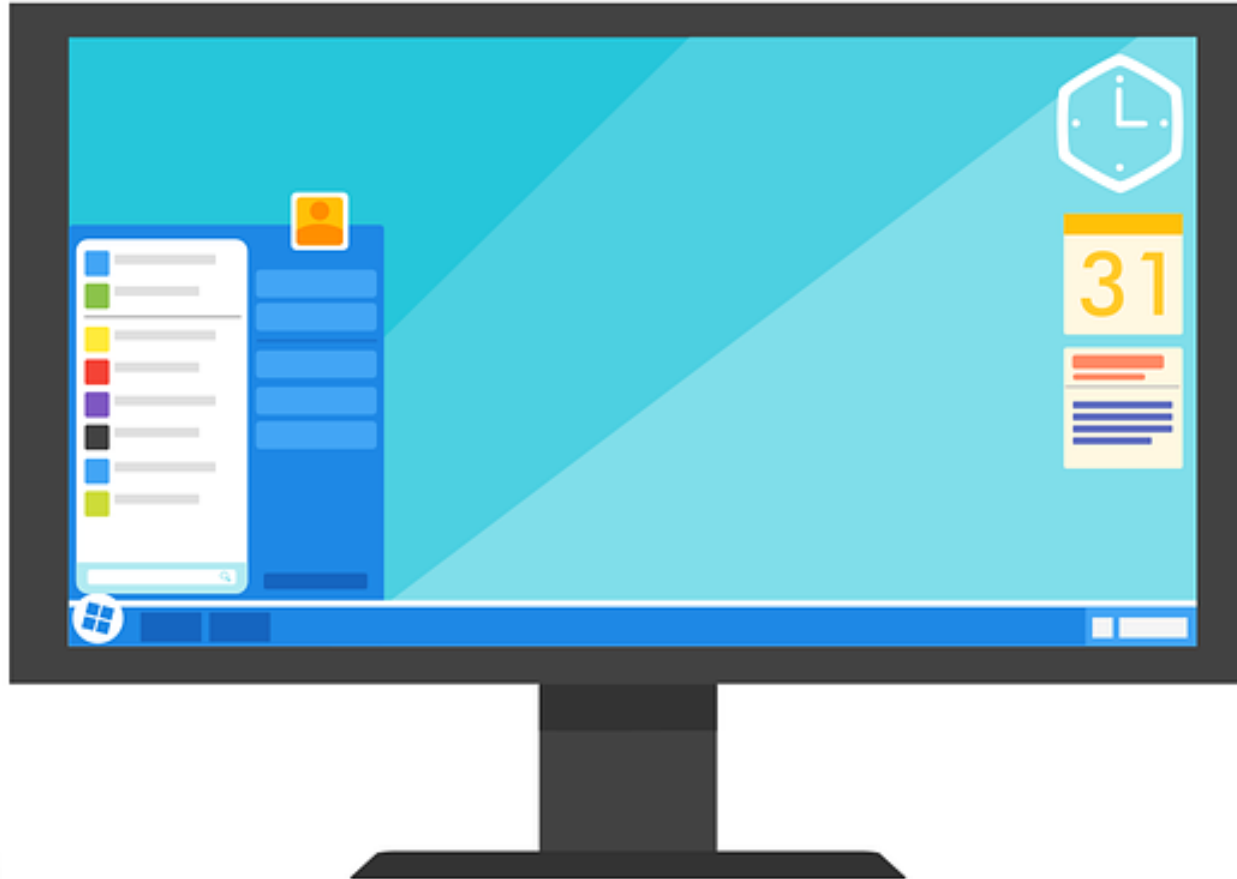


بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

برنامه نویسی کامپیوتر با محوریت آموزش نرم افزار MATLAB



مدرس: مجتبه قربانے | کارشناسے ارشد مهندسے برق



دانشگاه تربت حیدریه

ویرایش پنجم - پاییز ۹۸

آموزش نرم افزار MATLAB

فصل چهارم ...

توابع در MATLAB

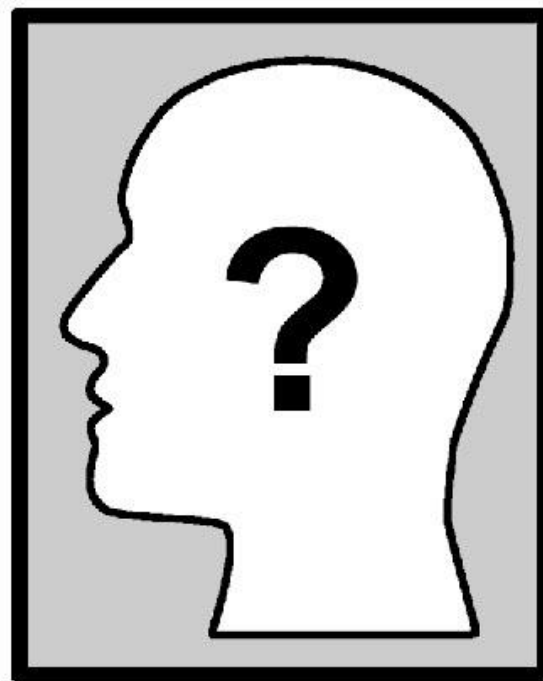
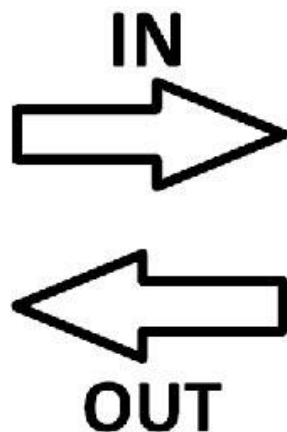
www.ieeei.blog.ir



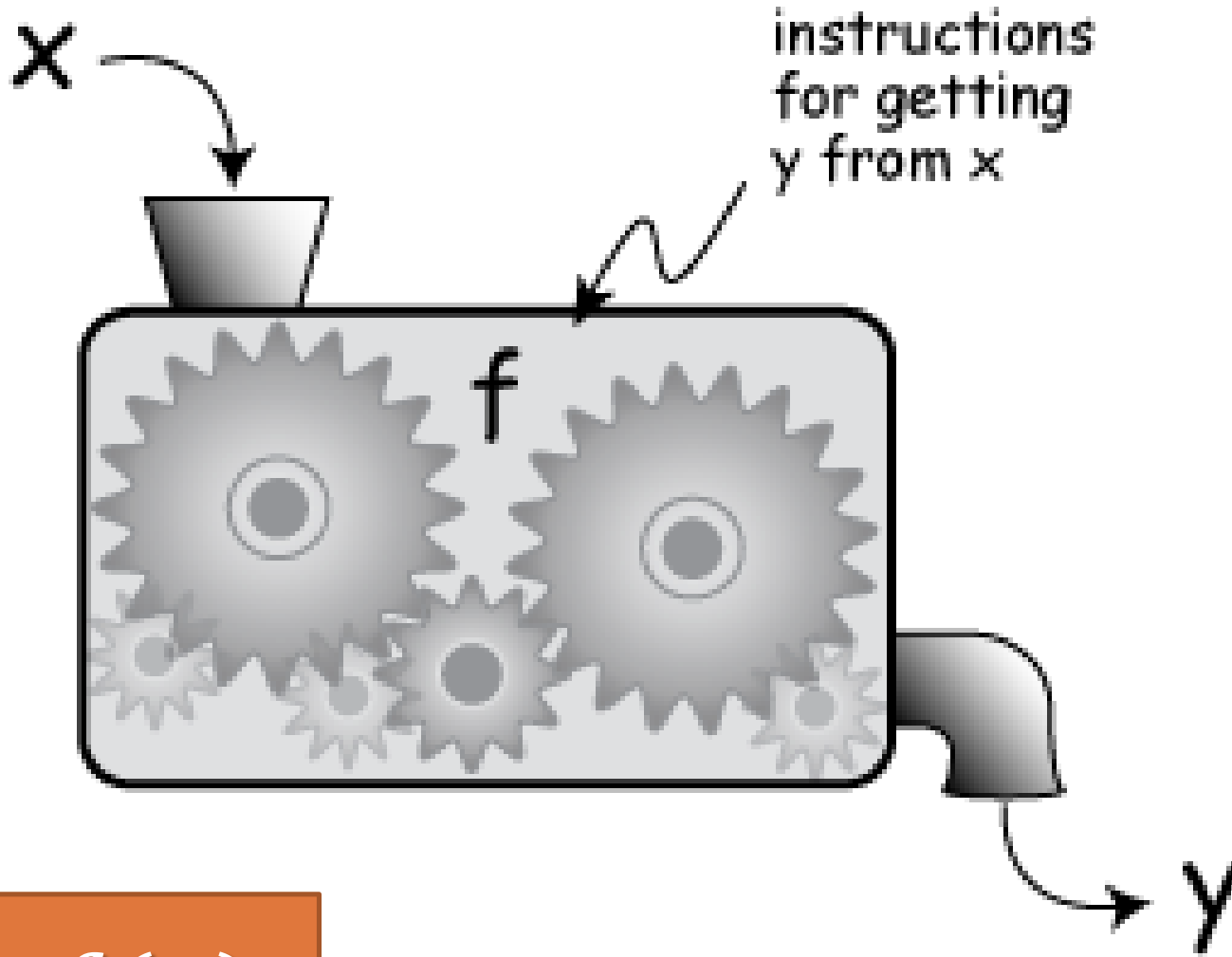
استفاده از متلب :

- تا کنون با فضای کلی و نحوه محاسبات در متلب آشنا شده ایم، اما در این فصل چگونگی استفاده از دستورات مختلف در متلب توضیح داده خواهد شد.

هنر خوب دستور دادن...



تابع :



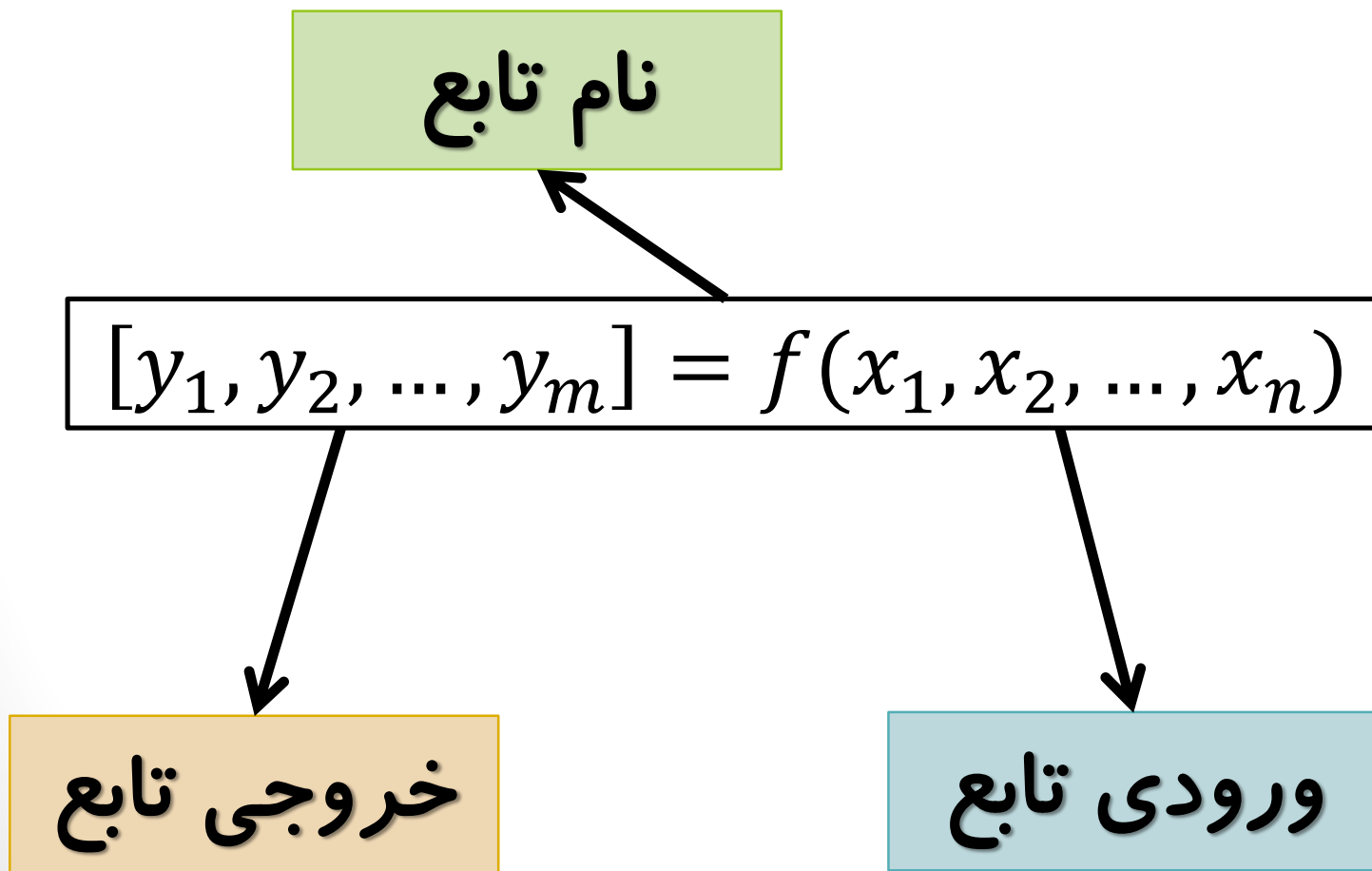
$$y = f(x)$$





تابع در متلب :

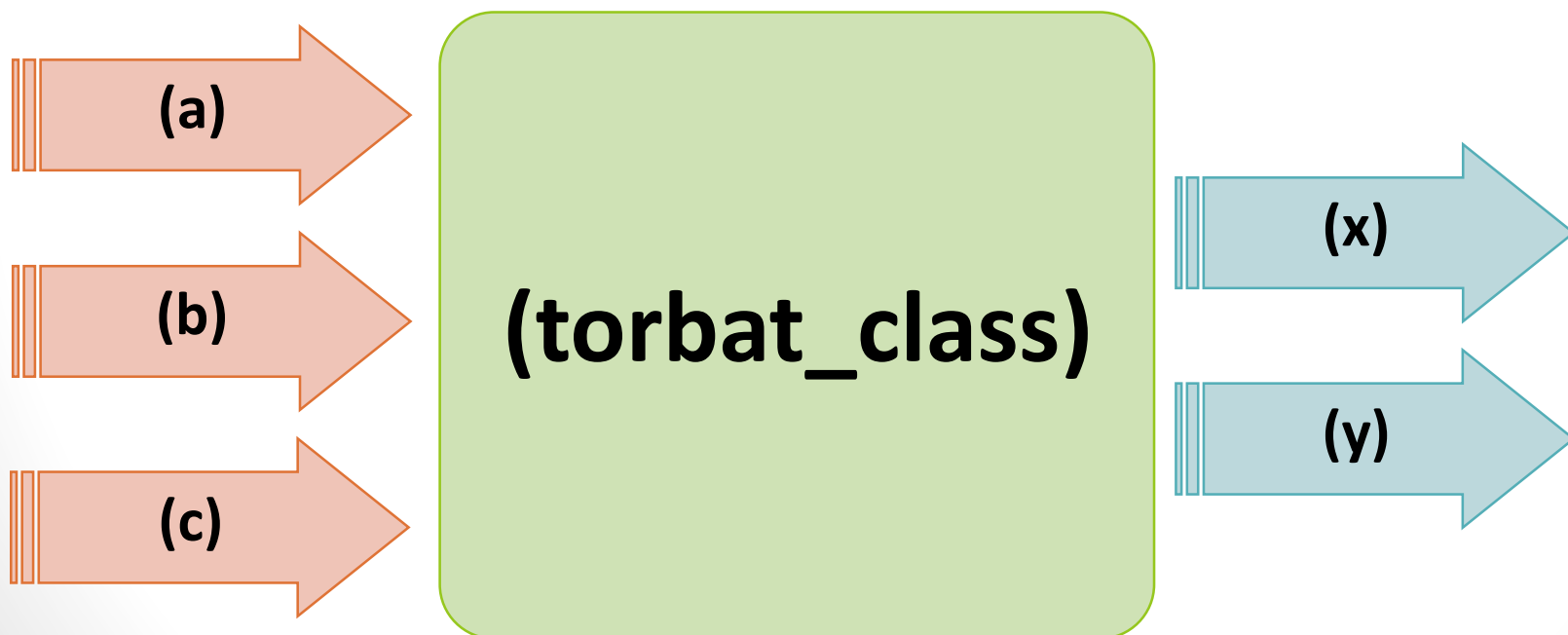
- قاعده کلی برای نمایش توابع در متلب به صورت زیر می باشد.



تابع در متلب :

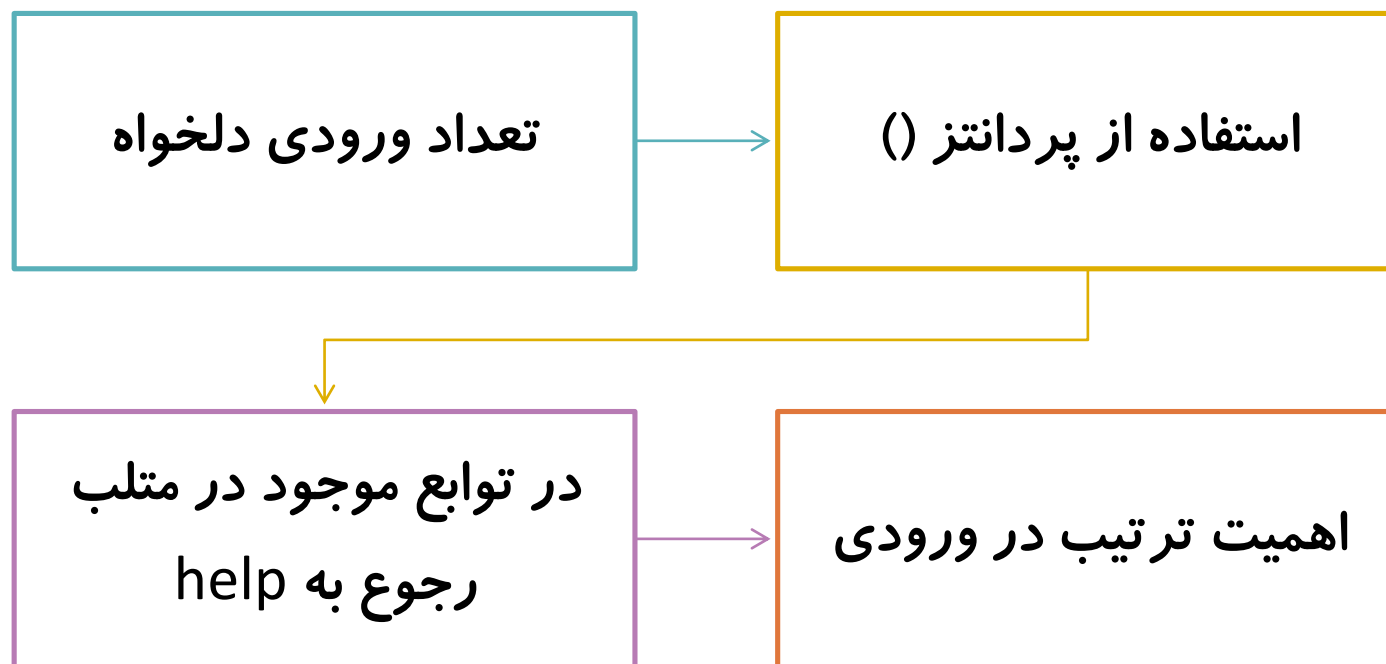
- برای مثال اگر تابع زیر را در متلب تعریف کنیم ، ورودی و خروجی آن به صورت زیر خواهد بود.

```
function [ x , y ] = torbat_class( a , b , c )
```



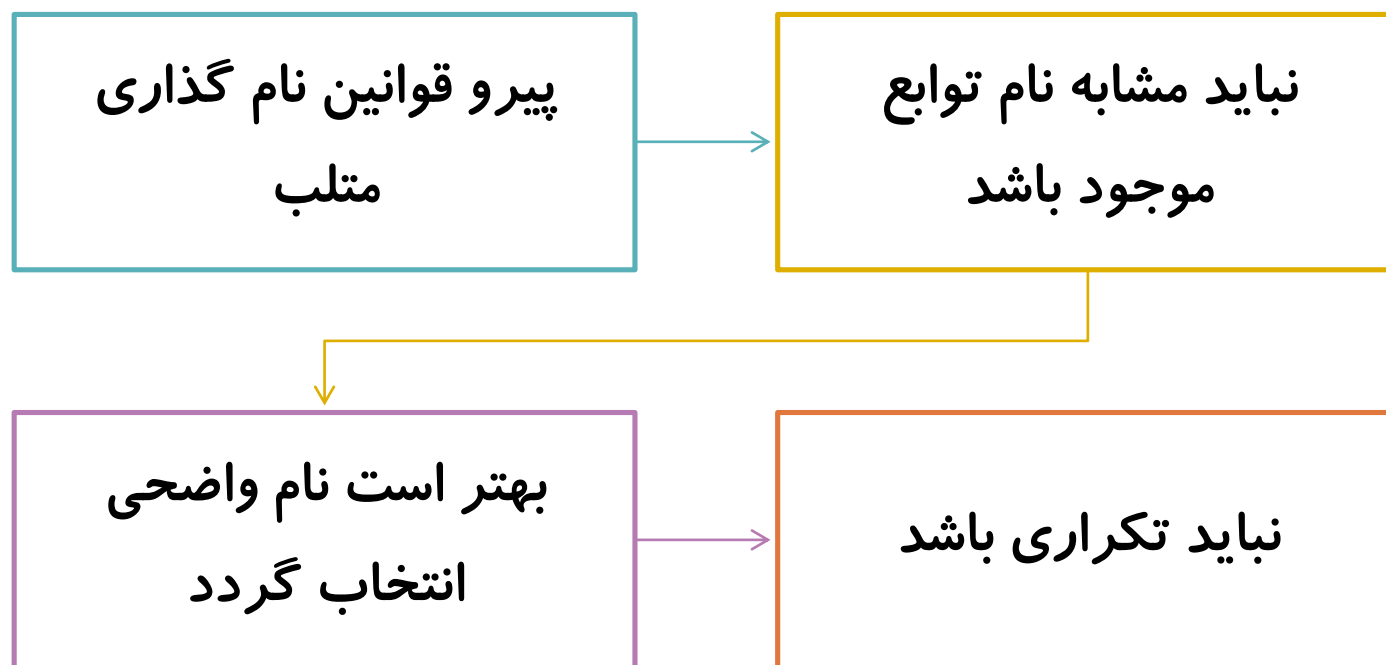
ورودی در توابع متلب :

- در ورودی توابع مختلف باید نکات زیر را مد نظر قرار داد.



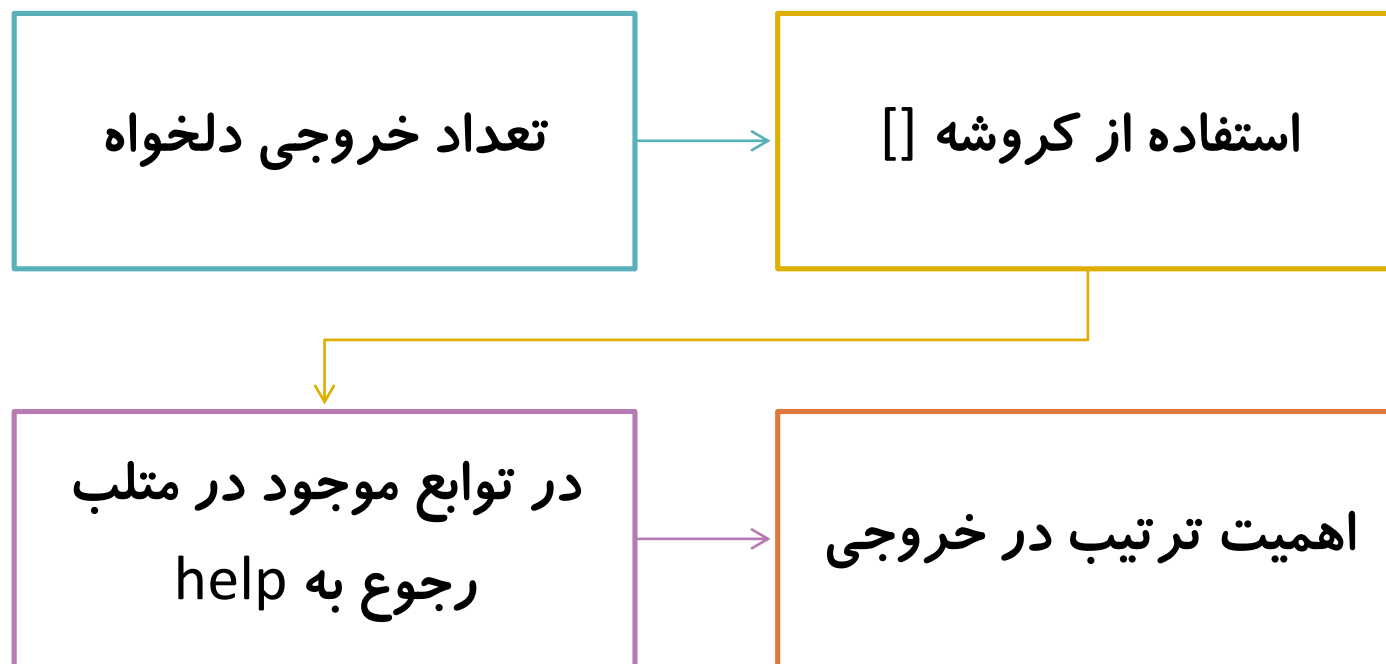
نام گذاری در توابع متلب :

- در نام گذاری توابع مختلف باید نکات زیر را مد نظر قرار داد.



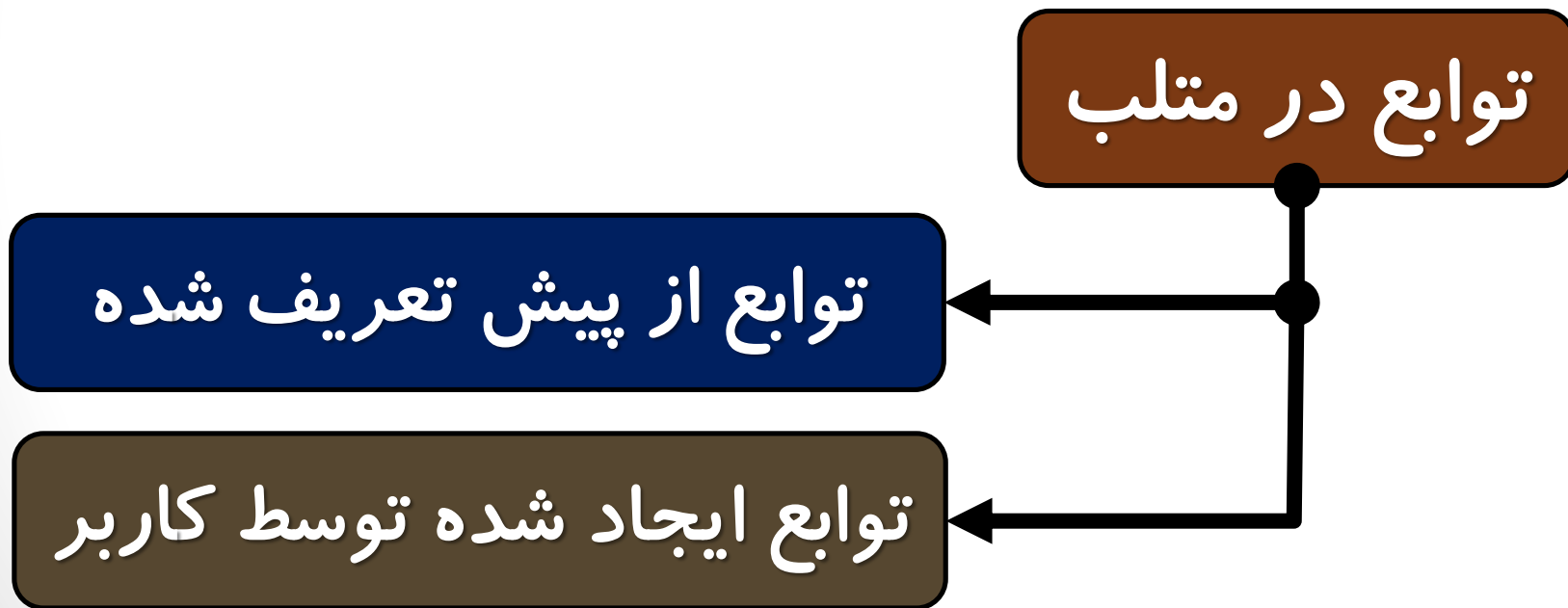
خروجی در توابع متلب :

- در خروجی توابع مختلف باید نکات زیر را مد نظر قرار داد.



استفاده از توابع در متلب:

- اگر بخواهیم از یک تابع در متلب استفاده کنیم باید قبلا آن تابع توسط متلب شناخته شده باشد.
- به طور کلی توابع در متلب به دو گروه تقسیم می شوند:



توابع از پیش تعریف شده در متلب :

- نرم افزار متلب مجموعه ای بسیار کامل از توابع مختلف را شامل می شود.
- این توابع به هنگام نصب نرم افزار در محل نصب متلب بر روی سیستم کپی می شوند.
- به علت وجود همین توابع بسیار زیاد است که متلب یکی از بهترین ابزارهای محاسبه برای پژوهشگران محسوب می گردد.
- در ادامه برخی از توابع کاربردی معرفی خواهد شد.



معرفی توابع کاربردی - ماتریس :

ones(n,m)	ماتریس n در m با درایه های یک
ones(n)	ماتریس n در n با درایه های یک
zeros(n,m)	ماتریس n در m با درایه های صفر
zeros(n)	ماتریس n در n با درایه های صفر
eye(n)	ماتریس همانی با بعد n
rand(n,m)	ماتریس n در m با درایه های تصادفی
rand(n)	ماتریس n در n با درایه های تصادفی
randperm(n)	اعداد ۱ تا n به صورت تصادفی در یک بردار سطری
randi([a,b],n,m)	اعداد صحیح بین a و b در یک ماتریس n در m



مثال :

```
>> a1=ones(2,4)
```

```
a1 =  
     1     1     1     1  
     1     1     1     1
```

```
>> a2=ones(3)
```

```
a2 =  
     1     1     1  
     1     1     1  
     1     1     1
```

```
>> a3=zeros(1,5)
```

```
a3 =  
     0     0     0     0     0
```

```
>> a4=zeros(3)
```

```
a4 =  
     0     0     0  
     0     0     0  
     0     0     0
```

```
>> a5=eye(4)
```

```
a5 =  
     1     0     0     0  
     0     1     0     0  
     0     0     1     0  
     0     0     0     1
```

```
>> a6=rand(1,4)
```

```
a6 =  
     0.9502     0.0344     0.4387     0.3816
```

```
>> a7=rand(2,2)
```

```
a7 =  
     0.7655     0.1869  
     0.7952     0.4898
```

```
>> a8=randperm(5)
```

```
a8 =  
     5     1     2     3     4
```

```
>> a9=randi([-2,2],3,2)
```

```
a9 =  
     1    -2  
     1     0  
    -2     2
```



معرفی توابع کاربردی - ماتریس :

سایز ماتریس x	$[m,n]=size(x)$
بزرگترین عدد و موقعیت بزرگترین عدد	$[a,b]=max(x)$
کوچکترین عدد و موقعیت کوچکترین عدد	$[c,d]=min(x)$
سایز بردار x	$L=length(x)$
حاصل جمع مولفه های x	$M=sum(x)$
حاصل ضرب مولفه های x	$P=prod(x)$
به صورت صعودی مرتب می کند.	$[a,b]=sort(x)$



مثال :

```
>> B=randi([0,9],5,4)
B =
     5     3     1     9
     5     5     1     9
     1     4     2     4
     8     0     4     4
     6     2     0     3

>> b1=size(B)
b1 =
     5     4

>> [b2,b3]=size(B)
b2 =
     5
b3 =
     4

>> b1=length(B)
b1 =
     5

>> [b2,b3]=length(B)
Error using length
Too many output arguments.
```

```
>> b4=min(B)
b4 =
     1     0     0     3

>> [b4,b5]=min(B)
b4 =
     1     0     0     3
b5 =
     3     4     5     5

>> b6=min(min(B))
b6 =
     0

>> b7=max(B)
b7 =
     8     5     4     9

>> [b8,b9]=max(B)
b8 =
     8     5     4     9
b9 =
     4     2     4     1

>> b10=max(max(B))
b10 =
     9
```



مثال :

```
>> F=randi([0,20],3,4)
```

```
F =
```

17	8	13	9
16	11	13	0
1	8	6	20

```
>> f1=sum(F)
```

```
f1 =
```

34	27	32	29
----	----	----	----

```
>> f2=sum(sum(F))
```

```
f2 =
```

122

```
>> f3=prod(F)
```

```
f3 =
```

272	704	1014	0
-----	-----	------	---

```
>> f4=sort(F)
```

```
f4 =
```

1	8	6	0
16	8	13	9
17	11	13	20

```
>> [f5,f6]=sort(F)
```

```
f5 =
```

1	8	6	0
16	8	13	9
17	11	13	20

```
f6 =
```

3	1	3	2
2	3	1	1
1	2	2	3



برخی توابع کاربردی - رابط کاربری:

دستورات پاک کردن

`clc` پاک کردن صفحه نمایش

`clear` پاک کردن (delete) تمامی متغیرها

`clear x` پاک کردن (delete) متغیر `x`



برخی توابع کاربردی - مثلثاتی:

توابع مثلثاتی

کسینوس هیپربولیک	$\cosh(x)$	معکوس کسینوس، رادیان	$\operatorname{acos}(x)$
کتانژانت، رادیان	$\cot(x)$	معکوس کسینوس، درجه	$\operatorname{acosd}(x)$
کتانژانت، درجه	$\cotd(x)$	معکوس کسینوس هیپربولیک	$\operatorname{acosh}(x)$
کتانژانت هیپربولیک	$\coth(x)$	معکوس کتانژانت، رادیان	$\operatorname{acot}(x)$
کسکانت، رادیان	$\csc(x)$	معکوس کتانژانت، درجه	$\operatorname{acotd}(x)$
کسکانت، درجه	$\cscd(x)$	معکوس کتانژانت هیپربولیک	$\operatorname{acoth}(x)$
کسکانت هیپربولیک	$\operatorname{csch}(x)$	معکوس کسکانت، رادیان	$\operatorname{acsc}(x)$
سکانت، رادیان	$\sec(x)$	معکوس کسکانت، درجه	$\operatorname{acscd}(x)$
سکانت، درجه	$\secd(x)$	معکوس کسکانت هیپربولیک	$\operatorname{acsch}(x)$
سکانت هیپربولیک	$\operatorname{sech}(x)$	معکوس سکانت، رادیان	$\operatorname{asec}(x)$



مثال :

```
>> a1=sin(0)
a1 =
    0
>> a2=cos(0)
a2 =
    1
>> a3=sin(90)
a3 =
    0.8940
>> a4=cos(90)
a4 =
   -0.4481
>> a5=sin(pi/2)
a5 =
    1
>> a6=cos(pi/2)
a6 =
   6.1232e-17
```

```
>> b1=sind(0)
b1 =
    0
>> b2=cosd(0)
b2 =
    1
>> b3=sind(90)
b3 =
    1
>> b4=cosd(90)
b4 =
    0
>> b5=sind(pi/2)
b5 =
    0.0274
>> b6=cosd(pi/2)
b6 =
    0.9996
```



برخی توابع کاربردی – نمایی:

توابع نمایی

e^x (ورودی حقیقی و مختلط)	<code>exp (x)</code>
لگاریتم در مبنای e (ورودی حقیقی و مختلط)	<code>log (x)</code>
لگاریتم در مبنای ۲ (ورودی حقیقی و مختلط)	<code>log2 (x)</code>
لگاریتم در مبنای ۱۰ (ورودی حقیقی و مختلط)	<code>log10 (x)</code>
لگاریتم در مبنای e (ورودی فقط اعداد حقیقی مثبت)	<code>reallog (x)</code>
جذر (ورودی فقط اعداد حقیقی نامنفی)	<code>realsqrt (x)</code>
جذر (ورودی حقیقی و مختلط)	<code>sqrt (x)</code>
$\sqrt[y]{x}$	<code>nthroot (x, y)</code>



مثال :

```
>> a=exp(1)
a =
    2.7183
>> b=log(2)
b =
    0.6931
>> c=sqrt(9)
c =
     3
>> d=nthroot(9,2)
d =
     3
>> e=9^(1/2)
e =
     3
```

```
>> g=exp(exp(exp(1)))
g =
    3.8143e+06
>> h=log(exp(log(exp(1))))
h =
     1
>> i=sqrt(sqrt(sqrt(256)))
i =
     2
```



برخی توابع کاربردی – محاسباتی:

توابع گرد کردن

گرد کردن به سمت صفر	<code>fix(x)</code>
گرد کردن به سمت منفی بی نهایت	<code>floor(x)</code>
گرد کردن به سمت مثبت بی نهایت	<code>ceil(x)</code>
قدر مطلق	<code>abs(x)</code>
گرد کردن به سمت نزدیک ترین عدد صحیح	<code>round(x)</code>
نمایش عدد x با d رقم اعشار	<code>vpa(x, d)</code>



مثال :

```
>> a=fix(pi)
a =
     3
>> b=fix(-pi)
b =
    -3
>> c=floor(pi)
c =
     3
>> d=floor(-pi)
d =
    -4
>> e=ceil(pi)
e =
     4
>> f=ceil(-pi)
f =
    -3
```

```
>> g=abs(pi)
g =
     3.1416
>> h=abs(-pi)
h =
     3.1416
>> i=round(pi)
i =
     3
>> j=round(-pi)
j =
    -3
>> k=vpa(pi,6)
k =
     3.14159
```



برخی توابع کاربردی – محاسباتی:

توابع ریاضیات گسسته

تجزیه x به عوامل اول	<code>factor(x)</code>
$x!$	<code>factorial(x)</code>
بزرگ‌ترین مقسوم‌علیه مشترک x و y	<code>gcd(x, y)</code>
کوچک‌ترین مضرب مشترک x و y	<code>lcm(x, y)</code>
نمایش ۱ در صورت اول بودن x و در غیر این صورت، نمایش ۰	<code>isprime(x)</code>
$\binom{x}{y}$	<code>nchoosek(x, y)</code>
نمایش اعداد اول از ۲ تا x	<code>primes(x)</code>



برخی توابع کاربردی - اعداد مختلط:

توابع اعداد مختلط

i	$\sqrt{-1}$
j	$\sqrt{-1}$
<code>abs(z)</code>	محاسبه مقدار قدر مطلق
<code>angle(z)</code>	محاسبه مقدار زاویه بر حسب رادیان
<code>conj(z)</code>	محاسبه مزدوج
<code>imag(z)</code>	نمایش قسمت موهومی
<code>real(z)</code>	نمایش قسمت حقیقی
<code>isreal(z)</code>	اگر z عدد حقیقی باشد، مقدار ۱ و اگر متلط باشد، صفر را برمی گرداند.
<code>complex(a,b)</code>	ایجاد یک عدد مختلط به فرم $a+bi$



برخی توابع کاربردی - رشته ها:

findstr	درون یک رشته را برای یک کلمه جستجو می کند.
ischar	بررسی می کند که آیا ورودی کارکتر است ؟
strcmp	مقایسه رشته ها را انجام می دهد.
strcmpi	مقایسه رشته ها بدون حساسیت به بزرگی و کوچکی حروف
strrep	جستجو و جایگزینی
isletter	بررسی می کند که یک کارکتر حرف است یا خیر
isspace	به ازای فضای خالی یک را بر میگرداند.
int2str	تبدیل یک مقدار اسکالر به یک آرایه کارکتری
num2str	تبدیل یک مقدار اسکالر به یک آرایه کارکتری (کاربردی تر)
upper	تمامی کارکترهای حرف درون یک رشته به حروف بزرگ تبدیل می کند.
lower	تبدیل به حروف کوچک

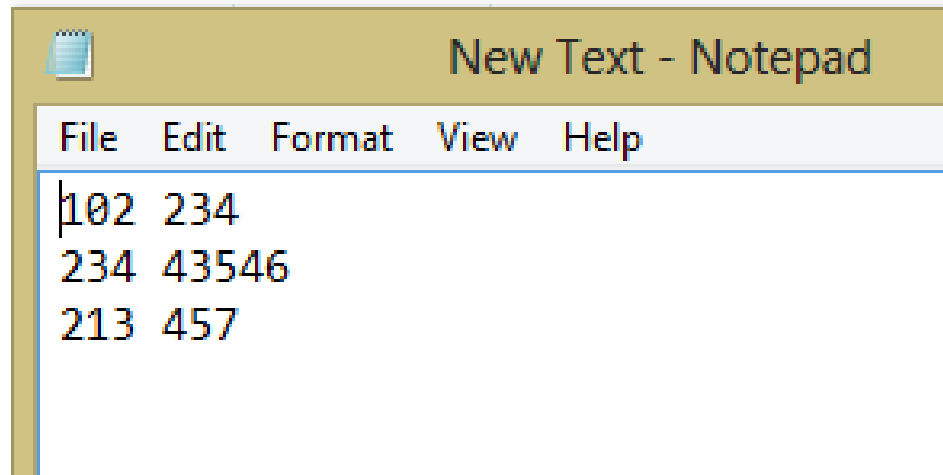


برخی توابع کاربردی - ارتباط با فایل:

مقادیر موجود در فضای کاری (work space) را در فایل matlab.m ذخیر می کند.	save
تمام مقادیر فضای کاری را در aaa.m ذخیر می کند.	save aaa
مقادیر فضای کاری که در فایل bbb.m ذخیره شده بودند را باز می کند.	load bbb
فایل ccc را پاک میکند.	delete ccc.txt
فایل ddd را باز می کند.	open ddd
نام تمامی m فایل های موجود در مسیر را نشان می دهد.	whos
تمامی دستورات نوشته شده در محیط command به جز علامت >> را در فایل text ذخیر می کند.	diary file
لیست متغیرهای موجود در فضای کاری را نشان می دهد.	whos
به دنبال یک m فایل مشخص شده می گردد.	lookfor
باز کردن فایل اکسل ذخیره شده.	xlsread



مثال از نحوه ارتباط با فایل :



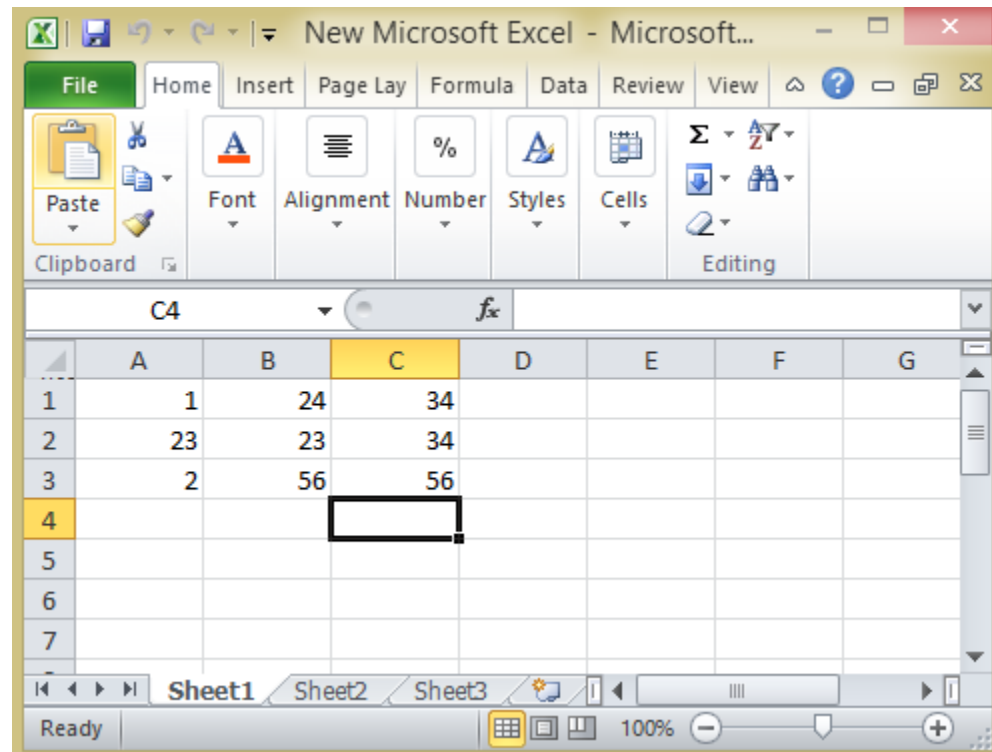
```
>> a=load('New Text.txt')
```

```
a =
```

```
102      234
234      43546
213      457
```



مثال از نحوه ارتباط با فایل :



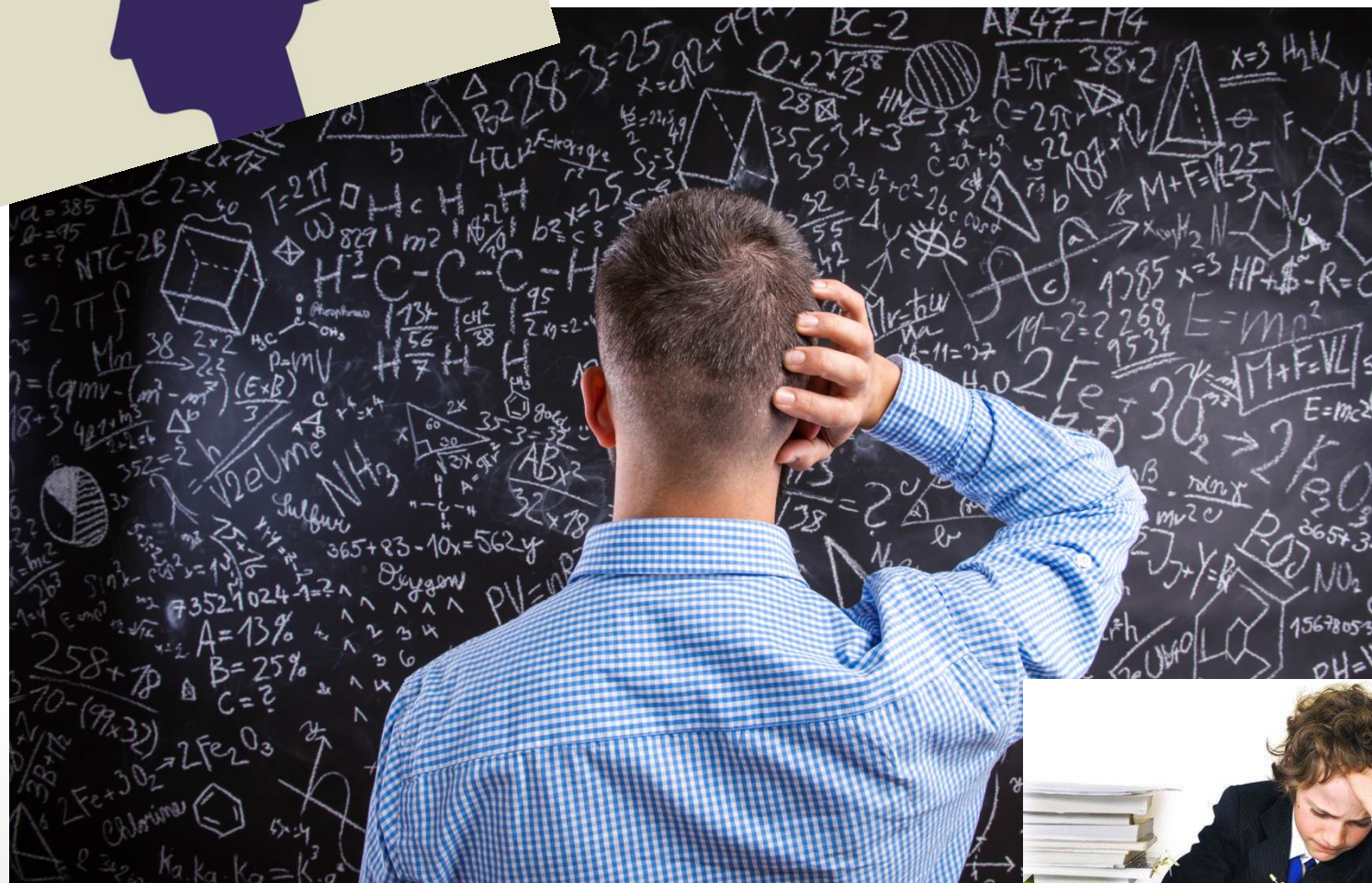
```
>> b=xlsread('New Microsoft Excel.xlsx')
```

```
b =
```

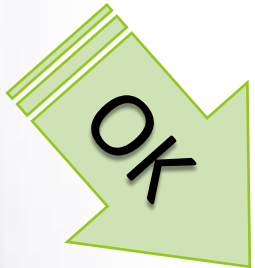
```
     1     24     34
    23     23     34
     2     56     56
```



توابع بی شمار متلب :



بهترین کمک یار در فراگیری مطلب :



Help



ساختن تابع دلخواه :

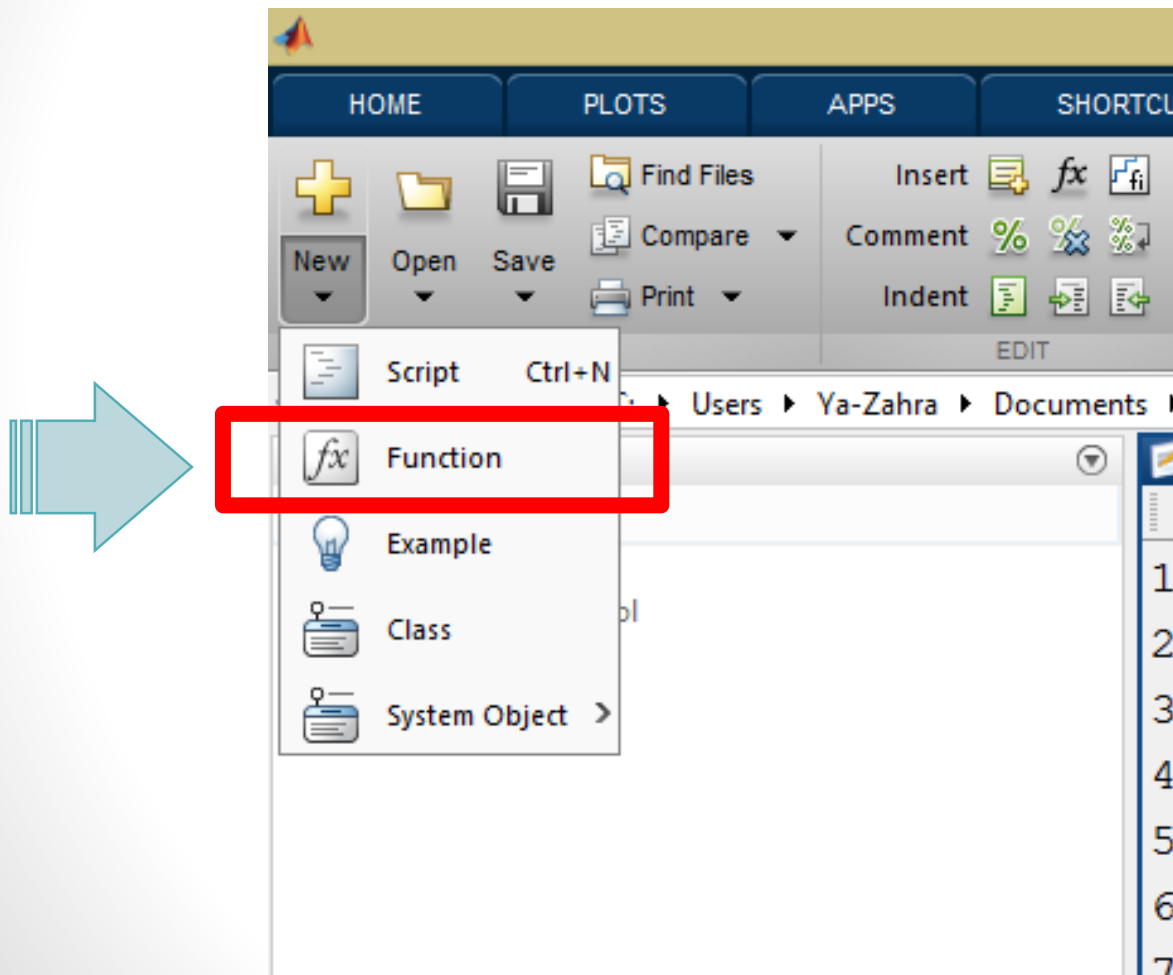
- در بسیاری از موارد لازم است تا متناسب با نیاز توابع مختلفی تعریف و ساخته شود. از سویی دیگر لازم است تا برای حل مسئله و یا انجام پروژه ای ، الگوریتمی خاص انجام گیرد ، لذا تعریف و ساختن توابع مورد نظر یکی از ضروری ترین بخش های هر زبان برنامه نویسی می باشد.
- در ادامه روند تعریف توابع مختلف در متلب تشریح خواهد شد.

توابع ایجاد شده توسط کاربر



ایجاد تابع در متلب :

- راحت ترین روش برای ایجاد یک تابع در متلب استفاده از قسمت نواز ابزار این نرم افزار می باشد.



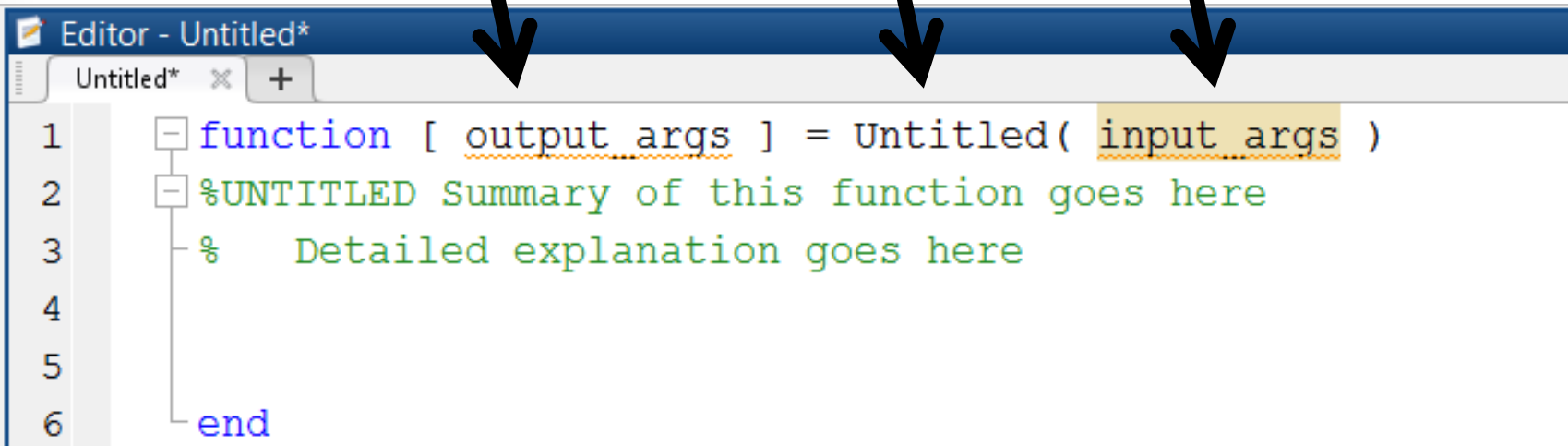
محیط کدنویسی تابع (function):

$$[y_1, y_2, \dots, y_m] = f(x_1, x_2, \dots, x_n)$$

خروجی تابع

نام تابع

ورودی تابع



```
Editor - Untitled*
Untitled* x +
1 function [ output_args ] = Untitled( input_args )
2 %UNTITLED Summary of this function goes here
3 % Detailed explanation goes here
4
5
6 end
```



محل ذخیره سازی تابع :

- برای استفاده از توابع نوشته شده ، باید این توابع را ساخت و سپس استفاده نمود. برای ساخته شدن یک تابع باید بر روی گزینه (Run) کلیک نمود. در صورتی که اولین بار باشد که تابع ساخته شود لازم است تا تابع را ذخیره کرده و سپس اجرا نماییم.

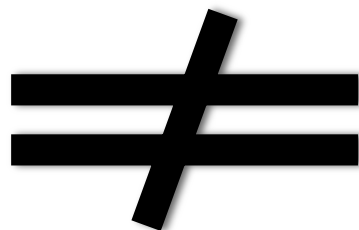


- توجه شود که پس از ذخیره سازی یک تابع لازم است تا `current folder` در همان مسیری باشد که تابع ذخیره شده است و گرنه تابع اجرا نخواهد شد . به عبارتی دیگر تنها توابعی قابل اجرا شدن هستند که در قسمت `current folder` قابل مشاهده باشند.



تفاوت مهم :

- دقت شود که تا وقتی که برنامه ای ایجاد نگردد (کامپایل کردن) از آن برنامه نمی توان استفاده نمود. در متلب نیز این قاعده وجود دارد و لذا همواره باید برنامه را قبل از استفاده اجرا به عبارتی run نمود.



- در run کردن ، ابتدا برنامه ذخیره شده و سپس اجرا می گردد ولی در save کردن فقط برنامه ذخیره می گردد و اجرا نمی شود.



مثال :

- اکنون برنامه زیر را می خواهیم به صورت مرحله به مرحله نوشته و ایجاد نماییم.

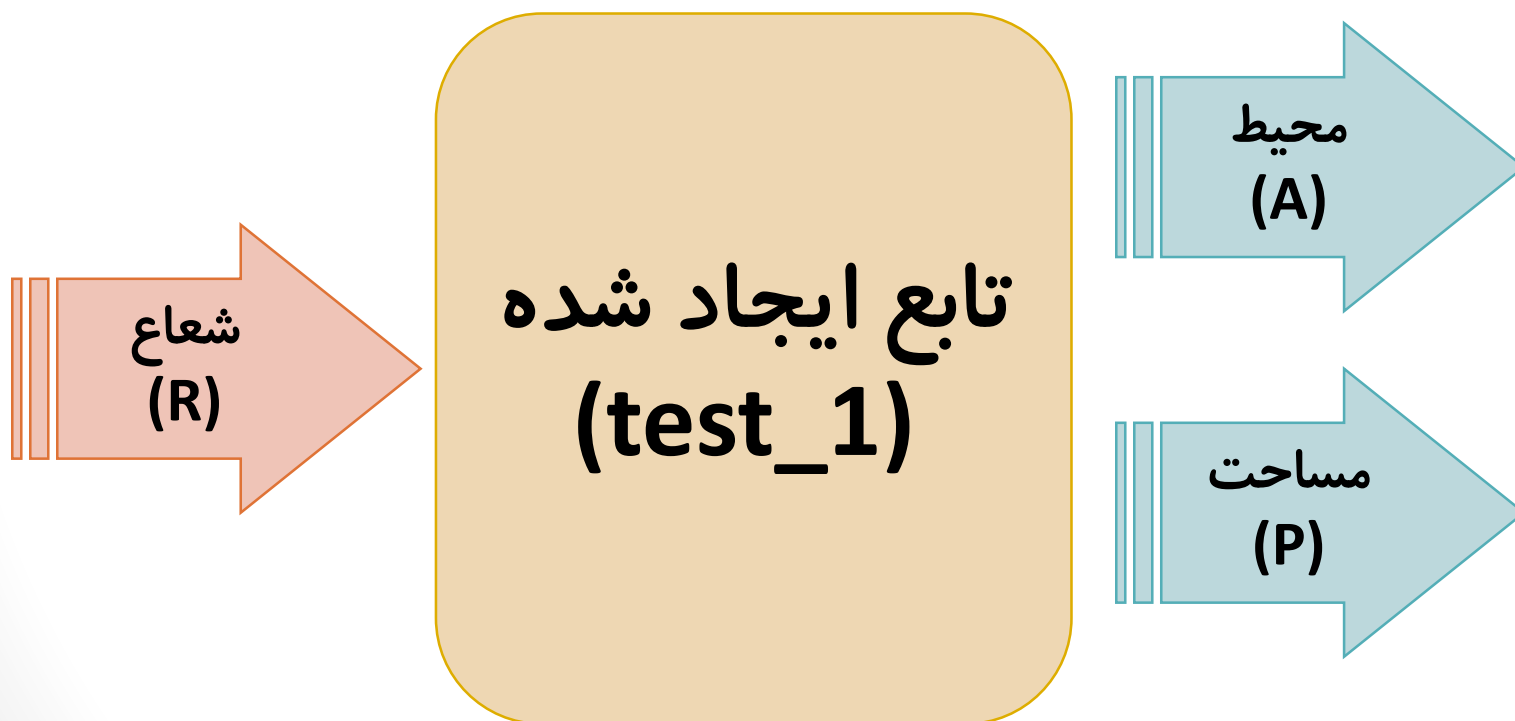
مثال ۱

برنامه ای بنویسید که شعاع یک دایره را گرفته و سپس مساحت و محیط آن را نمایش دهد.



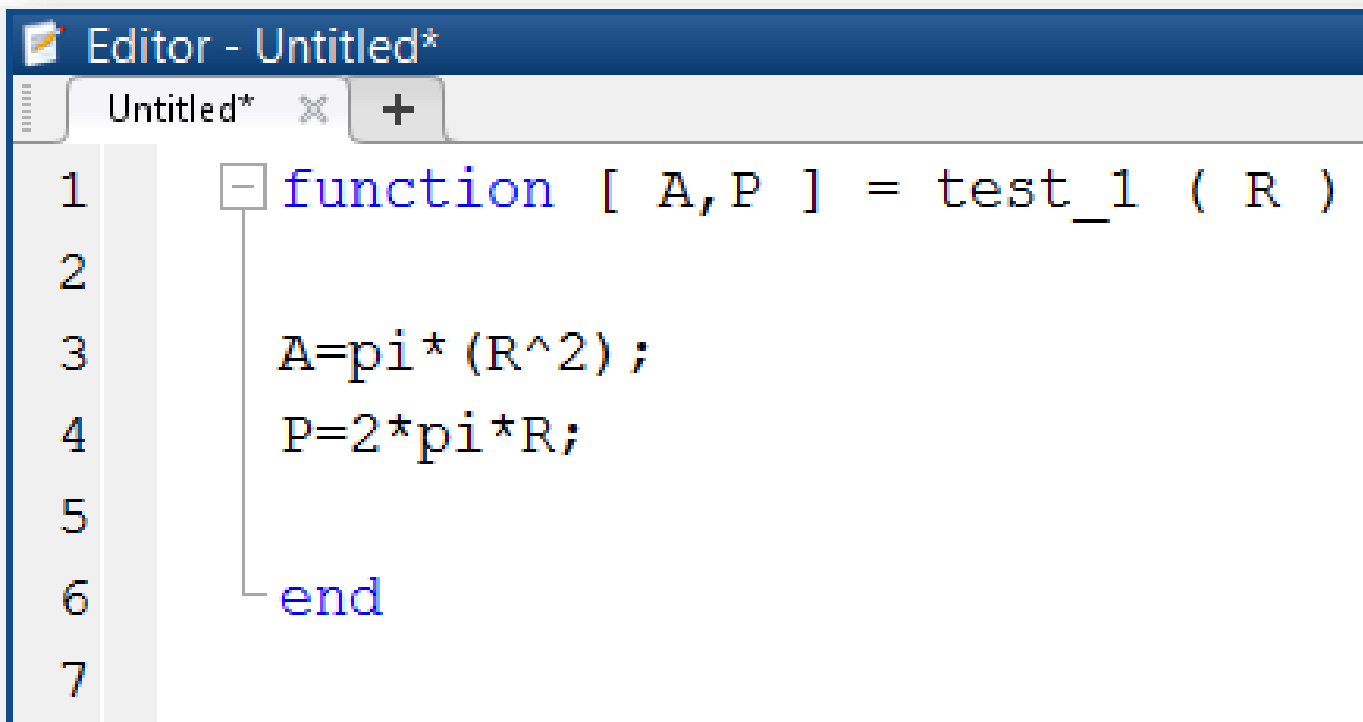
گام اول - انتخاب ورودی و خروجی :

- در این برنامه فرض می کنیم که شعاع به عنوان ورودی این تابع و نیز مقادیر محیط و مساحت محاسبه شده به عنوان خروجی این تابع باشد.



گام دوم - پیاده سازی الگوریتم :

- متناسب با الگوریتم در نظر گرفته شده ، دستورات در محیط function نوشته می شود.
- نام مورد نظر مطابق با قواعد نام گذاری در متلب باید انتخاب شود.
- ورودی های تابع در قسمت ورودی و خروجی های تابع باید قسمت خروجی ها قرار گیرد.

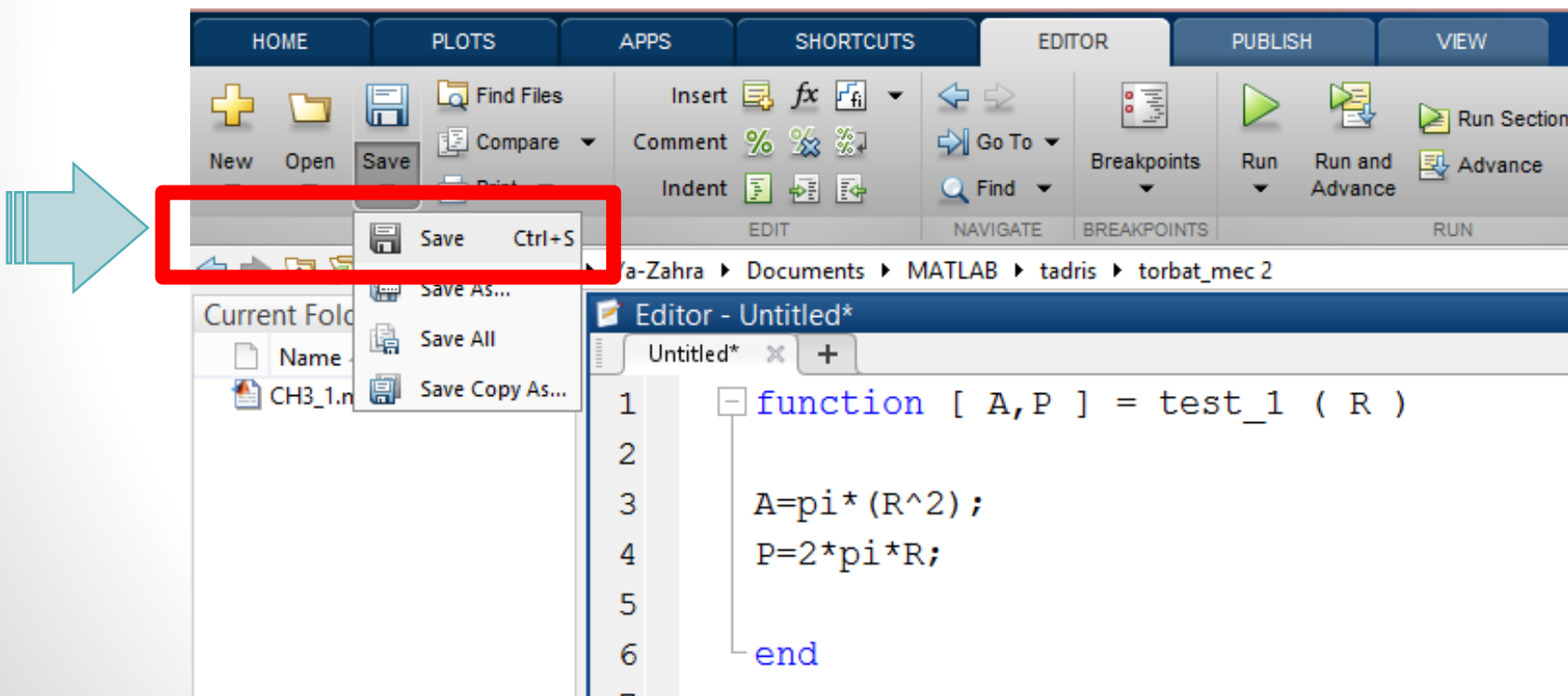


```
Editor - Untitled*
Untitled* x +
1 function [ A,P ] = test_1 ( R )
2
3     A=pi*(R^2);
4     P=2*pi*R;
5
6 end
7
```



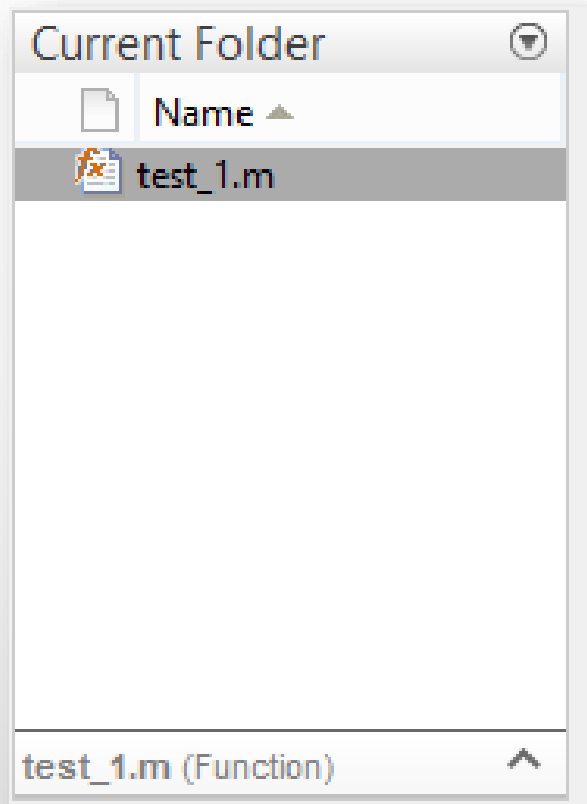
گام سوم - ذخیره سازی تابع :

- بهتر است در ابتدا و یا بعد از نوشتن هرچند خط کد ، تابع ذخیره شود.
- توصیه می شود تابع را در محل مناسب و از پیش در نظر گرفته شده ذخیره نمایید. در صورت عدم مشاهده تابع ذخیره شده در محل current folder ، تابع ساخته شده اجرا نخواهد شد.



گام چهارم - استفاده از تابع :

- در صورتی که تابع ساخته شده در محل current folder ایجاد شده باشد، می توان از تابع استفاده نمود.



```
Command Window

>> test_1
Error using test_1 (line 3)
Not enough input arguments.

>> [masahat,mohit]=test_1(10)

masahat =

    314.1593

mohit =

    62.8319
```



مثال :

- برنامه ای بنویسید که یک عدد دو رقمی را گرفته و سپس مقلوب آن را نشان دهد.

Command Window

```
>> q=maghloob_1(27)
q =
    72
>> q2=maghloob_1(45)
q2 =
    54
>> maghloob_1(1)
ans =
    10
fx >> |
```



بررسی ورودی و خروجی مثال:

- در این برنامه یک ورودی و یک خروجی داریم.



کد برنامه :

```
Editor - C:\Users\Ya-Zahra\Documents\MATLAB\tadris\Torbat_Un
maghloob_1.m*  ✕  +
1  function [ m ] = maghloob_1( n )
2
3  a=mod(n,10);
4  b=(n-a)/10;
5  m=a*10+b;
6
7  end
8
9
10
```



مثال :

- برنامه مقلوب یک عدد را که در مثال قبل توضیح داده شده بود، این بار برای یک عدد سه رقمی بنویسید.

Command Window

```
>> q1=maghloob_2(123)
q1 =
    321
>> q2=maghloob_2(876)
q2 =
    678
>> q3=maghloob_2(1)
q3 =
    100
>> maghloob_2(313)
ans =
    313
```



کد برنامه :

```
Editor - C:\Users\Ya-Zahra\Documents\MATLAB\tadris\Torbat_Uni\m
maghloob_1.m* x maghloob_2.m x +
1 function [ m ] = maghloob_2( n )
2
3 a=mod(n,10) ;
4 b=(n-a)/10 ;
5 c=mod(b,10) ;
6 d=(b-c)/10 ;
7 m= a*100+c*10+d;
8
9 end
10
11
```



استفاده از برنامه های ایجاد شده:

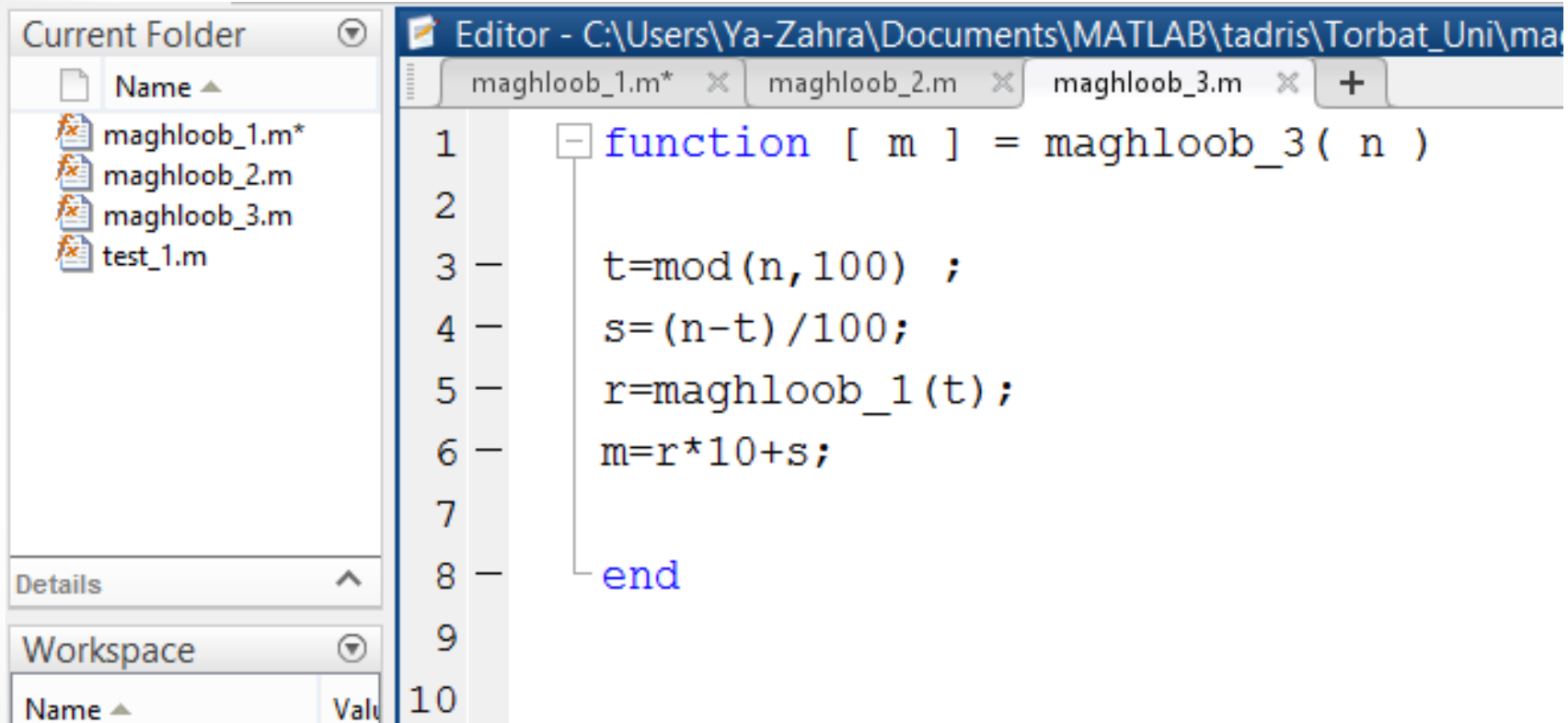
- در صورت تعریف یک تابع و ذخیره آن در محل current folder می توان از برنامه در سایر برنامه ها نیز استفاده کرد.

مثال :

- اکنون باتوجه به برنامه مقلوب اعداد دو رقمی، برنامه مقلوب اعداد سه رقمی را بنویسید.



کد برنامه :

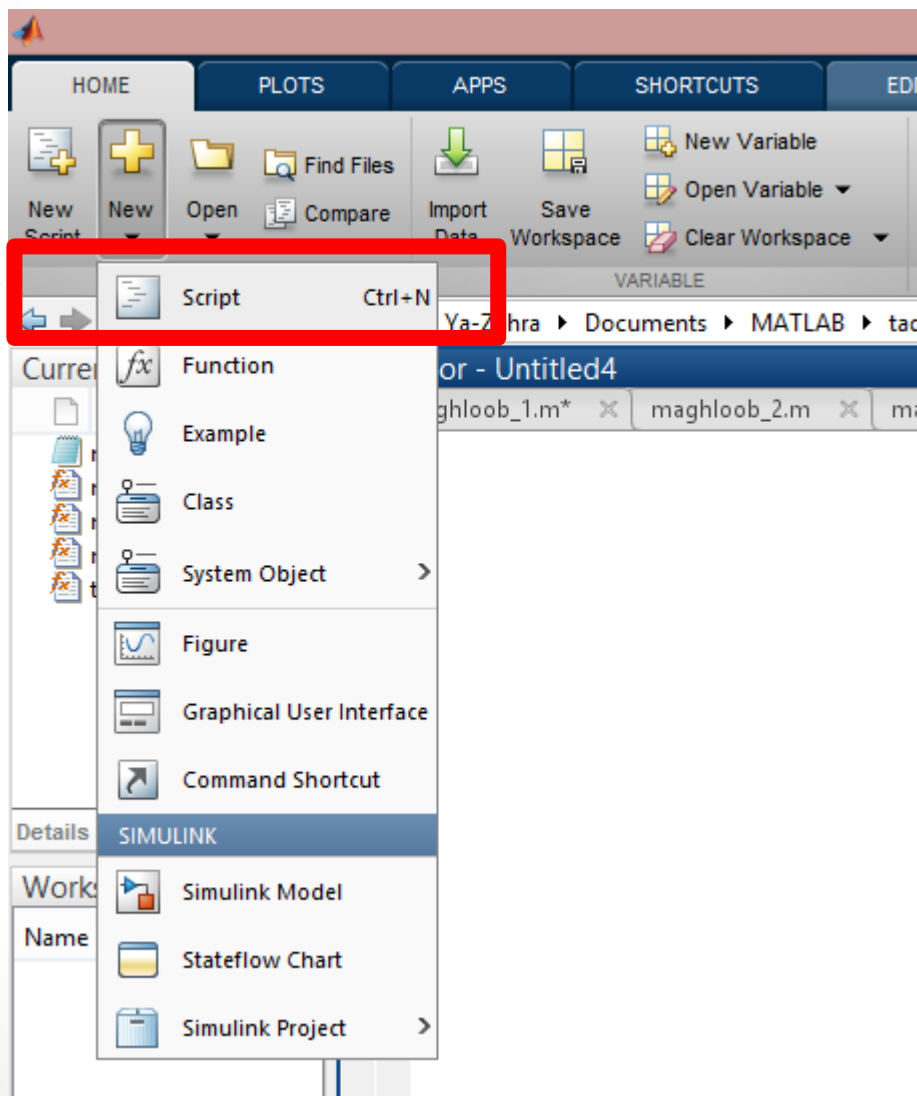


```
Editor - C:\Users\Ya-Zahra\Documents\MATLAB\tadris\Torbat_Uni\ma
maghloob_1.m* x maghloob_2.m x maghloob_3.m x +
1 function [ m ] = maghloob_3( n )
2
3 t=mod(n,100) ;
4 s=(n-t)/100;
5 r=maghloob_1(t);
6 m=r*10+s;
7
8 end
9
10
```



محیط کد نویسی (script):

- به طور کلی محیط اسکریپت را مجموعه ای توابع مختلف دانست.
- در صورتی که خواسته باشیم مجموعه ای عملیات مختلف انجام گیرد می توان از این محیط استفاده نمود.



Script یا Function :

- می توان یک پروژه را به یک درخت تشبیه کرد. تنه اصلی این پروژه همان قسمت اسکریپت و تمام شاخه و ساقه های دیگر این پروژه همان function خواهند بود.
- به عبارتی دیگر هر پروژه یم قسمت اصلی (script) و چندین بخش فرعی (function) را دارا خواهد بود.
- محیط های script و function تفاوت های در زمینه سرعت عملکرد ، حافظه مورد استفاده را دارند.



مثال :

- می خواهیم برنامه ای بنویسیم که میانگین سه عدد را محاسبه نماید. این برنامه را هم در محیط function و هم در محیط script می نویسیم.

```
mean_1.m  x  +
1  function [ M ] = mean_1( a,b,c )
2
3  -      M=(a+b+c)/3;
4
5  -      end
6
7
```

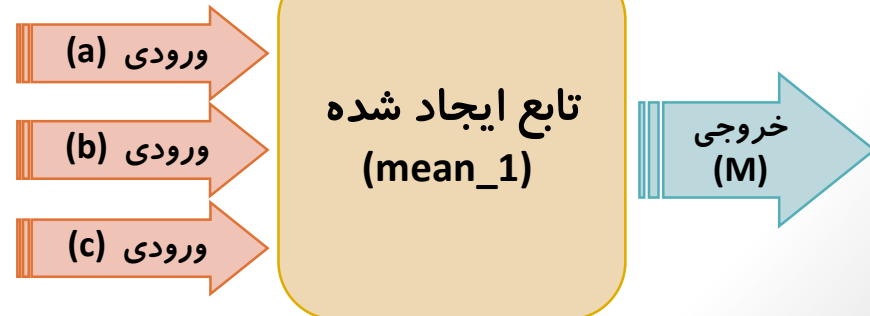
function

Command Window

```
>> M=mean_1(10,14,30)
```

```
M =
```

```
18
```



مثال :

- در صورتی که بخواهیم همین برنامه را در محیط اسکریپت بنویسیم، می توان به صورت مقابل عمل کرد.

```
mean2.m  x  +
1 -      clc
2
3 -      a=input('a == ');
4 -      b=input('b == ');
5 -      c=input('c == ');
6 -      M=(a+b+c)/3
7
```

Command Window

a == 10

b == 14

c == 30

M =

18

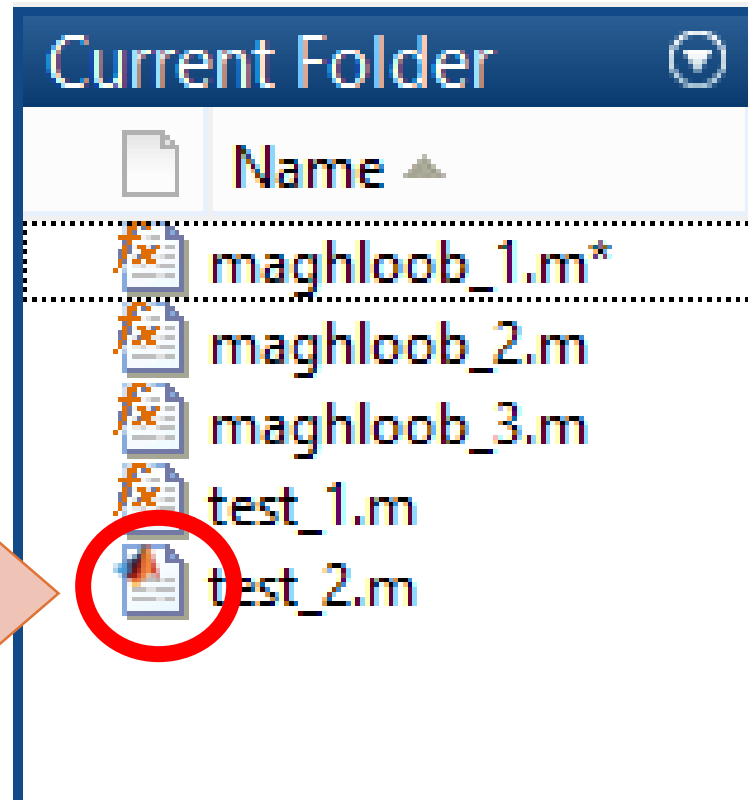
>>

script



مثال :

- برنامه ای در محیط اسکریپت بنویسید که ابتدا تمام متغیرها را پاک کرده و سپس صفحه را پاک نماید. سپس یک عدد سه رقمی را از کاربر گرفته و بعد مقلوب آن را با استفاده از برنامه های قبل محاسبه نماید.



به تفاوت آیکون ها دقت کنید.



کد برنامه :

```
Editor - C:\Users\Ya-Zahra\Documents\MATLAB\tadris\Torbat_Uni\test_2.m
maghloob_1.m* x maghloob_2.m x maghloob_3.m x test_2.m x +
1 - clear all
2 - clc
3
4 - n=input('yek adad 3 raghami ra vared konid == ');
5 - a1= maghloob_2 (n)
6 - a2= maghloob_3 (n)
7
8
```

```

yek adad 3 raghami ra vared konid == 123
a1 =
    321
a2 =
    321
>>
```



مثال :

Command Window

```
your number ?= -pi/2
```

```
a1 =
```

```
0.0000 + 1.2533i
```

```
a2 =
```

```
1.5708
```

```
a3 =
```

```
2.4674
```

```
a4 =
```

```
1.5708
```

```
a5 =
```

```
-1
```

```
fx >> |
```

• برنامه ای بنویسید که یک عدد را از کاربر گرفته و سپس مراحل زیر را به ترتیب انجام دهد.

۱. رادیکال آن عدد را محاسبه نماید.
۲. یک قدر مطلق آن عدد را محاسبه نماید.
۳. مربع آن عدد را محاسبه نماید.
۴. باقی مانده تقسیم آن عدد را بر عدد π محاسبه نماید.
۵. سینوس آن عدد را محاسبه نماید.



کد برنامه :

```
Editor - C:\Users\Ya-Zahra\Documents\MATLAB\tadris\
test_3.m × +
1 — clear all
2 — clc
3
4 — n=input('your number ?= ');
5 — a1=sqrt(n)
6 — a2=abs(n)
7 — a3=n^2
8 — a4=mod(n,pi)
9 — a5=sin(n)
```



مثال:

- خروجی برنامه ی زیر چیست ؟

```
text_1.m* x +
1 -   clc
2 -   clear all
3 -   close all
4
5 -   text1='ABCDEFGHIJKLM';
6 -   text2='NOPQRSTUVWXYZ';
7 -   text3(1)=text2(end-1);
8 -   text3(2)=text1(1);
9 -   text3(3)='-';
10 -  text3(4)=text3(2);
11 -  text3(5)=text1(sqrt(2*8)*3);
12 -  text3(6)=text1(end-sqrt(5*2-1)-1);
13 -  disp(text3)
14
```

نمونه سوال
امتحان حانی



مثال :

```
text_1.m*  ✕  +
1 -      clc
2 -      clear all
3 -      close all
4
5 -      text1='ABCDEFGHJKLM';
6 -      text2='NOPQRSTUVWXYZ';
7 -      text3=text2(5:10);
8 -      text4=text1(1:3:12);
9 -      text5='-';
10 -     text6=[text3,text4(3:end)];
11 -     text7=[text6(1:2:end),text5,text4(end:-2:1)];
12 -     disp(text7)
13
```



مثال :

```
text_1.m* x +
1 -   clc
2 -   clear all
3 -   close all
4
5 -   text1='ABCDEFGHIJKLM';
6 -   text2='NOPQRSTUVWXYZ';
7 -   text3=text2(5:10);
8 -   text4=text1(1:3:12);
9 -   text5='-';
10 -  text6=[text3,text4(3:end)];
11 -  text7=[text6(1:2:end),text5,text4(end:-2:1)];
12 -  disp(text7)
13
```

Command Window

RTVG-JD



منابع مفید :

- ۱- تمرینتمرینتمرین
- ۲- جزوه آموزشی متلب دانشگاه صنعت آب و برق ، سرافراز
- ۳- کاملترین مرجع آموزشی و کاربردی matlab ، مهندس علمداری ، انتشارات نگارنده دانش
- ۴- سایر منابع موجود در اینترنت
- ۵- ...

