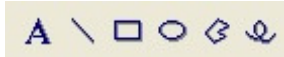


## Annotation

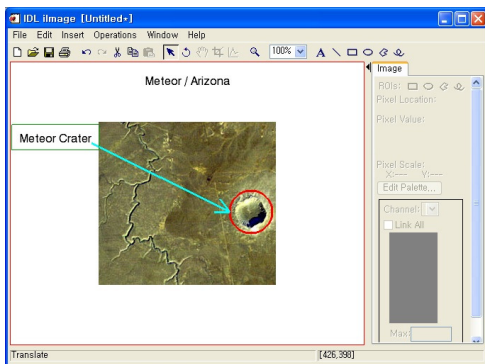


iTools의 Annotation 관련 버튼

iTools에서 그림과 같은 도구버튼을 통해 글자, 선, 다각형 등을 입력할 수 있었습니다. 이 기능을 그대로 프로그램을 통해 구현할 수 있는 프로시저가 다음과 같습니다.

- iText : 문자 Annotation 입력
- iPolyLine : 선(화살표 포함) Annotation 입력
- iPolygon : 다각형(사각형 포함) Annotation 입력
- iEllipse : 타원(원 포함) Annotation 입력

```
FILE=FILEPATH('meteo_crater.jpg', subdir=['examples', 'data'])
read_jpeg, file, img
iimage, img, dimension=[500, 400], view_title='Meteor/Arizona'
iEllipse, 30, 215, 103, /DATA, COLOR=[255, 0, 0], $
    FILL_BACKGROUND=0, THICK=3
iPolyLine, [[-10, 200],[185, 103]], /DATA, ARROW_STYLE=1,$
    ARROW_SIZE=0.1, THICK=3, COLOR=[0, 255, 255]
iText, 'Meteor Crater', -10, 200, /DATA, ALIGN=1.0
```



다이렉트 그래픽 체계에서, XYOUTS, ARROW 등으로 처리하던 일과 유사합니다. 각 프로시저의 자세한 사용법은 도움말을 활용하도록 하십시오.

## 화면 저장

iTools의 현재 화면을 읽어 JPEG, TIFF, PNG 등의 그래픽 파일로 저장할 수 있는 프로시저가 iSave입니다. iSave는 iTools의 저장 형식인 .isv 포맷으로도 저장하는 기능이 있습니다만 확장자를 그래픽 포맷으로 지정하면, 해당 그래픽 파일로 저장합니다.

```
iSave, 'C:\MyDIR\test1.jpg'
```

```
iSave, 'C:\MyDIR\test2.jpg', resolution=100
```

:RESOLUTION 키워드 값을 변경해 가며 저장되는 그림의 크기가 어떻게 되는지 확인해 보세요.

## iSave

:파일 지정이 없으면 앞에 사용한 파일에 덮어씁니다.

```
iSave, /SAVE_AS
```

:SAVE\_AS 키워드를 사용하면 파일 이름 대화창이 열립니다.

```
iSave, 'C:\MyDIR\test4.isv'
```

:.isv 파일은 iTools를 현재 상태로 저장하는 방식입니다.

:생성된 파일을 더블클릭하면 iTools가 그대로 복원됩니다.

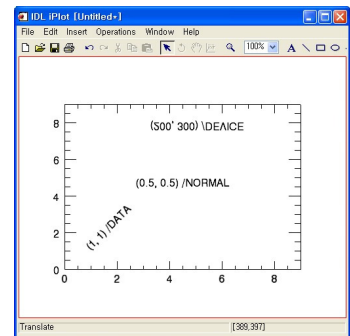
## 좌표계

iTools에서도 다이렉트 그래픽 체계와 같이, Normal, Device, Data 좌표계를 사용할 수 있습니다.

- Data : 표출된 데이터의 좌표(축의 눈금을 따름)
- Normal : 화면 크기에 무관. 0.0~1.0의 범위를 가짐
- Device : 좌측 하단부터 픽셀수.

```
iPlot, indgen(10), /SCATTER, dimension=[500, 400]
iText, '(1,1) /DATA', 1, 1, /DATA, ORIENTATION=45
iText, '(0.5,0.5) /NORMAL', 0.5, 0.5, /NORMAL, ALIGN=0.5
iText, '(200,300) /DEVICE', 200, 300, /DEVICE, UPDIR=[0.0,-1.0]
```

Orientation 키워드는 글자를 써나가는 방향(각도), ALIGN 키워드는 문자열을 정렬하는 위치(0.0은 좌측정렬, 0.5는 가운데 정렬, 1.0은 우측정렬), UPDIR 키워드는 글자를 뒤집는 방향을 의미합니다.



좌표계 간의 변환은

iConvertCoord() 함수를 이용하면 편리합니다.

```
print, iConvertCoord(200, 300, /DEVICE, /TO_DATA)
3.250000 8.0714286 0.000000
```

## 객체의 ID

iTool을 동시에 여러 개 생성할 경우도 있고, 하나의 iTool에 여러개의 구성요소를 올리는 경우도 있습니다. 마우스 컨트롤에서는 이들을 마우스로 잡으면 되지만, 프로그래밍으로 컨트롤할 때에는 각 구성요소의 고유 ID를 이용하여 지명합니다.

1) iTools나 구성요소를 생성할 때, ID를 받아 놓습니다.

```
iPlot, indgen(10), IDENTIFIER=id1
```

```
print, id1
```

```
/TOOLS/PLOT TOOL
```

```
iText, 'Center', 0.5,0.5, /NORMAL, IDENTIFIER=id2
```

```
print, id2
```

```

/TOOLS/PLOT TOOL/WINDOW/VIEW_1/ANNOTATION LAYER/TEXT
2) iGetCurrent() 함수로 현재 iTool의 ID를 받습니다.
iPlot, RANDOMU(SEED, 10)
idplot=iGETCURRENT()
3) iGetID() 함수로 원하는 ID를 찾아냅니다.
print, IGETID('TEXT')
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/ANNOTATION LAYER/TEXT
print, IGETID('PLOT')
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/PLOT
print, IGETID('DATA SPACE/*')
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/PLOT
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/PLOT/SYMBOL
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/AXES
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/AXES/AXISO
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/AXES/AXIS1
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/AXES/AXIS2
/TOOLS/PLOT TOOL/WINDOW/VIEW_1/VISUALIZATION LAYER/DATA SPACE/AXES/AXIS3

```

### iTools 선택 (Focusing)

다이렉트 그래픽의 WSET, WSHOW 역할을 하는 프로시저는 iSetCurrent입니다. 지정 ID가 가리키는 iTool로 포커스를 잡습니다. /SHOW 키워드를 사용하면 화면의 맨 앞으로 가져와 보여주는 일을 함께 수행합니다. ID를 지정하지 않고 iTools 관련 프로시저를 실행하게 되면 iSetCurrent로 지정한 iTool에 적용이 됩니다.

```
iSetCurrent, id1, /SHOW
```

### 회전/스케일링/이동

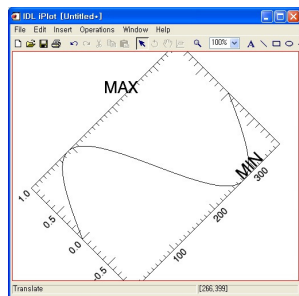
```

iSurface, dist(100), IDENTIFIER=id
for l=0, 35 do iRotate, id, 10, /YAXIS
iScale, id, 0.5, /Z
iScale, id, 2.0, /Z
iTranslate, id, 0.1, 0.2, 0.3, /NORMAL
iTranslate, id, -0.1, -0.2, -0.3, /NORMAL

```

### 데이터에 고정된 Annotation

객체의 ID절을 보면 VISUALIZATION LAYER와 ANNOTATION LAYER가 따로 존재하는 것을 알 수 있습니다. 이 경우 VISUALIZATION 레이어의 크기를 변경하거나 회전 또는 이동할 때, ANNOTATION이 함께 움직이지 않고 따로 놓게 됩니다. Annotation 관련 명령문을 실행할 때, /VISUALIZATION 키워드를 사용하면, Visualization 레이어에 올려져서 데이터의 변화에 함께 반응할 것입니다. 그림에서 MIN 은 MAX와 달리 그래



프를 따라 함께 회전하고 있습니다.

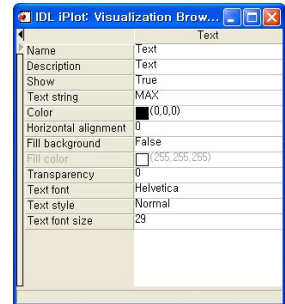
```

iPlot, sin(findgen(361)*!dtor), identifier=id1
iText, 'MAX', 90, 1.0, /DATA
iText, 'MIN', 270, -1.0, /DATA, /VISUALIZATION
iRotate, id1, 45

```

### 속성 항목 조사

iTools에서 마우스로 대상을 더블클릭하면 속성창이 열리며 사용자가 이를 변경할 수 있습니다. Text Annotation의 속성창은 그림과 같은데, 속성들의 항목을 프로그래밍에서 알아내는 방법은 iGetProperty와 \_REGISTERED 키워드를 이용하는 것입니다.



```

iText, 'MAX', 90, 1.0, /DATA,
IDENTIFIER=idtext
iGetProperty, idtext, _REGISTERED=props
print, props

```

```

NAME DESCRIPTION HIDE STRING COLOR ALIGNMENT
FILL_BACKGROUND FILL_COLOR TRANSPARENCY FONT_INDEX
FONT_STYLE FONT_SIZE

```

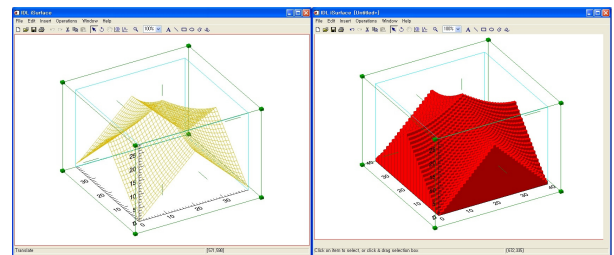
### 속성의 지정 및 변경

```

iSurface, dist(40), STYLE=1 ;Wire Mesh
id=iGetID('surface')
iGetProperty, id, color=c
print, c ; [225, 184, 0]
iSetProperty, id, color=[255, 0, 0], STYLE=6 ;RED&LegoFilled

```

처음부터 키워드를 이용하여 속성을 지정해 줄 수 있습니다. iGetProperty를 이용하여 속성값을 알아내고,



iSetProperty를 이용하면 속성값을 변경할 수 있습니다.

### iTools 제거

다이렉트 그래픽의 WDELETE와 유사한 개념으로 iDelete 문을 이용하여 지정 iTool을 닫습니다.

```

iPlot, indgen(10), IDENTIFIER=id1
iSurface, dist(10), IDENTIFIER=id2
iDelete, id1

```