

초급 활용

UI(User Interface)

- Rollout, Button, Spinner, Check Box 등과 같이 유저가 직접 조작하고 만질수 있는 인터페이스와 컨트롤.

```
rollout test "hello world~"
(
)
createdialog test
```



```
rollout test "hello world~"
(
    button test_btn '버튼'
)
createdialog test width:100 height:100 bgcolor:green
```



```
rollout test "hello world~"
(
    button test_btn '버튼' pos:[10,10] width:90 height:80
)
createdialog test width:120 height:100 bgcolor:green
```



Createdialog 이벤트 예제

CD\초급 활용판\Chapter_01\01_CreateDialog events.ms

```
rollout test "hello world~"
(
    label x_la "x: " pos:[70,50] -- label 넣기 (x)
    label y_la "y: " pos:[120,50] -- label 넣기 y)

    on test mousemove pos do -- 마우스를 움직였을 때 이벤트
    (
        x_la.text = "x: " + pos.x as string -- label의 text에 마우스 위치(x) 를 출력
        y_la.text = "y: " + pos.y as string -- label의 text에 마우스 위치(y) 를 출력
    )

    on test open do -- 대화상자의 open 이벤트
    (
        print "open~"
    )

    on test close do -- 대화상자의 close 이벤트
    (
        print "close~"
    )

    on test lbuttondown pos do -- 마우스 왼쪽 클릭 이벤트
    (
        messagebox "마우스 왼쪽 클릭"
    )

    on test mbuttondown pos do -- 마우스 가운데 버튼 클릭 이벤트
    (
        messagebox "마우스 가운데 클릭"
    )

    on test rbuttondown pos do -- 마우스 오른쪽 클릭 이벤트
    (
        messagebox "마우스 오른쪽 클릭"
    )
)
createdialog test width:200 height:150
```

예제 04

● Button 만들기

```
rollout test "hello world-"
{
    button my_btn "Button"

    on my_btn pressed do
    (
        messagebox "누르기"
    )

    on my_btn rightclick do
    (
        messagebox "오른쪽 버튼 누르기"
    )
}
createdialog test
```



실행 결과

예제 07

● spinner 사용하기

```
rollout test "hello world-"
(
    spinner sp1 "spinner입니다" range:[-50,100,5] scale:2 type:#integer

    on sp1 changed val do
    (
        format "changed %\n" val
    )
)
createdialog test
```



spinner

● 해당 예제는 부록 CD\초급 활용편\Chapter_1\07_spinner.ms 에 있습니다.

● Edittext의 사용하기

```
rollout test "hello world-"
(
    edittext et "Edittext"

    on et changed txt do          -- 한 글자씩 입력한 경우의 이벤트
    (
        print txt
    )

    on et entered txt do        -- enter를 입력한 경우의 이벤트
    (
        format "entered: %\n" txt
    )
)

createdialog test
```



edittext

예제 09

checkbox 사용하기

```
rollout test "hello world-"
(
    checkbox chk "checkboxbox 입니다" width:200

    on chk changed state do
    (
        format "checkboxbox %\n" state
    )
)

createdialog test width:200 height:70
```



checkboxbox

예제 10

• checkbox 사용하기

```
❶ rollout test "hello world-"
❷ (
❸     checkbox chkbtn "checkboxbutton" tooltip:"체크버튼 입니다" highlightColor:(color 255 0 255)
❹
❺     on chkbtn changed state do
❻     (
❼         format "checkboxbutton %\n" state
❽     )
❾
❿     on chkbtn rightclick do
⓫     (
⓫         chkbtn.text = "hi --" -- chkbtn의 텍스트를 바꾼다.
⓫     )
⓫ )
⓫ createdialog test height:50
```



checkboxbutton

예제 11

● Radiobutton 사용하기

```
rollout test "hello world-"
(
    radiobuttons rad1    "radiobuttons1 입니다" labels:#{ "a", "b", "c" }
    radiobuttons rad2    "radiobuttons2 입니다" labels:#{ "min", "max", "center" } default:3 pos:[40,50]

    on rad1 changed val do
    (
        format "rad1: %\n" val
    )

    on rad2 changed val do
    (
        format "rad2: %\n" val
    )
)

createdialog test height:100 width:200
```



실행 결과

예제 12

● slider 사용하기

```
rollout test "hello world-"  
(  
    slider sli "slider 입니다" range:[-100,100,0]  
  
    on sli changed val do  
    (  
        print val  
    )  
)  
  
createdialog test height:100 width:200
```

-- Slider 조절 이벤트



slider

● 해당 예제는 부록 CD\초급 활용편\Chapter_1\12_slider.ms 에 있습니다.

예제 13

● Listbox 사용하기

```
rollout test "hello world-"
(
    listbox li "listbox" items:#{ "one", "two", "three", "four", "five" }

    on li selected val do
        format "selected: %\n" val

    on li doubleClicked val do
        mgs = "doubleClicked: " + val as string
        messagebox mgs

createdialog test height:200 width:200
```



listbox

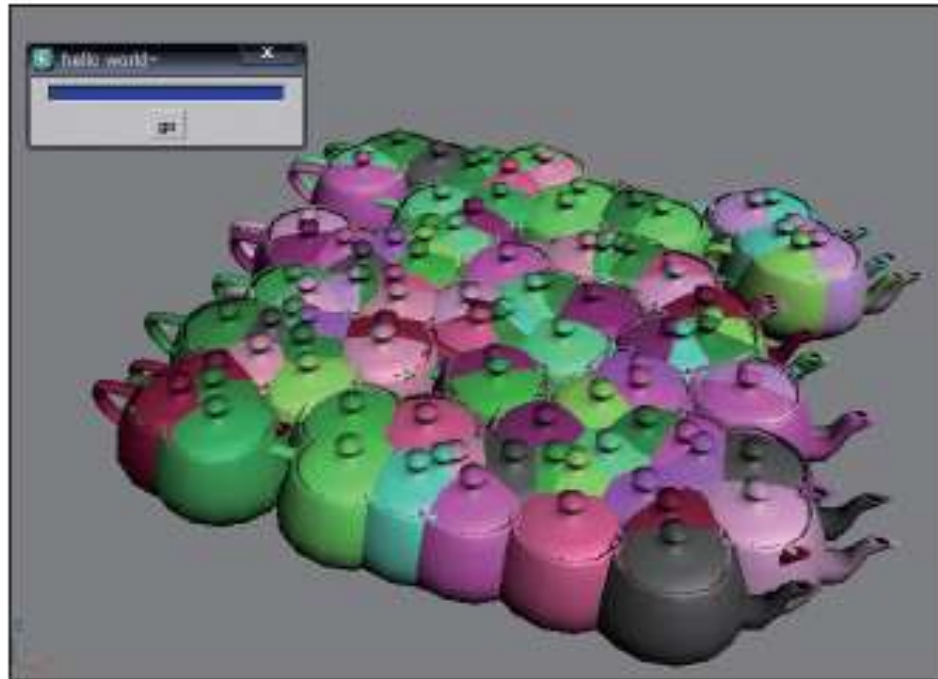
ProgressBar 사용하기

CD\초급 활용편\Chapter_1\15_progressbar.ms

```

1 rollout test "hello world~"
2 {
3     progressbar pbar                -- progressbar 생성
4     button go_btn "go"             -- 버튼 생성
5
6
7     on pbar clicked val do          -- progressbar를 클릭했을 때 발생 이벤트
8     (
9         format "clicked: %\n" val
10    )
11
12
13    on go_btn pressed do             -- 'go' 버튼을 누르면..
14    (
15        delete $temp*               -- temp란 이름을 가진 모든 오브젝트를 지운다.
16
17        -- 랜덤으로 주전자 100개 생성
18        for i in 1 to 100 do
19        (
20            rnd = random [-100,-100,0] [100,100,0]
21            tea = teapot pos:rnd name:("temp" + i as string)
22            pbar.value = i
23        )
24    )
25 }
26
27 createdialog test    width:200

```



progressBar

● Colorpicker의 사용 방법

```
colorpicker <name> [ <caption> ] [color:<color>] [alpha:<boolean>]  
[fieldWidth:<number>] [height:<number>] [modal:<boolean>] [title:<string>]
```

- [color:<color>] : color 값을 설정할 수 있습니다.
- [alpha:<boolean>] : 팔레트 대화상자에 alpha 값을 추가합니다.
- [fieldWidth:<number>] : 팔레트를 선택하는 color 박스의 넓이를 설정합니다.
- [height:<number>] : 높이를 설정합니다.
- [title:<string>] : 팔레트 대화상자의 title을 설정합니다.

● Colorpicker의 이벤트

```
on <colorpicker> changed <arg> do <expr>
```

Colorpicker의 color 값이 바뀌었을 때의 이벤트입니다.

예제 16

Colorpicker 사용하기

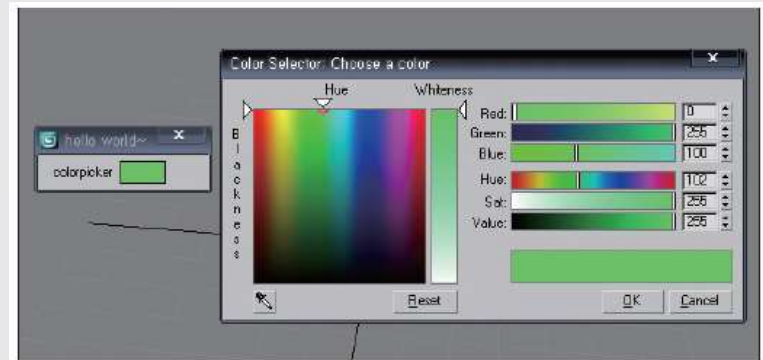
해당 예제는 부록 CD\초급 활용편\Chapter_1\16_colorpicker.ms 에 있습니다.

```
rollout test "hello world~"
(
    colorpicker cp "colorpicker" color:(color 0 255 100)

    on cp changed co do          -- colorpicker가 바뀌게 되면 발생하는 이벤트
    (
        format "color: %\n" co

        -- box01이 있으면 wirecolor를 바꾼다
        if $box01 != undefined do
            $box01.wirecolor = co
    )
)

createdialog test width:150
```



colorpicker

● Dropdownlist의 사용 방법

```
dropdownlist <name> [ <caption> ] [items:<array_of_strings>]  
[selection:<number>] [height:<number>]
```

- [items:<array_of_strings>] : String 배열로 된 아이টে을 넣습니다.
- [selection:<number>] : 초기 선택 넘버입니다.
- [height:<number>] : 리스트의 높이 값입니다. 픽셀 단위가 아니라 라인 단위입니다. 기본 값은 10으로 설정되어 있습니다.

● Dropdownlist의 이벤트

```
on <dropdownlist> selected <arg> do <expr>
```

List를 선택했을 때의 이벤트입니다.

예제 17

● Dropdownlist 사용하기

```
rollout test "hello world~"
(
    dropdownlist dd "dropdownlist" items:#{ "1", "2", "3", "4", "5", "6", "7" } height:5

    on dd selected val do          -- 오브젝트를 위해 Pick 했을 때 이벤트
    (
        format "dropdownlist:  %\n" val
    )
)

createdialog test    width:150
```



dropdownlist

● Bitmap의 사용 방법

```
bitmap <name> [ <caption> ] \  
[ fileName:<filename_string> | bitmap:<bitmap> ]
```

- [fileName:<filename_string> | bitmap:<bitmap>] : filename 또는 bitmap 넣는다.

● Bitmap 사용하기

```
rollout test "hello world~"  
(  
    local fname = "C:\\Program Files\\Autodesk\\3ds Max 2008\\maps\\Backgrounds\\deco.jpg"  
    bitmap map "Bitmap" filename:fname  
)  
  
createdialog test width:1070 height:768
```

● 해당 예제는 부록 CD\초급 활용편\Chapter_1\19_bitmap.ms 에 있습니다.

timer 사용 방법

```
timer <name> [ interval:<number> ] [ active:<boolean> ]
```

- [interval:<number>] : 얼마 간격으로 동작할 지 결정합니다. 기본 값은 1000이며 1000은 1초입니다.
- [active:<boolean>] : 타이머의 동작 여부를 설정할 수 있습니다.

timer의 이벤트

```
on <timer> tick do <expr>
```

일정 시간 간격으로 이벤트가 발생합니다.

● timer의 사용하기

이 예제는 대화상자가 만들어지면서 box가 "myBox"란 이름으로 만들어집니다.

그 다음에 1초 간격으로 box를 계속 Z축으로 이동합니다.

CD\초급 활용편\Chapter_1\20_timer.ms

```

1 rollout test "hello world-"
2 (
3     local count = 0
4     timer clock "timer" interval:1000    -- 1초 간격
5     label test "1"
6
7     on clock tick do                    -- 시간 간격으로 발생하는 이벤트
8     (
9         count +=1
10        test.text = count as string
11        $myBox.pos += [0,0,10]          -- mybox란 이름의 box를 이동한다.
12    )
13
14    on test open do                    -- rollout을 오픈했다면 발생하는 이벤트
15    (
16        box name:"myBox"
17    )
18
19 )
20
21 createdialog test

```

예제 21

● groupBox의 사용하기

```
❶ rollout test "hello world-"
❷ (
❸     spinner spin1 "Spin1: " pos:[30,25] width:100
❹     spinner spin2 "Spin2: " pos:[180,25] width:100
❺     groupBox group1 "Group 1" pos:[5,5] width:140 height:50
❻     groupBox group2 "Group 2" pos:[150,5] width:140 height:50
❼ )
❽
❾ createdialog test width:300
```



groupBox

○ UI control을 왼쪽 오른쪽 정렬하기

```
rollout test "hello world~"  
(  
    button btn1 "button1" align:#left  
    spinner sp1 "spinner1" width:70 align:#right  
)  
  
createdialog test width:200 height:100
```



control의 정렬

across를 사용하면 지정한 숫자만큼 control item을 한 줄에 넣을 수 있습니다.

```
rollout test "hello world~"  
(  
    button btn1 "button1"    across:3  
    button btn2 "button2"  
    button btn3 "button3"  
)  
  
createdialog test  width:200 height:100
```



across : 1



across : 2



across : 3

controls를 숨기거나 보이게 하기

```
1 rollout test "hello world~"
2 (
3     local toggle = false
4
5     groupbox gb "" width:170 height:80
6     button btn1 "button1" pos:[30,30]
7     checkbox chk1 "checkbox1" pos:[100,30]
8     button btn2 "visible" pos:[60,100] width:80
9
10
11 on btn2 pressed do
12 (
13     -- btn2를 누르면 숨고 보이는 것이 토글된다.
14     if toggle == false then
15 (
16         -- btn1과 chk1를 숨긴다.
17         btn1.visible = false
18         chk1.visible = false
19
20         toggle = true                -- toggle 변수는 true
21     )
22 else
```



```

23         (
24             -- btn1과 chk1를 보이게 한다
25             btn1.visible = true
26             chk1.visible = true
27
28             toggle = false
29         )
30     )
31 )
32
33 createdialog test width:200 height:130

```



visible의 true



visible의 false

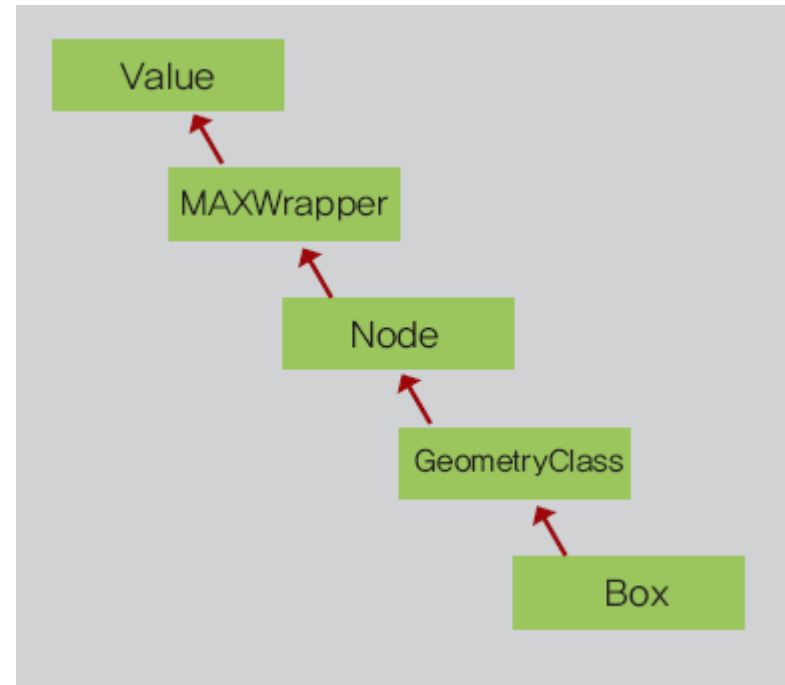
Object 다루기

오브젝트를 다른 용어로 Node라고 말합니다.

Node는 정확히 설명하면 Node로부터 상속 받은 Class를 말합니다.

Node의 서브 class는 다음과 같습니다.

- GeometryClass
- Shape
- Light
- Camera
- Helper
- SpacewarpObject
- System



```
b=box()          -- box를 변수에 인스턴스로 대입
```

```
b=box width:10 length:100      -- width, length를 붙인 경우
```

```
classof b          -- class 알아내기
```

```
Box                -- 출력 결과
```

```
classof (classof b)  -- 부모 class 알아내기 (superclassof 를 사용해도 된다)
```

```
GeometryClass      -- 출력 결과
```

```
classof (classof (classof (classof (classof b))))  -- 최상위 부모 class 알아내기
```

```
Value              -- 출력 결과
```

```
showProperties b    -- 속성 출력
```

```
.height : float      -- 출력 결과
```

```
.length : float
```

```
.lengthsegs : integer
```

```
.width : float
```

```
.widthsegs : integer
```

```
.mapcoords : boolean
```

```
.heightsegs : integer
```

```
.realWorldMapSize : boolean
```

```
false
```

```
getPropNames b          -- 속성(Properties) 출력
```

```
-- 출력 결과(배열로 리턴)
```

```
#[#height, #length, #lengthsegs, #width, #widthsegs, #mapcoords, #heightsegs,  
#realWorldMapSize)
```

```
b.height                -- height 속성 액세스
```

```
25.0                    -- 출력 결과
```

```
b.height = 10           -- height 속성 변경
```

```
getProperty b "height"  -- height 속성 알아내기
```

```
10.0                    -- 출력 결과
```

```
setProperty b "height" 15 -- height 속성 설정
```

```
hide <node> -- mapped
```

오브젝트를 Hide합니다.

```
unhide <node> [doLayer:<boolean>] -- mapped
```

```
freeze <node> -- mapped
```

```
unfreeze <node> [doLayer:<boolean>] -- mapped
```

```
deselect <node> -- mapped
```

오브젝트의 선택을 해제합니다.

```
select <node> -- mapped
```

오브젝트를 선택합니다.

```
sphere pos: [-100,0,0] name: "mySphere01"
```

```
sphere pos: [0,0,0] name: "mySphere02"
```

```
sphere pos: [100,0,0] name: "mySphere03"
```

```
dummy pos: [-100,100,0] boxsize: [20,20,20] name: "myDummy01"
```

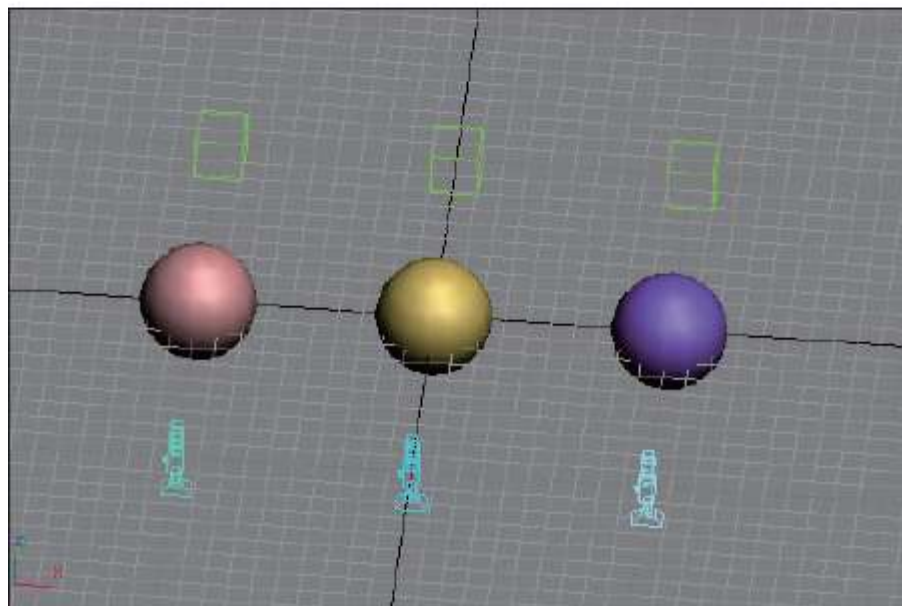
```
dummy pos: [0,100,0] boxsize: [20,20,20] name: "myDummy02"
```

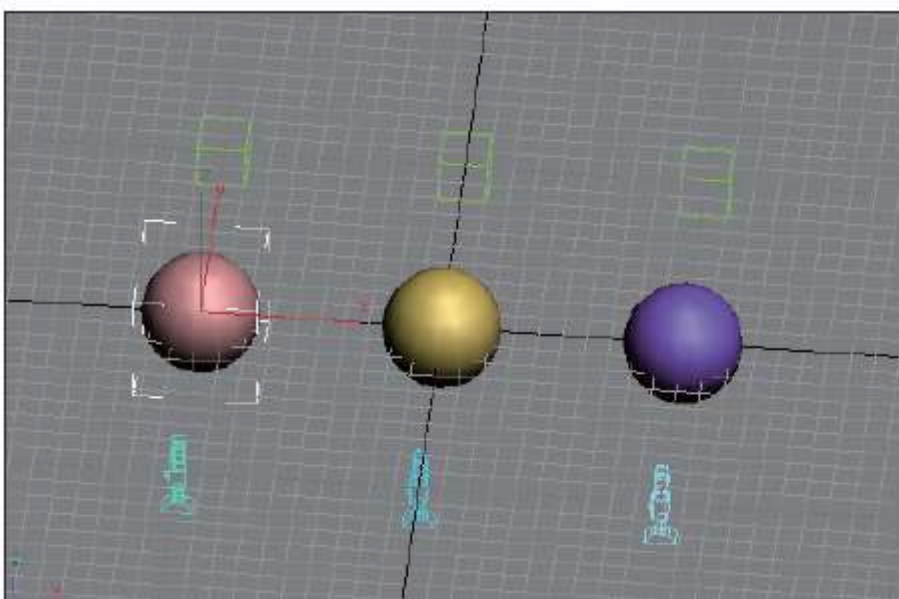
```
dummy pos: [100,100,0] boxsize: [20,20,20] name: "myDummy03"
```

```
freecamera pos: [-100,-100,0] name: "myCamera01"
```

```
freecamera pos: [0,-100,0] name: "myCamera02"
```

```
freecamera pos: [100,-100,0] name: "myCamera03"
```





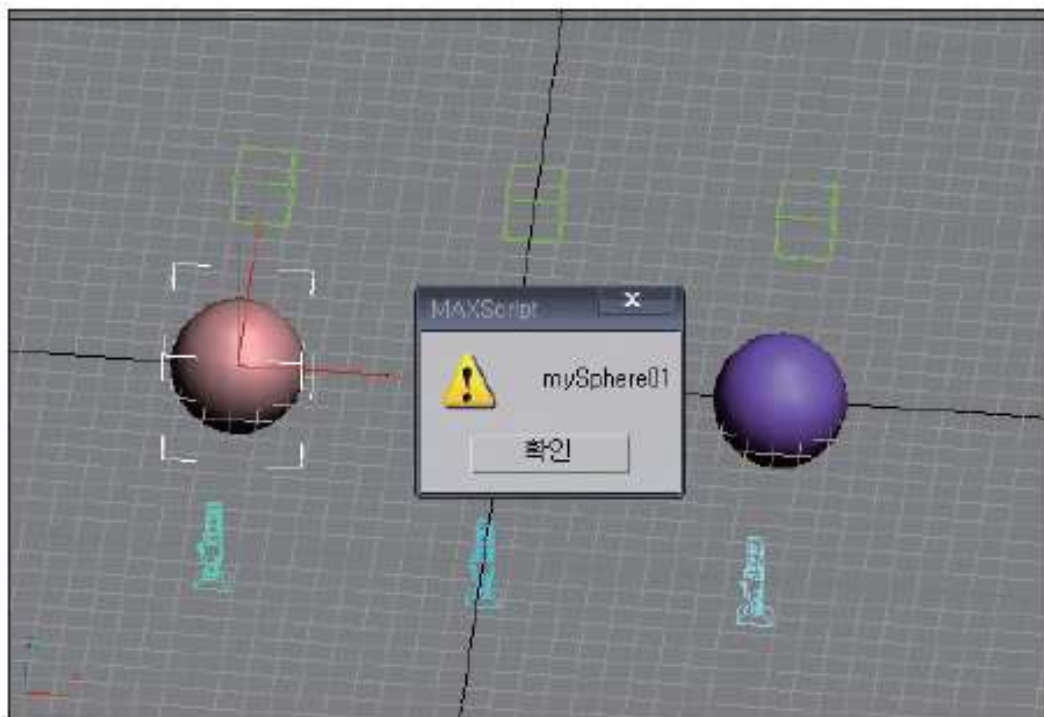
mysphere01을 이름으로 선택하려면 이름 앞에 \$를 붙여서 사용하면 됩니다.

```
Select $mysphere01
```

만약 오브젝트 이름이 "the sphere" 처럼 중간에 공백이 있다면 \$ '...' 사용하면 됩니다.

```
Delete $'the sphere'  
Select $'the sphere'
```

현재의 선택한 오브젝트를 알아내고 싶다면 단독으로 \$ 만 사용하면 됩니다.



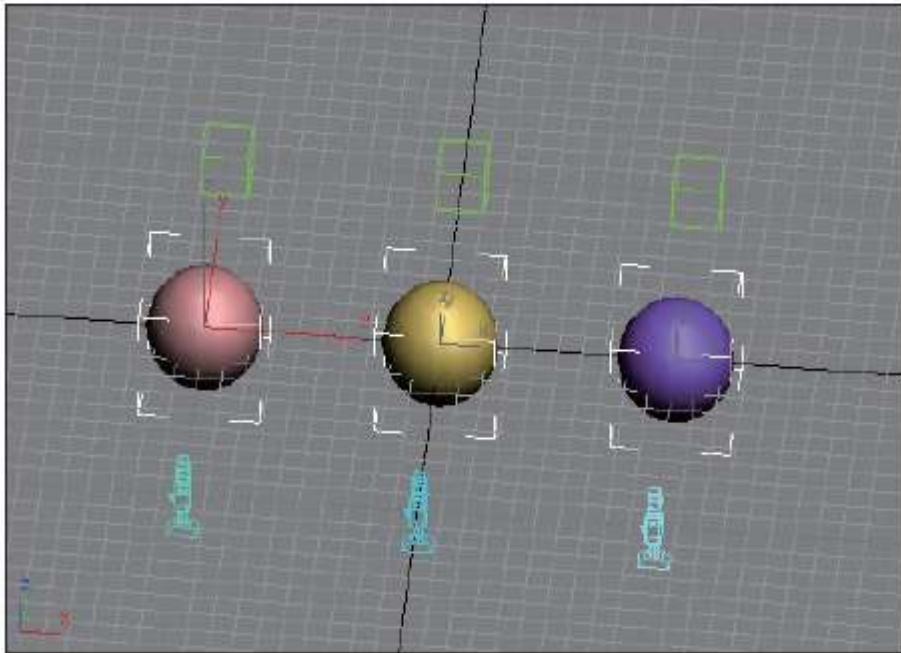
```
n = $.name  
MessageBox n
```

만약 동시에 3개를 선택하려면 배열을 사용하면 됩니다.

```
select #($mySphere01, $mySphere02, $mySphere03)
```


현재의 상황에서 다음의 예제는 모두 동일한 결과를 얻어냅니다.

<code>select \$mySphere*</code>	-- 앞 글자가 mySphere로 된 오브젝트를 전부 선택
<code>select \$*Sphere*</code>	-- 중간에 Sphere란 이름을 가진 오브젝트 전부 선택
<code>select \$mySphere??</code>	-- 앞 글자가 mySphere로 되어 있고 뒤에 두 자리를 모를 때 전부 선택

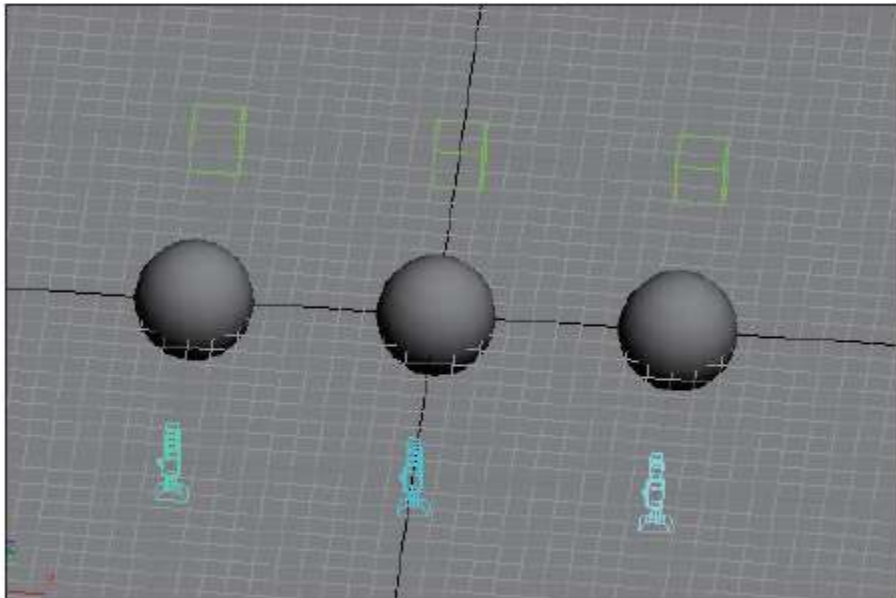


ObjectSet은 특정 class만 이용할 수 있습니다.

```
select objects          -- 오브젝트 전체를 선택
select helpers          -- helpers 선택
select cameras          -- cameras 선택
```

Freeze/unfreeze 또한 ObjectSet에 반응합니다.

```
freeze $mySphere*       -- mySphere*를 freeze 한다
freeze cameras          -- 모든 카메라를 freeze 한다
unfreeze $myCamera*     -- myCamera*를 unfreeze 한다
```



ObjectSet를 배열로 변환할 수도 있습니다.

```
objs = objects as array
```

-- ObjectSet를 배열로 변환

```
str = ""
```

-- str 변수를 string으로 초기화

```
for i in objs do
```

```
(
```

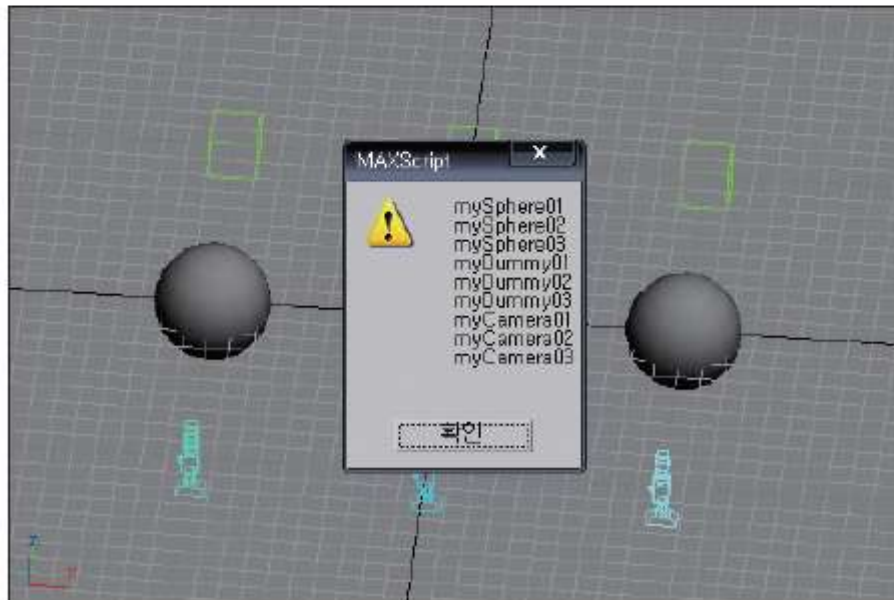
```
    str += i.name + "\n"
```

-- str에 차곡차곡 넣는다

```
)
```

```
messagebox str
```

-- messagebox에 출력



point2와 point3(벡터)

point2와 point3는 3ds Max에서 매우 자주 사용되는 값으로 [5,0] [10,20,30]처럼 표현합니다.
그 중 point3는 오브젝트 이동에 사용됩니다.

```
tea = teapot pos:[100,0,0]      -- 주전자를 [100,0,0]에 생성
tea.pos = [50,0,10]            -- 주전자를 [50,0,10]으로 이동
tea.pos.x = 30                  -- 주전자의 x 위치를 30으로 이동
```

● point3의 사용 방법

```
point3 <x> <y> <z>
<color> as point3
```

```
point3 0 10 0
red as point3
```

```
[0,10,0]      -- 출력 결과
[255,0,0]
```

point3는 다음과 같이 덧셈, 뺄셈, 곱셈, 나눗셈이 가능합니다.

```
<point3> == <point3>
<point3> != <point3>

<point3> + <point3_or_number>
<point3> - <point3_or_number>
<point3> * <point3_or_number>
<point3> / <point3_or_number>
```

```
[10,10,20] + [20,30,50]
[20,50,5] * 3
[100,0,-100] == [100,0,-100]
```

-- 벡터끼리의 덧셈

-- 벡터에 스칼라(scalar) 곱

```
[30,40,70]
```

-- 출력 결과

```
[60,150,15]
```

```
true
```

copy <point3>

벡터를 복사한다.

length <point3>

벡터의 크기를 알아낸다.

distance <point3> <point3>

2개의 벡터에 거리를 구한다.

random <point3> <point3>

랜덤으로 벡터를 구한다.

다음은 두 벡터의 거리 값을 구하는 예제입니다.

```
a = [10,20,30]
b = [100,30.5,41.3]
distance a b  -- 결과 91.3123
distance b a  -- 결과 91.3123
length (b-a)  -- 결과 91.3123
length (a-b)  -- 결과 91.3123
```

Move / Rotate / Scale

```
move <node> <point3> -- mapped
```

오브젝트를 이동합니다.

```
scale <node> <point3> -- mapped
```

오브젝트의 스케일을 변형합니다.

```
rotate <node> <angle> <axis_point3> -- mapped -- angle in degrees
```

물체의 이동.

```
move <node> <point3>
```

예제 01

오브젝트를 World축으로 이동하는 가장 쉬운 방법입니다.

```
sp = sphere pos:[0,0,0]      -- sphere를 [0,0,0] 위치에 생성
sp.pos = [0,0,100]          -- sp에 위치를[0,0,100]으로 옮긴다.
```

다음은 현재의 위치에서 일정한 위치만큼 상대적으로(relative) 증가하는 방법입니다.

```
sp = sphere pos:[100,0,0]
sp.pos += [100,0,0]          -- sp에 위치를 [100,0,0] 만큼 증가한다.
```

```
sp = sphere pos:[100,0,0]
move sp [100,0,0]            -- sp에 위치를 [100,0,0] 만큼 증가한다.
```

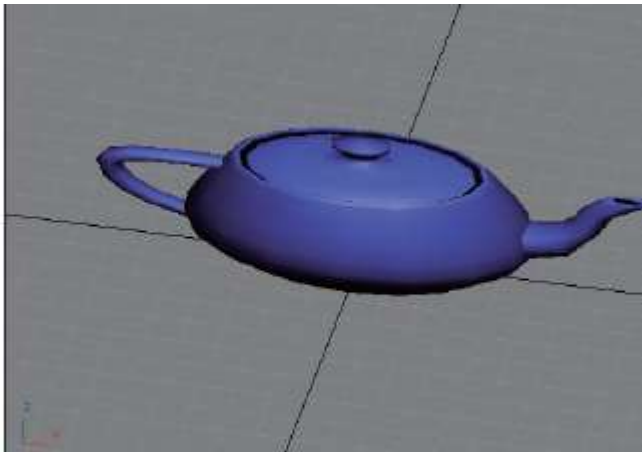

예제 04

CD\초급 활용 편\Chapter_3\04_rotate.ms

다음 예제를 실행해 봅시다.

```
t = teapot()           -- 주전자 생성
t.rotation = angleaxis 90 [1,0,0]  -- 주전자를 x축으로 90도 돌린다
```

```
t = teapot()  
t.scale = [2,1,1]    -- x축으로 2배 늘린다.
```



```
t = teapot()  
scale t [1,1,2]    -- z축으로 2배 늘린다.
```



Animation

다음과 같은 절차를 Script로 만들어 보겠습니다.

1. 주전자를 [0,0,0]에 만든다.
2. 애니메이션 버튼을 켜다.
3. 슬라이더를 10frame으로 이동한다.
4. 주전자를 [100,0,0]으로 이동한다.
5. 슬라이더를 20frame으로 이동한다.
6. 주전자를 [100,100,0]으로 이동한다.
7. 슬라이더를 30frame으로 이동한다.
8. 주전자를 [0,0,0]으로 이동한다.
9. 애니메이션 버튼을 끈다.

```
tea = teapot()           -- 주전자를 [0,0,0]에 생성
animButtonState = true   -- ani 버튼을 켜다
sliderTime = 10f         -- slider를 10f로 이동
tea.pos = [100,0,0]      -- 주전자를 [100,0,0]로 이동
sliderTime = 20f
tea.pos = [100,100,0]
sliderTime = 30f
tea.pos = [0,0,0]
animButtonState = false  -- ani 버튼을 끈다
```

```
tea = teapot()
animate on
(
    at time 10 tea.pos = [100,0,0]
    at time 20 tea.pos = [100,100,0]
    at time 30 tea.pos = [0,0,0]
)
playAnimation()           -- play Animation 버튼 누르기
```

예제 08

box가 원형으로 애니메이션이 되는 예제입니다.

```
mybox = box()
radius = 50
num = 101
step = 360 / num as float

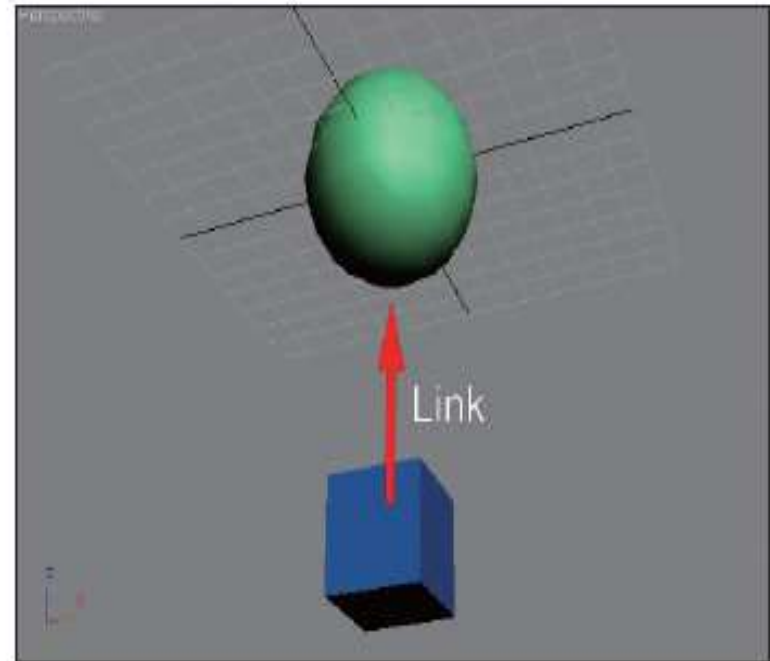
for i in 1 to num do
(
    -- circle 공식 적용
    x = radius * sin(i * step)
    y = radius * cos(i* step)
    with animate on
    (
        at time (i-1) mybox.pos =[x,y,0]
    )
)
-- 0 부터 100까지 애니메이션 한다.
```

CD\초급 활용 편\Chapter_3\08_animate.ms

Link

다음은 box를 sphere에 링크하는 예제입니다.

```
o1 = sphere()  
o2= box pos:[0,0,-100]  
o2.parent = o1    -- box를 sphere에 링크시킨다.
```



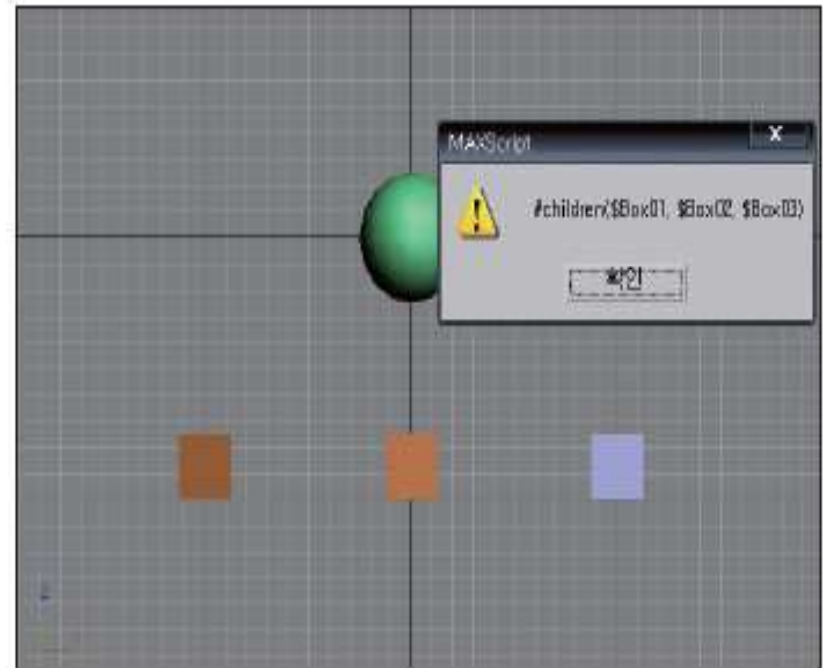
link 하기

부모 오브젝트는 하나이지만 자식 오브젝트는 여러 개가 될 수 있기 때문에 배열 형태 (NodeChildrenArray)로 나오게 됩니다.

```
o1 = sphere ()
o2= box pos: [0,0,-100]
o3= box pos: [-100,0,-100]
o4= box pos: [100,0,-100]

-- box들을 sphere에 링크한다.
o2.parent = o1
o3.parent = o1
o4.parent = o1

-- 자식 오브젝트들을 string으로 변환해서 messagebox로 출력한다.
messagebox (o1.children as string)
```



자식 오브젝트들

오브젝트들을 attach 하지 않고 한꺼번에 선택하려면 group 단위로 묶는 것이 편합니다.
Script에서 group 하려면 다음과 같이 처리하면 됩니다.

```
-- box를 11개 만들고 배열에 모은다.
boxs = for i in 0 to 10 collect
(
    s = sin (i*36)*50
    box pos:[i*50,s,0]          -- box를 sin곡선으로 배치
)

group boxs name:"myGroup"      -- group으로 만든다.
```

ungroup 또한 간단합니다.

```
ungroup $myGroup              -- 현재 level의 그룹을 ungroup 한다
explodeGroup $myGroup         -- 모든 level의 그룹을 ungroup한다(그룹안에 그룹이 있는 경우).
```


복사

```
t = teapot()           -- 주전자 생성
myCopy = copy t        -- copy한다.
myReference = reference t -- reference한다.
myInstance = instance t -- instance한다.
```

```
b = box()
```

```
boxArray = #()
```

```
for i in 1 to 10 do
```

```
(
```

```
    rcolor = random white black
```

```
-- random으로 color를 결정
```

```
    for j in 1 to 10 do
```

```
    (
```

```
        ib = instance b
```

```
-- instance로 복사
```

```
        ib.pos = [i*50, j*50, 0]
```

```
-- 오브젝트 위치를 변경
```

```
        ib.wirecolor = rcolor
```

```
-- wirecolor 줄단위로 입힌다.
```

```
        append boxArray ib
```

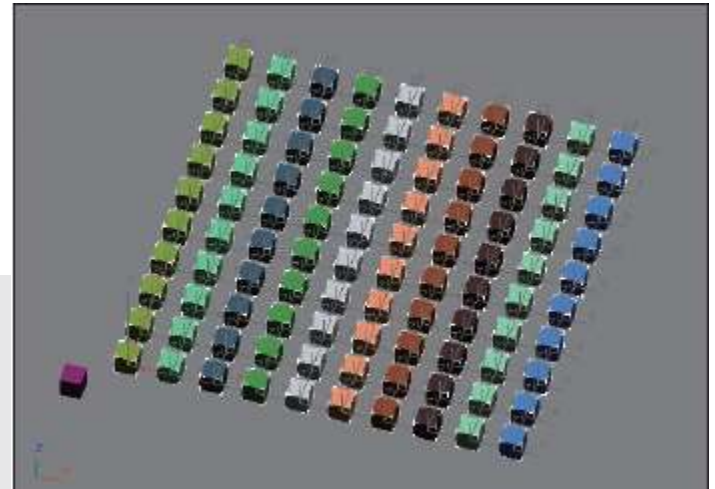
```
-- box를 boxArray에 모은다.
```

```
    )
```

```
)
```

```
select boxArray
```

```
-- 모은 오브젝트들을 선택
```



Mytool

```
❶ rollout batch_rename "batch rename"
❷ (
❸     edittext br_et ""
❹     button rename_btn "rename"
❺
❻ on rename_btn pressed do                -- 일괄적으로 rename 한다.
❼     (
❽         count = 0
❾         undo on
❿             for i in selection do        -- 선택한 오브젝트만 적용한다
⓫             (
⓫             count += 1
⓫             na = formattedprint count format:"02d" -- 숫자를 01, 02, 03 이런식으로 붙인다
⓫             i.name = br_et.text + na as string
⓫             )
⓫         )
⓫     )
⓫ )
```

해당 예제는 부록 CD\초급 활용편\Chapter_3\myTool.ms 에 있습니다.



```

18 rollout transform_rollout "Transform"
19 (
20     button zero_btn "[0,0,0]"
21     button rotate1_btn "rotate 90" pos:[10,40]
22     button rotate2_btn "rotate -90" pos:[70,40]
23
24     on zero_btn pressed do -- 오브젝트 위치를 [0,0,0]으로
25     (
26         undo on
27         for i in selection do
28             (
29                 i.pos = [0,0,0]
30             )
31         )
32
33     on rotate1_btn pressed do -- 90도 회전
34     (
35         undo on
36         rotate selection (EulerAngles 0 0 90)
37     )
38
39     on rotate2_btn pressed do -- 반대로 90도 회전
40     (
41         undo on
42         rotate selection (EulerAngles 0 0 -90)
43     )
44 )
45 )
46

```



```
47 rollout pivot_rollout "Pivot"
```

```
48 (
```

```
49     button zero_btn "[0,0,0]" pos:[20,20]
```

```
50     button center_btn "center" pos:[70,20]
```

```
51
```

```
52     on zero_btn pressed do                                -- pivot의 위치를 [0,0,0]으로
```

```
53     (
```

```
54         for i in selection do
```

```
55             i.pivot = [0,0,0]
```

```
56     )
```

```
57
```

```
58     on center_btn pressed do                            -- pivot의 위치를 center로
```

```
59     (
```

```
60         for i in selection do
```

```
61             i.pivot = i.center
```

```
62     )
```

```
63 )
```

```
64
```



```

65 rollout wirecolor_rollout "wirecolor"
66 (
67     button red_btn "RED"    pos:[5,20]    width:30
68     button green_btn "GREEN" pos:[40,20]    width:50
69     button blue_btn "BLUE"   pos:[95,20]    width:35
70
71     on red_btn pressed do      -- wirecolor를 red로
72     (
73         for i in selection do
74             i.wirecolor = red
75     )
76
77     on green_btn pressed do    -- wirecolor를 green으로
78     (
79         for i in selection do
80             i.wirecolor = green
81     )
82
83     on blue_btn pressed do     -- wirecolor를 blue로
84     (
85         for i in selection do
86             i.wirecolor = blue
87     )

```



```

90 rollout properties_rollout "Properties"
91 (
92     checkbox renderable_cb "Renderable"
93     checkbox castShadow_cb "Cast Shadows"
94
95     on renderable_cb changed state do          -- renderable 토글
96     (
97         for i in selection do
98             i.renderable = state
99         )
100
101     on castShadow_cb changed state do          -- cast shadow 토글
102     (
103         for i in selection do
104             i.castShadows = state
105         )
106     )
107

```



```

108 rollout hide_rollout "hide/freeze"
109 (
110     button hide_btn "hide" pos:[10,10] width:50
111     button unhide_btn "unhide All" pos:[65,10]
112     button freeze_btn "freeze" pos:[10,40] width:50
113     button unfreeze_btn "unfreeze All" pos:[65,40]
114
115     on hide_btn pressed do -- hide 한다.
116     (
117         for i in selection do
118             hide i
119     )
120
121     on unhide_btn pressed do -- unhide all 한다.
122     (
123         max unhide all
124     )
125
126     on freeze_btn pressed do -- freeze 한다.
127     (
128         for i in selection do
129             freeze i
130
131         clearselection() -- 선택을 해제한다.
132     )
133
134     on unfreeze_btn pressed do -- unfreeze all 한다.
135     (
136         max unfreeze all
137     )
138 )

```



```

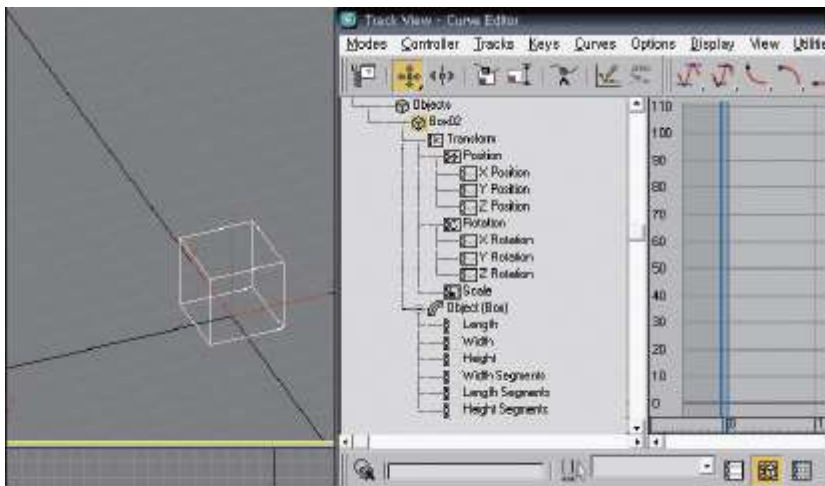
14 rollout boundDummy_rollout "Bound Dummy"
15 (
16     button dummy_btn "dummy"
17
18     fn makeDummy obj =
19     (
20         size = obj.max - obj.min      -- 사이즈를 구한다.
21         dummy boxsize:size pos:obj.pos -- dummy 생성
22     )
23
24     on dummy_btn pressed do           -- 바운딩 dummy 만들기
25     (
26         for i in selection do
27             makeDummy i
28     )
29 )
30
31 main = newRolloutFloater "my tools" 150 550 -- newRolloutFloater를 만든다.
32 addRollout batch_rename main           -- rollout들을 추가한다.
33 addRollout transform_rollout main
34 addRollout pivot_rollout main
35 addRollout wirecolor_rollout main
36 addRollout properties_rollout main
37 addRollout hide_rollout main
38 addRollout boundDummy_rollout main

```



Animation Control

```
b = box()
select b
max treeview                                     -- track view를 연다.
macros.run "Track View" "LaunchFCurveEditor"     -- track view를 연다(위와 같은 명령).
```



track view 안에서 controller를 확인할 수 있다.

```

source_obj = box()                -- 박스 생성
target_obj = sphere()            -- 스피어 생성

with animate on                  -- 박스의 애니메이션 (0, 25, 50)
(
    at time 25 source_obj.pos.controller.y_position = 100
    at time 50 source_obj.pos.controller.y_position= 200
)

with animate on                  -- 스피어의 애니메이션 (0, 10, 20)
(
    at time 10 target_obj.pos.controller.x_position = 60
    at time 20 target_obj.pos.controller.x_position = 100
)

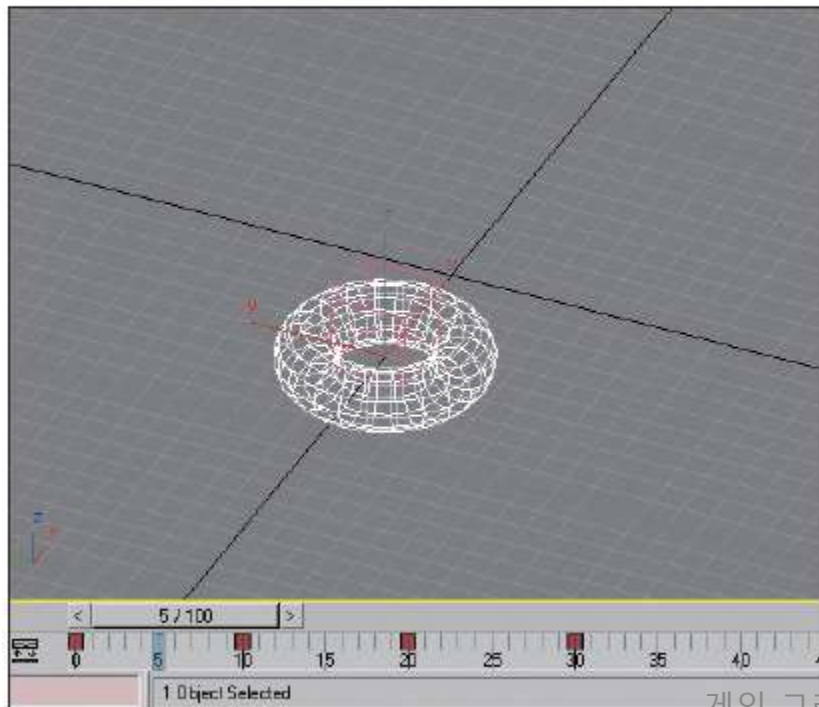
```

```

b = box()
animate on
(
    at time 10 move b [-100,0,0]
    at time 20 move b [0,100,0]
    at time 30 move b [0,0,100]
)
tor = torus name:"torus01"
tor.position.controller = b.position.controller

```

-- instance로 복사



controller의 instance

게임 그래픽스 & 애니메이션

만약 copy로 복사하고 싶다면 간단하게 다음과 같이 하면 됩니다.

```
delete $torus01          -- torus를 지운다.  
tor = torus name:"torus01"  -- torus를 다시 생성  
tor.position.controller = copy b.position.controller  -- copy로 복사
```

controller의 keys를 편집하려면 keys의 정보를 알아야 합니다.

Keys는 controller.keys 속성으로 금방 알아낼 수 있습니다.

앞에서 생성한 torus가 남아 있으니 박스만 남겨 두고 지우겠습니다.

```
delete $torus01  
b.position.controller.keys  -- keys  
-- #keys(0f, 10f, 20f, 30f)  -- 출력 결과
```

controller에는 keys를 직접 추가하고 지우고 이동하는 함수가 있습니다.

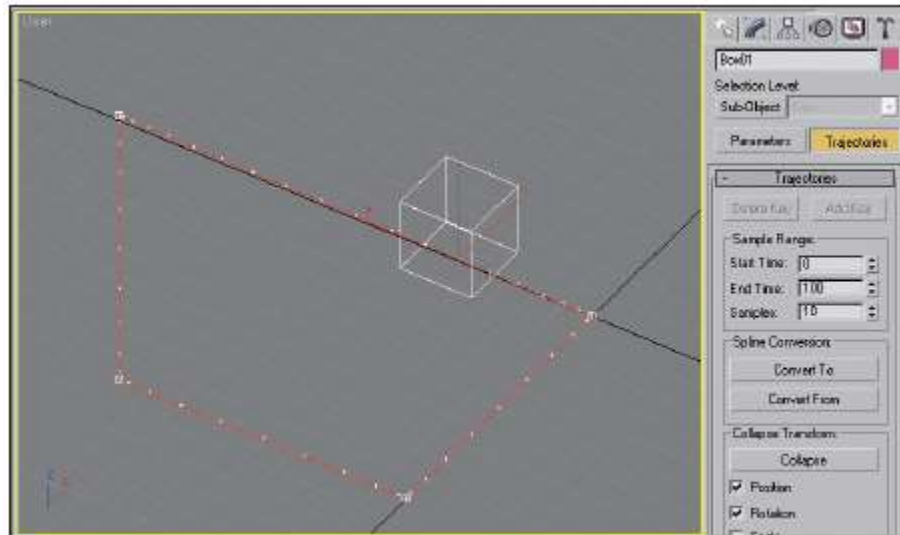
addNewkey는 keys를 추가하는 함수입니다.

```
con = b.position.controller
```

-- position controller를 con 변수에 저장

```
addNewkey con 50f
```

-- 50 프레임에 키를 추가한다.



50프레임에 key를 추가했지만 value 값을 넣지 않아 애니메이션은 그대로입니다. 다음과 같이 입력해 봅시다.

```
gkey = getkey con 5
```

```
gkey.value = [0,0,0]
```

```
interval <start_time> <end_time>
```

controller에 시간을 늘리거나 줄이는 과정을 Script로 자유자재로 제어할 수 있습니다.

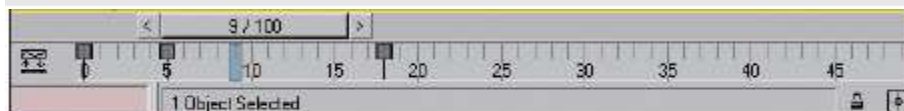
```
b=box()  
at time 10 animate on b.width=50      -- 10프레임에 width 애니메이션  
at time 20 animate on b.width=120     -- 20프레임에 width 애니메이션  
bhc=b.width.controller  
deletetime bhc (interval 5 10)         -- 5에서 10 사이를 지운다.
```



deleteTime 사용

이번에는 insertTime 함수를 이용하여 늘려보겠습니다.

```
insertTime bhc 10 3    -- 10 프레임에 3만큼 넣는다.
```



insertTime 사용

예제 01

화면을 Reset하고 일정 부분만큼 키를 지우는 예제를 알아보겠습니다.
Deletekeys 함수를 이용하면 keys의 일부분을 지우는 것이 가능합니다.

CD\초급 활용편\Chapter_4\01_controller.ms

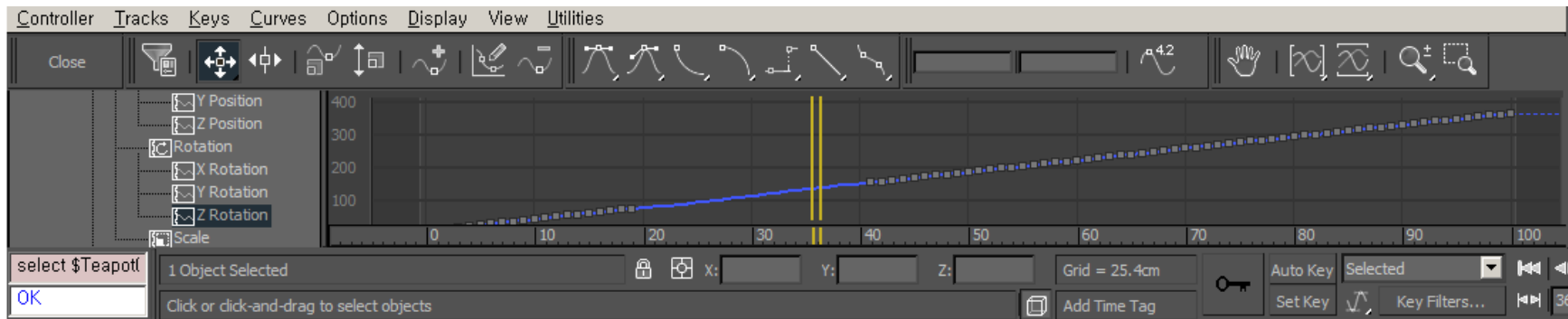
```
t = teapot()

animate on
(
    frame = animationRange.end - animationRange.start
    tic = frame as integer / 160.0          -- NTSC time 160 ticks은 1 frame입니다
    rot = 360 / tic                          -- 회전량을 계산
    for i in animationRange.start to animationRange.end do
    (
        at time i rotate t (eulerAngles 0 0 rot)    -- 주전자를 회전
    )
)

Con = t.rotation.controller.X_Rotation.controller    -- x_rotation 콘트롤러
selectkeys Con (interval 20f 40f)                    -- 20f에서 40f 키를 선택
deletekeys Con #selection                             -- 선택한 keys를 지운다

Con = t.rotation.controller.Y_Rotation.controller    -- y_rotation 콘트롤러
selectkeys Con (interval 20f 40f)
deletekeys Con #selection

Con = t.rotation.controller.Z_Rotation.controller    -- z_rotation 콘트롤러
selectkeys Con (interval 20f 40f)
deletekeys Con #selection
```



Key를 지우는 예제를 좀더 짧게 하는 방법도 있습니다.

다음 예제를 보면 Deletekeys, moveKeys 등의 함수는 서브 controller까지 검색하여 처리한다는 것을 알 수 있습니다.

```
t = teapot()

animate on
(
    frame = animationRange.end - animationRange.start
    tic = frame as integer / 160.0          -- NTSC time 160 ticks은 1frame입니다
    rot = 360 / tic                        -- 회전량을 계산
    for i in animationRange.start to animationRange.end do
    (
        at time i rotate t (eulerAngles 0 0 rot)    -- 주전자를 회전
    )
)

Con = t.controller          -- Position_Rotation_Scale 컨트롤러
selectkeys Con (interval 20f 40f)
deletekeys Con #selection
```


Modifiers

Modifier List에 있는 여러 가지 기능들을 script를 이용하여 사용하여본다.

자주 사용되는 Modifiers 함수

```
addModifier <node> <modifier> [before:index] -- mapped
```

node에 modifier를 추가합니다. before 키워드 옵션을 붙인 경우 modifier 스택의 최상위에서 인덱스만큼 내려간 위치에 modifier를 추가합니다.

```
deleteModifier <node> <modifier_or_index>
```

node에 modifier를 삭제합니다.

```
collapseStack <node> -- mapped
```

node의 Stack으로부터 modifiers를 collapse합니다.

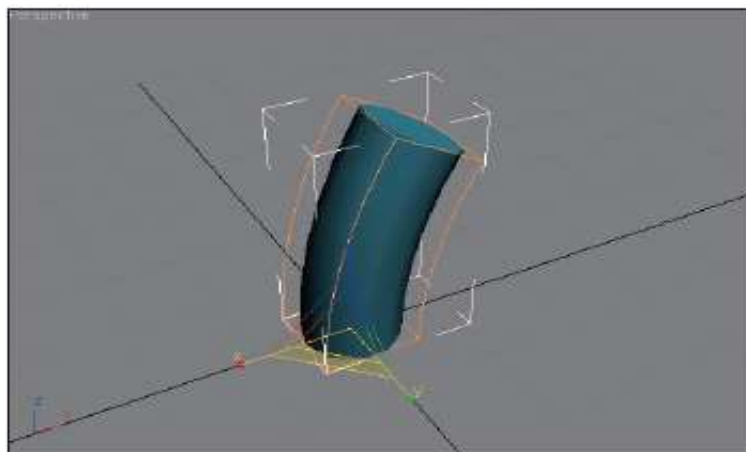


예제 01

cylinder에 Bend modifiers를 추가하는 Script를 작성해 보겠습니다.

```
c = cylinder heightsegs:5 sides:10 radius:5
select c
max modify mode
myBend = Bend()
addModifier c myBend
$.modifiers[#bend].angle = 30
```

```
-- cylinder 생성
-- cylinder 선택
-- modify 모드로 간다.
-- bend를 myBend 변수에 저장
-- modifier를 추가한다.
-- angle 값을 30 적용
```



bend 적용 결과

modifiers를 하나 적용해봤으니 지워보겠습니다.

(초급 활용편\Chapter_5\01_bend.ms 에 있습니다.

```
ben = c.modifiers[#bend]
deleteModifier c ben
```

```
-- #bend란 이름의 modifiers를 ben 변수에 저장
-- modifiers 지우기
```

Modifier Stacks

```
b = box()  
addModifier b (UVWmap())  
addModifier b (UVWmap())  
addModifier b (UVWmap())  
collapseStack b           -- collapse Stack
```

다른 modifiers를 적용해 봅시다.

```
b = box()  
addModifier b (UVWmap())      -- UVW map 적용  
addModifier b (twist())       -- twist 적용  
addModifier b (ffd_4x4x4())   -- ffd_4x4x4 적용
```

Modifier stacks.

Listener에서 modifiers 내용을 확인해보면 다음과 같습니다.

```
b.modifiers          -- modifier array를 얻어낸다.  
#modifiers(FFD 4x4x4:FFD 4x4x4, TwistMod:Twist, MapMod:UVW Mapping)  -- 출력 결과
```

여기서 두번째의 twist modifiers를 알아낼려면 다음과 같이 하면 됩니다.

```
b.modifiers[2]        -- 배열 인덱스로 알아낸다.  
b.modifiers[#twist]    -- #twist란 이름으로 알아낸다.
```

만약 angle 값을 적용하고 싶다면 아래와 같이 angle 속성을 수정해주면 됩니다.

```
b.modifiers[#twist].angle = 35      -- angle을 35로 수정  
b.twist.angle = 35                  -- 위와 같은 결과
```

만약 특정 modifiers가 적용되었는지 알아보려면 validModifier 함수를 사용하면 됩니다.

```
validModifier b b.modifiers[#ffd_4x4x4]  
true          -- 출력 결과
```

Modifier 복사

다음 예제는 box 1과 box 2를 만들고 modifiers 스택을 instance로 복사하는 예제입니다.

```
b1 = box pos:[0,0,0] heightsegs:2  
b2 = box pos:[50,0,0] heightsegs:2  
addModifier b1 (bend())
```

-- bend 적용

-- 만약 copy로 하고 싶다면 다음 copy 명령어를 앞에 붙여두면 된다.

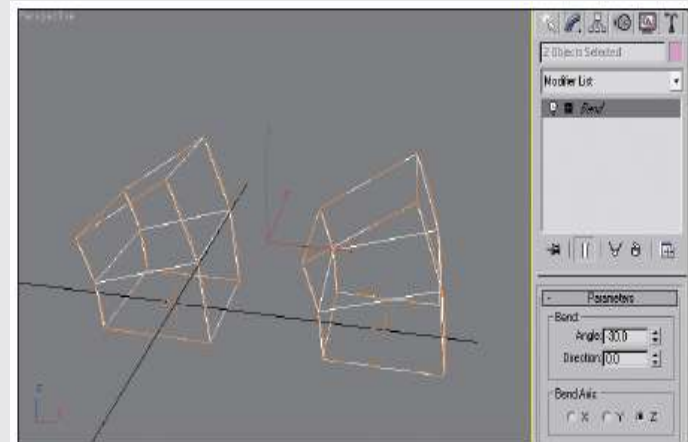
```
-- copyBend = copy b1.modifiers[#bend]
```

```
copyBend = b1.modifiers[#bend]
```

```
addModifier b2 copyBend
```

-- instance로 복사

```
copyBend.angle = -30
```



bend를 instance로 복사

Editable Mesh

- 선택한 vertex를 바닥에 [0,0,0]에 정렬할때,
- 좌우 대칭형 character의 반쪽을 지울때등에 script 사용.

```
p = plane()  
convertToMesh p  
print p.verts
```

```
#verts(#all : $Editable_Mesh:Plane01 @ [0.000000,0.000000,0.000000])    -- 출력 결과
```

```
p.verts[1]  
p.edges[1]  
p.faces[1]
```

-- 출력 결과

```
-- #verts(1 : $Editable_Mesh:Plane01 @ [0.000000,0.000000,0.000000])  
-- #edges(1 : $Editable_Mesh:Plane01 @ [0.000000,0.000000,0.000000])  
-- #faces(1 : $Editable_Mesh:Plane01 @ [0.000000,0.000000,0.000000])
```

```

p.verts.count      -- vertex 개수를 알아낸다.
p.numverts         -- vertex 개수를 알아낸다.
getNumVerts p      -- vertex 개수를 알아낸다. 전부 동일한 결과

```

```

25                -- 출력 결과

```

```

p.faces.count      -- face 개수를 알아낸다.
p.numfaces         -- 역시 face 개수를 알아낸다.
getNumFaces p      -- 역시 face 개수를 알아낸다.

```

```

32                -- 출력 결과

```

```

p.verts[1].pos     -- 1번 vertex 위치를 알아낸다.
getvert p 1        -- 역시 1번 vertex 위치를 알아낸다.

```

```

[-12.5, -12.5, 0] -- 출력 결과

```

face 구조에 대해 알아보려면 getface를 이용하면 됩니다.

```

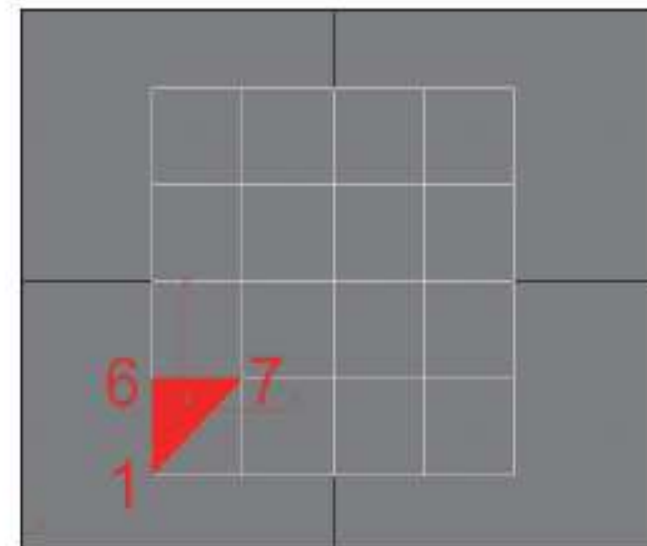
face = getface p 1

```

```

[6, 1, 7]         -- 출력 결과

```



아래는 plane을 만들고 1번 vertex 위치를 알아내는 예제입니다.

```
p = plane()  
p.mesh.verts[1].pos
```

`[-12.5, -12.5, 0]` -- 출력 결과

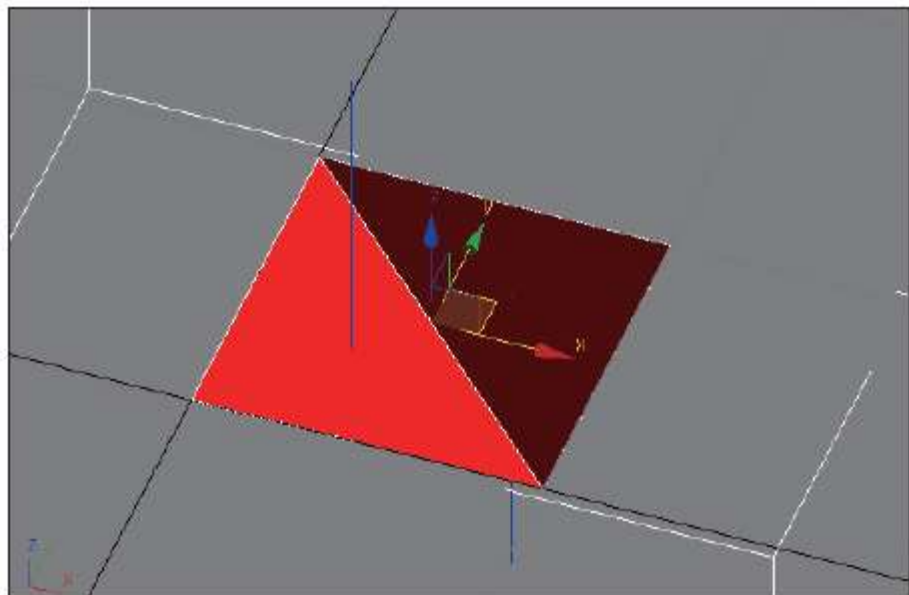
```
p = plane()  
addmodifier p (mesh_select())  
p.verts[1].pos
```

`[-12.5, -12.5, 0]` -- 출력 결과

CD\초급 활용판\Chapter_6\01.msi

mesh 함수를 이용하여 사각형 만들기

```
mesh vertices:#([0,0,0],[10,0,0],[0,10,0],[10,10,0]) faces:#([1,2,3],[2,4,3])
```



face 구조가 시계 방향이면 면이 뒤집힌다.

실행해보면 삼각형 두 개로 이루어진 사각형이 만들어 집니다. 첫 번째 옵션인 vertices에는 각각의 vertex 위치 정보가 들어가고 두 번째 옵션인 faces에는 삼각형을 이루는 vertex 구조 인덱스가 2개 들어가 있는 것을 알 수 있습니다. 근데 자세히 보면 face를 이루는 구조가 [1,2,3], [2,4,3] 인데 이것이 반시계 방향으로 구성되어 있는 것을 알 수 있습니다.

만약 거꾸로 시계 방향으로 이루어지면 어떻게 될까요? 다음 예제는 두번째 face가 시계 방향으로 이루어져 있습니다.

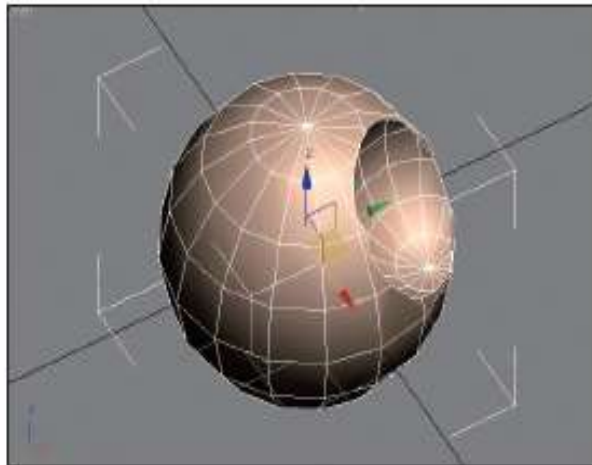
```
mesh vertices:#([0,0,0],[10,0,0],[0,10,0],[10,10,0]) faces:#([1,2,3],[2,3,4])
```

```
<mesh> + <mesh> -- union  
<mesh> - <mesh> -- difference  
<mesh> * <mesh> -- intersection
```

CD\초급 활용편\Chapter_6\04_mesh difference.ms, 04_verts.ms

예제 04

다음은 두 개의 Editable Mesh를 - (마이너스)하는 예제입니다.



mesh - mesh 결과

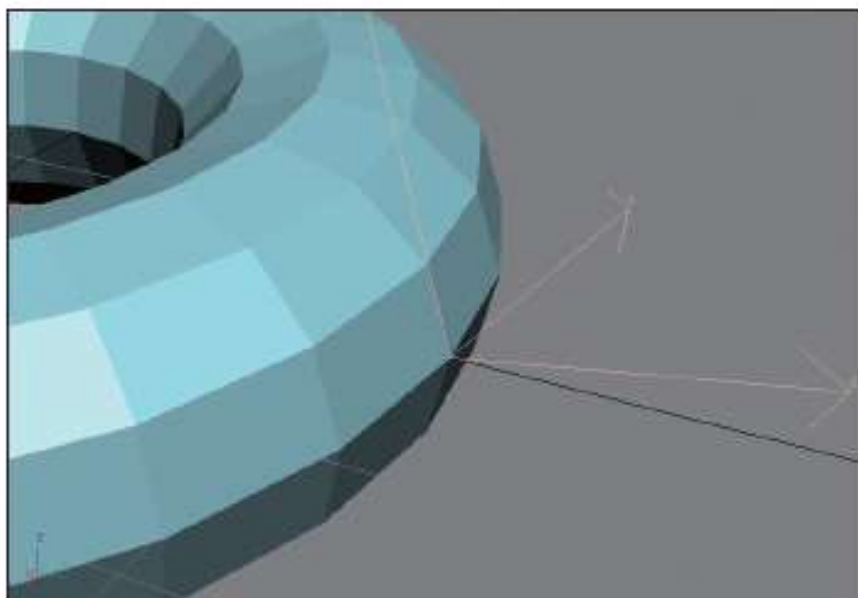
```
sp1 = sphere pos: [0,0,0]  
sp2 = sphere radius:15 pos: [15,15,15]  
convertToMesh sp1  
convertToMesh sp2  
sp1 - sp2  
delete sp2
```

예제 05

다음은 1번 vertex의 위치와 노말 벡터(Normal)을 얻어와서 그 위치에 point를 생성하는 예제입니다.

```
obj = torus()
convertToMesh obj
vertNum = 1
pos = getvert obj vertNum
Normal = getNormal obj vertNum
point pos:pos dir:Normal axistripod:true cross:false
```

- torus 생성
- Editable_Mesh로 변환
- 버텍스 인덱스는 1
- vertex 위치를 알아낸다.
- Normal을 알아낸다.
- point 생성



CD\초급 활용 편\Chapter_6\05_vert Normal.ms

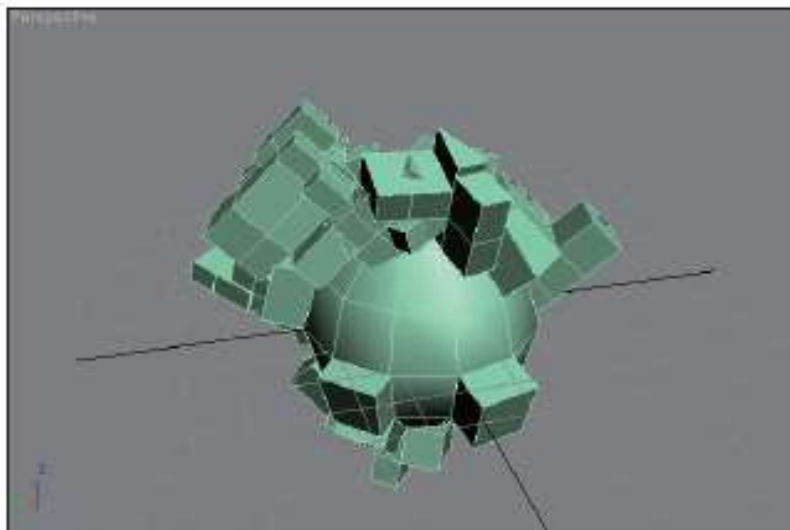
Vertex Normal

예제 11

아래 예제는 랜덤으로 Faces를 extrude하는 예제입니다.

Editable Poly를 편집하는 Script 중 meshTools는 이러한 짧은 Script들의 모음입니다.

```
p = sphere segments:12
convertToPoly p
for i = 1 to 100 do                -- 1 부터 100까지 for문 돌리기
(
                                -- 랜덤으로 face 선택하기
    polyop.setFaceSelection p #{(random 1 p.faces.count)}
    p.extrudeFaces (random 5.0 10.0)    -- extrude Face 한다.
)
```



랜덤 faces extrude

CD\초급 활용편\Chapter_6\11_random ExtrudeFaces.ms

File Access

```
teapot()
```

```
saveMaxFile "c:\\test.max" -- 맥스 파일을 세이브한다.
```

```
saveMaxFile "c:\\test02.max" useNewFile:true
```

```
resetMaxFile() -- reset 하기
```

```
loadMaxFile "c:\\test.max" quiet:true -- 맥스 파일을 로딩한다.
```

3ds Max System Directories

<filetype_name>

● Getdir 함수의 사용방법

GetDir <filetype_name>

#animations	-	#plugcfg
#archives	-	#preview
#autoback		#proxies
#defaults		#renderassets
#downloads	-	#renderPreset
#drivers		#renderoutput
#export		#scene
#expression		#scripts
#font		#sound
#help		#startupScripts
#image		#temp
#import		#ui
#matlib		#userMacros
#maxData	-	#userScripts
#maxroot		#userStartupScripts
#maxstart		#userIcons
#Photometric	-	#vpost

```
getdir #animations  
getdir #help  
getdir #ui  
getdir #maxroot
```

-- 출력 결과

```
"C:\Documents and Settings\Administrator\My Documents\3dsmax\sceneassets\animations"  
"C:\Program Files\Autodesk\3ds Max 2008\help"  
"C:\Program Files\Autodesk\3ds Max 2008\UI\  
"C:\Program Files\Autodesk\3ds Max 2008\"
```

Bitmap 과 color의 표현

예제 01

아래 예제는 box로 r, g, b 색상을 단계적으로 생성하여 보여 주는 예제입니다.

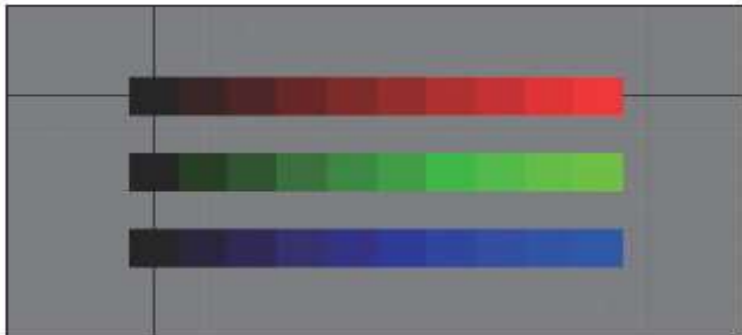
맨 위에 width 변수를 수정하면 좀더 많은 단계의 box를 생성하니 수정하면서 실행해 보기 바랍니다.

```
width = 10

widthStep = 255.0/(width-1)
for i in 0 to width-1 do
(
    p = [i*10,0,0]
    c = color (i*widthStep) 0 0
    box length:10 width:10 height:10 pos:p wirecolor:c

    p = [i*10,-20,0]
    c = color 0 (i*widthStep) 0
    box length:10 width:10 height:10 pos:p wirecolor:c

    p = [i*10,-40,0]
    c = color 0 0 (i*widthStep)
    box length:10 width:10 height:10 pos:p wirecolor:c
)
```



color <r> <g> [<a>]

- <r>: 붉은색(red)을 나타냅니다. 사용 범위는 0~255까지 입니다.
- <g>: 초록색(green)을 나타냅니다. 사용 범위는 0~255까지 입니다.
- : 파란색(blue)을 나타냅니다. 사용 범위는 0~255까지 입니다.
- [<a>]: Alpha 값입니다. 사용 범위는 0~255까지 입니다.

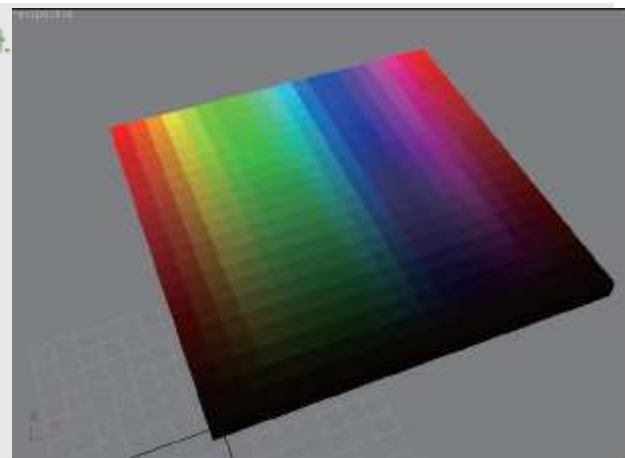
다음 예제는 color 값을 조절하여 color picker 다이얼로그 같은 색을 box로 표현해 주는 스크립트입니다. Color.v와 color.s, color.h가 어떻게 변해가는지 관찰해 보고 위쪽에 width 변수와 height 변수를 수정해서 다시 실행합니다.

만약 스크립트를 디버깅하려면 중간중간에 print나 format을 넣어 변수에 어떤 값이 들어가는지 살펴보는 것도 좋습니다.

```
delete $tempBox*           -- tempBox란 이름을 가진 box가 있을 경우 지운다.
width = 20                 -- 가로 box 개수 설정
height = 20               -- 세로 box 개수 설정

wstep = 255.0/(width-1)    -- 증가되는 wstep 계산
vstep = 255.0/(height-1)  -- 증가되는 vstep 계산
for i in 0 to width-1 do
(
    for j in 0 to height-1 do
    (
        p = [i * 10, j * 10, 0]          -- box의 위치 계산
        myColor = black                  -- 초기 color는 black
        myColor.v = j * vstep            -- color value를 계산
        myColor.s = 255                  -- color의 채도값 255
        myColor.h = i * wstep            -- color hue 값 계산

        -- box 생성
        box length:10 width:10 height:10 pos:p \
            name:"tempBox" wirecolor:myColor
    )
)
)
```



box의 r, g, b 배치

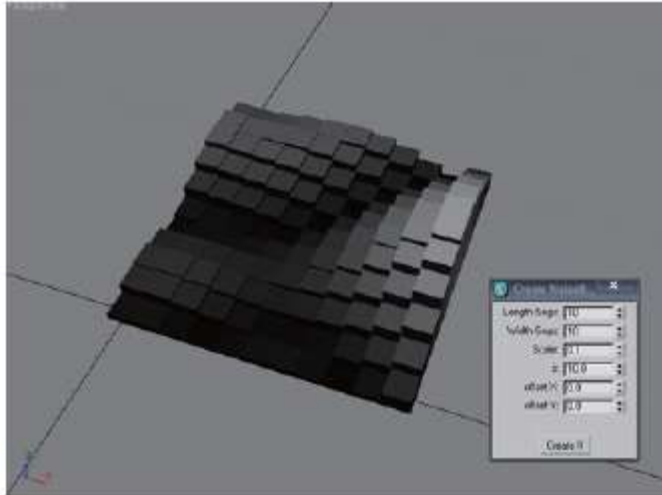
예제 03

이제 좀더 재미있는 예제를 진행해 봅시다. 이전과 동일한 box를 사각형 모양으로 배치하는 예제이지만 noise4 함수를 사용하여 박스의 높이와 색깔을 결정합니다. 박스 높이가 낮으면 검정색에 가까워지고 반대로 높다면 흰색에 가까워 집니다. 박스 생성 후에 옵션은 실시간으로 조정 가능합니다. 주의할 점은 Length Segs, Width Segs 수치를 너무 올리면 버벅이므로 적당히 올리는 것이 좋습니다.

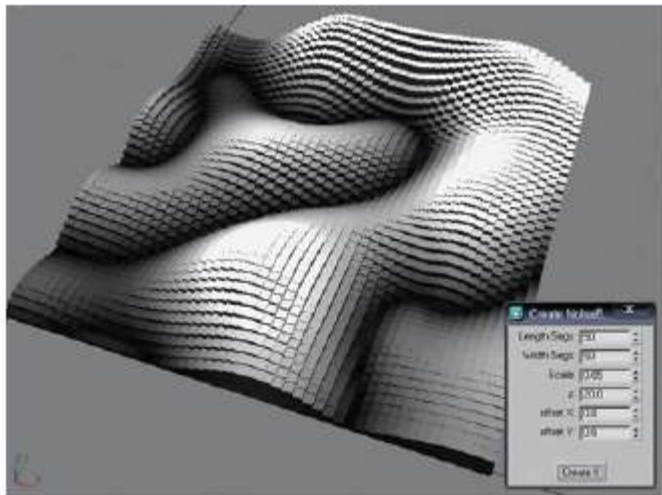
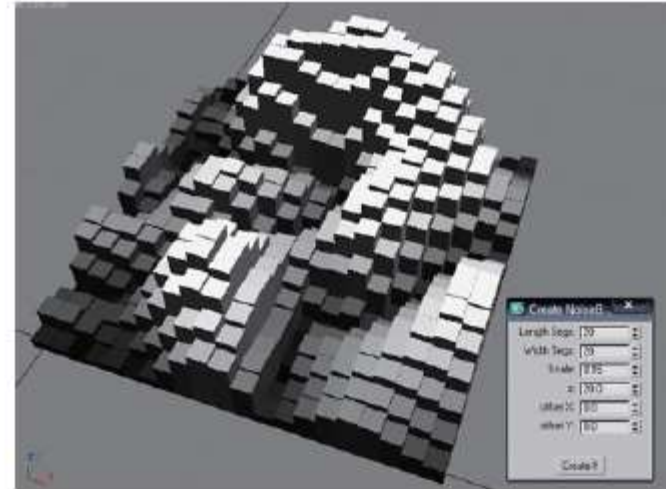
```
noise4 <point3> <phase_float>
```

예제에 사용된 Noise4의 사용 방법은 다음과 같습니다.

noise4는 3차원 노이즈를 수학적으로 생성해주는 함수로 리턴 값은 -1.0 ~ 1.0 사이의 값을 가집니다. 이와 비슷한 함수로 noise3, turbulence, fractalNoise 함수가 준비되어 있으며 스크립트 레퍼런스에도 검색하면 나옵니다.



Create NoiseBox 실행 결과



해당 예제는 부록 CD\초급 활용편\Chapter_10\03_CreateNoiseBox.ms 에 있습니다.

```

1 -- Noise로 박스를 배치하기 위한 함수
2 fn makeNoiseBoxFn Lengthseg Widthseg scaleV randV z offset =
3 (
4     if randV == undefined do return undefined
5
6     delete $tempNoiseBox*                -- $tempNoiseBox 이름의 오브젝트 모두 지우기
7
8     -- 2중 for문
9     for y in 0 to Lengthseg do
10     (
11         for x in 0 to Widthseg do
12         (
13             n = noise4 ([x*scaleV,y*scaleV,0]+offset) randV -- noise4를 이용해서 box에 높이 설정
14             hei = abs (n*z) -- box의 height 값 설정. 여기서는 (-)숫자가 나올수도 있기 때문에 abs를 사용한다.
15
16             -- box color 설정
17             col = [50,50,50]*hei
18             if col.x >= 255 or col.y >= 255 or col.z >=255 do
19             (
20                 col = [255,255,255]
21             )
22
23             -- box 생성
24             box width:1 length:1 height:hei pos:[x,y,0] name:"tempNoiseBox"
25             wirecolor:col
26         )
27     )
28
29
30 try( destroydialog CreateNoiseBox_roll)catch() -- 다이얼로그가 있다면 지우기
31
32 rollout CreateNoiseBox_roll "Create NoiseBox"
33 (
34     local randomNum
35     -- random 숫자를 저장하기 위한 변수

```

```

41
42 spinner Lengthseg_spi "Length Segs: " range:[2,1000,10] type:#integer
43 spinner Widthseg_spi "Width Segs: " range:[2,1000,10] type:#integer
44 spinner scale_spi "Scale: " range:[0,1000,0.1] scale:0.01
45 spinner z_spi "z: " range:[0,1000,10]
46 spinner offsetX_spi "offset X: " range:[-1000,1000,0] scale:0.01
47 spinner offsetY_spi "offset Y: " range:[-1000,1000,0] scale:0.01
48 button create_btn "Create !!" offset:[0,20]
49
50
51 on Lengthseg_spi changed val do
52     makeNoiseBoxFn Lengthseg_spi.value Widthseg_spi.value scale_spi.value
53     randomNum z_spi.value \
54         [offsetX_spi.value,offsetY_spi.value,0]
55
56 on Widthseg_spi changed val do
57     makeNoiseBoxFn Lengthseg_spi.value Widthseg_spi.value scale_spi.value
58     randomNum z_spi.value \
59         [offsetX_spi.value,offsetY_spi.value,0]
60
61 on scale_spi changed val do
62     makeNoiseBoxFn Lengthseg_spi.value Widthseg_spi.value scale_spi.value
63     randomNum z_spi.value \
64         [offsetX_spi.value,offsetY_spi.value,0]

```

```

55
56     on z_spi changed val do
57         makeNoiseBoxFn    Lengthseg_spi.value    Widthseg_spi.value    scale_spi.value
randomNum    z_spi.value    \
58             [offsetX_spi.value,offsetY_spi.value,0]
59
60     on offsetX_spi changed val do
61         makeNoiseBoxFn    Lengthseg_spi.value    Widthseg_spi.value    scale_spi.value
randomNum    z_spi.value    \
62             [offsetX_spi.value,offsetY_spi.value,0]
63
64     on offsetY_spi changed val do
65         makeNoiseBoxFn    Lengthseg_spi.value    Widthseg_spi.value    scale_spi.value
randomNum    z_spi.value    \
66             [offsetX_spi.value,offsetY_spi.value,0]
67
68     on create_btn pressed do
69         (
70             randomNum = random -1.0 1.0
71
72             -- makeNoiseBoxFn 함수 호출
73             makeNoiseBoxFn    Lengthseg_spi.value    Widthseg_spi.value    scale_spi.value
randomNum    z_spi.value    [0,0,0]
74         )
75 )
76 createdialog CreateNoiseBox_roll

```

bitmap 저장 형식을 avi으로 지정하여 애니메이션한 후 저장하는 예제입니다. 예제를 실행하기 전에 파일이 저장 될 "c:/temp/" 폴더가 있는지 확인해 보고, 없다면 만들어 줍니다.

```

theTeapot=teapot()
animate on
    at time 30
        rotate theTeapot 360 z_axis

-- 카메라 생성
cam=targetcamera pos:[200,0,100] target:theTeapot
renderFrames=#{1..30} -- 프레임 저장 범위
-- bitmap 생성..파일은 avi로 한다.
b=bitmap 160 120 filename:"c:/temp/test.avi"

-- 프레임 범위 안에서 렌더링
for i = 1 to renderFrames.count do
(
    if renderFrames[i] then
    (
        at time i -- 해당 프레임으로 이동
        -- 렌더링 후 bitmap에 저장
        render 160 120 camera:cam to:b
        save b
    )
)
close b -- bitmap 닫기
RAMPlayer "c:/temp/test.avi" ""

```