

بسم الله الرحمن الرحيم

الدرس الثالث :

الأرقام النصوص والمصفوفات في MAX Script 7 :

ذكرنا سابقاً أنه بوسعنا إدخال السكريبتات في نافذتين إحداهما MAXScript Listener Window والأخرى MAXScript Editor Window . فلنختار الأولى ونقوم بتطبيق والتعرف على قواعده بعجالة سريعة .

الأرقام في الماكس سكريبت :

يميز الماكس سكريبت بين نوعين من الأرقام وهي الأعداد الصحيحة Integer والأنواع الكسرية Float أو يمكن تسميته Floating Point . وهنا أيضاً إذا قمنا بإعطائه مجموعة من الأوامر على نوع معين فإن الناتج سيكون بنفس النوع . وإليك مثلاً على ذلك :

```

actionMan.executeAction 0 "40472" -- MAX Script: MAXScript Listener

Welcome to MAXScript.

-- مثال على نواتج العمليات الحسابية بنوعيه الصحيح والكسري --

5 + 7 -- Int + Int --> Int
12
5.0 + 7.0 -- Float + Float --> Float
12.0
5.0 + 7 -- Float + Int --> Float
12.0
|

```

فكما لاحظت فإن ناتج جمع عددين صحيحين هو عدد صحيح أيضاً وكذلك الحال مع الأعداد الكسرية ، ويمكن ملاحظة أن النوع الأشمل غالباً يسيطر على النوع الأخص كما في ناتج الجملة الثالثة حيث قمنا بجمع عددين إحداهما صحيح والآخر عشري فكان الناتج عشري .

النصوص في الماكس سكريبت :

تعتبر لغة الماكس سكريبت أساسها التعبير أي أنها تعتمد على التعبير في بناء كودها ، وكل جملة لها ناتج ، ولذلك إذا أردنا كتابة نص فإنه يتوجب علينا وضعه في علامتي تنصيص ( " ) ، وغالباً يكون ناتج هذه الجمل هو نفس النص المدخل فمثلاً لو أردنا تنفيذ كلمة ( جديد ) فهناك طريقتين لفعل ذلك كما في النافذة التالي :

```

MAXScript Listener
File Edit Search MacroRecorder Help
actionMan.executeAction 0 "40472" -- MAX Script: MAXScript Listener

Welcome to MAXScript.

-- مثال على طباعة النصوص بنوعيه

-- النوع الأول
جديد
undefined
[
-- النوع الثاني
جديد
جديد

```

فكما لاحظنا في النوع الأول قام بإعطائنا خطأ بكلمة ( غير معرف ) وقام بقبول النوع الثاني وطباعة الناتج بشكل صحيح .

#### المصفوفات في الماكس سكريبت :

يمكن استخدام المصفوفة لضم مجموعة من العناصر في مكان تجميع واحد ، ويمكن تعريف المصفوفة في الماكس سكريبت أنها مجموعة من القيم ، وهنا كل قيمة قد تختلف في نوعها فيمكن ضم العدد الصحيح والعشري والنصي في مصفوفة واحدة ، ويمكن إدخال العناصر بشكل منفرد عن العناصر الأخرى .

ويمكن تعريف المصفوفات عن طريق الرمز # ( ) بالصيغة التالية : #( value1 , value2 )

وبإمكاننا التعامل معها ببساطة مع ملاحظة أن القيمة الأولى يمكننا الوصول لها بـ Index بالقيمة 1 وكذلك بالنسبة لباقي العناصر [1] ... [n] .  
ويمكنك تمثيل الكلام السابق جميعه في المثال التالي :

```

MAXScript Listener
File Edit Search MacroRecorder Help

Welcome to MAXScript.

a = #( 5^2 , "نص" , 5 + 7 - 2 * 20 )
#(25, "نص", -28)
a[2] = "String"
String
a
#(25, "String", -28)
a[1] = a[1] + 5
30
a[2] = a[2] + " In MAXScript"
String In MAXScript
a
#(30, "String In MAXScript", -28)

```

**المعاملات الرياضية في الماكس سكريبت:**

يوجد كم كبير لأغلب الدوال الرياضية مثل الجيب وجيب التمام والـ PI و جميع الثوابت الرياضية .

لنتحدث قليلاً عن المعاملات الأساسية الأربعة فمثلاً في ضرب عددين ، يوجد عندي ثلاث دوال لعملية الضرب وذلك :

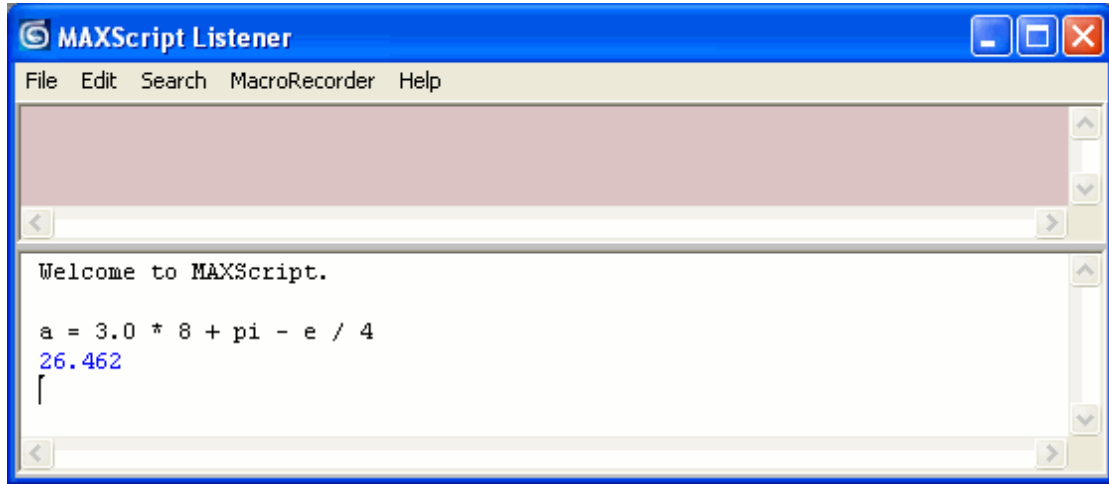
واحدة إذا كان العددين كلاهما عدد عشري .

واحدة إذا كان أحد العددين عدد صحيح والآخر عشري .

واحدة إذا كان العددين كلاهما عدد صحيح .

وهكذا لأغلب العمليات .

ولنكتب مثال على كيفية استخدام الأعداد والمعاملات الأربعة الأساسية و استخدام الثوابت الرياضية :



```

MAXScript Listener
File Edit Search MacroRecorder Help

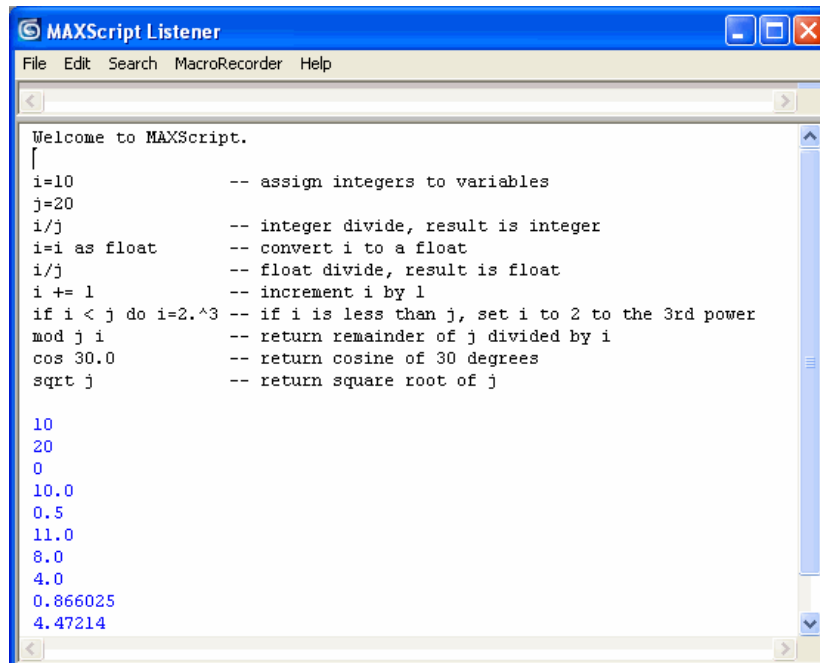
Welcome to MAXScript.

a = 3.0 * 8 + pi - e / 4
26.462
|

```

ويمكننا استخدام المتغيرات التالية :

- العمليات المثلثية مثل : Atan ، Sinh ، Cos ...
  - الدوال المتسامية مثل : Sqrt ، Log ، Exp ...
  - الأعداد العشوائية Random Number .
  - إلى غيرها من العمليات الحسابية .
- واليك مثال على استخدام الدوال الحسابية :



```

MAXScript Listener
File Edit Search MacroRecorder Help

Welcome to MAXScript.
|
i=10          -- assign integers to variables
j=20
i/j           -- integer divide, result is integer
i=i as float  -- convert i to a float
i/j           -- float divide, result is float
i += 1       -- increment i by 1
if i < j do i=2.^3 -- if i is less than j, set i to 2 to the 3rd power
mod j i      -- return remainder of j divided by i
cos 30.0     -- return cosine of 30 degrees
sqrt j       -- return square root of j

10
20
0
10.0
0.5
11.0
8.0
4.0
0.866025
4.47214

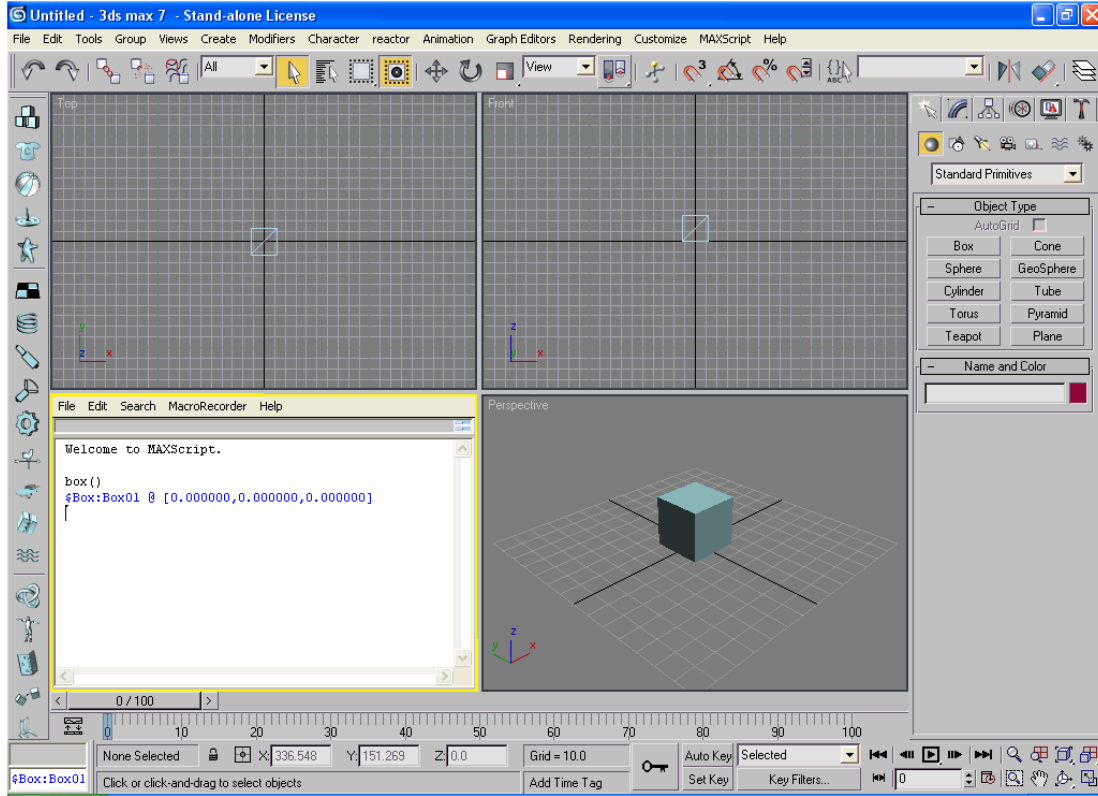
```

بالإضافة إلى معاملات الزيادة مثل :

< Var > += val -- إضافة القيمة val إلى المتغير Var  
 < Var > -= val -- تنقيص القيمة val إلى المتغير Var  
 < Var > \*= val -- ضرب القيمة val بالمتغير Var  
 < Var > /= val -- قسمة القيمة val على المتغير Var

إضافة صندوق إلى المشهد :

سنقوم إبتداءً من هذا القسم بالتعامل مع المشهد ثلاثي الأبعاد .  
 بإمكاننا إنشاء كائن الصندوق Box باستخدام الدالة الافتراضية القيم Box() ، فلو قمت  
 بكتابته بالنافذة لوحدت التالي :



كذلك بإمكانك إسناد هذه القيمة إلى متغير كالتالي :

**Box1 = box()**

وهنا لو قمت بكتابة Box() أو box() أو BoX() أو غيرها من الصيغ سيقبلها الماكس سكريبت لأنه لغة غير حساسة أي أن كل الجمل السابقة متشابهة .

كما لاحظت سابقاً أننا نقوم بإنشاء مربع ولكن؟؟ ماذا لو أردت أن تعطي خصائص للصندوق  
 أي الطول أو العرض أو حتى الارتفاع !!  
 يمكنك فعل ذلك بكتابة الجملة التالية :

**mybox = box length:20 width:20 height:20**

وهنا قمنا بإسناد الصندوق لمتغير ولاحظ أننا قمنا بإعطائه خصائص تصفه .  
 ولاحظ أنه تم إنشاء مخرجات على الشكل التالي :

`$Box:Box02 @ [0.000000,0.000000,0.000000]`

وهنا يمكن تقسيم السطر إلى قسمين :

القسم الأول : يمكن تمثيله بالمسار ( كمسار أي نافذة C:\windows\... ) لأنه يدل على أنه  
 يوجد متغير اسمه Box02 وهو من نوع Box ، ودائماً يبدأ بالرمز \$ .  
 القسم الثاني : وهي الإحداثيات لهذا الكائن من حيث المحاور الثلاثة X , Y , Z .

وهنا نكون قد أنهينا الدرس الثالث ، وإلى لقاء قريب في درس جديد .