while (a < 10 )  
    disp(a);  
    a = a+1;  
    if( a > 6)  
        error('You Go Wrong !');  
    end  
end
```


شرط های switch-case

عبارت مورد نظر
switch
وضعیت 1
دستورات 1
وضعیت 2
case
دستورات 2
.
.
.
otherwise
دستورات n ام
end

این دستور به ما این امکان را می دهد تا بتوانیم گروهی از دستورات را باهم اجرا کنیم .
هنگامی که می خواهیم گروهی از دستورات را که به استفاده های مکرر از یک شرط و با یک آرگومان مشترک بستگی دارند را به صورت شرطی مورد ارزیابی قرار دهیم ، از این شرط استفاده می کنیم . در واقع این شرط تصمیم گیری در میان چندین موضوع همسان را برعهده دارد و تا حدودی شبیه دستور شرطی if عمل می کند .

شرط های switch-case



شرط های
switch-case

```
grade =15;  
switch (grade)  
case 20  
disp('Excellent! ' );  
case 15  
disp('Well done' );  
case 10  
disp('You passed' );  
otherwise  
disp('Invalid grade' );  
end
```

```
Name ='Ali';  
switch (Name)  
case 'HAMED'  
disp('Name is : Hamed ' );  
case 'REZA'  
disp('Name is : Reza' );  
case 'Ali'  
disp('Name is : Ali' );  
otherwise  
disp('Invalid Name' );  
end
```

مثال هایی بیشتر

۱. برنامه ای بنویسید که نام شما و سن را به صورت روز/ماه/سال گرفته و بگوید که چند روز از تولد شما می گذرد.

۲. برنامه ای بنویسید که متغیر x را از کاربر گرفته و نمودار $y=\sin(x*\pi)$ را رسم کند .

۳. برنامه ای بنویسید که معادلات زیر را با ورودی ماتریس دلخواه b محاسبه کند .

$$\begin{bmatrix} 1 & 5 & 2 \\ 2 & 0 & 7 \\ -7 & 1 & -5 \end{bmatrix} \cdot \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix}$$

توابع در واقع همان M-file ها هستند. فرم کلی یک تابع در Matlab به صورت زیر است :

```
function [out1 , out2 , ...] = Function_Name (input1 , input2)
```

همانطور که مشخص است یک تابع بسته به عملکرد آن می تواند شامل یک سری ورودی و یا خروجی باشد. البته تابع می تواند شامل آرگومان های ورودی و یا خروجی نیز نباشد (تحت شرایطی!)

هدف از تابع نویسی در هر زبانی عبارت است از :

- عدم نیاز به تکرار استفاده از چندین خط کد در چند محل مختلف
- استفاده چندین باره از مجموعه ای از کدها در برنامه های مختلف
- ایجاد سهولت در عیب یابی برنامه

توابع متلب می توانند به تعداد دلخواه آرگومان های ورودی و خروجی داشته باشند .
برخی از خصوصیات و معیارهای مربوط به این آرگومان ها عبارتند از :
۱- تابع **M-file** می تواند آرگومان های ورودی و خروجی صفر داشته باشد .
۲- توابع می توانند توسط آرگومان های خروجی یا ورودی کمتری نسبت به تعدادی
که در خط تعریف تابع **M-file** مشخص شده ، فراخوانی شوند ولی نمی توانند با
آرگومان های بیشتری به آرگومان های ورودی یا خروجی که **M_file** را مشخص
می کنند ، فراخوانی شوند .

✓ عموماً نمی توان توابع را به صورت معمول اجرا کرد .

- ۱- هدف مون رو از نوشتن تابع مشخص کنیم .
- ۲- نامی را برای این تابع مان انتخاب کنیم .
- ۳- متغیرهای ورودی و خروجی را تعیین کنیم . (چند تا متغیر ورودی و خروجی داریم و برای هر کدام نام دلخواهی را به کار ببریم)
- ۴- نام فایل متلب مان باید هم نام ،نام تابع مان باشد .

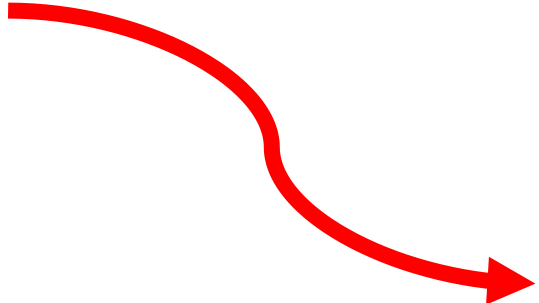
مثالهایی از تابع نویسی در متلب :

```
function z=Miyangin(x,y)
x=input('Enter X value :');
y=input('Enter Y value :');
z=(x+y)/2;
end
```

```
function Name_age(c,b)
x=input('Enter Your Name :' , 's');
y=input('Enter Your age :');
disp('Your Name is :');
disp(x);
disp('Your age is :');
disp(y);
end
```


مثالهایی از تابع نویسی در متلب :

```
My_x=5;  
My_y=4;  
My_t=2;  
My_Sum(My_x , My_y , My_t)
```



```
function [ z , k] = My_Sum( x,y,t)  
z=(x+y+t)*3  
k=x+5  
end
```

برخی از توابع می توانند هیچ ورودی و خروجی نداشته باشند. مثل تابع زیر :

```
function Mycircle()  
theta=0:pi/50:2*pi ;  
x=cos(theta);  
y=sin(theta);  
plot(x,y)  
end
```

nargout

مشخص کردن تعداد آرگومان های خروجی یک تابع

nargin

مشخص کردن تعداد آرگومان های ورودی یک تابع

```
function [ x,y ] = My_Fun( a,b,c,d )  
if nargin==2  
    x=a+b;  
    y=a*b;  
end  
if nargin==3;  
    x=a+b+c;  
    y=a*b*c;  
end  
if nargin==4;  
    x=a+b+c+d;  
    y=a*b*c*d;  
end  
end
```

حل معادلات دیفرانسیل

solve

این دستور، معادله جبری نمادین $f=0$ را حل می کند .

solve(f)

```
A=solve(' x^3-2*x-2=0 ');  
B=double(A)
```

solve(f1 , f2 , ...)

```
A=solve('x^2-y=2','y-2*x=5');  
X_ans=A.x(1)  
Y_ans=A.y(2)  
double(X_ans)
```

حل معادلات دیفرانسیل

dsolve

این دستور ، معادله دیفرانسیلی نمادین $f=0$ را حل می کند .

`dsolve('equation' , 'condition')`

$$\frac{d^2x}{dt^2} + \frac{5dx}{dt} + 3x = 7$$

```
syms x
```

```
x=dsolve(' D2x+5*Dx +3*x =7' , 'x(0)=3' , 'Dx(0)=4')
```

```
pretty(x)
```

$$a_n \frac{d^n x}{dt^n} + a_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \dots + a_1 \frac{dx}{dt} + a_0 x = f(t)$$

dsolve

معادله دیفرانسیلی

First order derivative $d/dt - D$
Second order derivative $d^2/dt^2 - D_2$

$$\frac{d^2 x}{dt^2} + 7 \frac{dx}{dt} + 10x = 20 \quad x(0) = 5 \quad \dot{x}(0) = 3$$

```
syms x
x=dsolve('D2x+7*Dx+10*x=20','x(0)=5','Dx(0)=3')
pretty(x)
```

حل معادلات دیفرانسیل

compose

با استفاده از این دستور می توان توابع ترکیبی ساخت.

compose (f , g)

```
syms x y
f=1+x*4;
g=y^2;
compose(f , g)
```


حل معادلات دیفرانسیل

finverse

معکوس تابعی را که به عنوان ورودی به آن می دهیم را برمی گرداند .

finverse (f)

```
syms x  
f=exp(x);  
finverse(f)
```

حل معادلات دیفرانسیل

diff(f)

مشتق عبارت f را نسبت به متغیر پیش فرض برمی گرداند .

diff(f)

```
syms x  
f=x+x^2+3*x+10;  
diff(f)
```

حل معادلات دیفرانسیل

`diff(f,'t')`

مشتق عبارت **f** را نسبت به متغیر **t** برمی گرداند .

`diff(f,'t')`

```
syms x t
f=x+x^2+3*t*x+10;
diff(f, 't')
```

حل معادلات دیفرانسیل

`diff(f,n)`

محاسبه مشتق مرتبه n ام

`diff(f, n)`

```
syms x
f=x^2+x^3+3*x+10;
diff(f, 3)
```

مشتق مرتبه سوم

حل معادلات دیفرانسیل

`diff(f,'t',n)`

محاسبه مشتق مراتب بالاتر

`diff(f,'t',n)`

```
syms x t
f=x^2+3*t^3*x;
diff(f, 't',3)
```

مشتق مرتبه سوم نسبت به متغیر t

حل معادلات دیفرانسیل

int(f)

انتگرال عبارت f را نسبت به متغیر پیش فرض برمی گرداند .

int(f)

```
syms x
f=x^2 + x;
int(f)
```

حل معادلات دیفرانسیل

`int(f , 't')`

انتگرال عبارت f را نسبت به متغیر t برمی گرداند .

`int(f,'t')`

```
syms x t
f=x^2 + x*t^2;
int(f,'t')
```

حل معادلات دیفرانسیل

`int(f,a,b)` ← انتگرال عبارت **f** را نسبت به متغیر پیش فرض در بازه **[a,b]** برمی گرداند .

`int(f,a,b)`

```
syms x
f=x^2 + x;
int(f , 0 , 3)
```


حل معادلات دیفرانسیل

انتگرال عبارت f را نسبت به متغیر t در بازه $[a,b]$ برمی گرداند .

`int(f, 't' , a , b)`

`int(f, 't' , a , b)`

```
syms x t
f=x^2 + x*t;
int( f , 't' , 0 , 4)
```

حل معادلات دیفرانسیل

`int(f, 'm' , 'n')`

انتگرال عبارت **f** را نسبت به متغیر پیش فرض در بازه **[m,n]** برمی گرداند .

`int(f, 'm' , 'n')`

```
syms x
f=x^2 + x*2;
int( f , '2' , '3')
```

حل معادلات دیفرانسیل

برای محاسبه انتگرال های دوگانه و یا چند گانه از دستورات `int` متوالی باید استفاده کرد .

$$\int_0^{\pi} \int_0^{\sin x} (x^2 + y^2) dy dx$$

```
syms x y
```

```
int ( int (x^2 + y^2 ,y,0,sin(x) ) ,0 ,pi)
```

حد (Limit)

`limit(f,x,A)`

برای محاسبه حد یک تابع به کار می رود .

`limit(f , x , A)`

$$\lim_{x \rightarrow A} f(x)$$

```
syms x
f=x^2 + x +5;
limit(f,x,0)
```

حد (Limit)

برای محاسبه حد راست یک تابع به کار می رود .

`limit(f,x,A , 'right')`

`limit(f, x , A , 'right')`

$$\lim_{x \rightarrow A^+} f(x)$$

$$\lim_{x \rightarrow 0^+} \frac{|x|}{x}$$

```
syms x
f= abs(x)/x;
limit(f,x,0 , 'right')
```

حد (Limit)

برای محاسبه حد چپ یک تابع به کار می رود .

`limit(f,x,A , 'left')`

`limit(f, x , A , 'left')`

$$\lim_{x \rightarrow A^-} f(x)$$

$$\lim_{x \rightarrow 0^-} \frac{|x|}{x}$$

```
syms x
f= abs(x)/x;
limit(f,x,0 , 'left')
```