



آموزش کامل برنامه نویسی متلب

گروه برنامه نویسی ایران متلب

IRAN MATLAB

iran-matlab.ir

ایران متلب

iran-matlab.ir

Contents

سرفصل

| | |
|--|----|
| سرفصل | ۲ |
| شروع کار با مطلب | ۱۰ |
| ۱-۳ آرایه های ساده : | ۱۰ |
| ۲-۳ آدرس دهی آرایه : | ۱۱ |
| ۳-۳ ساختار آرایه : | ۱۴ |
| روش اول استفاده از دستور کالن (:) | ۱۴ |
| روش دوم استفاده از دستور linspace : | ۱۵ |
| روش سوم استفاده از دستور logspace : | ۱۵ |
| ۳-۵ تغییر جهت آرایه ها : | ۱۶ |
| ۳-۶ آرایه های سطری و ستونی (ماتریسها) : | ۱۷ |
| توان رسانی ماتریسها : | ۱۸ |
| ۳-۷ عملیات ریاضی مابین چند ماتریس : | ۲۰ |
| ۳-۸ ماتریسهای همانی (واحد) و صفر : | ۲۲ |
| ۳-۹ آدرس دهی آرایه های ماتریس : | ۲۳ |
| ۳-۱۰ ماتریسهای چند بعدی : | ۳۳ |
| ۳-۱۱ چند تابع بسیار کاربردی در ماتریسها : | ۳۳ |
| ماتریس کرنوکه (keonecker tensor) : | ۳۶ |
| ۳-۱۲ ماتریسها (آرایه های چند بعدی) : | ۳۶ |
| توابع ریاضی | ۳۹ |
| ۲-۱ توابع ریاضی عمومی : | ۳۹ |
| مثال : حل یک مسئله ریاضی : | ۴۳ |
| ۲-۲ فرمت نمایش اعداد : | ۴۳ |
| - افزایش یا کاهش تعداد ارقام اعشار از محدوده تنظیم شده : | ۴۵ |
| ۲-۳ ذخیره اطلاعات از پنجره command window : | ۴۷ |
| ۳-۲ M-File مطلب : | ۴۸ |
| مثال حل مسئله یک پرتابه : | ۵۰ |

| | |
|---------|---|
| ۵۱..... | ۴-۲ دستورات <code>clc</code> و <code>clear all</code> : |
| ۵۲..... | دستور <code>clear all</code> : |
| ۵۴..... | مقدمه : |
| ۵۴..... | ۴-۱ رشته یا کاراکتر استرینگ (<code>character string</code>) : |
| ۵۷..... | ۶-۲ دستور <code>disp</code> : |
| ۵۷..... | ۶-۳ آرایه های سلولی (<code>cell arrays of string</code>) : |
| ۵۸..... | تابع <code>sprintf</code> : |
| ۶۰..... | ۶-۴ کنترل فرمت خروجی : |
| ۶۲..... | تابع <code>sprintf</code> : |
| ۶۳..... | تابع <code>fscanf</code> : |
| ۶۵..... | گرافیک و ترسیمات دوبعدی : |
| ۶۵..... | ۳-۱ مقدمه : |
| ۶۶..... | ۳-۱ نگاهی کلی به گرافیک مطلب : |
| ۶۸..... | تشریح یک ترسیم : |
| ۶۹..... | تغییر و تنظیم محورهای مختصات : |
| ۷۰..... | جزئیات مربوط به برچسب محورها : |
| ۷۱..... | جزئیات مربوط به نما : |
| ۷۳..... | دستورات پایه در ترسیمات : |
| ۷۴..... | تابع <code>plot</code> : |
| ۷۴..... | <code>plot(y)</code> : |
| ۷۴..... | <code>plot(x,y)</code> : |
| ۷۵..... | تغییر فرمت ترسیم خطوط : |
| ۷۶..... | رنگ خطوط : |
| ۷۹..... | تغییر ضخامت خطوط ترسیم : |
| ۸۰..... | اضافه کردن نمودار به ترسیمی که وجود دارد : |
| ۸۱..... | فقط ترسیم نقاط مفروض : |
| ۸۲..... | برچسب محورهای مختصات و عنوان : |
| ۸۳..... | ترسیم با دو محور <code>Y</code> : |

| | |
|----------|--|
| ۸۶..... | تنظیمات محورهای مختصات : |
| ۸۶..... | تعیین محدوده محورها: |
| ۸۹..... | ترسیم چندین نمودار در یک پنجره : |
| ۹۰..... | تابع subplot(m,n,p) : |
| ۹۲..... | نوشتن متن در پنجره ترسیم : |
| ۱۰۲..... | نوشتن متن در محدوده خارج از محورهای مختصات : |
| ۱۰۲..... | ایجاد ترسیمات خاص : |
| ۱۰۳..... | نمودارهای میله ای و ناحیه ای : |
| ۱۱۱..... | نمودارهای کیک (pie chart) : |
| ۱۱۳..... | هیستوگرام : |
| ۱۱۵..... | توابع ترسیم stem و stairs : |
| ۱۲۰..... | ترسیمات بردارهای جهت دار و سرعت : |
| ۱۲۰..... | ترسیم تابع compass : |
| ۱۲۱..... | ترسیم تابع feather : |
| ۱۲۴..... | تابع ترسیم quiver3 : |
| ۱۲۹..... | تابع ginput : |
| ۱۳۰..... | تابع ترسیم fill و fill3 : |
| ۱۳۴..... | حرکت و انیمیشن (Animation and Movies) : |
| ۱۳۴..... | روش EraseMode : |
| ۱۳۶..... | روش movie : |
| ۱۳۹..... | تابع ترسیم polar : |
| ۱۴۰..... | تابع ترسیم comet : |
| ۱۴۱..... | گرافیک و ترسیمات سه بعدی : |
| ۱۴۱..... | ۱ مقدمه : |
| ۱۴۱..... | ۲ تابع plot3 : |
| ۱۴۳..... | تابع ترسیم scatter3 : |
| ۱۴۴..... | تابع griddata : |
| ۱۴۴..... | تابع ترسیم bar3, bar3h : |

| | |
|----------|--------------------------------------|
| ۱۴۶..... | تابع ترسیم comet3 : |
| ۱۴۶..... | تابع ترسیم contour3 : |
| ۱۴۷..... | تابع ترسیم cylinder : |
| ۱۴۸..... | تابع ترسیم mesh : |
| ۱۵۱..... | تابع ترسیم tetramesh : |
| ۱۵۲..... | تابع ترسیم fill3 : |
| ۱۵۲..... | توابع ترسیم mesh, meshc, meshz : |
| ۱۵۴..... | تابع ترسیم pie3 : |
| ۱۵۴..... | تابع ترسیم quiver3 : |
| ۱۵۵..... | تابع ترسیم ribbon : |
| ۱۵۶..... | تابع ترسیم stem3 : |
| ۱۵۷..... | تابع ترسیم surf, surfc : |
| ۱۵۸..... | تابع ترسیم surf1 : |
| ۱۵۹..... | تابع ترسیم surfnorm : |
| ۱۶۰..... | تابع ترسیم waterfall : |
| ۱۶۲..... | تابع ترسیم tpaps : |
| ۱۶۳..... | تابع ترسیم ppform : |
| ۱۶۴..... | سایه زنی (shading) : |
| ۱۶۵..... | تابع ترسیم voronoi : |
| ۱۶۹..... | تابع ترسیم delaunay : |
| ۱۷۰..... | تابع ترسیم slice : |
| ۱۷۳..... | تابع ترسیم contourslice : |
| ۱۷۴..... | تحلیل پارامتری با مطلب. |
| ۱۷۴..... | ۱-۱ مقدمه : |
| ۱۷۴..... | ۲-۱ بیان سیمبولیک : |
| ۱۷۶..... | ۳-۱ بکارگیری معادلات جبری سیمبولیک : |
| ۱۷۸..... | ۴-۱ کاربری پیشرفته : |
| ۱۷۸..... | ترکیب توابع fog یا $f(g(x))$: |

| | |
|----------|--|
| ۱۸۰..... | محاسبه وارون : |
| ۱۸۰..... | محاسبه سیگما : |
| ۱۸۲..... | تبدیل سیمبولیک به عدد و برعکس : |
| ۱۸۲..... | تبدیل معادلات جبری به پارامتری و برعکس : |
| ۱۸۴..... | مقدار دهی به معادلات جبری : |
| ۱۸۵..... | محاسبه ریشه های معادلات جبری پارامتری : |
| ۱۸۸..... | مشتقگیری توابع پارامتری : |
| ۱۹۰..... | محاسبه ژاکوبین : |
| ۱۹۰..... | انتگرالگیری توابع پارامتری : |
| ۱۹۳..... | ۵-۱ ترسیمات توابع پارامتری : |
| ۱۹۳..... | ezplot : |
| ۱۹۴..... | ezplot3 : |
| ۱۹۵..... | ezpolar : |
| ۱۹۶..... | ezsurf : |
| ۱۹۷..... | ezsurfz : |
| ۱۹۷..... | ezmeshc : |
| ۱۹۸..... | ezmesh : |
| ۱۹۹..... | ezcontour : |
| ۲۰۰..... | ezcontourf : |
| ۲۰۱..... | محاسبه سری تیلور : |
| ۲۰۴..... | taylortool : |
| ۲۰۶..... | ۱-۶ چند دستور ساده و کاربردی در معادلات پارامتری : |
| ۲۰۷..... | تابع collect و horner و factor و expand : |
| ۲۰۸..... | تابع simplify و simple : |
| ۲۰۸..... | ۱-۷ حل دستگاه معادلات : |
| ۲۰۸..... | حل معادلات جبری : |
| ۲۱۱..... | حل دستگاه معادلات جبری : |
| ۲۱۴..... | ۱-۷ حل معادلات دیفرانسیل درجه اول : |

| | |
|----------|---|
| ۲۱۹..... | ۸-۱ تابع پله ای و تابع ضربه : |
| ۲۲۰..... | ۹-۱ تابع تبدیل لاپلاس : |
| ۲۲۱..... | ۱۰-۱ تابع تبدیل معکوس لاپلاس : |
| ۲۲۲..... | ۱۱-۱ تابع لاپلاسیین یا تابع دل (∇^2) : |
| ۲۲۴..... | ۱۲-۱ تابع انتگرال فوریه : |
| ۲۲۵..... | ۱۳-۱ تابع معکوس انتگرال فوریه : |
| ۲۲۶..... | ۱۴-۱ پنجره گرافیکی funtool : |
| ۲۲۷..... | توابع ریاضی |
| ۲۲۷..... | ۱-۱۳ مقدمه : |
| ۲۲۷..... | ۲-۱۳ خلاصه ای از تابع : |
| ۲۲۸..... | ۳-۱۳ نحوه ایجاد توابع ریاضی : |
| ۲۲۹..... | ۴-۱۳ روش دوم تعریف توابع ریاضی : |
| ۲۲۹..... | ۵-۱۳ ترسیم توابع ریاضی : |
| ۲۳۳..... | ۶-۱۳ توابع مینیموم و محاسبه صفرها : |
| ۲۳۳..... | مینیمایز کردن تابع تک متغیره : |
| ۲۳۴..... | پیدا کردن صفرهای یک تابع : |
| ۲۳۶..... | ۷-۱۳ انتگرالگیری عددی (Quadrature) : |
| ۲۳۶..... | تابع quad : |
| ۲۳۷..... | تابع quad1 : |
| ۲۳۷..... | تابع dblquad : |
| ۲۳۸..... | تابع triplequad : |
| ۲۴۰..... | عملگرهای منطقی و رابطه ای |
| ۲۴۰..... | ۱-۵ عملگرهای رابطه ای : |
| ۲۴۲..... | ۲-۵ عملگرهای منطقی : |
| ۲۴۲..... | ۳-۵ عملگرهای رابطه ای و منطقی : |
| ۲۴۴..... | حلقه ها |
| ۲۴۴..... | ۱-۵ مقدمه : |
| ۲۴۴..... | ۲-۵ حلقه for : |

| | |
|----------|--|
| ۲۴۷..... | ۳-۵ حلقه while : |
| ۲۴۷..... | ۴-۵ حلقه if-else-end : |
| ۲۵۱..... | ۵-۵ حلقه switch-case : |
| ۲۵۵..... | ۶-۵ حلقه tray-catch : |
| ۲۵۶..... | چند جمله ایها و درونیایی |
| ۲۵۶..... | ۱-۱۰ مقدمه : |
| ۲۵۶..... | ۲-۱۰ خلاصه ای از توابع چند جمله ای ها : |
| ۲۵۶..... | ۳-۱۰ ایجاد و بیان چند جمله ایها : |
| ۲۵۷..... | ریشه های چند جمله ای : |
| ۲۵۷..... | معادله مشخصه : |
| ۲۵۷..... | برازش یک چند جمله ای : |
| ۲۵۸..... | ضرب و تقسیم : |
| ۲۵۸..... | مشتق چند جمله ایها : |
| ۲۵۹..... | چند جمله ای منطبق بر منحنی : |
| ۲۶۰..... | پخش چند جمله ایها به تقسیمات جداگانه : |
| ۲۶۰..... | ۴-۱۰ درونیایی : |
| ۲۶۱..... | درونیایی تک بعدی : |
| ۲۶۱..... | درونیایی تک بعدی چند جمله ای : |
| ۲۶۲..... | توجه به سرعت و حافظه و همواری : |
| ۲۶۴..... | درونیایی بر اساس FFT : |
| ۲۶۴..... | درونیایی دوبعدی : |
| ۲۶۵..... | مقایسه روشهای درونیایی : |
| ۲۶۵..... | ۱- ایجاد تابع peaks در تجزیه و تحلیل پائین : |
| ۲۶۶..... | ۲- درونیایی با گره بندی های ریز : |
| ۲۶۶..... | ۳- درونیایی با بکارگیری روش همسایگی نزدیک : |
| ۲۶۷..... | ۴- درونیایی با بکارگیری روش bilinear : |
| ۲۶۷..... | ۵- درونیایی با بکارگیری روش bicubic : |
| ۲۶۹..... | درونیایی آرایه های چند بعدی : |

- ۲۶۹..... : درونیابی آرایه های سه بعدی
- ۲۷۰..... : درونیابی آرایه های چند بعدی
- ۲۷۰..... : گره بندی (شبکه بندی) آرایه های چند بعدی
- ۲۷۱..... : شبکه بندی مثلثی و درونیابی داده های پراکنده

ایران متلب

iran-matlab.ir

شروع کار با مطلب

۳-۱ آرایه های ساده :

مسئله محاسبه تابع $y = \sin(x)$ که در آن $0 \leq x \leq \pi$ واضح است که مجاسبه تابع y در تمامی نقاط غیرممکن می باشد و ما فقط میتوانیم تعداد تقسیمات خود را در این بازه افزایش دهیم مثلا برای x میتوانیم عبارات زیر را در نظر گرفته و مقدار y را بر اساس آن محاسبه نمائیم :

| | | | | | | | | | | | |
|-----|---|---------|---------|---------|---------|---------|---------|---------|---------|---------|-------|
| x | 0 | $.1\pi$ | $.2\pi$ | $.3\pi$ | $.4\pi$ | $.5\pi$ | $.6\pi$ | $.7\pi$ | $.8\pi$ | $.9\pi$ | π |
| y | 0 | .31 | .59 | .81 | .95 | 1.0 | .95 | .81 | .59 | .31 | 0 |

به پارامترهای مانند y و x که بیش از یک مقدار نسبت داده میشود : اصطلاحا آرایه گفته میشود ، آرایه های یکی از ابزارهای افزایش توانائی قدرت برنامه نویسی و راحتی عملکرد میباشد روند محاسباتی مثال فوق در مطلب بفرم زیر است :

```
clc;
clear all;
x=[0 .1*pi .2*pi .3*pi .4*pi .5*pi .6*pi .7*pi .8*pi .9*pi pi];
y=sin(x)
```

در صورتیکه برنامه فوق را در یک M-file نوشته و اجرا نمائید نتایج زیر بدست میاید :

```
y =
Columns 1 through 7
0    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511

Columns 8 through 11
0.8090    0.5878    0.3090    0.0000
```

همانطوریکه از مثال فوق دیده میشود شروع یک آرایه با کروه شش باز که اعداد و یا متغیرهای ما در داخل این دو کروه با فاصله از همدیگر و یا گذاشتن علامت کاما (،) مابین آنها و اتمام آرایه به کروه بسته تعریف میشود در واقع ما میتوانستیم x را به صورت زیر نیز تعریف نمائیم :

```
x=[0, .1*pi, .2*pi, .3*pi, .4*pi, .5*pi, .6*pi, .7*pi, .8*pi, .9*pi, pi];
(مانند  $y = \sin(x)$ )
```

زیر مجموعه آرایه و یا به عبارت دیگر تک به تک المانهای آن فراخوانی شده و در محاسبه دخالت داده میشوند

مثال : در صورتیکه داشته باشیم : $r = [0.5 \ 23 \ -11 \ 3]$; مطلوبست محاسبات : $y = r - 2$ و $e = r/2$ و

$$z = e - 2 * y + r/3$$

یک M-file باز کنید و برنامه زیر را در آن بنویسید :

```
clc;
clear all
r=[0.5 23 -11 3];
y=r-2
```

$$e=r/2$$

$$z=e-2*y+r/3$$

سپس آنرا اجرا نمائید نتایج زیر در صفحه Command مشاهده خواهد شد :

$$y = \begin{bmatrix} -1.5000 & 21.0000 & -13.0000 & 1.0000 \end{bmatrix}$$

$$e = \begin{bmatrix} 0.2500 & 11.5000 & -5.5000 & 1.5000 \end{bmatrix}$$

$$z = \begin{bmatrix} 3.0833 & -38.1667 & 24.1667 & -1.5000 \end{bmatrix}$$

برای المانهای آرایه محدودیت کاربرد و اندازه وجود ندارد برای نمونه در زیر آرایه ای که تعدادی از المانهای آن عدد مختلط میباشد مشاهده میگردد :

$$t=[1-2i \ 3 \ 4 \ 5+6i]$$

میتوان برای راحتی درک آرایه فوق را بصورت زیر نیز نوشت (البته هیچ فرقی از لحاظ نتیجه نخواهد داشت) :

$$t=[(1-2i) \ 3 \ 4 \ (5+6i)]$$

در مورد اعداد مختلط یاد آوری میشود i به عنوان $\sqrt{-1}$ در نظر گرفته میشود و در صورت بکارگیری توجه داشته باشید که این پارامترها (i , j) را قبلا در برنامه فوق برای نامگذاری پارامترها و یا ذخیره کردن مقادیر مشخصی بکار نبرید چرا که در این صورت مقادیر تعریف شده بجای آنها جایگزین گردیده و برنامه شما اجرای مطلوبی نخواهد داشت .

۳-۲ آدرس دهی آرایه :

حال x مثال قبلی را که بصورت زیر تعریف شده در نظر بگیرید :

$$x=[0, .1*pi, .2*pi, .3*pi, .4*pi, .5*pi, .6*pi, .7*pi, .8*pi, .9*pi, pi];$$

که شامل ۱۱ المان که در یک سطر و ۱۱ ستون اقرار دارند میباشد ، در ریاضیات به آرایه فوق آرایه یک سطر و ۱۱ ستون گفته میشود و با عبارت 1×11 نشان داده میشود . و یا برای سادگی به آن آرایه ای بطول ۱۱ نیز گفته میشود .

به المانهای یک آرایه اصطلاحاً زیر آرایه گفته میشود و برای آدرس دهی به یک زیر آرایه بدین صورت عمل میگردد :

(شماره ستون زیر آرایه مورد نظر) نام آرایه

مثلاً $x(1)$ یعنی اولین المان آرایه x که 0 میباشد و یا $x(4)$ اشاره به المان یعنی عدد $3*pi$. میباشد هر عملیاتی بر روی

آرایه ها انجام پذیرد خروجی نیز بصورت آرایه میباشد یعنی شما میتوانید بعد از انجام محاسبات ریاضی بر روی آرایه های که در یک

پارامتر خاص ذخیره کرده اید با آدرس دهی به المانهای آن نیز دست یابید :

مثال : برنامه زیر را در یک M-file نوشته و اجرا نمائید :

```
clc;
clear all;
u=[11 sin(pi/15) 1-2i (1+tan(2*pi/3))^(1/3) 27];
y=u*cos(pi/6)
t=y.^2+1
w=t+0.1*u-y/5
s=w(2)+0.1*t(1)
q=y(3)+1
e=t(2)
```

```
h=y(1:4)
```

خروجی برنامه بصورت زیر میباشد :

```
y =
```

```
Columns 1 through 5
```

```
9.5263 0.1801 0.8660 - 1.7321i 0.3903 + 0.6759i 23.3827
```

```
t =
```

```
1.0e+002 *
```

```
Columns 1 through 4
```

```
0.9175 0.0103 -0.0125 - 0.0300i 0.0070 + 0.0053i
```

```
Column 5
```

```
5.4775
```

```
w =
```

```
1.0e+002 *
```

```
Columns 1 through 4
```

```
0.9094 0.0102 -0.0132 - 0.0285i 0.0066 + 0.0047i
```

```
Column 5
```

```
5.4577
```

```
s =
```

```
10.1922
```

```
q =
```

```
1.8660 - 1.7321i
```

```
e =
```

```
1.0324
```

```
h =
```

```
9.5263 0.1801 0.8660 - 1.7321i 0.3903 + 0.6759i
```

نکته مثال فوق اینست که برای محاسبه $t=y \cdot ^2+1$ قبل از به توان رساندن y علامت دات (.) قرار دادیم، این بدین معنی است که تک تک المانهای آرایه y را به توان مورد نظر برساند و گذاشتن علامت دات در تمامی توان رسانندهای آرایه ها امری ضروری میباشد.

نکته دوم محاسبه $h=y(1:4)$ میباشد که منظور از این عبارت اینست که h آرایه ای است متشکل از المانهای اول و دوم و سوم و چهارم آرایه y یعنی شما برای نوشتن شماره چهار المان پشت سر هم از عبارت کالن (:) برای ایجاد تکرار استفاده کردیم عبارت $1:4$ یعنی از شماره 1 شروع کن و تا شماره 4 پیش برو. در واقع h را بایستی بصورت زیر تعریف میکردیم که ما برای راحتی در انجام محاسبات از عبارت کالن (:) استفاده نموده ایم:

$h=[y(1) \ y(2) \ y(3) \ y(4)]$

واضح است که روند آدرس دهی تک به تک در بسیاری مواقع امری غیر ممکن است مثلاً فرض کنید که شما آرایه ای به طول ۱۰۰۰۰۰ داشته باشید و بخواهید المانهای مابین محدوده ۲۵۰۰۰ تا ۷۱۰۰۰ را در با انجام یک سری عملیات در آرایه دیگری بکاربرید بدیهی است که آدرس دهی تک به تک امری غیرمنطقی است.

در واقع کلیت دستور بکارگیری دستو کالن (:) بصورت زیر است:

انتها : گام مورد نظر : ابتدا

برای درک بهتر مثال زیر را در یک M-file نوشته و اجرا نمائید.

```
clc;
clear all;
x=[ 1 -2 4 -21 38 11.5 9 8];
y=x(1:3)
z=x(2:6)
r=x(1:3:8)
t=x(7:-1:2)
p=x(2:2:7)
g=x(8:-1:1)
w=g-1
s=x(3:-1:1)*5-11
f=x([2 5 3 8 7])
l=x([2:4 7:-1:5])
```

بعد از اجرا خواهیم داشت:

```
Y =
     1     -2         4
z =
    -2     4    -21    38    11.5
r =
     1    -21         9
t =
     9    11.5    38   -21     4    -2
p =
    -2    -21    11.5
```

```

g =
    8    9   11.5   38   -21    4   -2    1
w =
    7    8   10.5   37   -22    3   -3    0
s =
    9   -21    -6
f =

   -2    38     4     8     9
l =

   -2     4   -21     9   11.5   38

```

میتوان برای ایجاد یک آرایه با المانهای با شماره‌های غیر منظم (مثلا در محاسبه $f = x([2 \ 5 \ 3 \ 8 \ 7])$) بکار رفته نیز استفاده نمود و یا همزمان با آدرس دهی غیر منظم از در صورت امکان از کالن نیز استفاده نمود (مثلا محاسبه l در مثال فوق $l = x([2:4 \ 7:-1:5])$)

۳-۳ ساختار آرایه :

در ابتدای معرفی آرایه ها مثال محاسبه سینوس یک زاویه ($0 \leq x \leq \pi$) را بیان کردیم و مقادیر آنها برای ۱۱ نقطه محاسبه نمودیم حال این سوال پیش میاید اگر تعداد نقاط درخواستی بیشتر میبود مثلا یک میلیون نقطه چگونه ما میتوانیم آرایه x را بیان کنیم بدون آنکه فضای زیادی را اشغال نمائیم و در ضمن در کمترین زمان ممکن بتوانیم آنها تشکیل دهیم برای اینکار سه دستور زیر را بیان میکنیم:

روش اول استفاده از دستور کالن (:) :

```

>> x=(0:0.1:1)*pi
x =

Columns 1 through 7

    0    0.3142    0.6283    0.9425    1.2566    1.5708    1.8850

Columns 8 through 11

    2.1991    2.5133    2.8274    3.1416

```