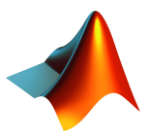


MATLAB 프로그래밍 및 실습

11강. 시각화 고급

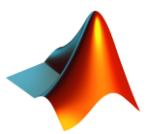
<https://bskyvision.com/845>

matplotlib 모듈 사용법 ≡ 매트랩의 그래픽 객체 사용법



오늘 수업을 듣고 나면 ()을 할 수 있다.

- 이미 그린 그래프의 속성 변경 (선 굵기, 마커 크기 등)
- figure 창의 모니터 상 위치 변경 (마우스 사용 X)
- figure 창의 배경색, 그래프 창의 색, 창 이름 변경
- plot으로 그리는 선의 기본 굵기 변경
- 그래프를 간단한 애니메이션으로 작성

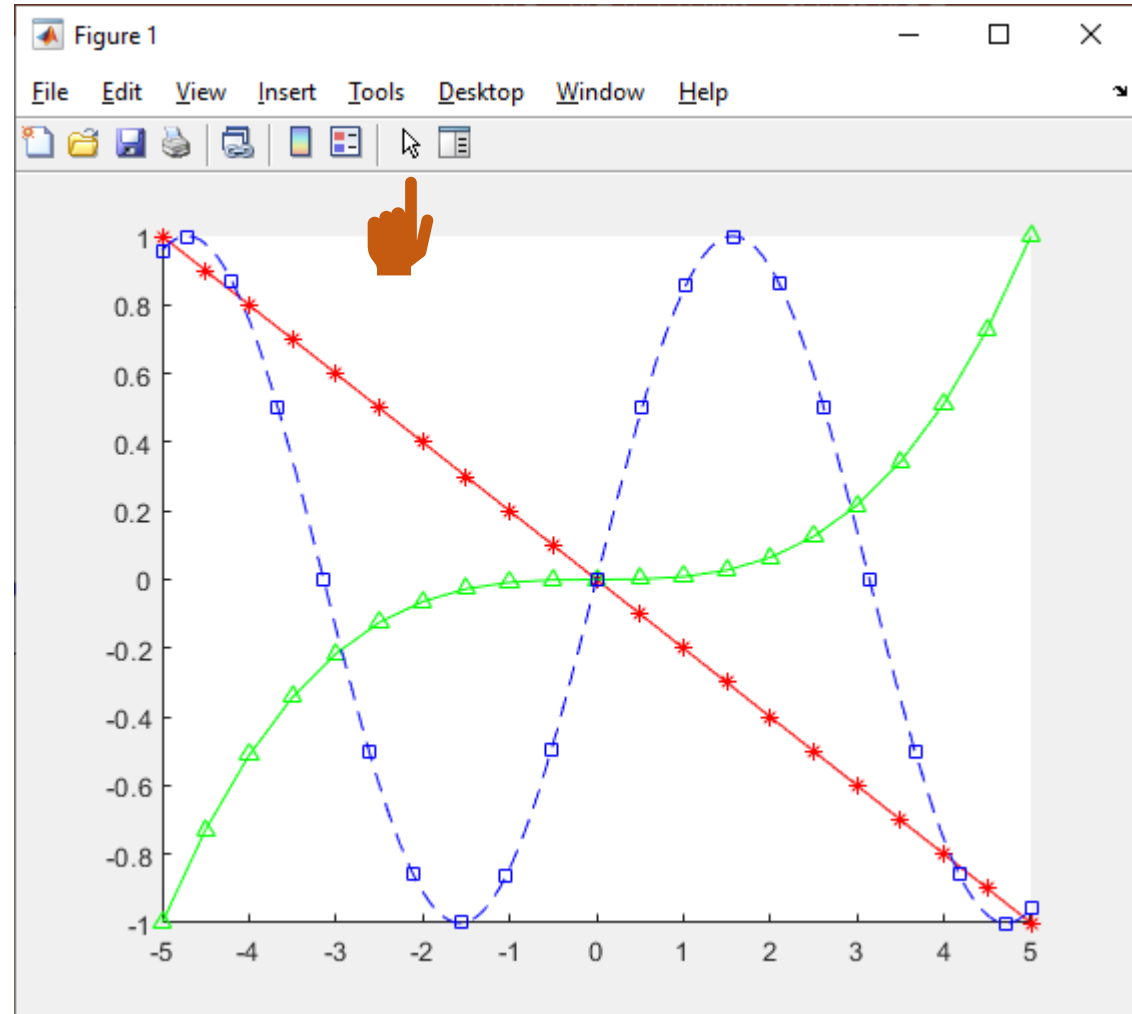


Property Inspector 띄우기

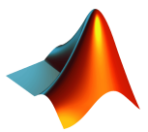
```
x = -5:0.5:5;  
y1 = -x/5;  
y2 = (x/5).^3;
```

```
figure, hold on,  
plot(x, y1, 'r-*')  
plot(x, y2, 'g-^')  
fplot(@sin, 'b--s')
```

- 화살표(Edit Plot)를 누르고
- 그래프 선 중 아무거나 더블클릭
- 속성 바꿔보기
- 다른 그래프 선도 바꿔보기
- 그래프 추가로 그려보고 속성 확인



※ plot, fplot에서 설정하는 property
= Property Inspector에 나오는 property

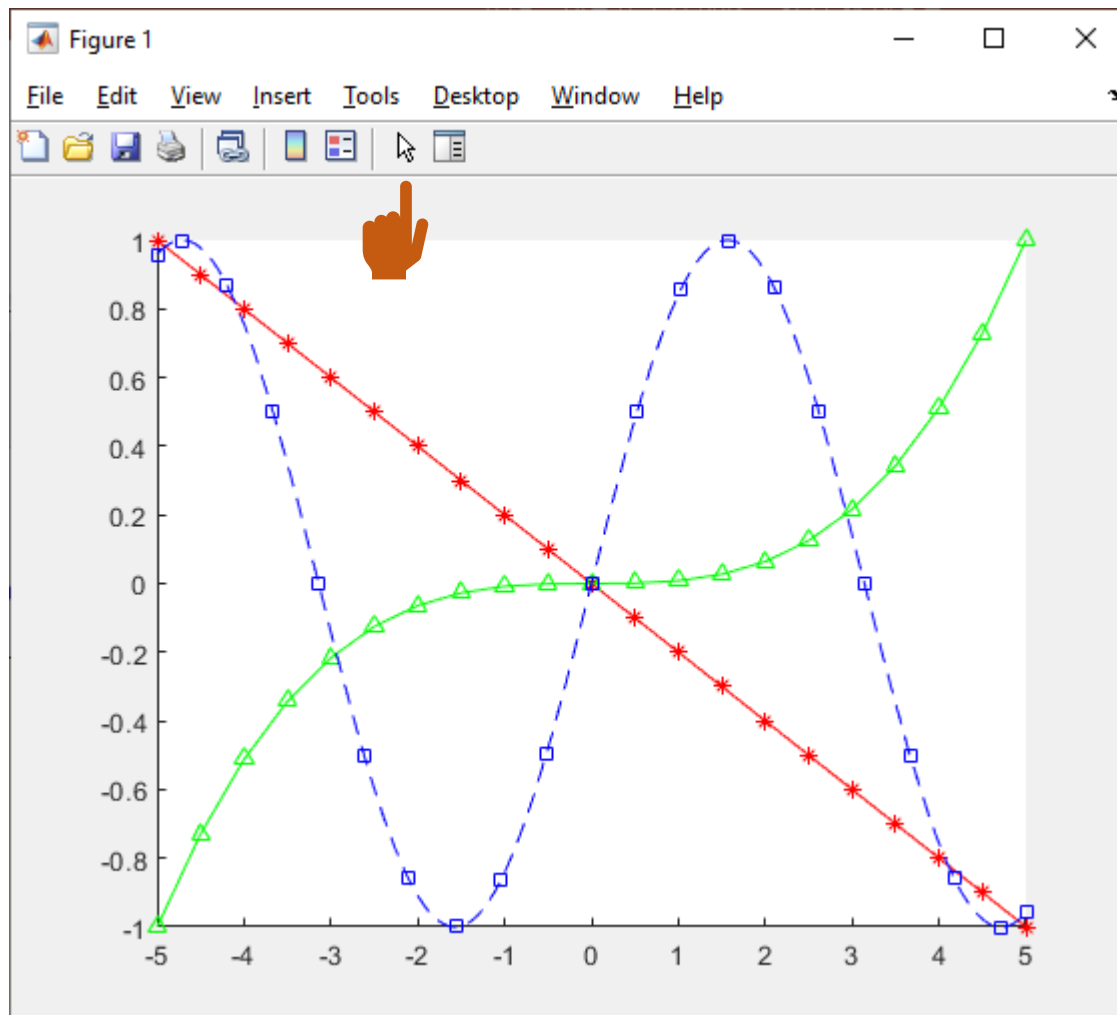


figure의 구성

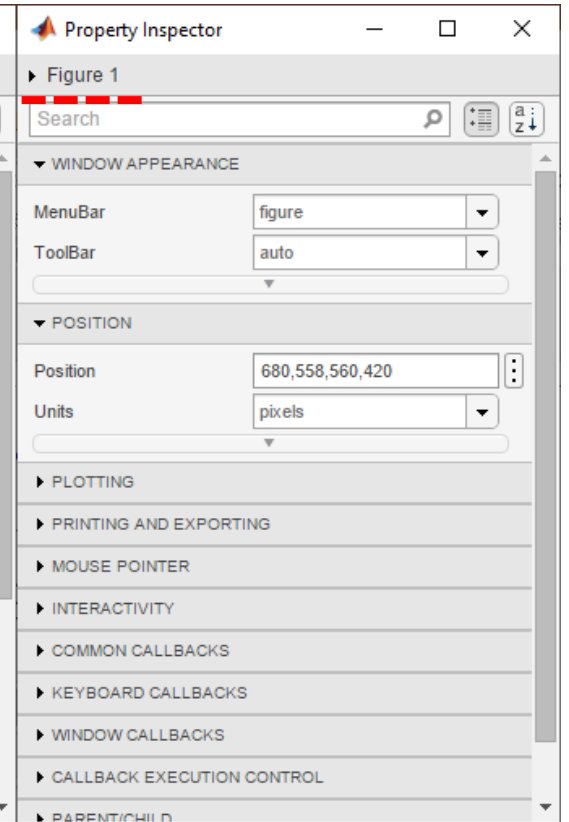
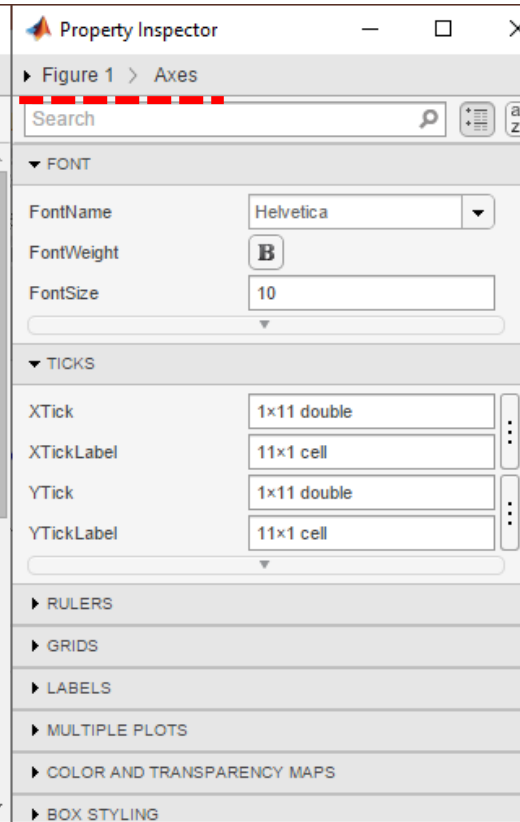
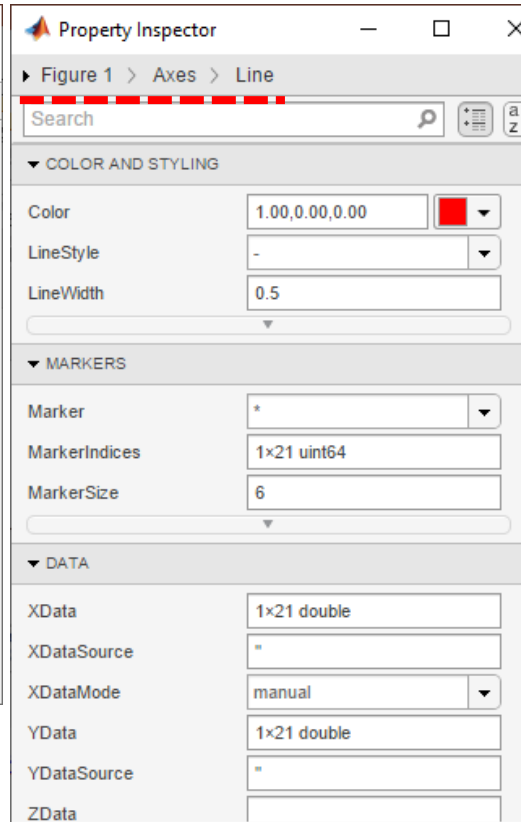
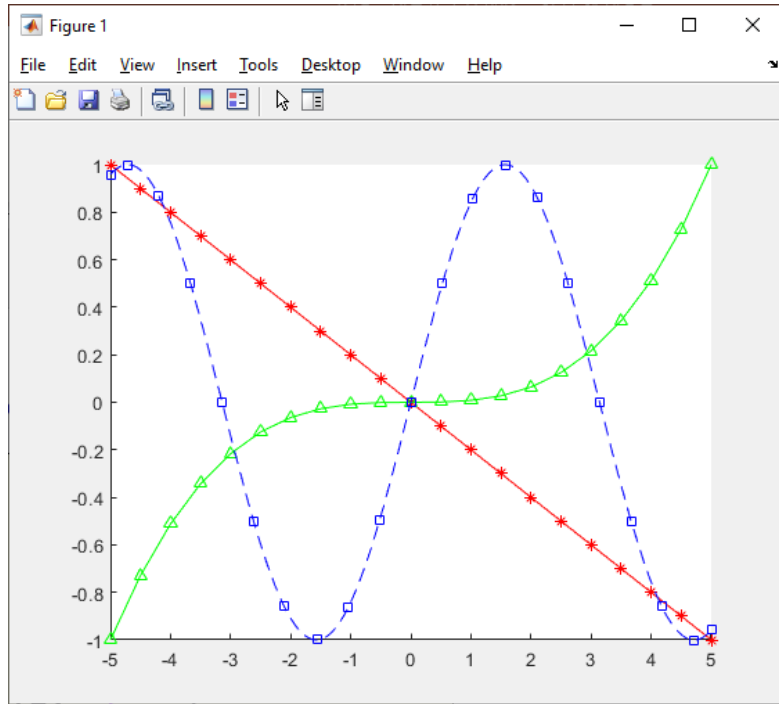
```
x = -5:0.5:5;  
y1 = -x/5;  
y2 = (x/5).^3;
```

```
figure, hold on,  
plot(x, y1, 'r-*')  
plot(x, y2, 'g-^')  
fplot(@sin, 'b--s')
```

- 화살표(Edit Plot)를 누르고
 - 1) 그래프 선 아무거나 클릭
 - 2) 그래프 없는 흰 영역을 클릭
 - 3) 그래프 바깥쪽을 클릭
- 차이는?

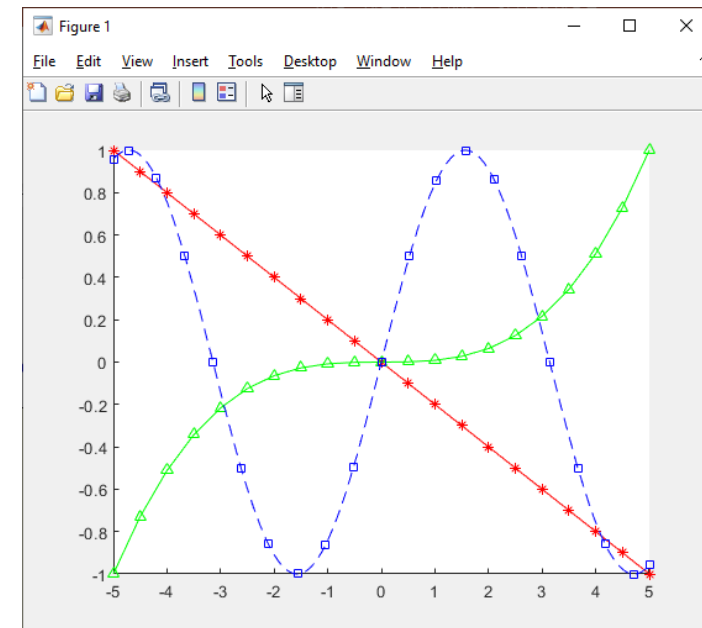
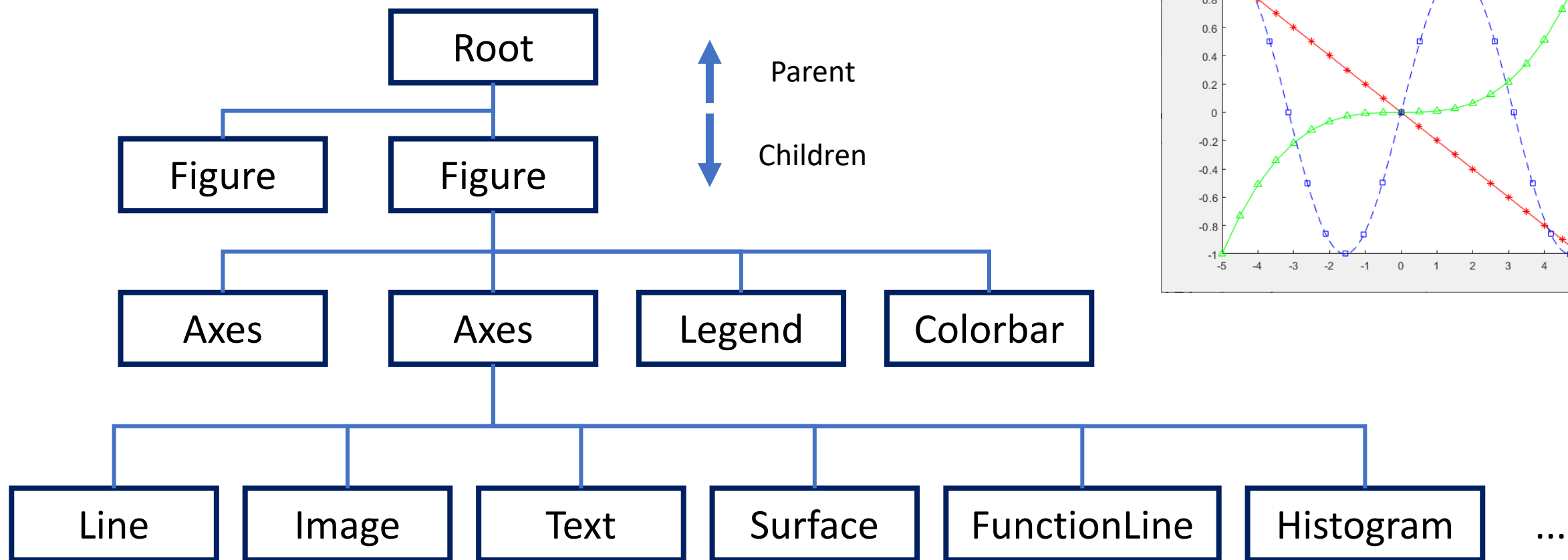


figure의 구성



- 1) 각각 설정할 수 있는 속성이 다르다.
- 2) Figure 1 밑에 Axes가 있고, Axes 밑에 Line이 있다...?
- 3) fplot으로 그린 건 Line이 아니다...?

figure의 구성 (계층구조)



- * 각각을 그래픽 객체(Graphics Object)라고 부른다.
 - Root 객체, Figure 객체, Axes 객체, Line 객체, ...
- * Figure 창 및 창에 포함된 모든 것은 그래픽 객체이다.

figure의 구성 (계층구조)

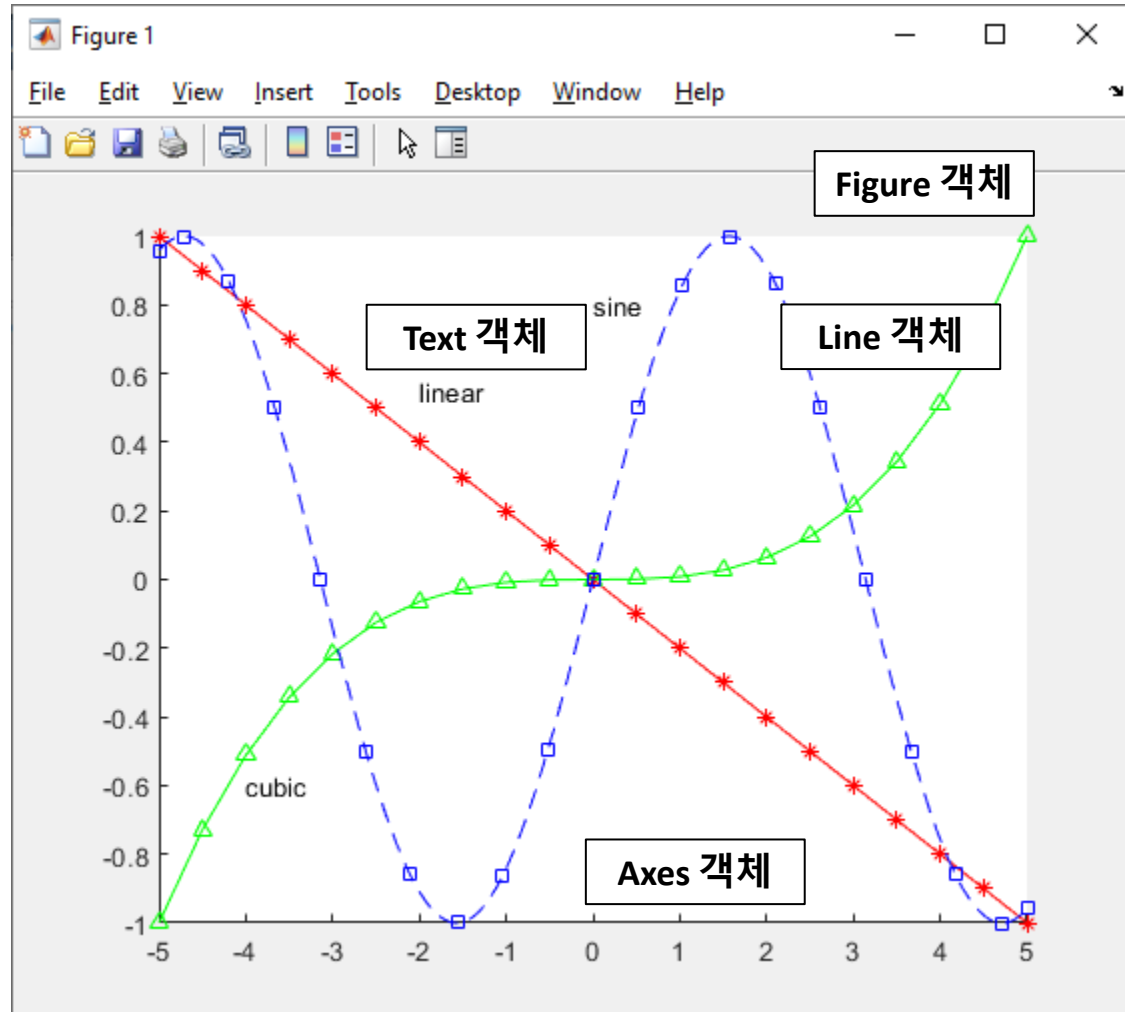


Figure 객체

Axes 객체

Line 객체

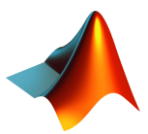
Line 객체

Line 객체

Text 객체

Text 객체

Text 객체



Line 객체란?

- 2차원 그래프의 선 속성을 설정하는 방법

```
x = -5:0.5:5;
```

```
y1 = x.^2;
```

```
y2 = x.^3;
```

```
figure, hold on,
```

```
plot(x, y1, 'r-*', 'linewidth', 2) % 그린 이후에 속성을 바꾸기 어려움
```

```
h = plot(x, y2, 'g--^'); % 이후에 속성을 바꾸기 쉬움 (세미콜론에 주의)
```

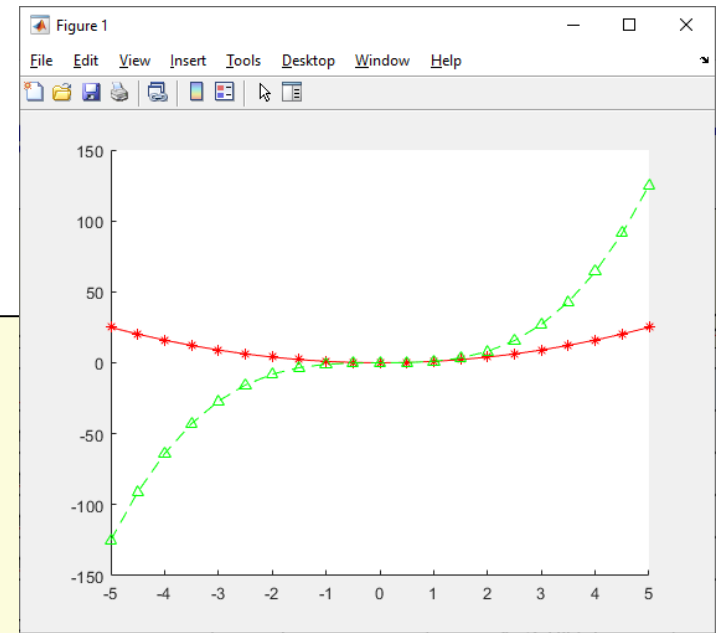
```
get(h, 'LineWidth') % plot으로 그린 Line의 선 두께를 확인
```

```
set(h, 'LineWidth', 2) % Line의 선 두께를 바꿈
```

```
h.Marker % plot으로 그린 Line의 마커 종류를 확인
```

```
h.Marker = 'd'; % 마커 종류 변경
```

```
get(h) % plot으로 그린 Line의 모든 속성을 확인
```



workspace에 h가 생김 (:: h = 변수)

plot(x,y) - Line 객체 생성

```
h = plot(x, y2, 'g--^');
```

변수 h에
대입해라.

이런 그래프(Line 객체)를 그려서
그 객체의 **핸들**을

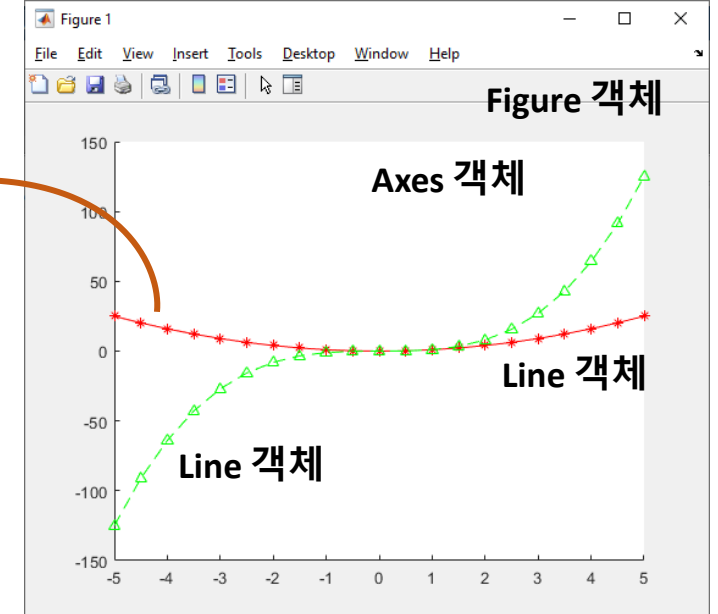
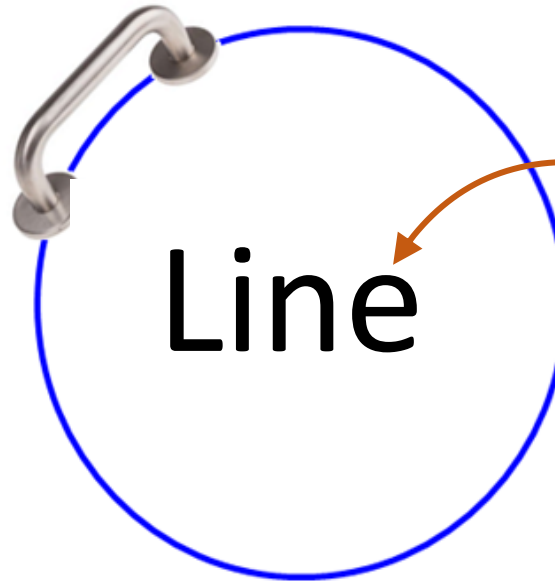
함수
핸들

```
function z = myadd(x, y)  
z = x + y;  
end
```

• 그래픽 핸들이란?

- 그래픽 객체에 **손잡이**를 단 것
- **변수**처럼 다룰 수 있음
- 함수에 인자로 전달 가능

```
get(h, 'LineWidth')  
set(h, 'Linewidth', 2)  
get(h)
```



그래픽 핸들 활용 - Line 객체 속성 읽기

- Line 객체의 속성 = plot으로 그린 Line의 속성

```
x = -5:0.5:5;  
y2 = x.^3;  
h = plot(x, y2, 'g--^');
```

% 방법1 - 핸들을 구조체처럼 다루기

h.Color % h.color -> error (case sensitive)

h.LineWidth

h.Marker

h.PropertyName

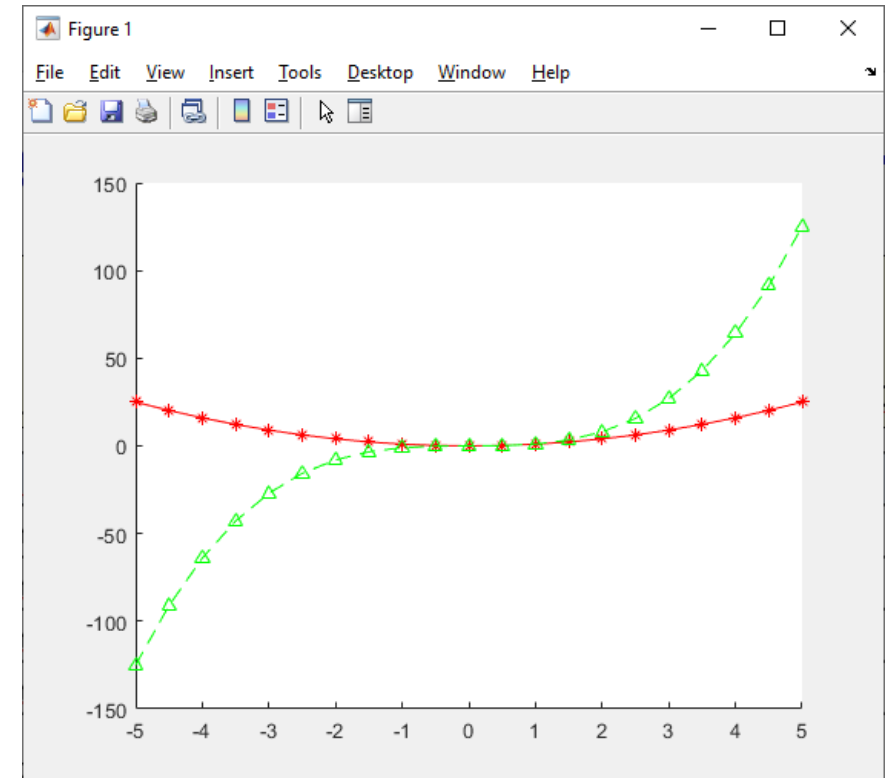
% 방법2 - get 사용

get(h, 'Color')

get(h, 'marker') % case insensitive

get(h) % h의 모든 속성

get(h, 'PropertyName')



그래픽 핸들 활용 - Line 객체 속성 변경

- Line 객체의 속성 = plot으로 그린 Line의 속성

```
x = -5:0.5:5;  
y2 = x.^3;  
h = plot(x, y2, 'g--^');
```

% 방법1 - 핸들을 구조체처럼 다루기

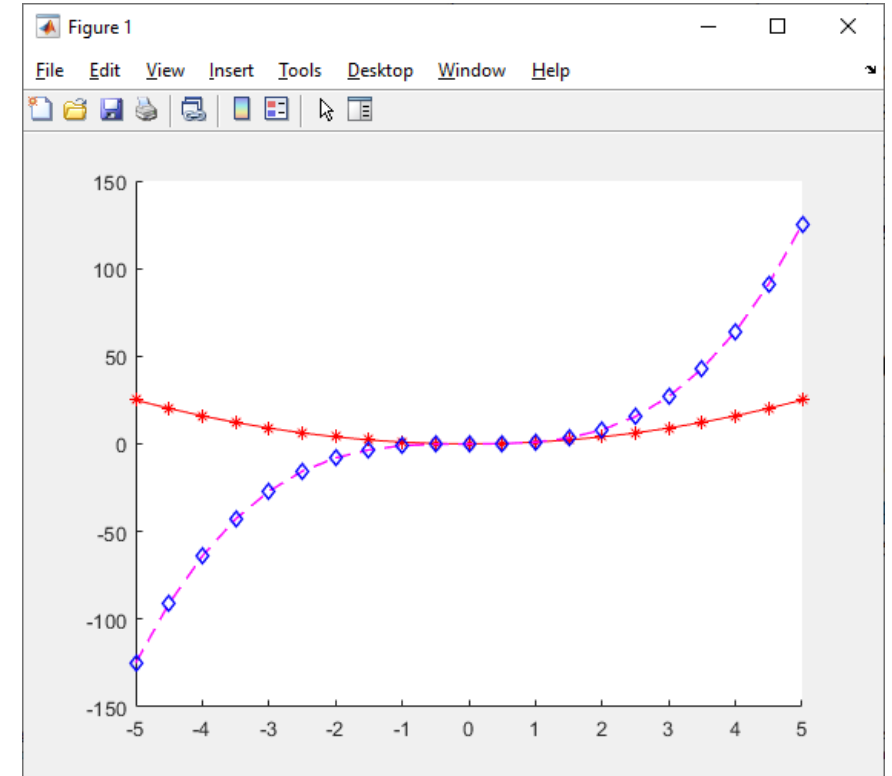
```
h.Color = 'k'; % h.color -> error (case sensitive)  
h.Color = [1 .5 0];  
h.LineWidth = 2;  
h.Marker = 'v';
```

`h.PropertyName = value;`

% 방법2 - set 사용

```
set(h, "color", 'm') % ''(char), ""(string) 모두 가능  
set(h, "markeredgecolor", 'b')  
set(h, "linewidth", 1)  
set(h, "marker", 'd')
```

`set(h, 'PropertyName', value)`



※ 속성 여러 개 동시 변경 가능

잠깐, 3강에서 배웠던 내용이...?

```
x = 0:0.1:1;  
y = x.^2;  
plot(x, y, 'LineStyle', 'propertyName', propertyValue)
```

	propertyName	의미	값	line specifier?	default
마커	marker	마커 종류	+o*.x^vsdph<>	Y	none
	markeredgecolor	마커 테두리 색	rgbmcykw 또는 1x3 벡터	N	auto
	markerfacecolor	마커 면 색		N	none
	markersize	마커 크기	양의 실수	N	6
선	color	선 색	rgbmcykw 또는 1x3 벡터	Y	[0 0.4470 0.7410]
	linestyle	선 스타일	- , -- , : , -.	Y	solid (-)
	linewidth	선 두께	양의 실수	N	0.5

- 핸들을 만들지 않을 경우

- 설정 가능한 속성은 7가지이며, 일단 만든 이후 변경이 어려움 (가능은 함)

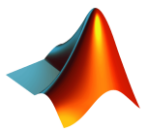
• 핸들을 만든 경우 -> 더 많은 속성 변경 가능



Line 객체의 주요 속성

Color	선 색 <code>rbgmcyk</code> 와 <code>[r, g, b]</code> 모두 가능
LineStyle	선 스타일 <code>-</code> <code>--</code> <code>:</code> <code>-.</code>
LineWidth	선 두께
Marker	마커 종류 <code>+o*.x^v</code> <code>sdph<></code>
MarkerEdgeColor	마커 테두리 색
MarkerFaceColor	마커 면 색
MarkerSize	마커 크기

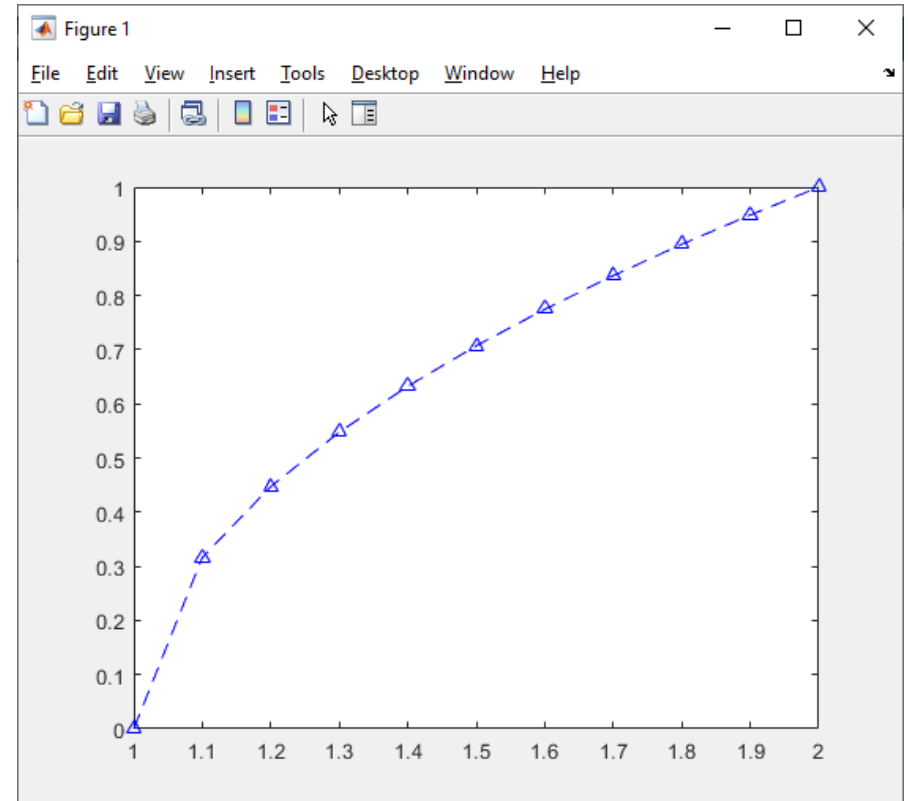
Parent	부모 Axes 객체
Type	객체 종류 (Line은 <code>'line'</code>)
XData	데이터 점들의 x좌표
YData	데이터 점들의 y좌표
ZData	데이터 점들의 z좌표



Line 객체의 XData, YData

```
x = 0:0.1:1;  
y = x.^2;  
  
figure,  
h = plot(x, y, 'b--^');  
  
disp([h.XData' h.YData'])  
  
h.YData = sqrt(x);  
h.XData = x+1;
```

- Line 객체의 XData, YData 변경
 - Line의 속성은 그대로 두고 점 위치만 바뀐다.
 - xlim, ylim, 또는 axis로 설정한 축 범위가 바뀌지 않는다.
 - 점 개수를 바꿀 수도 있다. (XData, YData 동시 변경 필요)



Surface 객체 (surf, mesh)

```
[xx, yy] = meshgrid(-1:0.1:1,-1:0.1:1);  
zz = xx.^2-yy.^2;
```

```
figure, hs = surf(xx,yy,zz);  
figure, hm = mesh(xx,yy,zz);
```

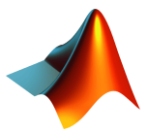
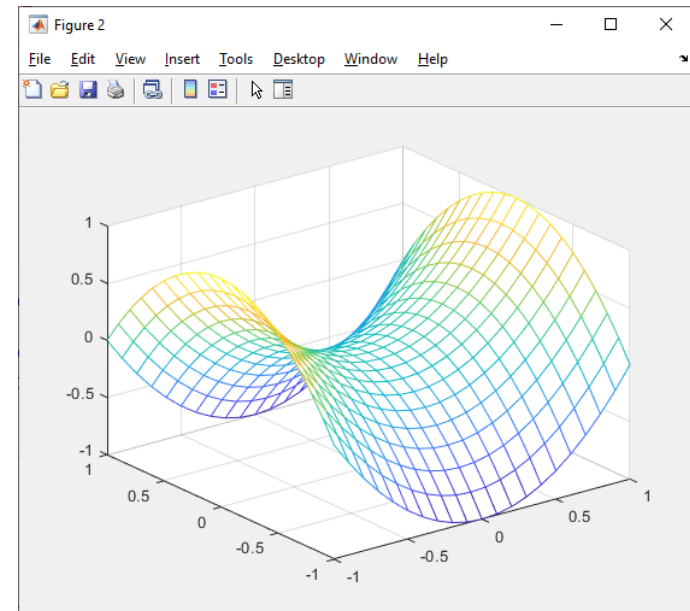
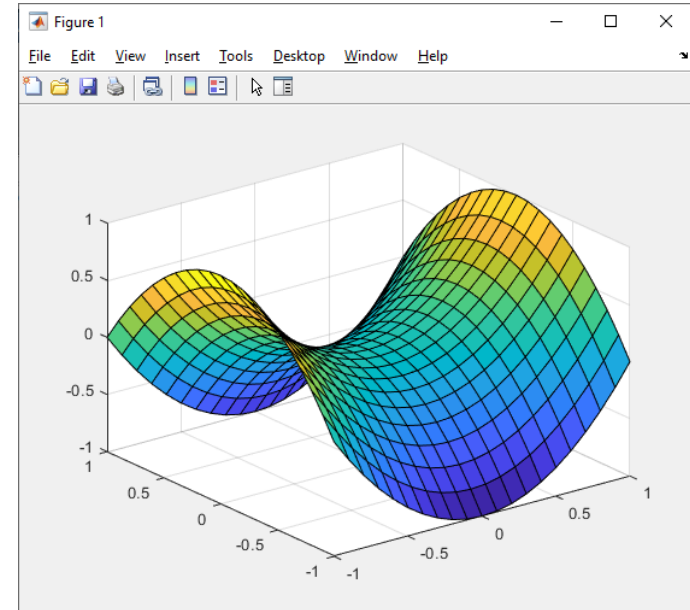
```
>> get(hs)
```

```
EdgeColor: [0 0 0]  
EdgeLighting: 'none'  
FaceColor: 'flat'  
FaceLighting: 'flat'
```

```
>> get(hm)
```

```
EdgeColor: 'flat'  
EdgeLighting: 'flat'  
FaceColor: [1 1 1]  
FaceLighting: 'none'
```

```
set(hs, 'EdgeColor', 'flat', ...  
      'EdgeLighting', 'flat', ...  
      'FaceColor', [1 1 1], ...  
      'FaceLighting', 'none')
```



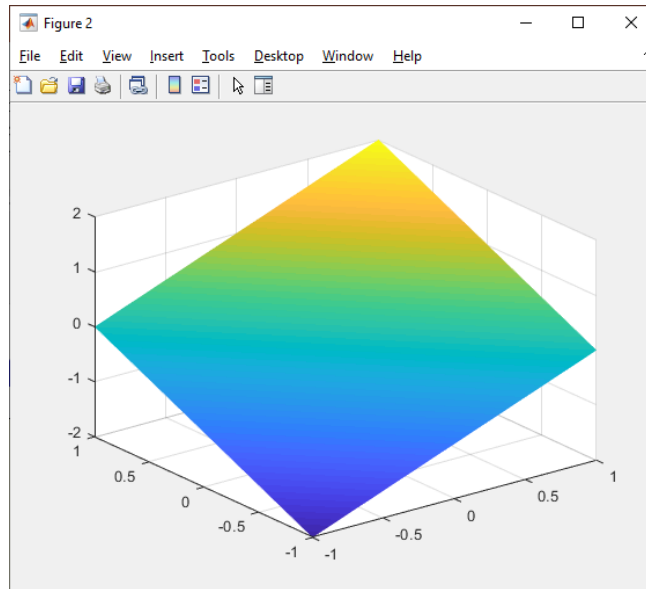
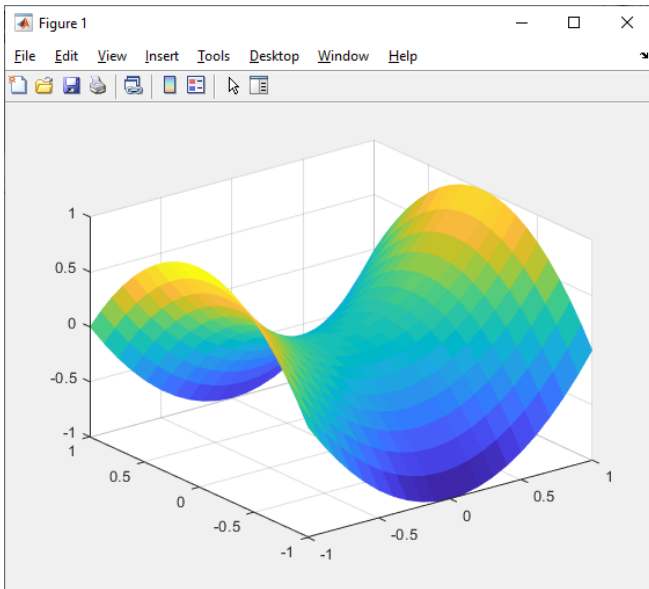
Surface 객체 (surf, mesh)

```
[xx, yy] = meshgrid(-1:0.1:1,-1:0.1:1);  
zz = xx.^2-yy.^2;
```

```
figure, hs1 = surf(xx,yy,zz); shading flat
```

```
figure, hs2 = surf(xx,yy,zz);  
hs2.EdgeColor = 'none'; % same as shading flat  
hs2.FaceColor = 'interp'; % same as shading interp
```

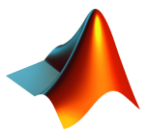
```
hs2.ZData = xx+yy;
```



Surface 객체의 주요 속성

EdgeAlpha	테두리 선 불투명도 'flat' 'interp'
Edgecolor	테두리 선 색 'none' 'flat' 'interp'
FaceAlpha	면 불투명도 'flat' 'interp'
FaceColor	면 색 'none' 'flat' 'interp'
LineStyle	선 스타일 - -- : -.
LineWidth	선 두께

Marker	포인트 마커 종류 <code>+o*.x^v</code> <code>sdph<></code>
MarkerEdgeColor	마커 테두리 색
MarkerFaceColor	마커 면 색
MarkerSize	마커 크기
Parent	부모 Axes 객체
Type	객체 종류 (Surface는 'surface')
XData	데이터 점들의 x좌표
YData	데이터 점들의 y좌표
ZData	데이터 점들의 z좌표



Text 객체

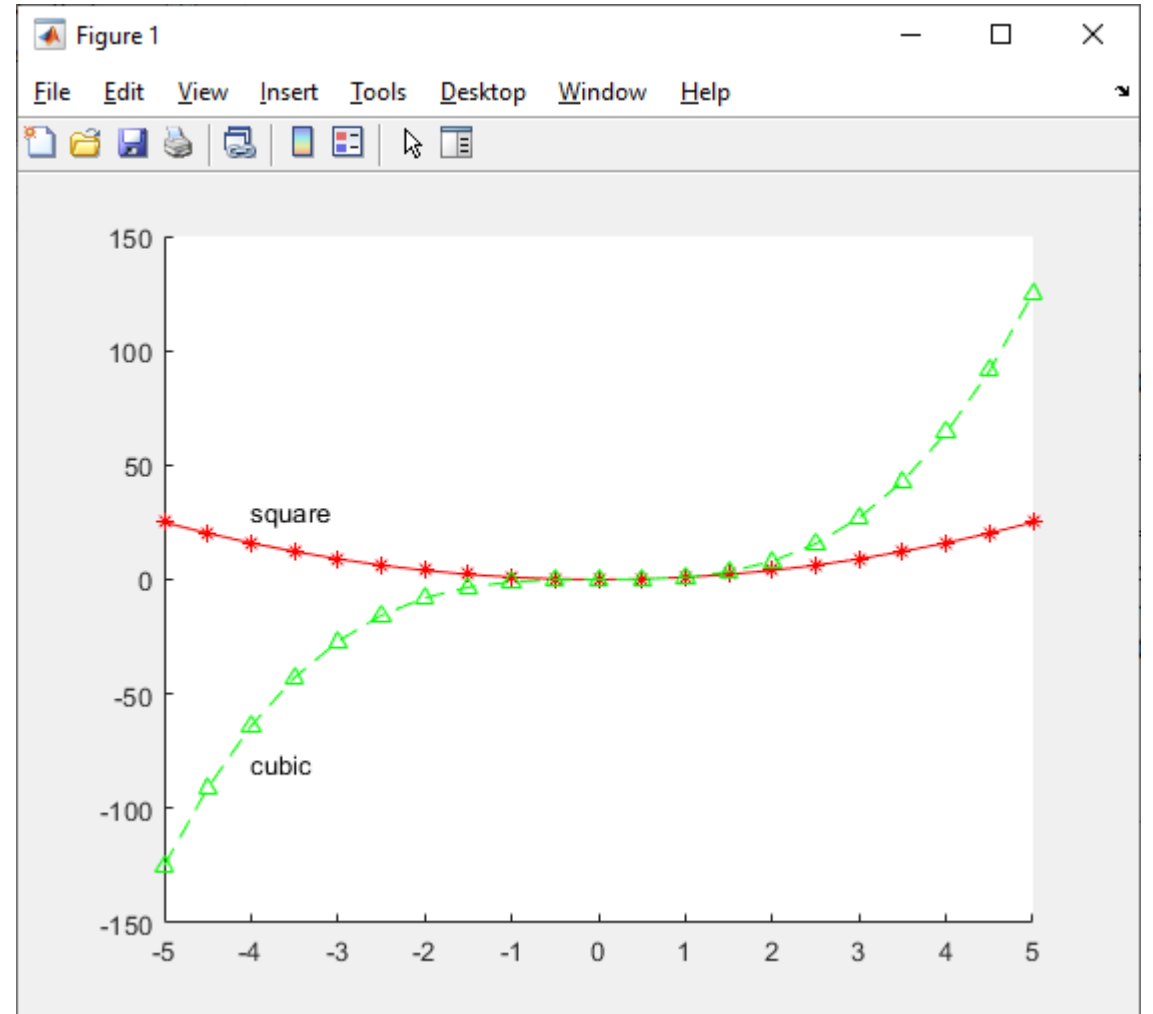
```
x = -5:0.5:5;
y1 = x.^2;
y2 = x.^3;

figure, hold on

plot(x, y1, 'r-*');
plot(x, y2, 'g--^');

h_t1 = text(-4, 30, 'square');
h_t2 = text(-4, -80, 'cubic');
```

- Text 위치 변경
 - Edit Plot으로
 - Property Inspector에서
 - `set(h, 'Position', [x, y, z])`
 - `h.Position = [x, y, z];`



Text 객체의 위치 - Position 속성

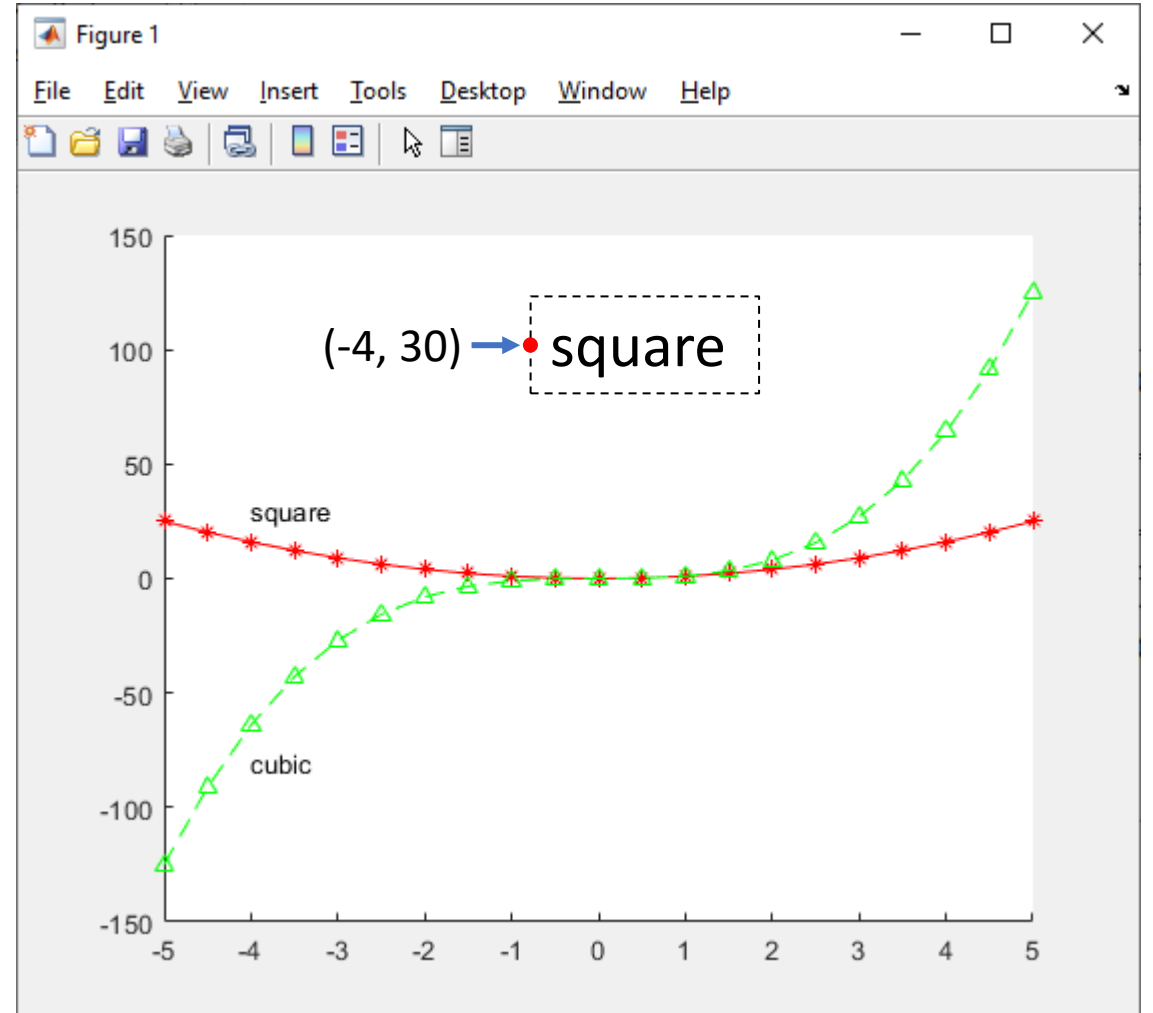
```
x = -5:0.5:5;  
y1 = x.^2;  
y2 = x.^3;
```

```
figure, hold on
```

```
plot(x, y1, 'r-*');  
plot(x, y2, 'g--^');
```

```
h_t1 = text(-4, 30, 'square');  
h_t2 = text(-4, -80, 'cubic');
```

```
>> h_t1.Position  
ans =  
    -4    30     0  
>> h_t2.Position  
ans =  
    -4   -80     0
```

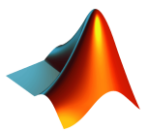


- text(x,y) -> Text 객체의 좌측 중앙점 위치
- Text 객체의 크기는 자동조절
- HorizontalAlignment, VerticalAlignment 속성으로 기준점 기준 객체 방향 변경 가능

Text 객체의 주요 속성

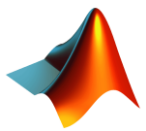
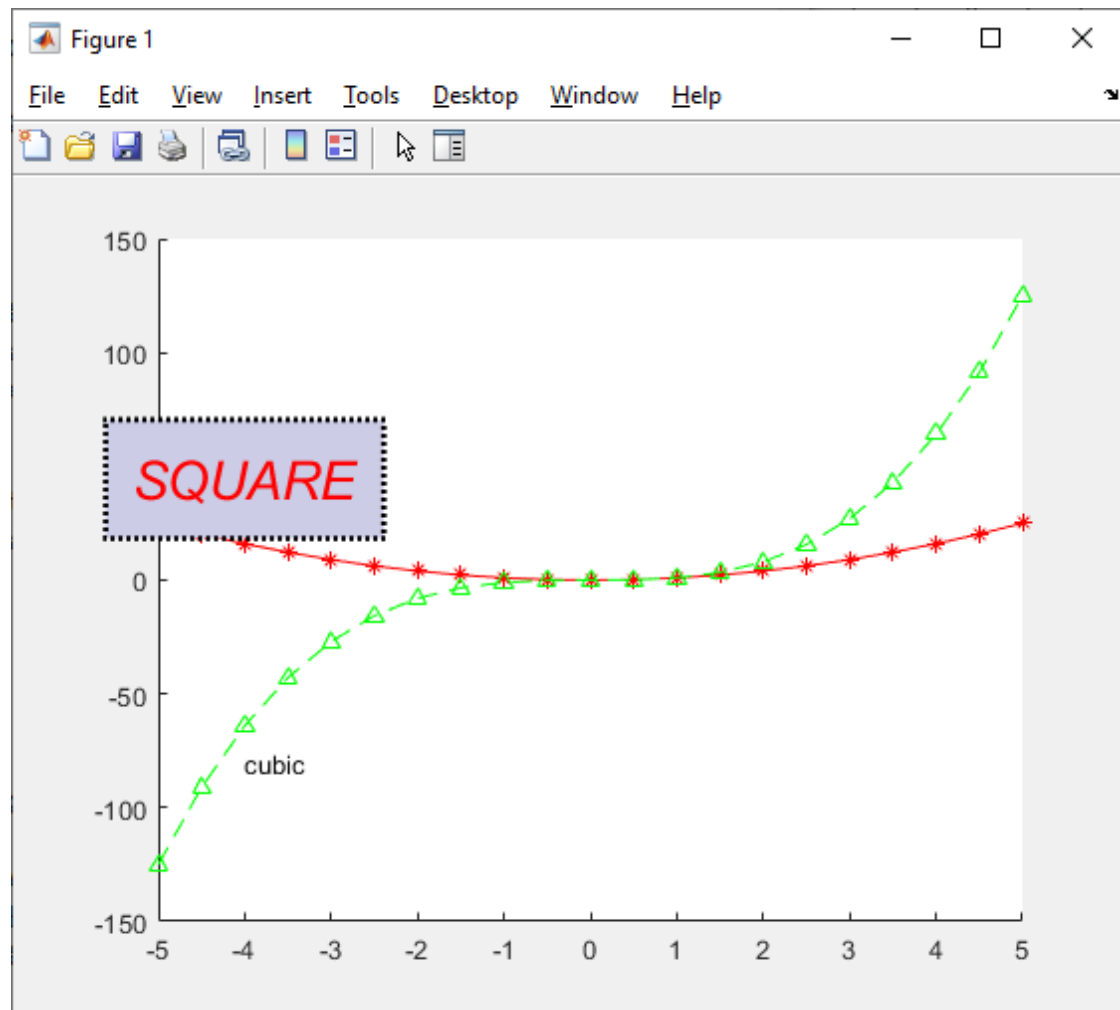
BackgroundColor	배경색
Color	글자색
EdgeColor	테두리색
FontAngle	폰트 기울기 'normal' 'italic'
FontName	글꼴
FontSize	글자크기
FontWeight	폰트 굵기 'normal' 'bold'
HorizontalAlignment	기준점 기준 가로정렬 방법

LineStyle	테두리 종류 (plot과 동일)
LineWidth	테두리 두께
Margin	텍스트-테두리 간 여백
Parent	부모 Axes 객체
Position	기준점의 위치 (Axes 좌표계 기준)
Rotation	반시계 방향 회전각 (deg)
String	문자열
Type	객체 종류 (Text는 'text')
VerticalAlignment	기준점 기준 세로정렬 방법



Text 객체의 주요 속성

```
h_t1.BackgroundColor = [.8 .8 .9];  
h_t1.Color = 'r';  
  
h_t1.EdgeColor = 'k';  
h_t1.LineStyle = ':';  
h_t1.LineWidth = 2;  
  
h_t1.Margin = 10;  
  
h_t1.FontAngle = 'italic';  
h_t1.FontSize = 20;  
  
h_t1.HorizontalAlignment = 'center';  
h_t1.VerticalAlignment = 'bottom';  
  
h_t1.String = 'SQUARE';
```



Lable, Title도 Text 객체이다.

```
x = -5:0.5:5;
y1 = x.^2;
y2 = x.^3;

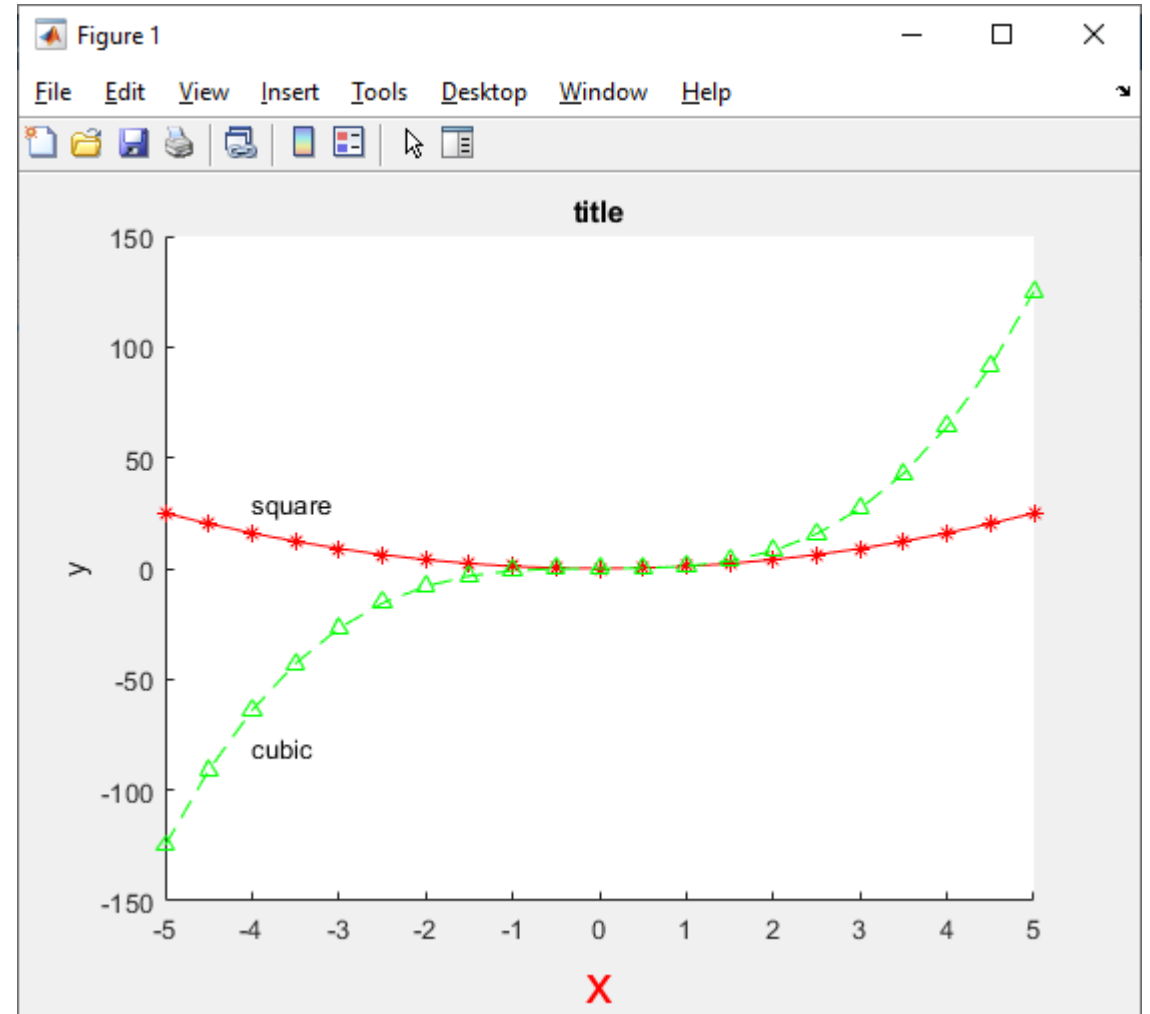
figure, hold on

plot(x, y1, 'r-*');
plot(x, y2, 'g--^');

h_t1 = text(-4, 30, 'square');
h_t2 = text(-4, -80, 'cubic');

h_xlabel = xlabel('x');
h_ylabel = ylabel('y');
h_ttl = title('title');

set(h_xlabel, 'fontsize', 20, 'color', 'r')
get(h_xlabel, 'position')
```



```
>> get(h_xlabel, 'position')
ans =
    0.0000 -171.3855  -1.0000
```

그 외에...

- FunctionLine

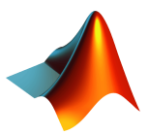
- `h = fplot(...);`
- <https://kr.mathworks.com/help/matlab/ref/matlab.graphics.function.functionline-properties.html>

- Image

- `h = imshow(...);`
- https://kr.mathworks.com/help/matlab/creating_plots/the-image-object-and-its-properties.html

- Histogram

- `h = histogram(...);`
- <https://kr.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.histogram-properties.html>

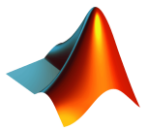


객체 속성 설정 관련 팁

- 탭 키로 자동완성 가능
 - h. 까지 누르고 탭
 - h.Co1 까지 누르고 탭
 - set(h," 까지 누르고 탭
 - set(h,"co1 까지 누르고 탭
- 속성의 설정가능 값 목록 보기

```
set(h,"LineStyle") % 설정 가능한 모든 LineStyle 보기  
set(h,"LineWidth") % 값 종류 무한대 -> 빈 cell array
```

- 일부러 엉뚱한 값을 넣고 에러메시지를 보는 방법도 가능 (h_t1.FontAngle = 1)
- 유한 옵션과 무한 옵션이 모두 가능한 속성도 있음 (예. Surface의 FaceColor)
- 속성 관련 자세한 내용은 도움말을 볼 것



figure의 구성 (계층구조)

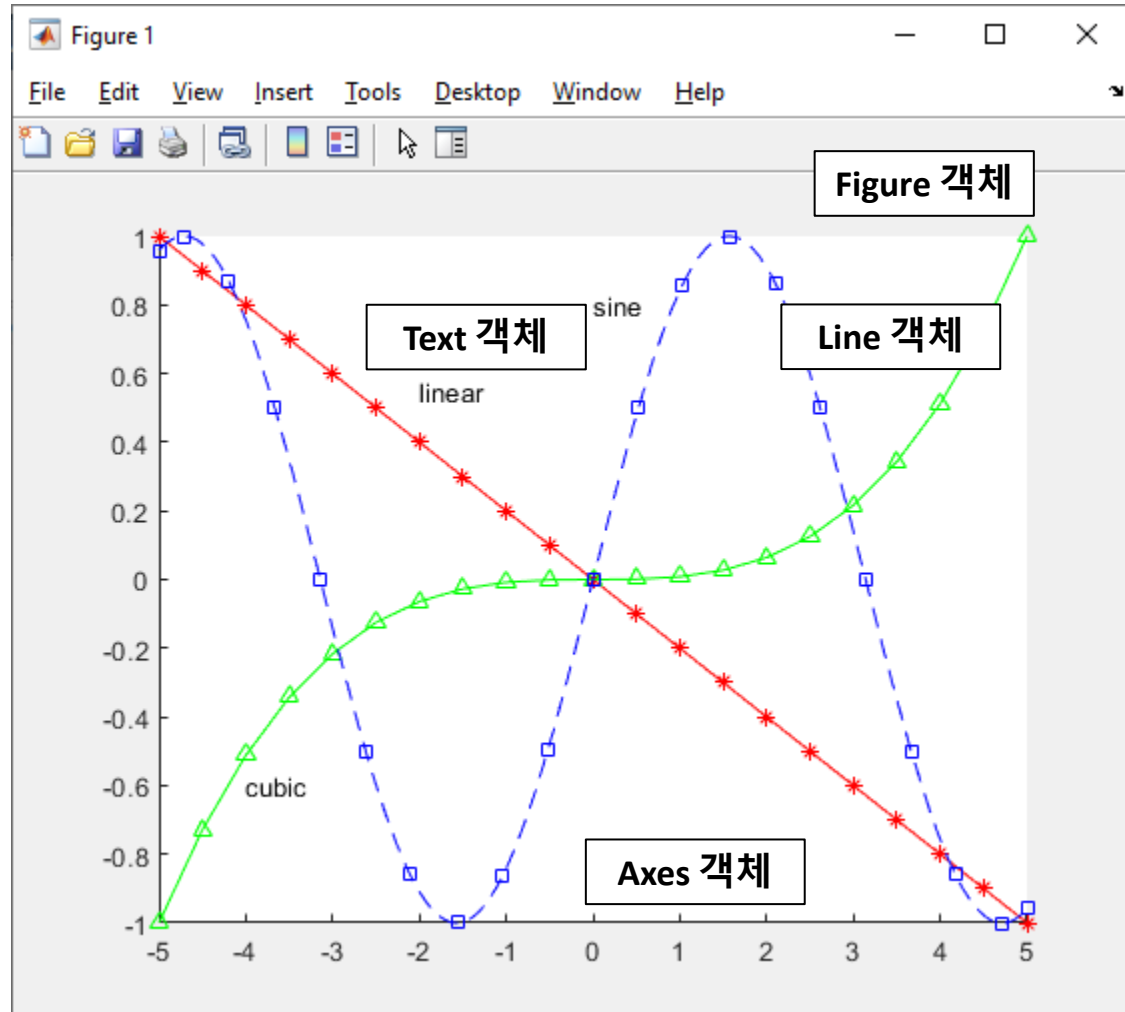


Figure 객체

Axes 객체

Line 객체

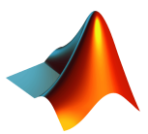
Line 객체

Line 객체

Text 객체

Text 객체

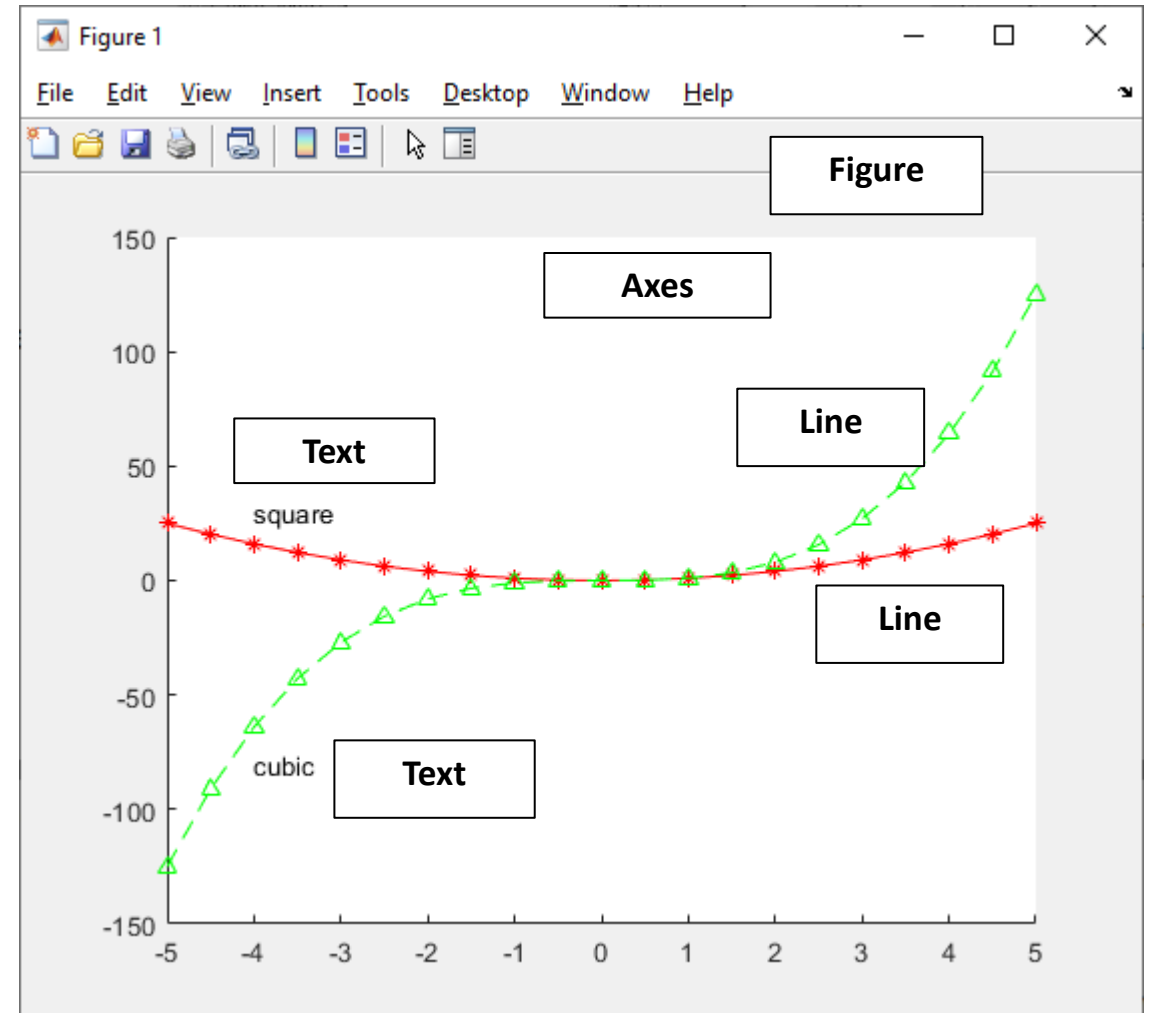
Text 객체



Axes 객체란?

- Plot Object(Line, Surface, Text 등)가 올라가는 영역
 - Axes = Plot Object의 Parent
 - Line, Surface, Text = Axes의 Children
 - Figure = Axes의 Parent
- Plot Object를 만들면 Axes 자동 생성

```
x = -5:0.5:5;  
y1 = x.^2;  
y2 = x.^3;  
  
figure, hold on,  
h_l1 = plot(x, y1, 'r-*');  
h_l2 = plot(x, y2, 'g--^');  
  
h_t1 = text(-4, 30, 'square');  
h_t2 = text(-4, -80, 'cubic');
```



Axes의 핸들을 얻는 방법

```
figure,
```

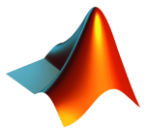
```
% getting axes handle: axes
```

```
ax = axes; % 현재 활성화된 Figure에 Axes를 새로 만들고 핸들을 변수 ax에 저장  
get(ax)
```

```
% getting axes handle: gca
```

```
get(gca) % 현재 활성화된 Axes의 핸들을 가져옴 (없으면 새로 만들고 가져옴)
```

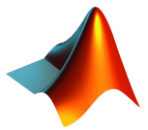
- `gca = get current axes`
 - 현재 활성화되어 있는 Axes의 핸들을 반환
 - Axes 핸들을 만들지 않고도 Axes에 접근 가능
 - `gca`는 구조체처럼 접근할 수 없으며 항상 `get`, `set` 필요
 - 활성화된 Figure가 바뀌면 `gca`도 해당 Figure의 Axes로 변경
- 열려있는 Figure 창이 없으면 -> Figure를 하나 만들고 Axes를 하나 만들면서 반환



활성화된 Axes란?

- Axes 활성화 방법
 - 방법1. 마우스로 Axes 객체를 클릭
 - 방법2. 만들어놓은 Axes 객체 핸들 ax에 대해서 axes(ax)
- 원하는 Axes에 plot 하는 방법

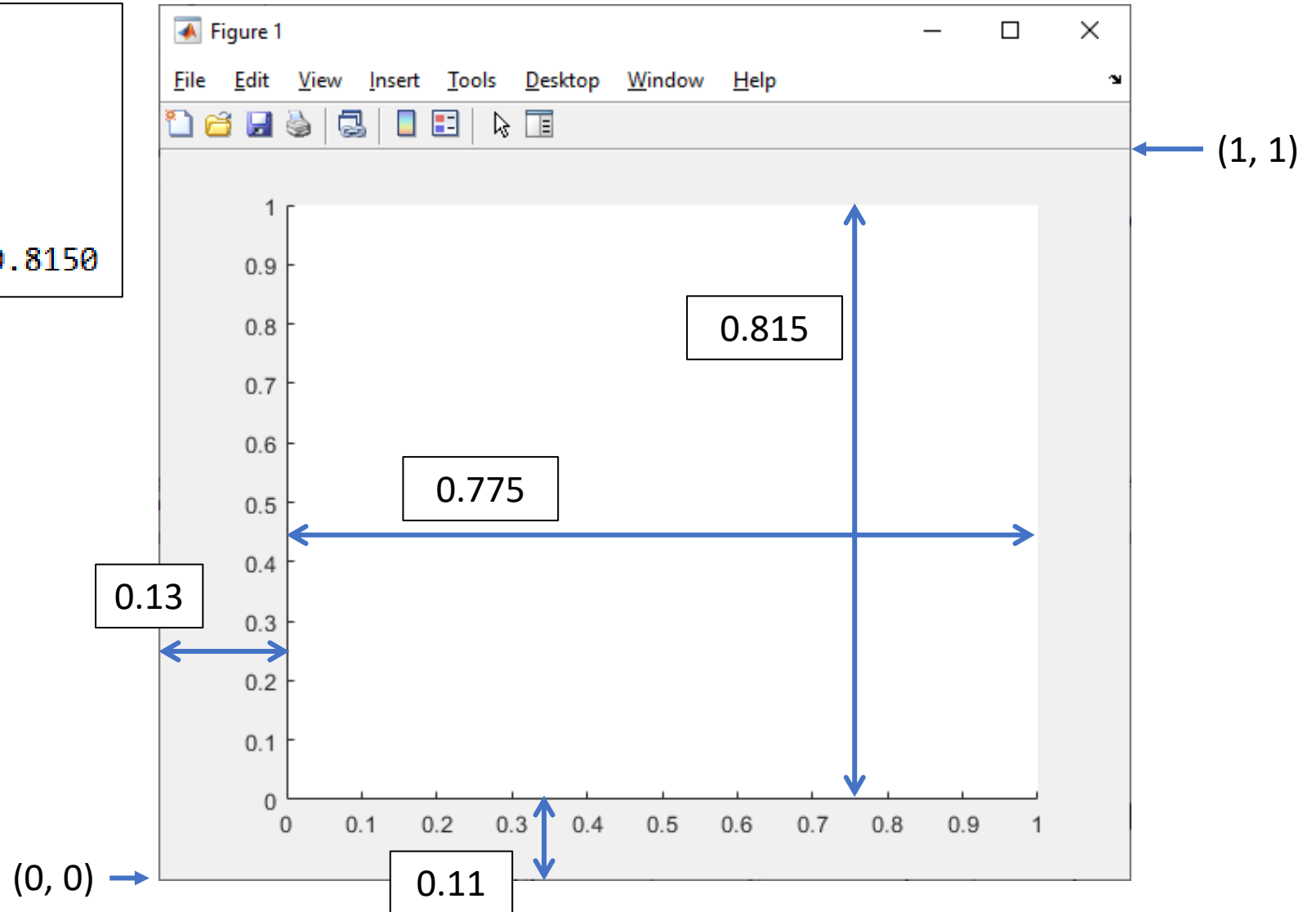
```
figure, ax1 = axes;  
figure, ax2 = axes;  
  
% 방법1: 원하는 Axes를 활성화하고 plot  
axes(ax1);  
plot(rand(10,1));  
  
% 방법2: plot에 Axes 객체를 지정  
plot(ax2, rand(10,1));
```



Axes의 Position

```
>> get(gca, 'units')  
ans =  
    'normalized'  
>> get(gca, 'position')  
ans =  
    0.1300    0.1100    0.7750    0.8150
```

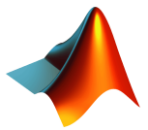
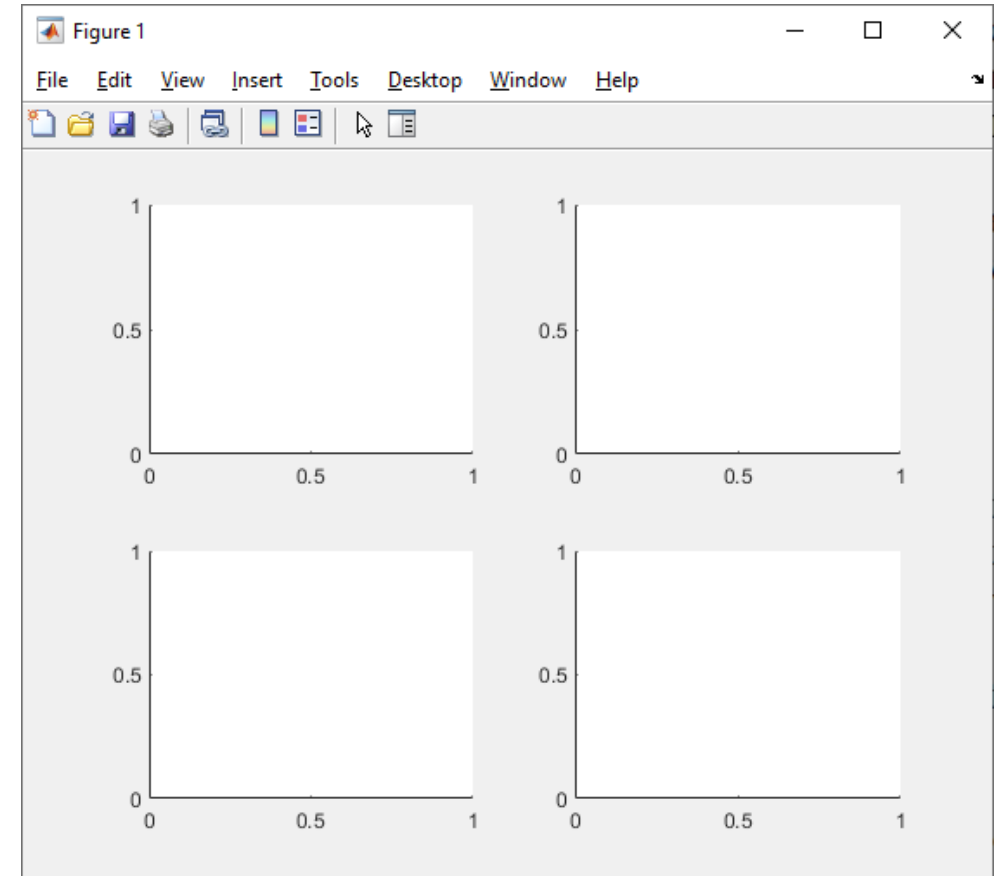
[left, bottom, width, height]



subplot의 비밀...!!

```
figure,  
h1 = subplot(2,2,1);  
h2 = subplot(2,2,2);  
h3 = subplot(2,2,3);  
h4 = subplot(2,2,4);
```

- subplot(m, n, p)
 - 현재 Figure에 Axes 여러 개를 만든다.
 - m, n, p를 이용하여 각 Axes의 위치를 알아서 계산해서 지정해준다.
 - 이미 만든 subplot에 대해서 subplot(m, n, p) = 해당 subplot(Axes)를 활성화
 - subplot의 Position을 바꿔보자.
 - 원하는 subplot을 활성화시키고 plot 해보자.



hold on으로 겹쳐 그리기?

```
x = -5:0.5:5;
y1 = x.^2;
y2 = x.^3;

figure,
get(gca, 'NextPlot')
hold on, % hold(gca, 'on')
get(gca, 'NextPlot')

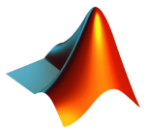
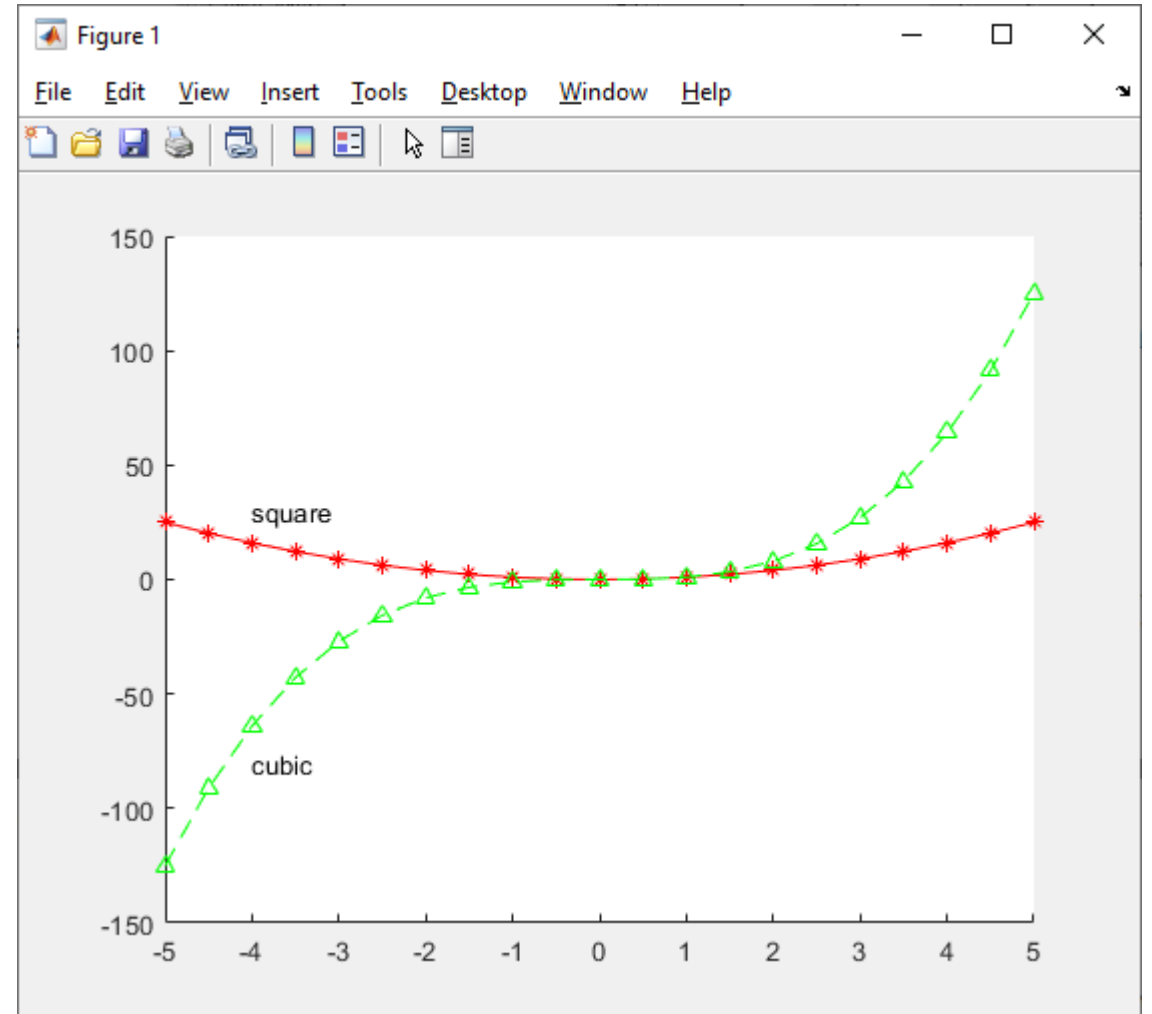
h_l1 = plot(x, y1, 'r-*');
h_l2 = plot(x, y2, 'g--^');

h_t1 = text(-4, 30, 'square');
h_t2 = text(-4, -80, 'cubic');

get(gca, 'Children')
```

- hold on

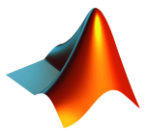
- Axes의 NextPlot 속성을 바꿈
- Children이 여러 개 생김 (Children = 그래픽 객체의 **배열**)



Axes 객체의 주요 속성

Box	Axes 테두리 'on' 'off'
Children	자식 객체
ColorOrder	연속 plot 시 자동 선 색 지정
FontAngle	폰트 기울기 'normal' 'italic'
FontName	글꼴
FontSize	글자 크기
FontWeight	폰트 굵기 'normal' 'bold'

NextPlot	다음 plot의 겹쳐 그리기 여부 'replace' 'add'
Parent	부모 Figure 객체
Position	Figure 상 Axes의 위치 및 크기
Type	객체 종류 (Axes는 'axes')
Units	Position의 단위
View	뷰 각도



Axes 객체의 주요 속성

```
figure,  
ax = axes;  
  
ax.Color = [.95 .95 .95];  
  
ax.Title.String = 'TITLEEEEE';  
ax.XLabel.String = 'XXX';  
  
ax.XAxisLocation = 'top';  
ax.XDir = 'reverse';  
ax.XColor = 'r';  
  
ax.XGrid = 'on';  
ax.XMinorGrid = 'on';  
ax.XMinorTick = 'on';  
ax.XTickLabelRotation = 45;  
ax.TickDir = 'out';  
  
ax.GridColor = 'b';  
ax.MinorGridColor = 'g';  
ax.GridAlpha = 1;  
ax.MinorGridAlpha = 1;  
ax.MinorGridLineStyle = '-';  
ax.XScale = 'linear';
```

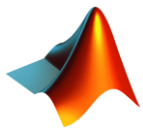
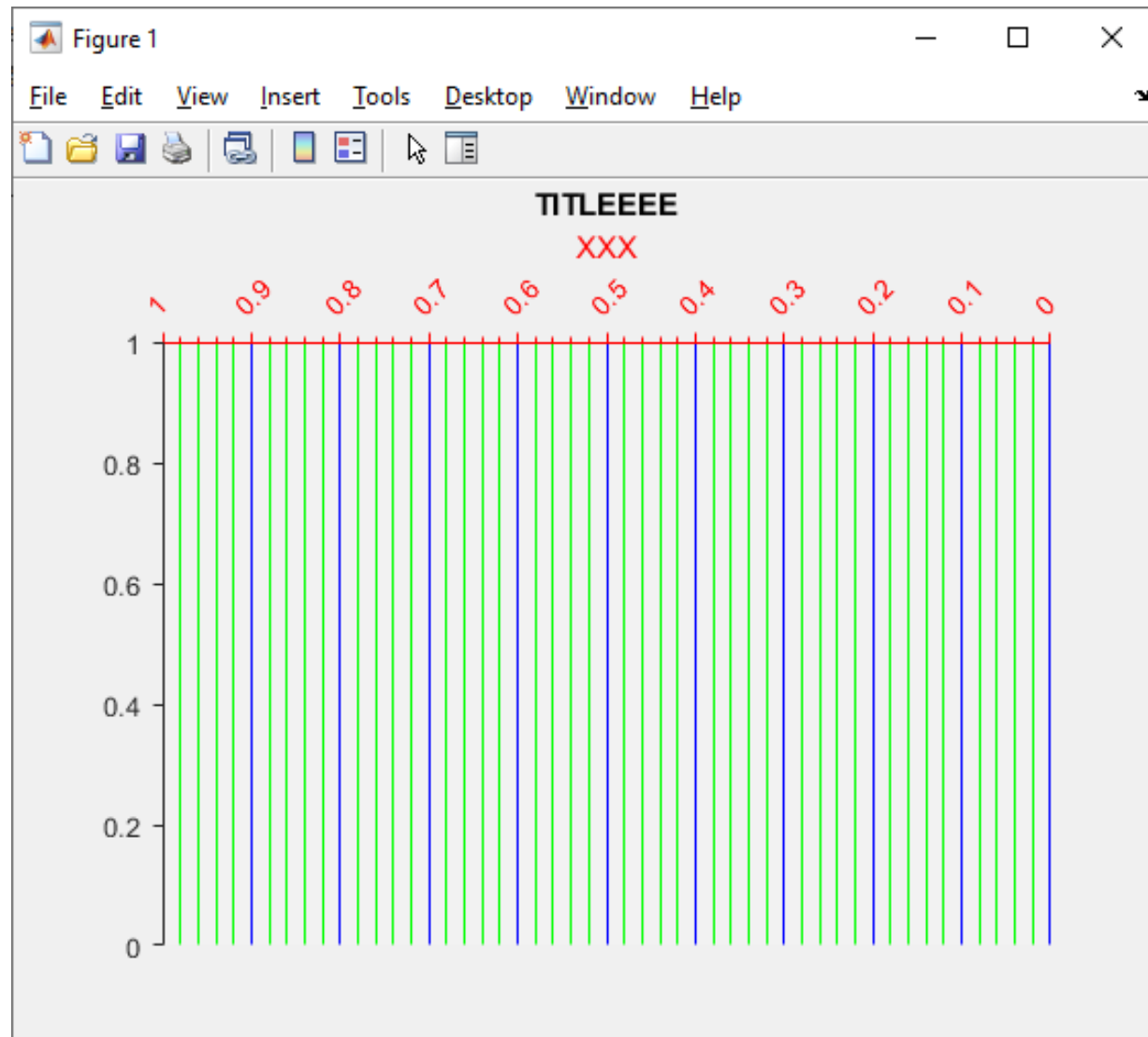
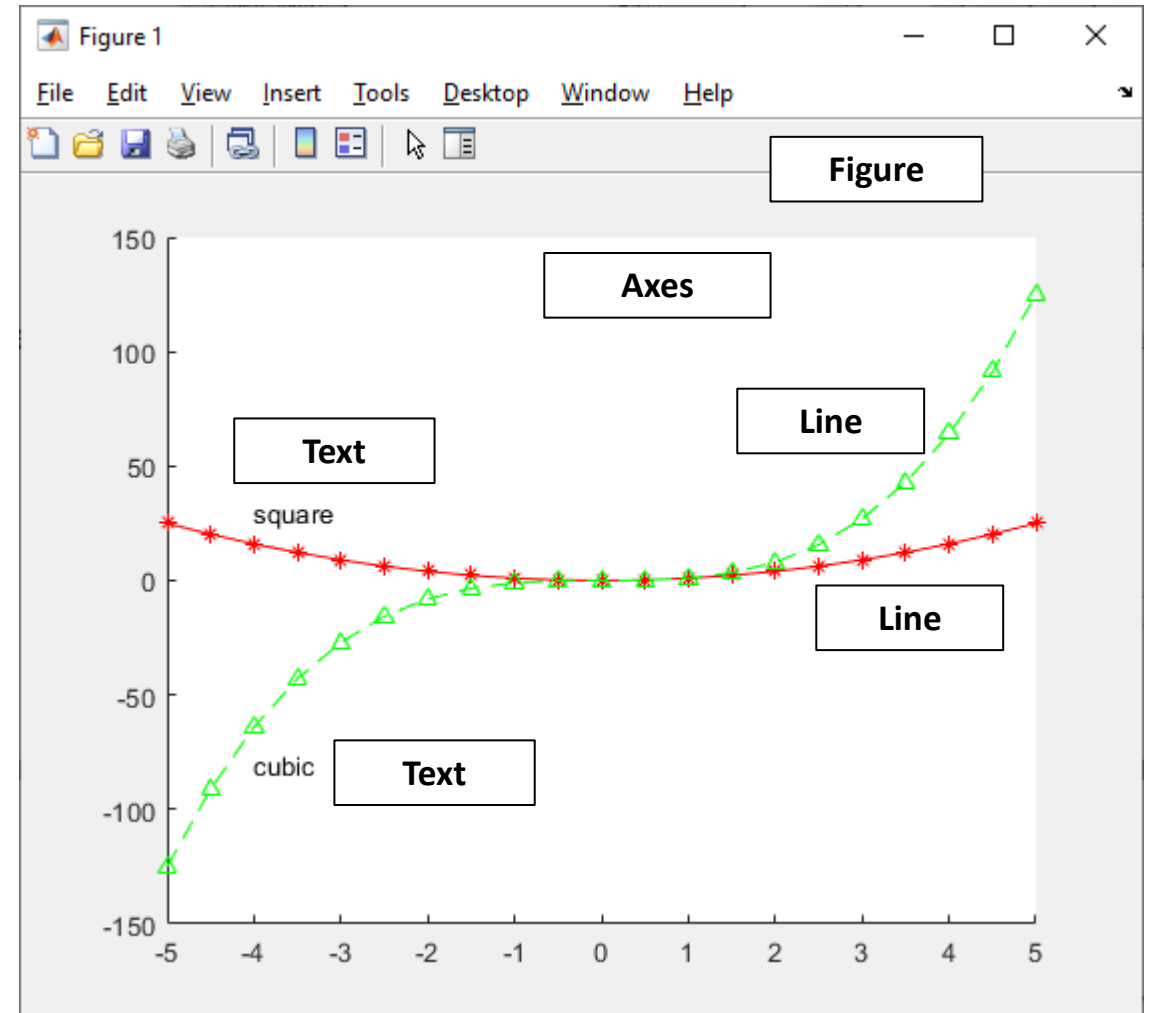


Figure 객체란?

- Axes가 올라가는 객체
 - Figure = Axes의 Parent
 - Axes = Figure의 Children
 - subplot -> Children(Axes)을 여러 개 만듦
- Figure가 없는 상태에서 Axes를 만들면 Figure 객체 자동 생성



Figure의 핸들을 얻는 방법

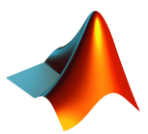
```
h = figure; % Figure를 새로 만들면서 핸들 반환
```

```
h = figure(1); % 1번 figure가 있으면 해당 핸들 가져옴 (없으면 새로 만들고 반환)
```

```
h = figure(10); % 10번 figure가 있으면 해당 핸들 가져옴 (없으면 새로 만들고 반환)
```

```
get(gcf) % 현재 활성화된 Figure의 핸들을 가져옴 (없으면 새로 만들고 가져옴)
```

- `gcf = get current figure`
 - 현재 활성화되어 있는 Figure 객체(`figure` 창)의 핸들을 반환
 - Figure 핸들을 만들지 않고도 Figure에 접근 가능
 - `gcf`는 구조체처럼 접근할 수 없으며 항상 `get`, `set` 필요
- Figure 활성화 방법
 - 방법1. 마우스로 Figure 창 클릭
 - 방법2. `figure(1)` -> 1번 figure를 활성화

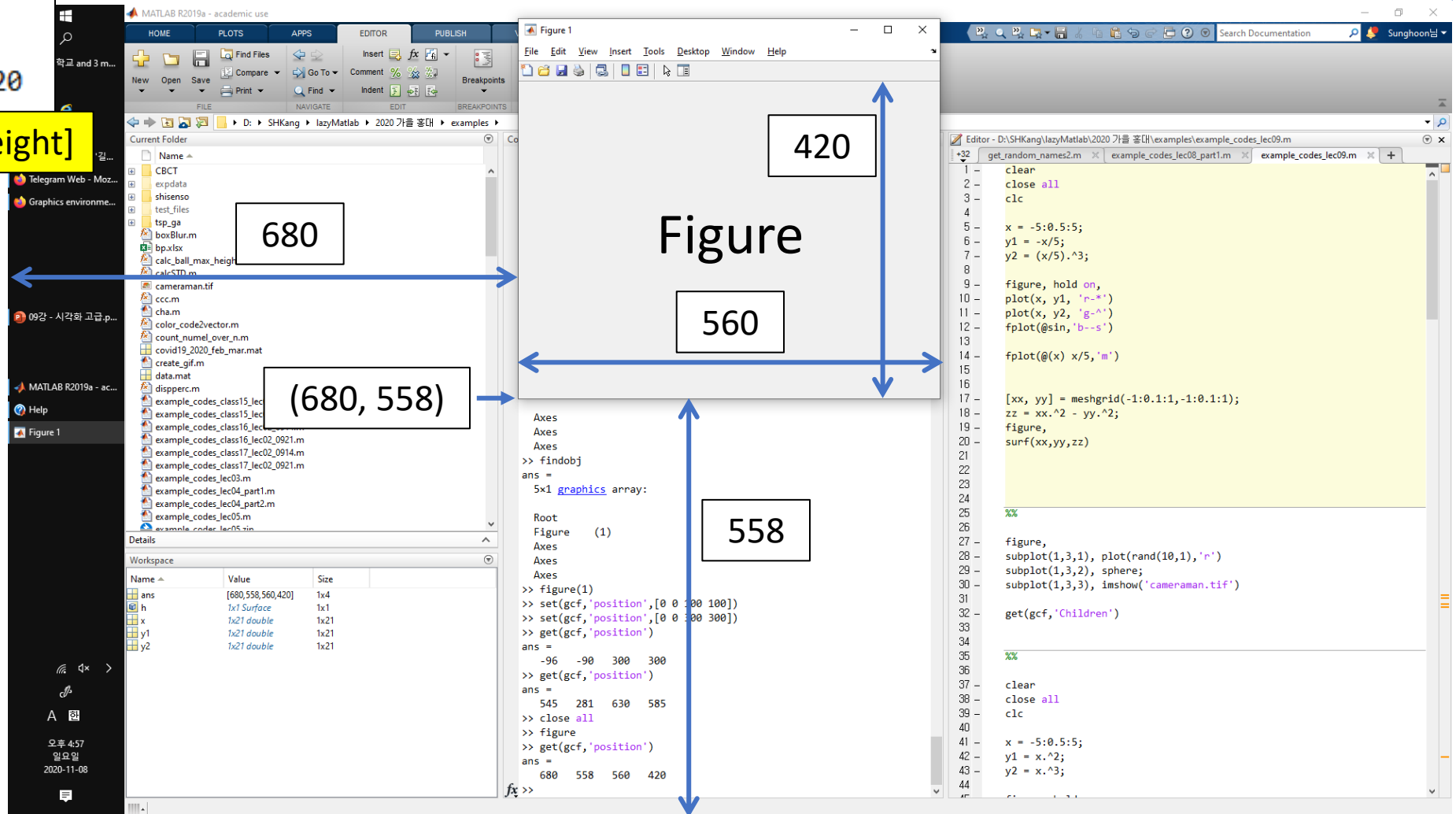


Figure의 Position (units가 pixel일 때)

```
>> get(gcf,'units')
ans =
    'pixels'
>> get(gcf,'position')
ans =
    680    558    560    420
```

[left, bottom, width, height]

모니터 해상도 (1920, 1080)



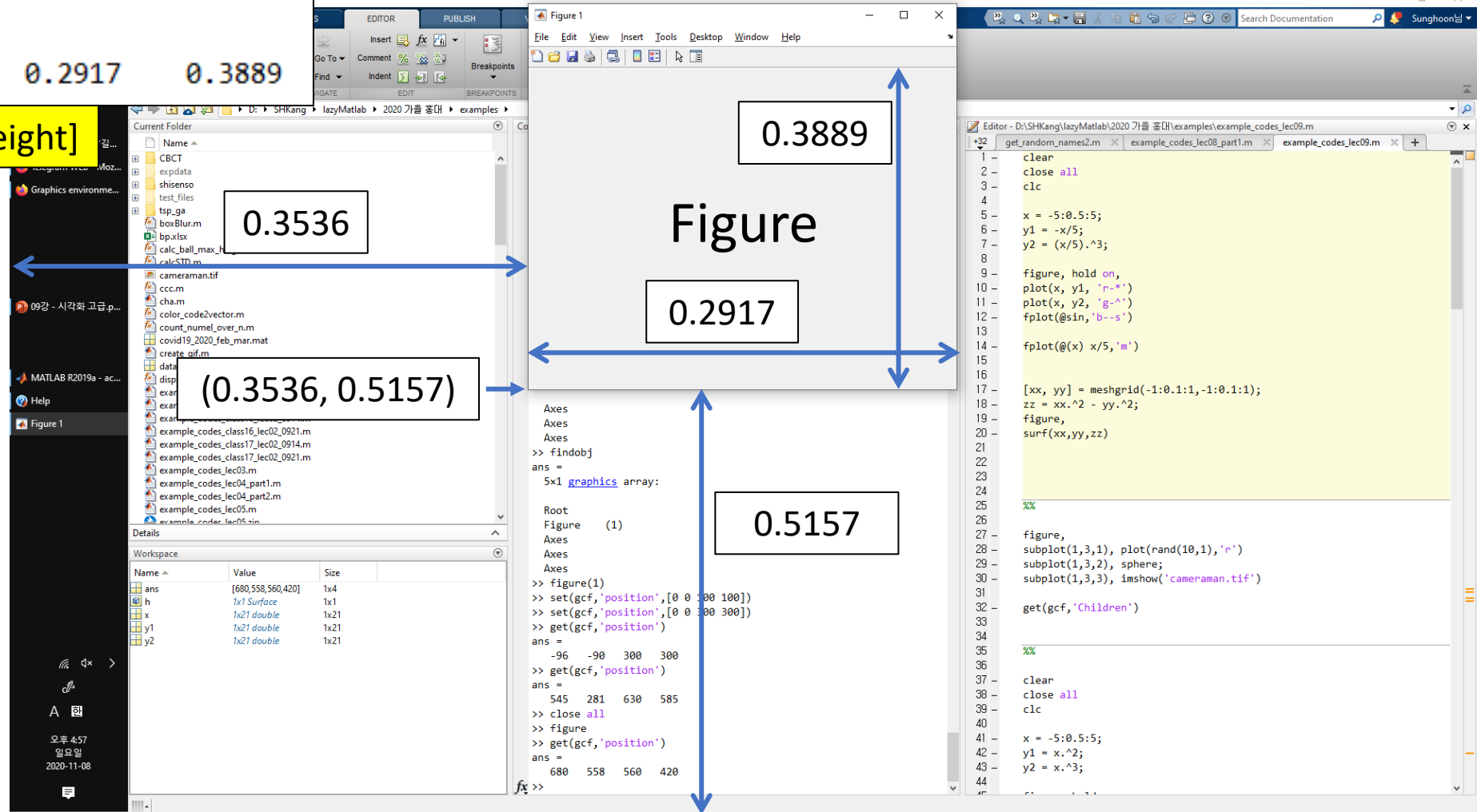
Figure의 Position (units가 normalized일 때)

```
>> get(gcf,'units')
ans =
    'normalized'
>> get(gcf,'position')
ans =
    0.3536    0.5157    0.2917    0.3889
```

[left, bottom, width, height]

※ 모니터 해상도에 상관없이
항상 같은 위치에 Figure를
띄울 수 있음

해상도 무관 고정값 (1, 1)



(0, 0) →

Figure의 Position 예제

```
dx = 0.1;
dy = 0.1;

x = -1:dx:3;
y = 1:dy:4;
[xx, yy] = meshgrid(x, y);
zz = xx.*yy.^2./(xx.^2+yy.^2);

cmaps = {'parula', 'jet', 'cool';
         'summer', 'winter', 'gray'};

for i=1:size(cmaps,1)
    for j=1:size(cmaps,2)
        figure,
        set(gcf, 'position', [200+(j-1)*300, 100+(i-1)*340, 300, 260])
        surf(xx, yy, zz), shading interp
        colormap(cmmaps{i,j})
        colorbar
        title(cmmaps{i,j})
    end
end
```

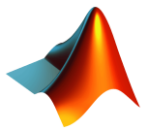
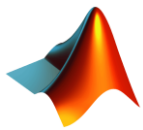


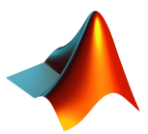
Figure 객체의 주요 속성

Children	자식 Axes 객체
Color	배경색
CurrentAxes	현재 활성화된 Axes
Name	Figure 창의 타이틀
NumberTitle	Figure 창 타이틀에 Figure 번호를 쓰지 말지 'on' 'off'
Parent	부모 Root 객체
Position	모니터 상 위치
Type	객체 종류 (Figure는 'figure')
WindowState	Figure 창 상태 'normal' 'maximized' 'minimized' 'fullscreen'
WindowStyle	Figure 창 스타일 'normal' 'modal' 'docked'



그래픽 객체의 기본값을 바꿀 수 있을까?

- Figure 창 배경색 기본값을 [.8 .8 .8]로 하고 싶다면?
- Axes 생성 시 글자크기를 기본 14로 하고 싶다면?
- Text 생성 시 자동으로 BackgroundColor를 [.9 .9 .9]로 하고 싶다면?
- Axes 내 Line들의 ColorOrder를 내 마음대로 하고 싶다면?



그래픽 객체 속성의 기본값 설정

`set(handle, DefaultPropertyName, value)`



Figure, Axes, Root 객체

h -> 핸들 h의 객체에 적용
gcf -> active Figure에 적용
gca -> active Axes에 적용
groot -> 모든 Figure, Axes에 적용

'default' + ObjectType + PropertyName

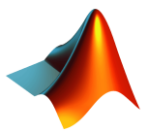
Figure의 배경색(Color)	-> 'defaultFigureColor'
Axes의 글자크기(FontSize)	-> 'defaultAxesFontSize'
Text의 글자기울기(FontAngle)	-> 'defaultTextFontAngle'
Axes의 ColorOrder	-> 'defaultAxesColorOrder'

* groot: 모든 그래픽 객체들의 최상위에 있는 root 객체 (**graphics root**)
- 매트랩에서 만드는 모든 그래픽 객체들의 기본값 설정 가능

https://kr.mathworks.com/help/matlab/creating_plots/default-property-values.html

<https://kr.mathworks.com/help/matlab/ref/groot.html>

<https://kr.mathworks.com/help/matlab/ref/matlab.ui.root-properties.html>



기본값 설정 예시

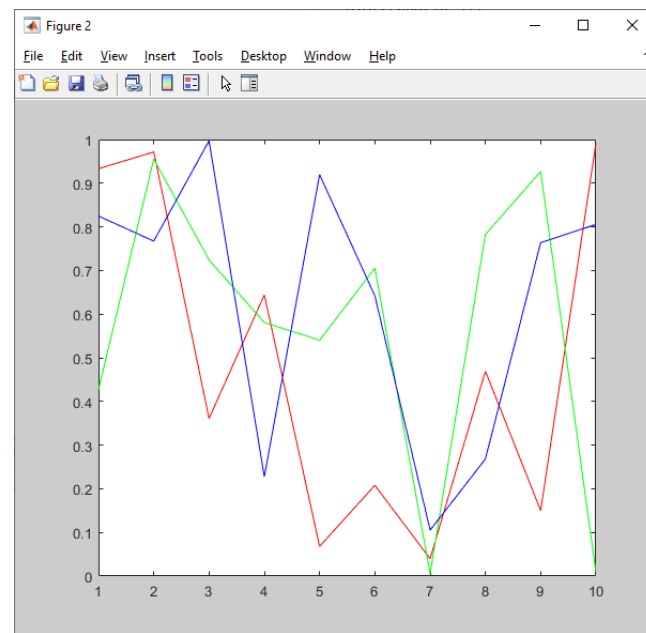
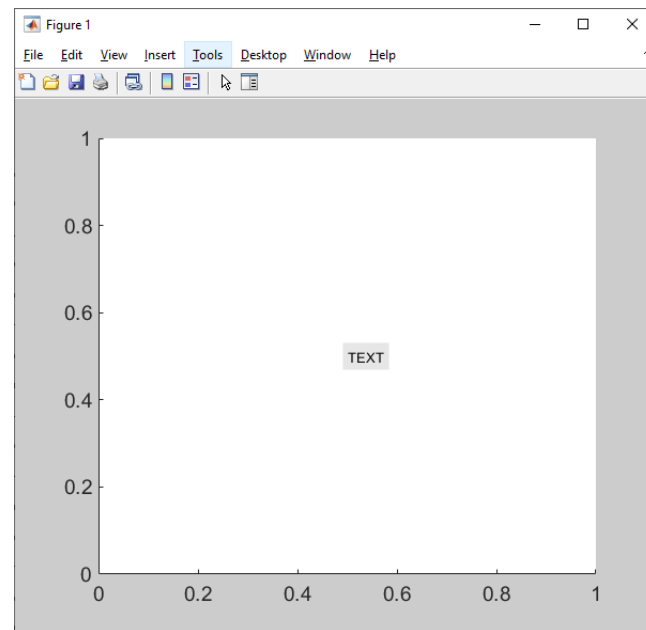
```
set(groot, 'defaultFigureColor',[.8 .8 .8])
figure

set(gcf, 'defaultAxesFontSize', 14)
ax = axes;

set(ax, 'defaultTextBackgroundColor', [.9 .9 .9])
text(.5, .5, 'TEXT')

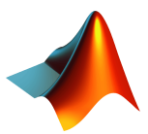
r = [1 0 0];
g = [0 1 0];
b = [0 0 1];
figure,
set(gcf, 'defaultAxesColorOrder',[r;g;b])
plot(rand(10,3))

set(gcf, 'defaultAxesColor', 'factory')
reset(groot)
```



객체 속성 설정 관련 팁

clf 또는 clf(h)	Figure 내 모든 Children 객체 지우기
cla 또는 cla(ax)	Axes 내 모든 Children 객체 지우기
delete(h)	객체 h 지우기 (괄호가 없으면 파일 삭제이므로 주의!)
findobj	현재 존재하는 모든 그래픽 객체 목록 특정 속성을 갖는 객체 검색 가능
startup.m	매트랩 실행할 때마다 자동으로 실행되는 파일 그래픽 객체 기본값을 여기에 적어둘 수 있음
그래픽 객체를 함수 인자로 전달 시	함수 안에서 객체 속성을 바꾸면 함수 밖에 적용된다! UserData를 이용하여 global 변수를 흉내낼 수도 있다.



예제 - 컬러 이미지의 RGB layer 시각화

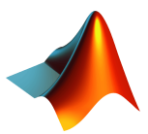
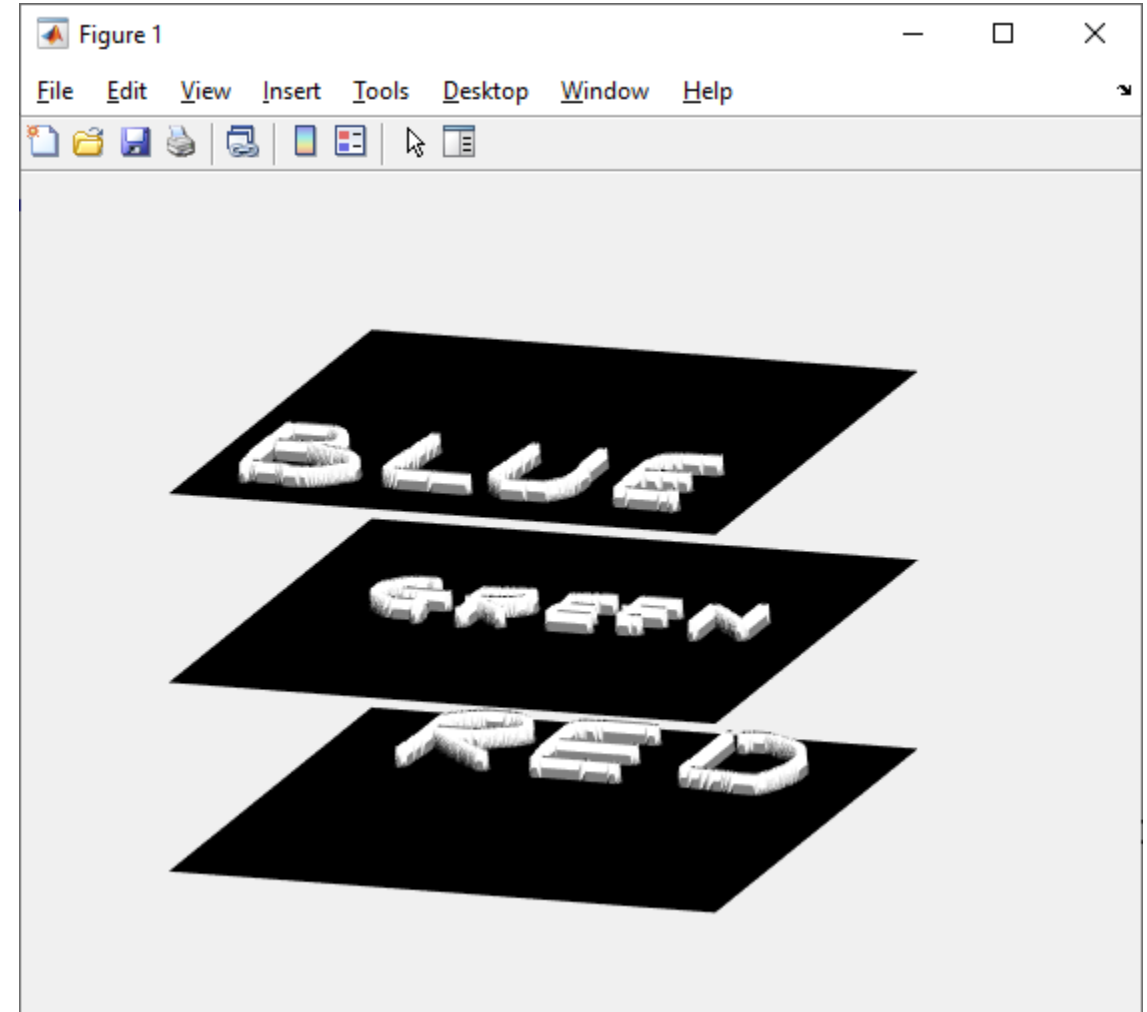
```
img = imread('RGB_test.png');

img_r = double(img(:,:,1));
img_g = double(img(:,:,2));
img_b = double(img(:,:,3));

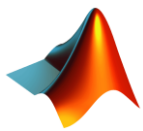
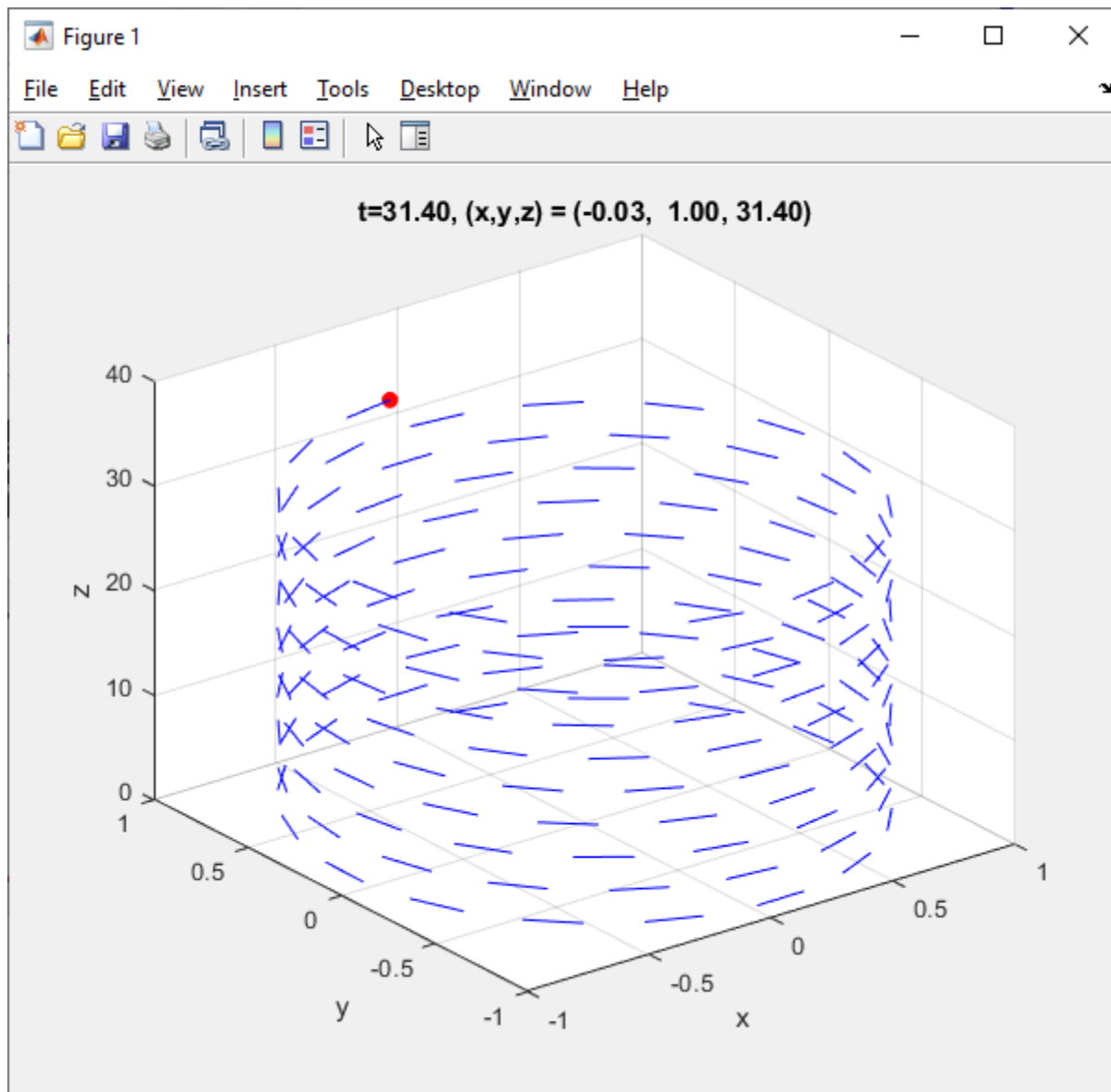
[xx,yy] = meshgrid(1:size(img,2),1:size(img,1));
yy = flipud(yy);

figure,

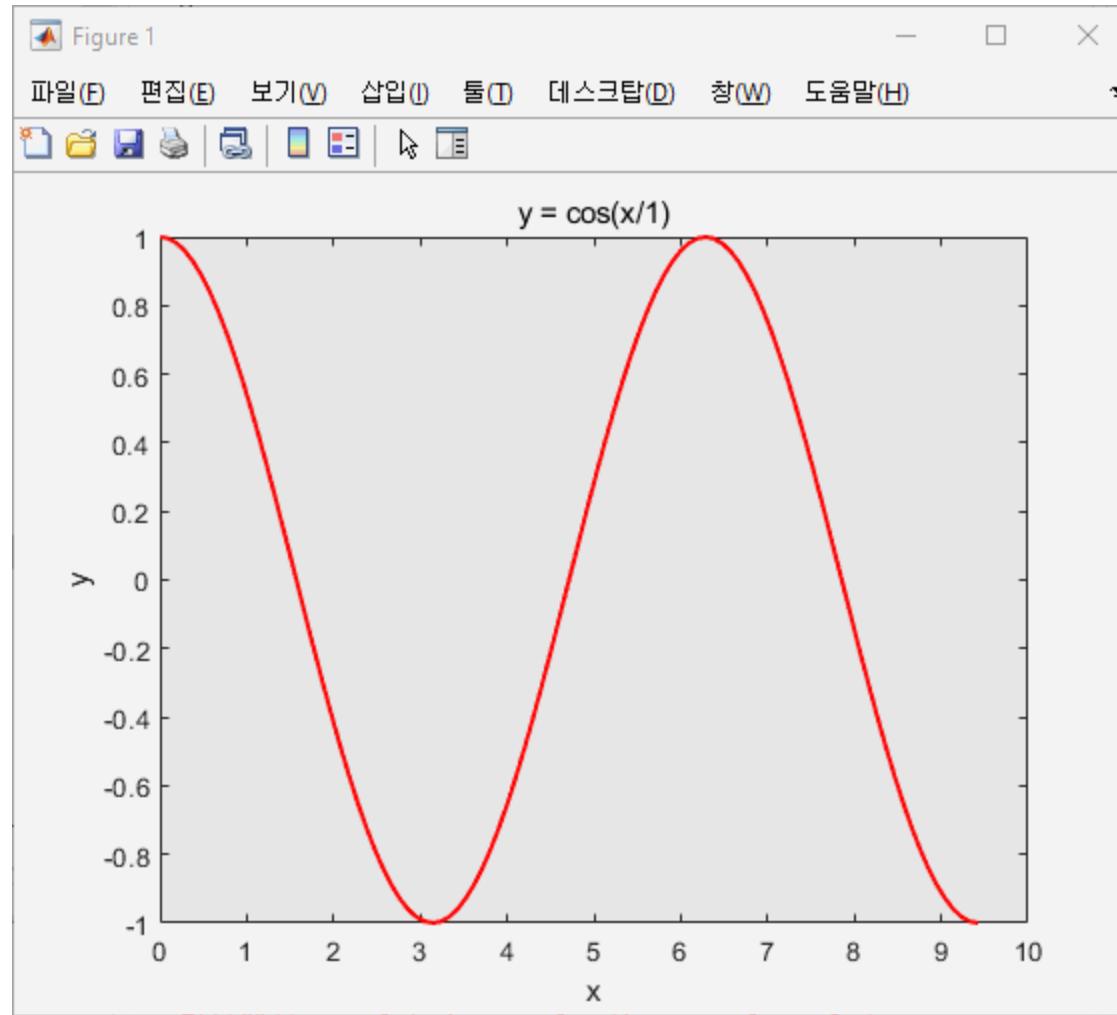
ax1 = axes; surf(xx,yy,img_r/255+ 0), shading flat,
axis off, view(17,20), colormap gray, zlim([0, 25])
ax2 = axes; surf(xx,yy,img_g/255+10), shading flat,
axis off, view(17,20), colormap gray, zlim([0, 25])
ax3 = axes; surf(xx,yy,img_b/255+20), shading flat,
axis off, view(17,20), colormap gray, zlim([0, 25])
```



예제 - 3D spiral



예제 - infinite cosine + gif 저장



<https://kr.mathworks.com/help/matlab/ref/getframe.html>

<https://kr.mathworks.com/help/matlab/ref/movie.html>

Q&A

