

初めての MAXScript 入門講座

3D Studio をご利用の方で、MAXScript って話は聞くけど実際何が
できるのって思っていないですか。

私もその1人です。これから数回に渡って実際に MAXScript の評価
をしてみます。そこでの、苦労話やノウハウを皆さんと共に考えて
見ましょう。

なお、本講座は、MAXScript の利用をを広める機会として提供して
います。ご意見、ご感想などがありましたら、”
webmaster@vizuser.com ”までご連絡ください。皆さんで MAXScript
講座を作って行きましょう。

2001年7月7日 vizuser.com mako 記

必要な環境：

- ・ 3D Studio VIZ R2 以降のバージョン
- ・ 3D Studio VIZ のヘルプ
- ・ Microsoft Windows NT または 2000 が稼動している PC

今回利用する MAXScript の参考文献：

- ・ Discreet の HP に MAXScript 入門のダウンロードページ
- ・ VIZ のオンラインヘルプ

MAXScript で何が出来るの？

MAXScript は、3D Studio MAX および 3D Studio VIZ の中に組み込まれているプログラム言語です。

R2 よりサポートされるようになりました。

幸いな事にMAX Script は、3D Studio MAX および 3D Studio VIZでほとんど共通に利用出来ます。将来MAXを使いたいなど思われているユーザもMAXScriptは有効に利用できるツールとなります。

利用可能な機能

- ・モデリング、アニメーション、マテリアル、レンダリングなどの新規作成や拡張。
- ・作成したスクリプトを用いてプラグインを構築する。
- ・ファイル入出力に関しては読み込み / 書き出しが可能。
- ・OLE オートメーションを利用して他のアプリケーションとのデータのやり取りが可能。
- ・通知イベント(プレ / ポスト シーン ファイルを開く、新規、リセット、シーン ファイルの保存、プレ / ポスト レンダリング、選択の変更など)を MAX Script 側で受け取る事が可能。

利用可能なユーザインターフェース

- ・カスタム ユーティリティのパネル ロールアウトやモードレスウィンドウが利用出来る。
- ・コマンドライン リスナー ウィンドウからの入力が可能。
- ・ツールバーのボタンにスクリプトをマクロとして割り当てる。
- ・標準のユーザインターフェースを拡張または置換する。

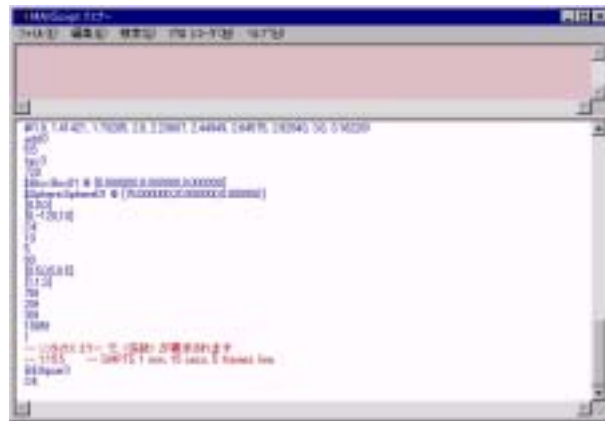
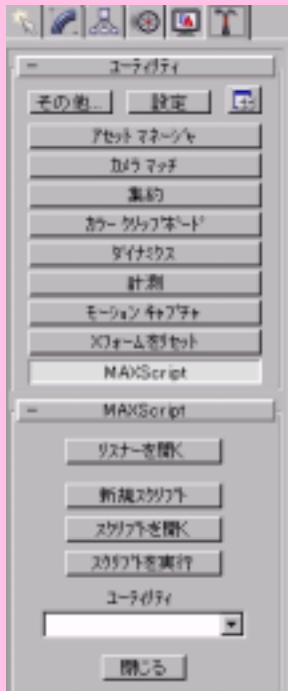
以上の事が出来れば、ほとんど全てのコントロールを MAXScript で出来る事になります。それでは、具体的にどうすればそれが利用出来るのかを見て行く事にしましょう。

リスナーを試してみよう

では、まずはMAXScriptの最初の第一歩を行ってみましょう。

MAXScriptは、[ユーティリティ]パネルの中の[MAXScript]ボタンをから開きます。

MAXScript ロールアウトの中に[リスナーを開く]ボタンがあります。これを選んでください。



これが、MAXScriptのリスナーです。

MAXScript リスナーはMAXScript 言語のインタープリタです。MAXScript リスナーにコマンドを入力すると直ちに実行されます。また、MAXScript リスナーは、テキスト エディタとコマンド プロンプト ウィンドウの機能を持っています。

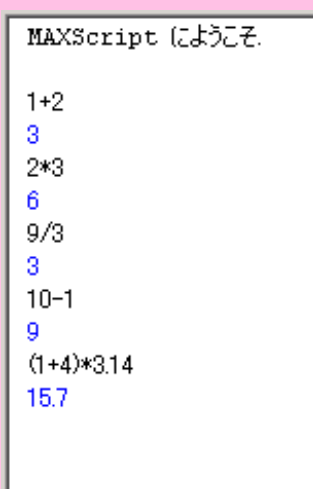
では、もっとも簡単なコマンドを行ってみましょう。MAXScript リスナで四則演算を行ってみます。

この様にまずは電卓として使えますね。

括弧をつけるとちゃんと演算の順位規則で処理してくれます。

リスナ内の入力にショートカットが利用できます。

- CNTL + X 選択したテキストの切り取り
- CNTL + C 選択したテキストのコピー
- CNTL + V クリップボードの内容の貼り付け



MAXScript の文法

ちょっと難しい話になりますが、MAXScriptで取り扱える文法について考えてみましょう。

MAXScriptの文法は非常にたくさんの構文から構成されています。詳細はオンラインヘルプを利用して、ここでは最小限必要となる項目に関して説明します。

データの型について

整数型

数値の型には整数と実数の 2 つのタイプがあります。

整数の範囲は、-2,147,483,648 ~ 2,147,483,647 です。

実数型

実数の範囲は、絶対値で 1.18E-38 ~ 3.40E38 までで、小数部の精度は 10E7 です。

文字型

文字列に関しては文字列の長さに制限はありません。

例 "3D Studio MAX"

特殊文字

特殊文字を指定するには¥を使用します。

```
¥"      -- ダブルクォーテーション
¥n      -- 改行文字
¥r      -- 改行
¥t      -- TAB
¥*      -- アスタリスク
¥?      -- 疑問符
¥¥      -- 単一の '¥'
¥%      -- アスタリスク
¥x{d}   -- 16 進数
```

Boolean 型

Boolean に関しては次のものがあります。

```
true
false
on      -- true と同じ
off     -- false と同じ
```

Boolean 演算子

演算子

```
not <boolean>
    ブール値が false の場合は true を返します。
    ブール値が true の場合は false を返します。
<boolean> and <boolean>
    両方のブール値が true の場合、
    true を返します。
<boolean> or <boolean>
    いずれかのブール値が true の場合、
    true を返します。
```

MAXScript の文法

Point3 型

Point3 型

3次元の座標値です。

[<expr>, <expr>, <expr>]

例

[x, y, z]

[100, 100, 100]

[sin x, cos y, tan z]

x_axis -- [1,0,0]と同等

y_axis -- [0,1,0]と同等

z_axis -- [0,0,1]と同等

プロパティ

<point3>.x : 浮動小数点 -- x 座標

<point3>.y : 浮動小数点 -- y 座標

<point3>.z : 浮動小数点 -- z 座標

Point2 型

Point2 型

2次元の座標値です。

[<expr>, <expr>]

例

[x, y]

[100, 100]

[sin x, cos x]

プロパティ

<point2>.x : 浮動小数点

<point2>.y : 浮動小数点

Time 型

Time 型

アニメーション時間値です。

例

7.5s : 72.5 秒

5m30s : 5 分 30 秒

150f : 150 フレーム

3:30.0 : 3 分 30 秒 0 フレーム (SMPTE)

プロパティ

<time>.ticks : ティックで表示する時間

<time>.frame : フレームで表示する時間

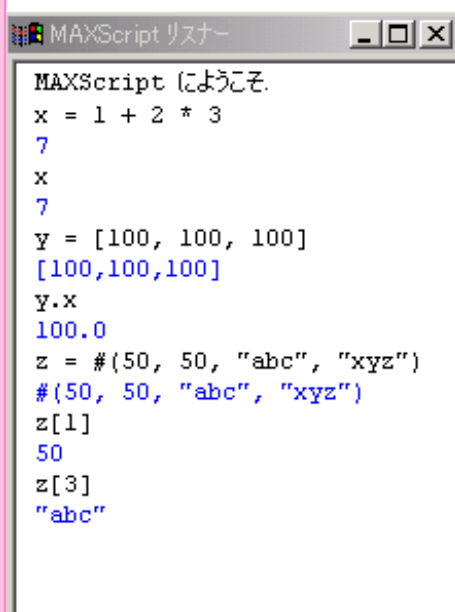
<time>.normalized : 正規分数で表示する時間

MAXScript の文法

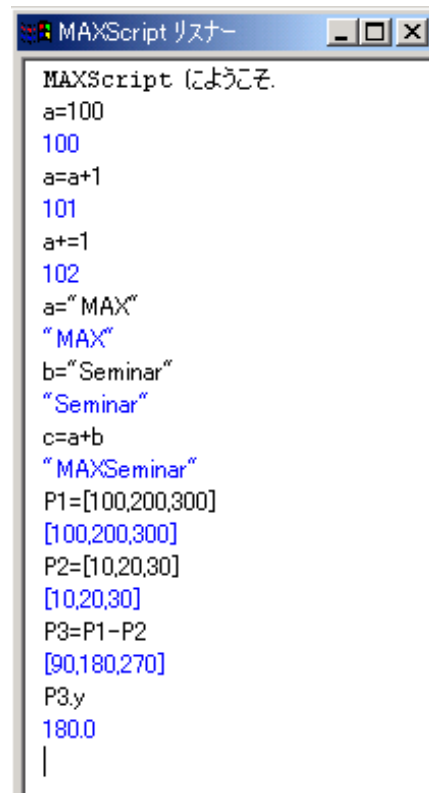
変数

一般的なプログラム言語では変数の型の宣言が必要です。ただ、MAXScript では、変数の型の宣言をすることもできますが、変数を使用される時に自動的に型の設定がされます。

<code>x = 1 + 2 * 3</code>	変数 x に代入
<code>y = [100, 100, 100]</code>	変数 y に 3D座標値を代入
<code>z = #(50, 50, "abc", "xyz")</code>	変数 z を配列とする



```
MAXScript リスナー
MAXScript (ようこそ。
x = 1 + 2 * 3
7
x
7
y = [100, 100, 100]
[100,100,100]
y.x
100.0
z = #(50, 50, "abc", "xyz")
#(50, 50, "abc", "xyz")
z[1]
50
z[3]
"abc"
```



```
MAXScript リスナー
MAXScript (ようこそ。
a=100
100
a=a+1
101
a+=1
102
a="MAX"
"MAX"
b="Seminar"
"Seminar"
c=a+b
"MAXSeminar"
P1=[100,200,300]
[100,200,300]
P2=[10,20,30]
[10,20,30]
P3=P1-P2
[90,180,270]
P3.y
180.0
|
```

また、変数の値の演算も可能です。

なお、変数名の命名規則は大体次の様になっています。

- MAXScript の予約キーワードは使えません。
- 大文字と小文字の区別はありません。
- 最初に数字が来る事はできません。
- スペースや特殊文字は使えません
- アンダースコア " _ " は使えます。