



# آموزش برنامه نویسی متلب





# به نام خدا

تقدیم به هموطنان عزیزم

متلب را با لذت یاد بگیر!



## آموزش برنامه نویسی متلب

آموزش برنامه نویسی متلب

موضوع: ادامه محاسبات نمادین در متلب

جلسه: بیستم (آخر)

مدرس : پدرام مشهدی زاده

متلب را ساده، آسان و شیرین بنوشید!!!



این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.



## آموزش برنامه نویسی متلب

ارتباط با ما:

سایت: [www.javapro.ir](http://www.javapro.ir)

ایمیل: [RAHMAN.ZARIE92@GMAIL.COM](mailto:RAHMAN.ZARIE92@GMAIL.COM)

کانال تلگرام:

[@javalike](https://t.me/javalike)

گروه پرسش و پاسخ برنامه نویسی :

[@javapro\\_ir](https://t.me/javapro_ir)



## آموزش برنامه نویسی متلب

جلسه نوزدهم به مباحث ابتدایی محاسبات نمادین پرداختیم. هم‌چنین راه‌های تعریف یک تابع نمادین و نحوه محاسبه این توابع به‌ازای مقادیر مختلف را یاد گرفتیم. این جلسه به ادامه مباحث جلسه گذشته و علمیات‌های ریاضی روی توابع از جمله حد و مشتق و انتگرال می‌پردازیم.

برای درک بهتر اینکه محاسبات نمادین متلب به مبحث تبدیل مقادیر عددی به نمادین می‌پردازیم. فرض کنیم یک بردار  $a$  مانند زیر تعریف کرده باشیم. با دستور `sym()` این بردار را می‌توان به یک بردار نمادین تبدیل کرد:

```
>>a = 1:10
>>sym(a)
ans =
[ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

تک تک عناصر بردار  $a$  به صورت نمادین درمی‌آیند. اما اگر این مقادیر به صورت اعشاری باشند متلب برای تبدیل آن‌ها به متغیر نمادین نزدیک‌ترین مقدار به آن‌را تخمین می‌زند. به مثال زیر توجه کنید:

```
>> a = pi
a =
    3.1416
>> sym(a)
ans =
pi
>> a = 2/3
a =
    0.6667
>> sym(a)
ans =
2/3
```



## آموزش برنامه نویسی متلب

$\text{sym}(a, 'e')$  مقدار خطای حاصل، یعنی اختلاف بین مقدار تخمین زده شده و مقدار واقعی را برمیگرداند:

```
>> sym(a, 'e')
ans =
2/3 - eps/6
```

### - ماتریس نمادین:

با دستور  $\text{sym}$  می توان به شکل زیر ماتریسی از نمادها را ایجاد کرد:

```
>> A = sym('m', [2 3])
A =
[ m1_1, m1_2, m1_3]
[ m2_1, m2_2, m2_3]
```

A نام ماتریس بوده و 'm' نام درایه ها و [2 3] ابعاد ماتریس مدنظر است. با این دستور نام هر درایه به طور خودکار تعیین می شود. مانند دستورات ماتریس عددی می توان با درایه های این ماتریس عمل نمود:

```
>> A(1)
ans =
M1_1
>> A(:)
ans =
m1_1
m2_1
m1_2
m2_2
m1_3
m2_3
```



## آموزش برنامه نویسی متلب

بنابراین می‌توان به راحتی به این درایه‌ها مقادیر موردنظر را تخصیص داد.

### - diff(expr) :

از عبارت expr به طور نمادین مشتق می‌گیرد.

```
>> syms x y
>> diff(3*x^2)
ans =
6*x
>> diff(sin(x))
ans =
cos(x)
```

### - diff(expr,var) :

اگر expr برحسب بیش از یک متغیر باشد می‌توان مشتق جزئی آن را برحسب متغیر موردنظر v گرفت:

```
>> diff(x^2+2*x*y+y^2,y)
ans =
2*x + 2*y
```

### - diff(...,n) :

اگر با آرگومان‌های بالا به کار رود، مشتق مرتبه nام را برمی‌گرداند:

```
>> diff(sin(x),2)
ans =
-sin(x)
>> diff(x^2+2*x*y+y^2,y,2)
ans =
2
```



## آموزش برنامه نویسی متلب

### - `int(expr,v)` :

انتگرال نامعین عبارت `expr` را نسبت به متغیر `v` اخذ می‌کند.

```
>> int(cos(x))
ans =
sin(x)
>> int(3*x^2)
ans =
x^3
```

### - `int(expr,v,a,b)` :

انتگرال را مشابه قبل از `a` تا `b` اخذ می‌کند. `a` و `b` می‌توانند یک عدد یا عبارت نه‌عادی که شامل `inf` و `-inf` نیز هست، باشند.

```
>> int(exp(-x),0,inf)
ans =
1
>>int(sin(y),y,x,1)
ans =
cos(x) - cos(1)
```

### - `symsum(expr)` :

اگر `expr` بیانگر جملات یک سری باشد، مجموع جملات آن را از 0 تا `n-1` محاسبه می‌کند:

```
>> syms n
>> symsum(n+1)
ans =
```





## آموزش برنامه نویسی متلب

$$n^2/2 + n/2$$

برای این که مجموع جملات حسابی از 1 تا n را حساب کنیم، به جای  $expr$ ،  $n+1$  گذاشتیم، که حاصل همان مقدار مورد انتظار  $n \frac{n+1}{2}$  گردید.

**- `symsum(expr,a,b)` :**

مقادیر  $a$  و  $b$  محدوده محاسبه سری  $expr$  را بیان می کنند:

```
>> symsum(1/n^2,1,inf)
ans =
pi^2/6
```

**- `symsum(expr,v)` :**

مجموع سری  $expr$  که بر حسب بیش از یک متغیر بیان شده است را به ازای  $v$  از 0 تا  $v-1$  حساب می کند.

```
>>syms n k
>>symsum(n^k/sym('k!'),k,0,inf)
ans =
exp(n)
```

**- نکته: برای محاسبه فاکتوریل می توان از `sym('k!')` استفاده کرد.**

**- `solve(eqn)` :**

معادله  $eqn$  را که یک معادله جبری است، حل می کند. مثال:

```
>> solve(x == sin(x))
ans =
0
```

این دستور به شکل زیر نیز می تواند به کار رود:

```
>>solve(x-sin(x))
```



## آموزش برنامه نویسی متلب

هرگاه عبارت سمت راست نداشته باشد، به طور پیش فرض معادله را برای حالت مساوی با 0 حل می کند.

**- مثال: به دست آوردن ریشه های چند جمله ای معادله زیر:**

$$(x - 2)(x + 1) = 0$$

```
>> solve( (x-2)*(x+1) )
ans =
     2
    -1
```

**- solve(eqn,var) :**

متغیری که قرار است بر حسب آن معادله حل شود، توسط آرگومان var تعیین شده در غیر این مانند حالات قبل یک مقدار پیش فرض را انتخاب می کند.

**- مثال: حل معادله زیر:**

$$(ax^2 + bx + c = 0)$$

```
>> syms a b c x
>> solve(a*x^2+b*x+c)
ans =
-(b + (b^2 - 4*a*c)^(1/2))/(2*a)
-(b - (b^2 - 4*a*c)^(1/2))/(2*a)
>> pretty(ans)
```

```
+-          +-
|          |
|      2    1/2 |
|  b + (b  - 4 a c)  |
```



## آموزش برنامه نویسی متلب

$$\frac{b^2 - (b^2 - 4ac)}{2a}$$

- **assume()** :

در حل معادلات می توان فرض هایی را برای متغیر دلخواه خود در نظر بگیریم. برای مثال اگر بخواهیم معادله ای را به ازای تنها مقادیر مثبت  $x$  حل کنیم، باید فرض مثبت بودن  $x$  را برای متلب تعریف کنیم. برای این منظور به طریق زیر عمل می کنیم:

معادله زیر را در نظر بگیرید:

$$(x - 2)(x + 1) = 0, x > 0$$

```
>>syms f(x)
>>f(x) = (x-1)*(x+2);
>>assume(x>0)
>>solve(f,x)
ans =
1
```

شرط فرضی برای متغیرهای نمادین، هنگام تعریف آن ها از طریق دستور `sym()` یا `syms` نیز امکان پذیر است:



## آموزش برنامه نویسی متلب

```
>> syms x 'positive'  
>> sym('x','real')
```

- `assumeAlso(var,'')` :

شرطهای ثانویه به بعد را می تواند به یک متغیر اضافه کند:

```
>> syms f(x)  
>> f(x) = (x-1)*(x+2);  
>> assume(x>0)  
>> solve(f,x)  
ans =  
1  
>> f(x) = (x-1)*(x+2)*(x-.5);  
>> solve(f,x)  
ans =  
1  
1/2  
>> assume(x,'integer')  
>> solve(f,x)  
ans =  
1  
-2  
>> assumeAlso(x>0)  
>> solve(f,x)  
ans =
```



## آموزش برنامه نویسی متلب

1

- assumptions :

فرضیات تعریف شده را برمی گرداند:

```
>> assumptions  
ans =  
[ 0 < x, x in Z_]
```

پیروز و موفق باشید



آموزش برنامه نویسی متلب

سایت آموزشی رایگان جاواپرو

[www.JAVAPRO.ir](http://www.JAVAPRO.ir)

برنامه نویسی را با تجربه شخصی و به زبان خودهونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دخل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاواپرو به دور از اخلاق حرفه ای ست و حرام می باشد.