



# 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: آرایه ها در متلب

جلسه: ششم

مدرس : مدرسین جاواپرو

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

در [جلسه اول](#) گفتیم که MATLAB به معنای آزمایشگاه ماتریس است و همه چیز را می توان به گونه ای ماتریس در نظر گرفت. با توجه به مطالب گفته شده در جلسه قبل می توان گفت که همه چیز یعنی متغیرها. در این جلسه به موضوع آرایه ها و ماتریس ها می پردازیم.

یک آرایه را می توان یک ماتریس سطری یا ستونی تعریف کرد. بنابراین ابتدا برای راحتی با آرایه ها شروع به کار می کنیم.

یک آرایه سطری را به شکل زیر تعریف می کنیم:

```
>>a = [1 2 3]
a =
     1     2     3
```

آرایه ستونی:

```
>>a = [1;2;3]
a =
     1
     2
     3
```

عملیات ریاضی و منطقی را برای دو آرایه نیز می توان به کار برد. در این حالت این عملیات به طور نظیره- نظیر بین درایه های دو آرایه اعمال می شود. به مثال های زیر توجه کنید:

```
>>a = [1 2 3]; b = [-1 0 2];
>>a + b
ans =
     0     2     5
>>a - b
ans =
     2     2     1
```



## آموزش برنامه نویسی متلب

```
>>a*b
Error using *
Inner matrix dimensions must agree.
```

از آنجایی که آرایه‌ها نیز نوعی ماتریس هستند و ضرب ماتریس‌ها فقط به شرط برابر بودن تعداد ستون‌های ماتریس دوم و سطرهای ماتریس اول تعریف می‌شوند، با به کارگیری دستور فوق با پیغام خطا مواجه می‌شویم. چنانچه قصد ضرب یا تقسیم درایه به درایه داشته باشیم، بدین شکل عمل می‌کنیم:

```
>>a.*b
ans =
    -1     0     6
>>a./b
ans =
-1.0000     Inf     1.5000
```

دقت کنیم که در این حالت طول آرایه‌ها باید برابر باشند، مثلاً نمی‌توان یک آرایه سطری را با یک آرایه ستونی جمع نمود:

```
>>[1 2 3] + [1;2;3]
Error using +
Matrix dimensions must agree
```

برای تعریف آرایه‌ها با طول بیشتر می‌توان از " : " استفاده کرد:

```
start:step:end
```

که در این جا start مقدار ابتدایی و step گام افزایش و end مقدار پایانی است:

```
>>a = 0:1:8
a =
     0     1     2     3     4     5     6     7     8
```

چنانچه گام افزایشی ما یک باشد، به طور ساده‌تر داریم:



## آموزش برنامه نویسی متلب

```
>>a = 1:8
```

```
a =
```

```
1     2     3     4     5     6     7     8
```

دستور linspace برای ایجاد یک دنباله حسابی، با n جمله a تا b به کار می‌رود:

```
linspace(a,b,n)
```

مثال:

```
>>linspace(1,8,8)
```

```
ans =
```

```
1     2     3     4     5     6     7     8
```

```
>>linspace(1,10,5)
```

```
ans =
```

```
1.0000    3.2500    5.5000    7.7500   10.0000
```

تفاوت دستور linspace با عملگر " : " این است که در linspace مقدار پایانی یا b الزاماً در آخرین درایه آرایه قرار داده می‌شود ولی چنانچه step در عملگر " : " به گونه‌ای انتخاب شود که مقدار end در محدوده تعریف دنباله نباشد بزرگترین جمله دنباله که کوچکتر از end است آخرین درایه خواهد بود. برای روشن شدن موضوع به مثال زیر توجه نمایید:

```
>>1:2:10
```

```
ans =
```

```
1     3     5     7     9
```

```
>>linspace(1,10,5)
```

```
ans =
```

```
1.0000    3.2500    5.5000    7.7500   10.0000
```

هر دو دنباله فوق دارای پنج جمله با مقدار ابتدایی ۱، بوده و نقطه انتهایی ۱۰ در نظر گرفته شده است. با این تفاوت که دنباله اول به ۹ ختم شده، زیرا بزرگترین جمله دنباله است که از ۱۰ کوچکتر بوده است.



## آموزش برنامه نویسی متلب

دستور logspace از  $10^a$  تا  $10^b$ ،  $n$  جمله تولید می کند.

```
Logspace (a, b, n)
```

کاربرد آن در تولید نمودارهای لگاریتمی و محاسبات لگاریتمی می باشد:

```
>>logspace (0, 3, 4)
```

```
ans =
```

```
1          10          100          1000
```

- آرایه زیر را در نظر بگیرید:

```
>>a = 1:5
```

```
a =
```

```
1     2     3     4     5
```

می توان درایه های دلخواه آرایه را به اصطلاح استخراج نمود:

```
>>a (2)
```

```
ans =
```

```
2
```

می بینیم که کفایت اندیس درایه مورد نظر را در پرانتزی جلوی نام آرایه وارد کرد. اندیس درایه اول همواره یک بوده ولی چنانچه بخواهیم به درایه آخر دسترسی داشته باشیم و ندانیم طول آرایه چقدر است (یا برای راحتی کار) می توان از عبارت `end` استفاده کرد:

```
>>a (end)
```

```
ans =
```

```
5
```

برای استخراج چندین درایه آرایه نیز می توان از ":" استفاده کرد:

```
>>a (1:3)
```

```
ans =
```





## آموزش برنامه نویسی متلب

```

1      2      3
>>a (3:end)
ans =
3      4      5

```

درواقع باید اندیس درایه‌های موردنظر را در یک آرایه وارد نمود. برای مثال فرض کنید می‌خواهیم همه درایه‌ها به غیر از درایه وسط را در یک آرایه دیگر ذخیره کنیم:

```

>>b = a ([1 2 4 5])
b =
1      2      4      5

```

**تمرین:** با کمک دستور ":" و آرایه  $\alpha$ ، آرایه‌ای ایجاد کنید که درایه‌های  $a$  را از آخر به اول برگرداند.

**جواب:**

```

>>b = a (5:-1:1)
b =
5      4      3      2      1

```

**تمرین:** با کمک دستورات فوق دستوری بنویسید که درایه‌های فرد آرایه  $a$  را استخراج کند.

**جواب:**

```

>>a (1:2:5)
ans =
1      3      5

```

**تمرین:** آرایه زیر را در نظر بگیرید:

```
a = [1 3 -1 4 8 0 0 -5 7 9 4 5 7 1 3 6];
```

**دستوری بنویسید که به جز چهار درایه اول و آخر آرایه فوق بقیه درایه‌ها را استخراج کند.**

**جواب:**



## آموزش برنامه نویسی متلب

```
>>a (5:end-4)
```

```
ans =
```

```
8      0      0     -5      7      9      4      5
```

عملگر "" یک آرایه سطری را به ستونی و برعکس تبدیل می‌کند:

```
>>a'
```

```
ans =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
>>ans'
```

```
ans =
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

از دستورات گفته شده می‌توان برای جایگزینی برخی درایه‌ها با مقادیر دلخواه نیز استفاده کرد. برای مثال فرض کنید درایه‌های زوج آرایه فوق را می‌خواهیم برابر صفر قرار دهیم:

```
>>a (2:2:end) = 0
```

```
a =
```

```
1
```

```
0
```

```
3
```

```
0
```

```
5
```

یا اینکه بخشی از آرایه را به شکل زیر حذف نمود:

```
>>a (2:2:end) = []
```

```
a =
```

```
1
```

```
3
```

```
5
```

برای اینکه بهتر با [] آشنا شویم، به دستورات زیر توجه کنید:



## آموزش برنامه نویسی متلب

```
>> []
```

```
ans =
```

```
 []
```

```
whos ans
```

Name	Size	Bytes	Class	Attributes
ans	0x0	0	double	

**نکته:** اگر اندیس ورودی آرایه از طول آرایه بیشتر باشد، با پیغام خطا روبه‌رو می‌شویم:

```
>>a(4)
```

```
Index exceeds matrix dimensions.
```

دستورات دیگری نیز برای استخراج بخش دلخواهی از آرایه، وجود دارد که امکانات بیشتری را در اختیار ما قرار می‌دهد. دستور find شماره اندیس مربوطه که در شرط خاصی صدق می‌کند را برمی‌گرداند. برای مثال آرایه زیر را در نظر بگیرید:

```
>>a = 9      8      -3      4      -6      -9      5      1      0      9
      3      3      8      7      2
```

اگر بخواهیم درایه‌های منفی را از آرایه فوق استخراج کرده و در آرایه دیگری ذخیره کنیم، ابتدا از دستور find جهت استخراج اندیس درایه‌های منفی استفاده می‌کنیم:

```
>>b = find(a < 0)
```

```
b =
```

```
 3      5      6
```

نتایج به دست آمده را با آرایه a مقایسه کنید، می‌بینید که اندیس درایه‌های مورد نظر به دست آمده‌اند. حال برای آن‌که این درایه‌ها را در یک آرایه دیگر ذخیره کنیم، بدین گونه عمل می‌کنیم:

```
>>a_negative = a(b)
```

```
a_negative =
```



## آموزش برنامه نویسی متلب

-3      -6      -9

اکنون که درایه‌های منفی استخراج شده‌اند می‌خواهیم، این درایه‌ها از آرایه a حذف شوند:

```
>>a( find(a < 0) ) = []
```

```
a =
```

```
Columns 1 through 9
```

```
9      8      4      5      1      0      9      3      3
```

```
Columns 10 through 12
```

```
8      7      2
```

در جلسه بعد به معرفی ماتریس‌ها و کاربرد دستورات این جلسه در ماتریس‌ها و ادامه دستورات آن می‌پردازیم.

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.