تابع magic ماتریسی را ایجاد می‌کند که مجموع اعضای هر ستون، هرسطر و دو قطر مورب آنها باهم برابر است. برای یک ماتریس (۴×۴) مقدار این مجموع برابر با ۳۴ است. حال اعضای سطر ۲ را بر ۲ تقسیم و سپس اعضای ستون ۲ را با ستون ۴ جمع و نتیجه را با ستون ۴ جایگزین می‌کنیم:

```
>> z=magic(٤);
>> z(٢,:)=z(٢,:)/٢;
>> z(:,٤)=z(:,٤)+z(:,٢)
z =
    ١٦     ٢     ٣    ١٣
   2.5   ٥.٥     ٥   ٩.٥
     9     7     6    19
     4    14    15    15
```

برای صفر کردن اعضای مورب ماتریس اصلی بدین شکل عمل می‌شود:

```
>> z=magic(٤);
>> z=z-diag(diag(z))
z =
    ٠     ٢     ٣    ١٣
    5     ٠    ١٠     ٨
    9     7     0    12
    4    14    15     0
```

برای جایگزین کردن اعضای مورب ماتریس اصلی با مقدار ۵، از دستورات زیر استفاده می‌شود:

```
>> z=magic(٤);
>> z=z-diag(diag(z))+٥*eye(٤)
z =
    ٥     ٢     ٣    ١٣
    5     ٥    ١٠     ٨
    9     7     5    12
    4    14    15     5
```

برای جایگزین کردن مقادیر ۱۱، ۲۳، ۵۴ و ۶۱ با اعضای مورب ماتریس z می‌توان نوشت:

```
>> z=magic(٤);
>> z=z-diag(diag(z))+diag([١١ ٢٣ ٥٤ ٦١])
z =
   ١١    ٢    ٣   ١٣
    5   ٢٣   ١٠    ٨
    9    7   54   12
    4   14   15   61
```

متلب سایت سیم پاور دوتابع دارد که برای ایجاد ماتریس‌هایی که با تکرار یک اسکالر می‌توان ایجاد کرد، به کار برده می‌شوند. این توابع عبارتند از : repmat و meshgrid، شکل اصلی repmat عبارت است از:

```
>> repmat(x,y,c)
```

تکرار ستون‌های جایگزین شده x است. به‌طور مثال، تابع repmat می‌تواند یک بردار ستونی یا سطری با طول دلخواه که اعضای بردار دارای مقادیر عددی یکسانی هستند را ایجاد کند.

بنابراین، برای ایجاد یک بردار سطری w متشکل از ۶ مقدار عددی ۴۵,۷۲ می‌توان نوشت:

```
>> w=repmat(٤٥,٧٢،١،٦)
w =
   ٤٥,٧٢  ٤٥,٧٢  ٤٥,٧٢  ٤٥,٧٢  ٤٥,٧٢  ٤٥,٧٢
```

این بردار را همچنین همانگونه که در سایت سیم پاور توضیح دادم می‌توان به شکل زیر ایجاد کرد:

```
>> w(١،١:٦)=٤٥,٧٢;
```

برای ایجاد یک ماتریس (۱×۱) با این مقادیر می‌توان نوشت :

```
>> w=repmat(٤٥,٧٢،٣،٣);
```

همچنین همانگونه که در سایت سیم پاور توضیح دادم می‌توان این ماتریس را به شکل زیر ایجاد کرد:

```
>> w(١:٣،٣:١)=٤٥,٧٢;
```

هرکدام از این عبارات در پنجره فرمان، ماتریس زیر را تولید می‌کنند:

```
٤٥.٧٢٠٠   ٤٥.٧٢٠٠   ٤٥.٧٢٠٠
٤٥.٧٢٠٠   ٤٥.٧٢٠٠   ٤٥.٧٢٠٠
٤٥.٧٢٠٠   ٤٥.٧٢٠٠   ٤٥.٧٢٠٠
```

حال به بردار زیر توجه کنید:

$$S = [a_۱\ a_۲\ a_۳\ a_۴]$$

عبارت v = repmat(s,۳،۱) معادل عددی ماتریس زیر را ایجاد می‌کند.

$$v = \begin{bmatrix} a_۱ & a_۲ & a_۳ & a_۴ \\ a_۱ & a_۲ & a_۳ & a_۴ \\ a_۱ & a_۲ & a_۳ & a_۴ \end{bmatrix}$$

سه سطر بردار v را ایجاد می‌کند که هر سطر دارای چهار ستون است.

عبارت repmat(s,۳،۲) معادل عددی ماتریس زیر را ایجاد می‌کند.

$$v = \begin{bmatrix} a_۱ & a_۲ & a_۳ & a_۴ & a_۱ & a_۲ & a_۳ & a_۴ \\ a_۱ & a_۲ & a_۳ & a_۴ & a_۱ & a_۲ & a_۳ & a_۴ \\ a_۱ & a_۲ & a_۳ & a_۴ & a_۱ & a_۲ & a_۳ & a_۴ \end{bmatrix}$$

به عبارت دیگر، دستور زیر یک ماتریس سه ستونی معادل عددی بردار ستونی s' را ایجاد می‌کند، که هر ستون دارای چهار سطر است.

```
>> v=repmat(s',۱،۳);
```

$$v = \begin{bmatrix} a_۱ & a_۱ & a_۱ \\ a_۲ & a_۲ & a_۲ \\ a_۳ & a_۳ & a_۳ \\ a_۴ & a_۴ & a_۴ \end{bmatrix}$$

عبارت repmat(s',۱،۳) معادل عددی ماتریس زیر را نتیجه می‌دهد.

$$v = \begin{bmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \\ a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \end{bmatrix}$$

اگر دو بردار t و s سطری باشند، آنگاه عبارت

```
>> [u,v]=meshgrid(s,t)
```

نتیجه یکسانی مانند دوسطر زیر را ارائه می‌دهد:

```
>> u=repmat(s,length(t),l)
>> v=repmat(t´,length(s))
```

به عبارت دیگر، u و v ماتریس‌هایی از مرتبه$(\text{length(t)} \times \text{length(s)})$ می‌باشند. بنابراین اگر

$s = [s_1\ s_2\ s_3\ s_4]$

$t = [t_1\ t_2\ t_3]$

آنگاه دستور زیر معادل عددی دو ماتریس (۳×۴) را ایجاد می‌کند.

```
>> [u,v]=meshgrid(s,t)
```

$$u = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \end{bmatrix}$$

$$v = \begin{bmatrix} t_1 & t_1 & t_1 & t_1 \\ t_2 & t_2 & t_2 & t_2 \\ t_3 & t_3 & t_3 & t_3 \end{bmatrix}$$

تابع meshgrid همچنین همانگونه که در سایت سیم پاور توضیح دادم می‌تواند تنها یک ماتریس مانند زیر را برگرداند:

```
>> w=meshgrid(s,t)
```

که w=u را ایجاد می‌کند که u ماتریس است.

حال دو تابع fliplr(a) و flipud(a) که به ترتیب برای جابجایی سطرها و ستون‌ها به کار می‌روند، معرفی میشوند. ماتریس (۲×۵) زیر را ملاحظه کنید:

$$\mathrm{a} = \begin{bmatrix} a_{۱۱} & a_{۱۲} & a_{۱۳} & a_{۱۴} & a_{۱۵} \\ a_{۲۱} & a_{۲۲} & a_{۲۳} & a_{۲۴} & a_{۲۵} \end{bmatrix} \rightarrow (۲ \times ۵)$$

که با جمله زیر ایجاد می‌گردد:

```
>> a= [a۱۱ a۱۲ a۱۳ a۱٤ a۱٥; a۲۱ a۲۲ a۲۳ a۲٤ a۲٥]
```

بنابراین

$$\mathrm{fliplr(a)} = \begin{bmatrix} a_{۱۵} & a_{۱۴} & a_{۱۳} & a_{۱۲} & a_{۱۱} \\ a_{۲۵} & a_{۲۴} & a_{۲۳} & a_{۲۲} & a_{۲۱} \end{bmatrix} \rightarrow (۲ \times ۵)$$

$$\mathrm{flipud(a)} = \begin{bmatrix} a_{۲۱} & a_{۲۲} & a_{۲۳} & a_{۲۴} & a_{۲۵} \\ a_{۱۱} & a_{۱۲} & a_{۱۳} & a_{۱۴} & a_{۱۵} \end{bmatrix} \rightarrow (۲ \times ۵)$$

$$\mathrm{flipud(fliplr(a))} = \begin{bmatrix} a_{۲۵} & a_{۲۴} & a_{۲۳} & a_{۲۲} & a_{۲۱} \\ a_{۱۵} & a_{۱۴} & a_{۱۳} & a_{۱۲} & a_{۱۱} \end{bmatrix} \rightarrow (۲ \times ۵)$$

همچنین همانگونه که در سایت سیم پاور توضیح دادم می‌توان نتایج توابع fliplr(a)و flipud(a) را با علامت دو نقطه (:) به دست آورد. به طور مثال :

```
>> c=fliplr(a)
```

نتیجه یکسان با عبارت زیر را ایجاد می‌کند.

```
>> c=a(:,length(a):-۱:۱)
```

..... به بردار زیر توجه کنید:

$$c = [a\ \text{fliplr}\ (a)]' = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{21} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \\ a_{15} & a_{25} \\ a_{14} & a_{24} \\ a_{13} & a_{23} \\ a_{12} & a_{22} \\ a_{11} & a_{11} \end{bmatrix}$$

دو سطر یکسان و همانند ۵ و ۶ را ایجاد می‌کند. حذف یکی از این سطرها، با مجموعه یک سطر مقدار خالی مانند عبارت [] انجام می‌شود که فضای بین براکت ها خالی است. بنابراین عبارت

```
>> c(length(a),:)=[]
```

```
>> c(length(a)+۱,:)=[]
```

بردار زیر را ایجاد می‌کند:

$$c = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \\ a_{14} & a_{24} \\ a_{13} & a_{23} \\ a_{12} & a_{22} \\ a_{11} & a_{21} \end{bmatrix}$$

که c یک ماتریس (۹×۲) است. عبارت c(length(a),:)=[] به این معنی است که تمام ستون های سطر با عدد length(a) در c با مقدار [] تعیین شده اند. (در این مورد، حذف شده اند). اگرچه میدانیم طول a، ۵ است این مزیت خوبی است که متلب سایت سیم پاور اجازه انجام شمارش را می‌دهد. از این رو، از تابع length(a) استفاده می‌شود.

علائم فوق را با معرفی نتایج سه عملگر مختلف متلب سایت سیم پاور بیشتر توضیح می‌دهیم. اول، دو ماتریس (۲×۵) با نام a و b را به صورت زیر در نظر بگیرید.

$$\mathrm{b} = \begin{bmatrix} b_{۱۱} & b_{۱۲} & b_{۱۳} & b_{۱۴} & b_{۱۵} \\ b_{۲۱} & b_{۲۲} & b_{۲۳} & b_{۲۴} & b_{۲۵} \end{bmatrix} \quad \mathrm{a} = \begin{bmatrix} a_{۱۱} & a_{۱۲} & a_{۱۳} & a_{۱۴} & a_{۱۵} \\ a_{۲۱} & a_{۲۲} & a_{۲۳} & a_{۲۴} & a_{۲۵} \end{bmatrix}$$

حال کاربرد این سه عملگر را ملاحظه کنید: