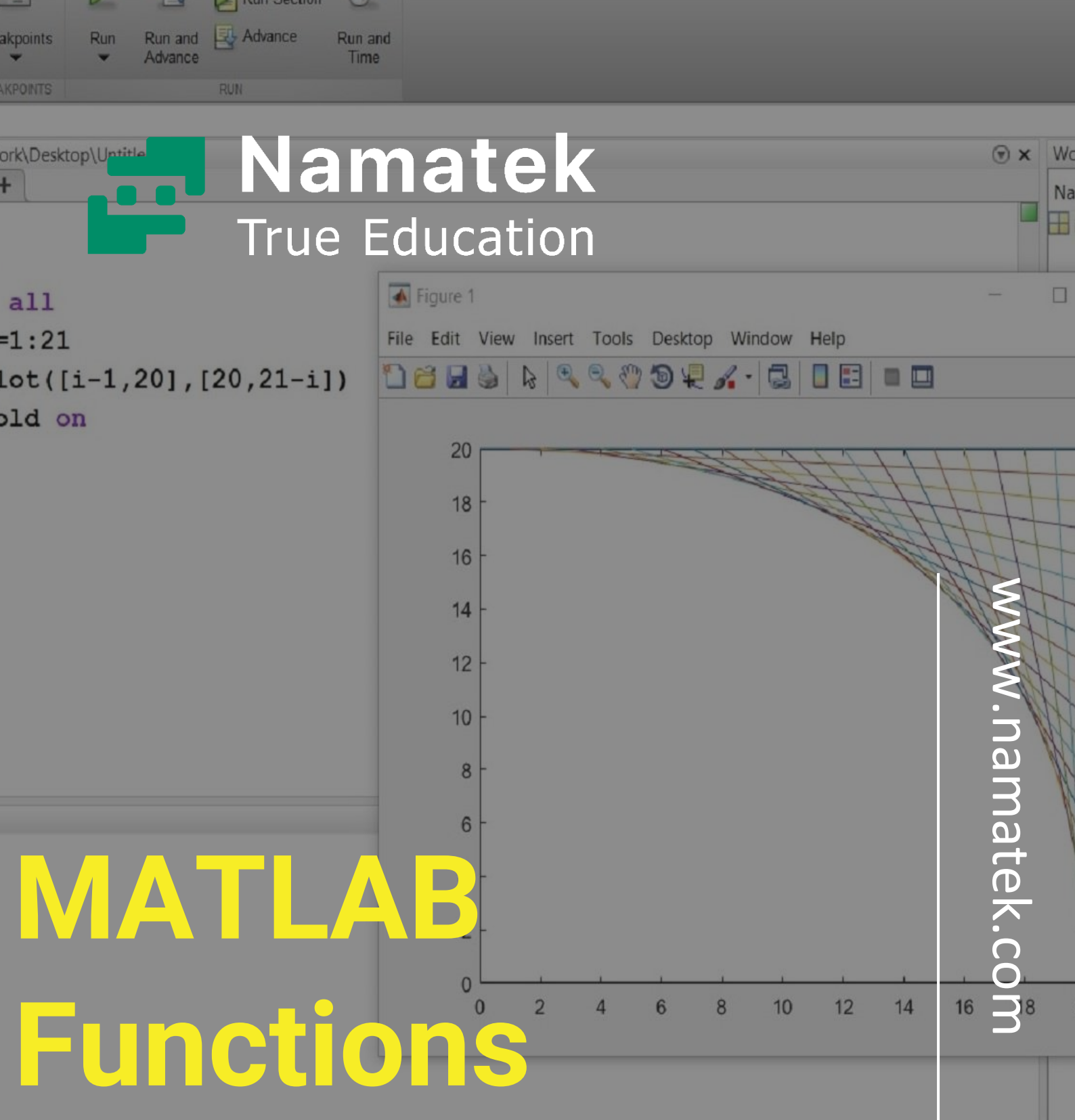


MATLAB Functions

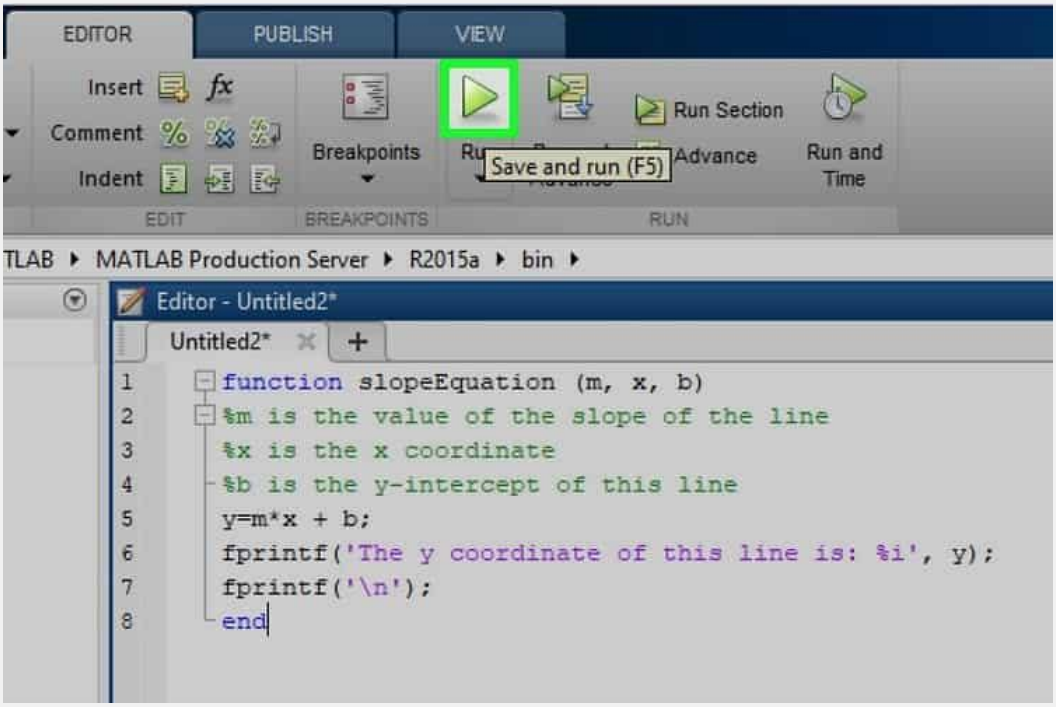
www.namatek.com

توابع متلب

فهرست مطالب

متلب یک برنامه نرم افزاری اختصاصی است که توسط ماتکریکس لابیراتوری (Matrix Laboratory) توسعه یافته است. این نرم افزار برای وظایف پیچیده ای مانند تجزیه و تحلیل داده ها، تنظیم داده ها و اجرای الگوریتم ها استفاده می شود که برای سهولت استفاده از آن می توان از توابع متلب استفاده کرد. در این مقاله قصد داریم به معرفی توابع متلب، کاربردها، انواع و مزایای استفاده از آن ها بپردازیم، پس با ما همراه باشید.

تابع در متلب چیست؟



یک تابع مجموعه ای از دستورالعمل ها است که به منظور انجام یک کار خاص استفاده می شود. توابع یکی از ابزارهای کار متلب هستند. توابع متلب به کاربر کمک می کنند کدی ایجاد کند که بتواند به صورت مکرر از آن استفاده کند که این یکی از قدرتمندترین ویژگی های متلب است.

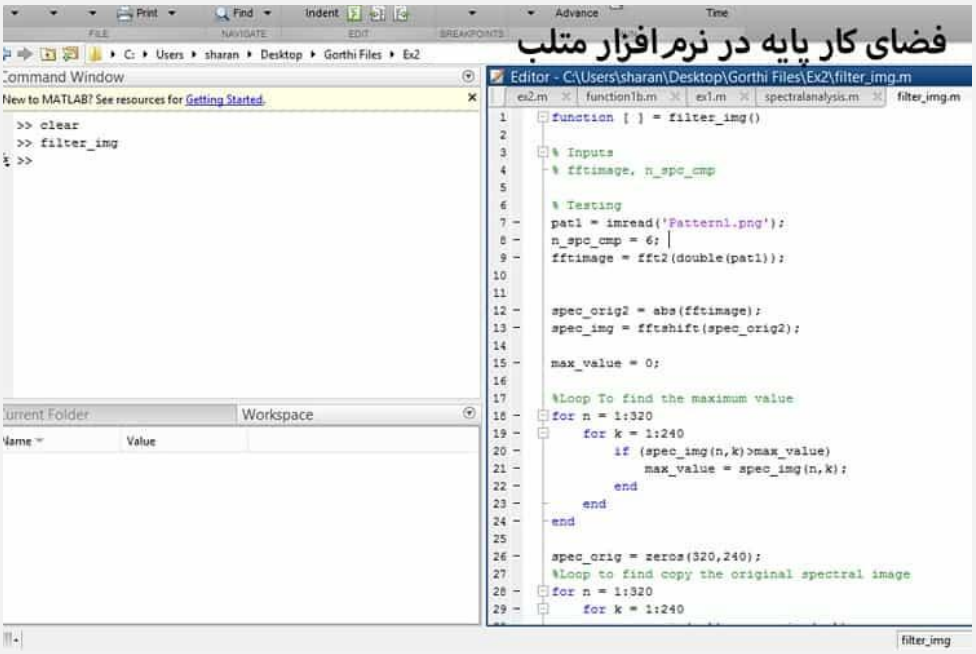
در بسیاری از زبان ها، می توان دستورات را در فایل هایی به نام اسکریپت (Script) ذخیره کرد؛ سپس می توان با تایپ نام این اسکریپت ها در یک

پنجره ترمینال، آنها را اجرا کرد. در متلب نیز می توان دستورات را در فایل هایی با نام توابع ذخیره کرد. توابع انعطاف پذیری بیشتری نسبت به اسکریپت ها دارند و به کاربر این امکان را می دهند که مقادیر ورودی را ارسال کنند و مقادیر خروجی را بازگرداند.

توابع متلب در فایل های اسکریپت جداگانه مشخص می شوند و شامل تعاریف و دستورات تابع هستند. نام تابع و نام فایل باید یکسان باشد و همیشه در انتهای فایل تعریف می شود. به توابع، متد نیز گفته می شود.

متد یک بلوک از کد است که هنگام فراخوانی توسط کاربر اجرا می شود و شامل فضای کار پایه ای است که مربوط به بخش دستورات است. توابع می توانند بیش از یک آرگومان ورودی را بپذیرند و حتی ممکن است بیش از یک آرگومان خروجی را بازگردانند.

توابع متلب در کدام بخش از نرم افزار قرار دارند؟



توابع متلب در فایل های جداگانه ای تعریف می شوند و باید هم نام فایل باشند. این توابع روی متغیرهایی در فضای کاری خود عمل می کنند که

فضای کاری محلی نامیده می شوند. فضای کاری محلی با فضای کاری که در خط فرمان متلب به آن دسترسی دارید، متفاوت است و فضای کار پایه نامیده می شود.

کاربرد توابع متلب

کاربرد توابع متلب عبارت اند از:

- خواندن کد را آسان تر می کند.
- از ورود اشکلات به برنامه جلوگیری می کند.

قوانین توابع معتبر در متلب

قوانینی وجود دارد که برای نام های توابع معتبر و ذخیره تابع باید رعایت شوند:

- نام توابعی که با حروف الفبا آغاز می شوند و می توانند شامل عدد یا زیر خط باشند، معتبر در نظر گرفته می شود.
- تابع را می توان در یک فایل تابعی ذخیره کرد که شامل تعاریف تابع است و نام فایل باید با اولین نام تابع موجود در فایل مطابقت داشته باشد.
- می توان تابعی را که شامل تعاریف و دستورات تابع است ذخیره کنیم. توابع باید در انتهای فایل وجود داشته باشند و نام فایل اسکریپت نمی تواند نامی مشابه با تابع موجود در فایل داشته باشد.

برای نشان دادن پایان کار تابع باید از کلمه کلیدی پایان (end) استفاده شود.

اجزای توابع متلب

رفتار توابع با مؤلفه هایی که در زیر توضیح داده شده، تعریف می شوند:

1) کد متلب

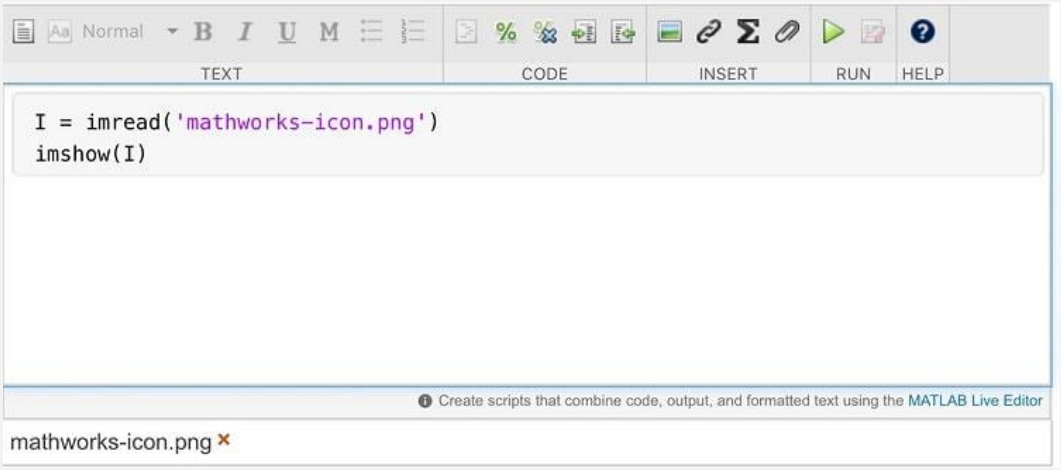
```
#include "myMult.h"

void myMult(double a[3][4], double b[4][5], double c[3][5])
{
  int i0;
  int i1;
  int i2;
  for (i0 = 0; i0 < 5; i0++) {
    for (i1 = 0; i1 < 3; i1++) {
      c[i1][i0] = 0.0;
      for (i2 = 0; i2 < 4; i2++) {
        c[i1][i0] += b[i2][i0] * a[i1][i2];
      }
    }
  }
}
```

کدهای متلب، کدی را تعریف می کنند که هنگام ارزیابی تابع invoke اجرا می شوند. کد را می توان به عنوان یک دستور ابزاری که روی دستگاه نوشته می شود یا به عنوان کد نرم افزار متلب تعریف کرد. اگر کد به عنوان یک دستور ابزاری تعریف شود، تعریف ابزار، گرفتن آرگومان های ورودی خواهد بود. تمامی نمادها در دستور ابزاری با مقدار ورودی تابع invoke جایگزین خواهند شد. برای مثال اگر تابعی با آرگومان ورودی start تعریف شده باشد، فرمان ابزار به صورت Data: Start تعریف می شود و مقدار شروع با ۱۰ به تابع invoke ارسال می شود، آنگاه دستور Data: Start 10 روی ابزار نوشته خواهد شد. اگر کد به عنوان یک دستور ابزار تعریف شده باشد، دستور ابزار نیز می تواند برای بازگشت یک آرگومان خروجی تعریف شود. آرگومان خروجی را می توان به صورت داده های عددی یا متنی برگرداند.

در صورتی که کدی به عنوان کد نرم افزار متلب تعریف شده باشد، می توان تعیین کرد که کدام دستورات به دستگاه ارسال شوند و داده هایی که از دستگاه به دست می آیند را در صورت لزوم دستکاری، تنظیم یا تجزیه و تحلیل کرد.

2) متن راهنما



متن راهنما اطلاعاتی در مورد عملکرد ارائه می دهد.

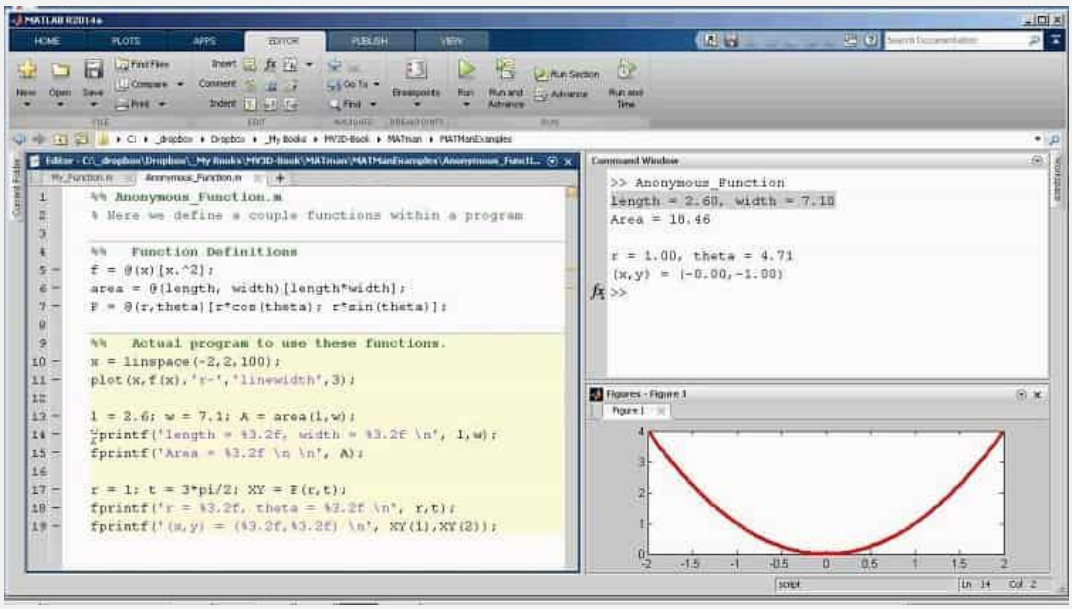
انواع توابع متلب

توابع متلب دارای انواع مختلفی هستند که از پرکاربردترین آن ها می توان به موارد زیر اشاره کرد:

- ناشناس
- فرعی
- تو در تو
- خصوصی

که در ادامه به بررسی هر یک از این توابع می پردازیم.

توابع ناشناس (Anonymous Functions)



یکی از توابع متلب، توابع ناشناس هستند که یک تابع درون خطی یک متغیر خروجی است. این نوع توابع می توانند حاوی چندین آرگومان ورودی و خروجی باشند. کاربر نمی تواند به توابع ناشناس از خارج فایل دسترسی پیدا کند. کاربران می توانند توابع ناشناس را در خط فرمان یا درون اسکریپت یا فایل تابع تعریف کنند. به این ترتیب می توان توابع ساده را بدون نیاز به ایجاد فایل برای آن ها به وجود آورد. این توابع در فایل برنامه ذخیره نمی شوند؛ اما با متغیری که نوع داده آن function – handle است، در ارتباط خواهند بود.

توابع ناشناس را می توان به صورت زیر نوشت:

```
Fun = @(argumentlist)
```

1) مثال

```
mul = @(x,y) x*y
res1 = mul (2,3)
```

res2 = mul (4,3)

res3 = mul (4,5)

وقتی خروجی را اجرا کنیم، نتیجه زیر به دست خواهد آمد:

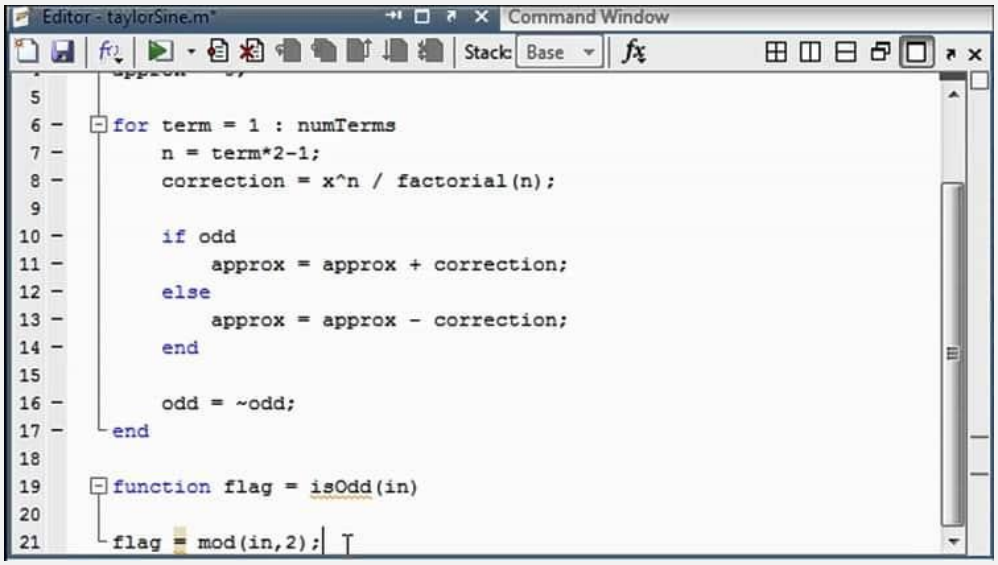
res1 = 6

res2 = 12

res3 = 20

توابع ناشناس را می توان بدون ورودی یا حتی چند ورودی و خروجی نوشت. اگر تابع ورودی نداشت، می توان از یک پرانتز خالی برای فراخوانی این تابع استفاده کرد.

توابع اولیه و فرعی (Primary and Secondary Functions)



هر تابعی غیر از توابع ناشناس باید در یک فایل تعریف شود. هر فایل تابع شامل یک تابع اولیه مورد نیاز است که ابتدا ظاهر می شود و هر تعداد تابع فرعی اختیاری، بعد از تابع اصلی می آید و توسط آن استفاده می شود.

توابع فرعی دارای یک تابع اصلی هستند که در خط اول کد ظاهر می شوند. توابع اولیه را می توان خارج از فایلی که آن ها را تعریف می کند، از خط فرمان یا از توابع دیگر فراخوانی کرد؛ اما توابع فرعی را نمی توان از خط فرمان یا توابع دیگری که خارج از فایل توابع است فراخوانی کرد. این توابع تنها برای تابع اصلی و سایر توابع فرعی که در فایل تابعی (که آن ها را تعریف می کند)، قابل مشاهده هستند.

1) مثال

می خواهیم تابعی با نام quadratic بنویسیم که ریشه های یک معادله درجه دوم را محاسبه کند. این تابع ۳ ورودی دارد که عبارت اند از:

- ضریب درجه دوم
- ضریب خطی
- جمله ثابت

فایل تابع quadratic .m شامل تابع اصلی درجه دوم و دیسک تابع فرعی است که تفکیک کننده را محاسبه می کند. یک فایل تابع quadratic .m ایجاد کرده و کد زیر را در آن تایپ کنید:

```
function [x1.x2] = quadratic (a, b, c)
this function returns the roots of %
a quadratic equation %
It takes 3 input arguments %
which are the co – efficient of x2, x and the %
constant term %
It returns the roots %
```

d = disc (a, b, c)

x1 = (-b + d)/ (2*a)

x2 = (-b -d)/ (2*a)

end % end of quadratic

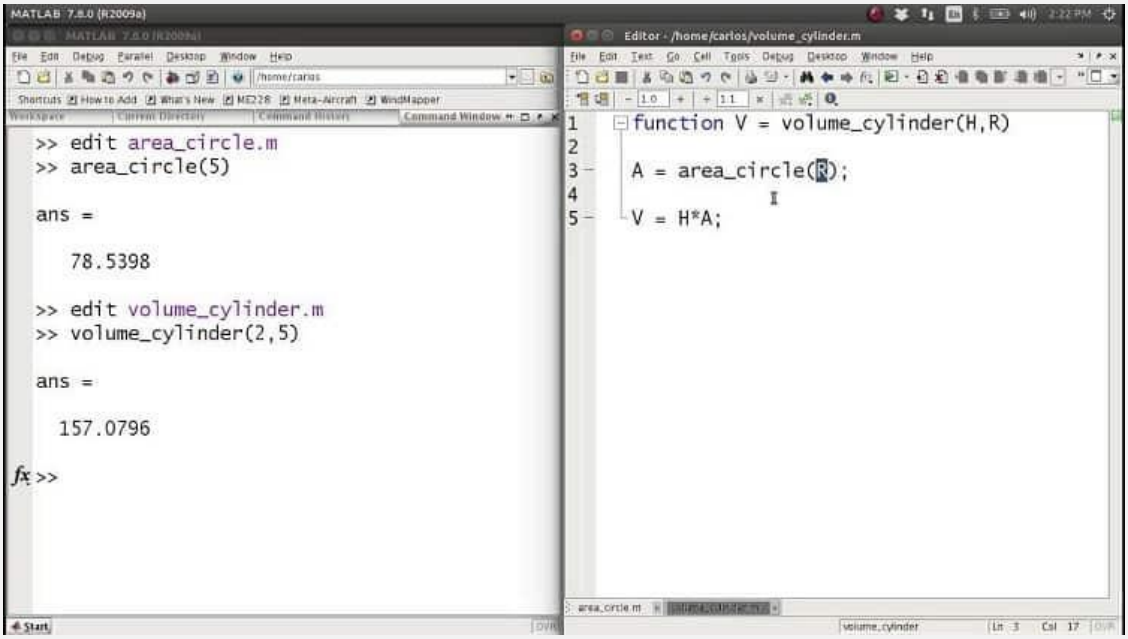
می توانید تابع فوق را از خط فرمان با استفاده از علامت – فراخوانی کنید.

Quadratic (2, 4, -4)

در این حالت متلب دستور بالا را اجرا و نتیجه زیر را ارائه می دهد:

ans = 0.7321

توابع تو در تو(Nested Functions)



توابعی که در یک تابع یا تابع والد دیگری تعریف می شوند را توابع تو در تو می نامند. آن ها می توانند از متغیرهایی که در تابع والد تعریف شده اند استفاده کنند یا آن ها را تغییر دهند. این نوع توابع در محدوده تابع والد تعریف می شوند و می توانند به فضای کاری ای که در آن تعریف شده اند،

دسترسی داشته باشند. الزامات خاصی وجود دارد که توابع تو در تو باید از آن ها پیروی کنند، که عبارت اند از:

- همه توابع به دستور پایان نیاز ندارند. با این حال، به منظور تو در تو کردن هر تابع، دستور پایان برای هر تابعی باید نوشته شود.
- نمی توان توابع تو در تو را داخل هر دستور کنترلی مانندif -else ، case switchو غیره تعریف کرد.
- این نوع توابع را می توان مستقیماً با نام یا با استفاده از هر دسته تابعی فراخوانی کرد.

```
function current
nestfun1
nestfun2
function nestfun1
x = 7
end
function nestfun2
x =4
end
```

یک تابع تو در تو از دستور زیر پیروی می کند:

```
function x = A (p1, p2)
...
B (p2)
function y = B (p3)
...
```

```
end
...
end
```

1) مثال

مثال را با تابع دیسک مثال توابع فرعی ادامه می دهیم؛ اما این بار تابع دیسک یک تابع تو در تو خواهد بود. یک فایل تابع m . quadratic2 ایجاد کنید و کد زیر را در آن تایپ کنید:

```
function [x1, x2] = quadratic2 (a, b, c)
this function disc % nested function
d = sqrt (b^2 – 4*a*c)
end % end of function disc
disc
x1 = (-b + d) / (2*a)
x2 = (-b – d) / (2*a)
end % end of function quadratic2
```

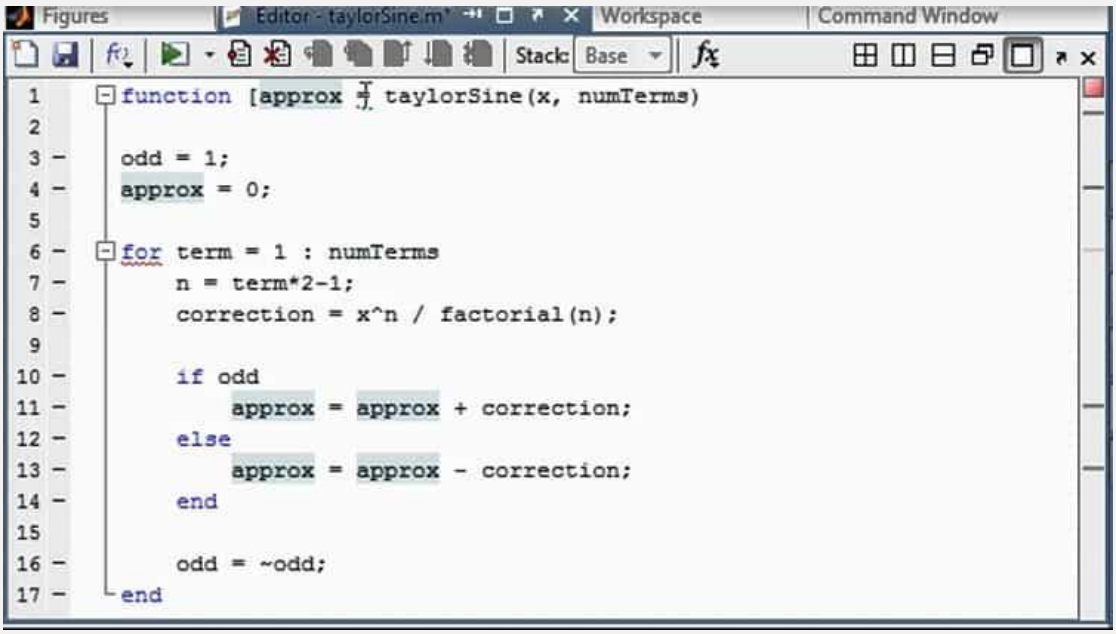
می توان تابع فوق را از خط فرمان با استفاده از علامت – فراخوانی کرد:

```
Quadratic2 (2, 4, -4)
```

متلب، دستور بالا را اجرا می کند و نتیجه زیر را نشان خواهد داد:

```
ans = 0.7321
```

توابع خصوصی (Private Functions)



تابع خصوصی یک تابع اصلی است که تنها برای گروه معدودی از توابع دیگر قابل مشاهده است. این نوع توابع در توابع فرعی قرار دارند و با کلمه کلیدی private مشخص می شوند. توابع خصوصی تنها برای توابعی قابل مشاهده هستند که در پوشه والد باشند یا توابع موجود در پوشه ای که در بخش بالایی زیر پوشه خصوصی قرار دارند. این توابع، زمانی مفید هستند که بخواهیم دامنه عملکرد را محدود کنیم.

توابع خصوصی را نمی توان از خط فرمان یا از توابع خارج از پوشه فراخوانی کرد.

1) مثال

در این مثال می خواهیم تابع درجه دوم را بازنویسی کنیم؛ اما این بار، تابع دیسک محاسبه کننده، تفکیک کننده یک تابع خصوصی خواهد بود. برای این کار یک زیر پوشه با نام private در پوشه کاری ایجاد کنید. فایل تابع discr .m را در آن ذخیره کنید.

```
function dis = dis (a, b, c)
function calculates the discriminant %
dis = sqrt (b^2 – 4* a* c)
end % end of sub – function
```

یک تابع quadratic3 .m در پوشه کاری خود ایجاد کنید و کد زیر را در آن تایپ کنید:

```
function [x1.x2] = quadratic3 (a, b, c)
function returns the roots of %
a quadratic equation %
It takes 3 input arguments %
which are the co – efficient of x2, x and the %
constant term %
It returns the roots %
d = disc (a, b, c)
x1 = (-b + d)/ (2*a)
x2 = (-b - d)/ (2*a)
end % end of quadratic3
```

می توان تابع فوق را از خط فرمان با علامت – فراخوانی کرد:

```
quadratic3 (2, 4, -4)
```

متلب دستور بالا را اجرا کرده و نتیجه زیر را ارئه خواهد داد:

```
ans = 0.7321
```

متغیرهای جهانی(Global Variables)

متغیرهای جهانی را می توان توسط بیش از یک تابع به اشتراک گذاشت. به همین منظور متغیر را باید در تمامی توابع با عنوان global تعریف کنید. اگر کاربر بخواهد از فضای کار پایه به آن متغیر دست پیدا کند باید آن را در خط فرمان اعلام کند. اعلان جهانی باید قبل استفاده از متغیر در یک تابع رخ دهد. استفاده از حروف بزرگ به منظور نامگذاری متغیرهای جهانی برای تمایز آن ها از سایر متغیرها روش مناسبی خواهد بود.

مزایای استفاده از توابع متلب

علاوه بر قابلیت استفاده مجدد، توابع متلب دارای چند مزیت مهم دیگر نیز هستند که عبارت اند از:

- توابع متلب می توانند بیش از یک آرگومان (Argument) ورودی را بپذیرند و ممکن است بیش از یک آرگومان خروجی را برگردانند.
- توابع متلب روی متغیرهایی در فضای کاری خود کار می کنند؛ جدا از فضای کاری پایه کاربر. (محلی که در خط فرمان استفاده می شود.)

- این توابع را می توان در فایل های جداگانه ای برای اشتراک گذاری آسان با دیگران ذخیره کرد.