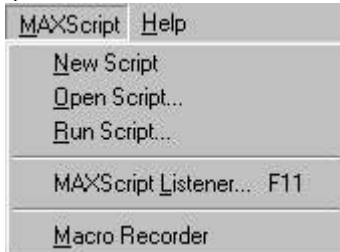
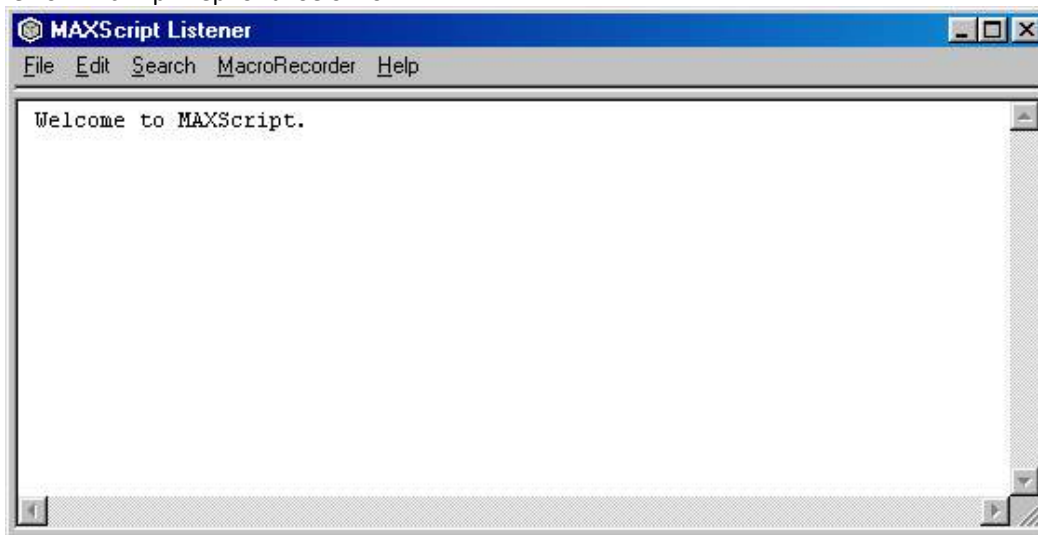


## Занятие 01


MAXScript - язык программирования, предназначенный, главным образом, для реализации потребностей, которые возникли у пользователя, но которых пока нет в текущей версии 3D Studio Max, или они есть, но реализованы не так, как хотелось бы. Самые простые знания и навыки в работе с ним можно получить, работая с окном интерпретатора MAXScript (MAXScript Listener). Для этого в главном меню 3ds max выберем пункт "MAXScript", а в появившемся меню – пункт "MAXScript Listener ... F11"



На экране появится примерно такое окно



это и есть интерпретатор. Если в нем набрать любую конструкцию языка и нажать клавишу **Enter**, то она (конструкция, конечно, а не клавиша) немедленно выполнится, а результат ее выполнения выведется в это же

окно, но синим цветом. Доступ к окну интерпретатора можно получить так же, выбрав закладку Utilities  и нажав в ней кнопку MAXScript. Откроется свиток,



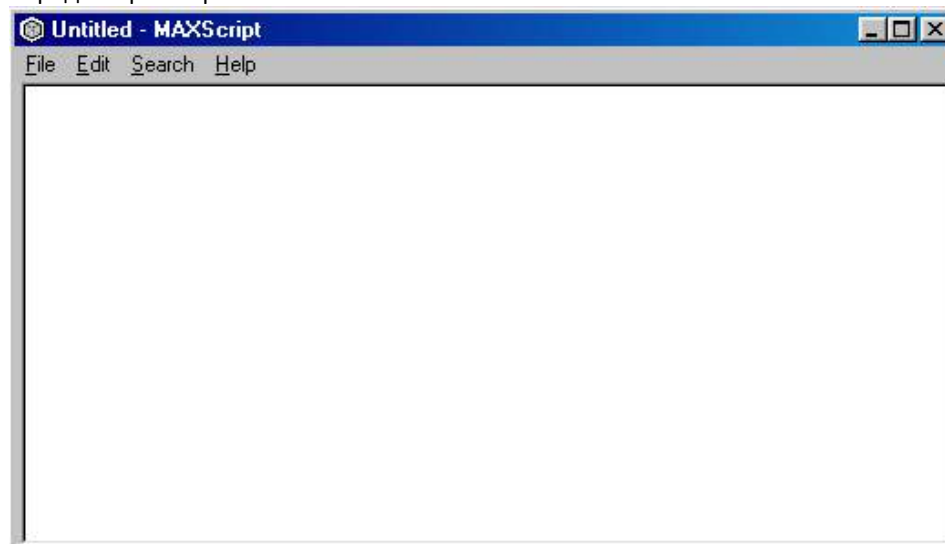
в котором надо нажать кнопку Open Listener.

Однако работать с окном интерпретатора довольно неудобно. Если Ваша программа состоит из большого числа строк, то в ней не мудрено допустить ошибку, а для ее исправления надо снова вводить всю программу. Кроме

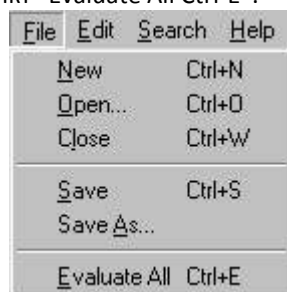
того, придумав прекрасную программу, Вы, наверное, захотите воспользоваться ей неоднократно. Глупо было бы записывать ее на бумажку и при надобности вводить ее каждый раз заново. Разумеется, MAXScript имеет средства для сохранения и загрузки программ. Более того, с его помощью можно дорабатывать пользовательский интерфейс 3ds max, но об этом далее. Пока воспользуемся меню окна интерпретатора. Выберем пункт File, а в открывшемся меню пункт New Script



(по сложившейся традиции, программа на MAXScript называется скрипт). Появится окно, напоминающее текстовый редактор Notepad.



В нем можно набрать любой сложности скрипт, и сохранить его в файл. Традиционное расширение для файлов, содержащих скрипты – ms. В меню File редактора скриптов, кроме обычных New, Open, Close, Save и Save As, есть пункт "Evaluate All Ctrl-E".



Выбрав его, Вы выполните скрипт, содержащийся в редакторе.

Нетрудно догадаться, что пункты Open Script и Run Script из меню File окна интерпретатора, служат для того, чтобы открыть имеющийся скрипт или выполнить его.

Впрочем, если хотите, можно набирать текст скрипта например в Notepad или подобном редакторе. Этих средств программирования вполне хватит для написания скриптов любой сложности.

Как и всякий язык, MAXScript имеет свою грамматику, которую и начнем изучать с операции присвоения значений переменным.

**Задание. Создать несколько примитивов, используя команды, вводимые в приемник. Задать параметры объектов и их положение.**

## Занятие 02

Редактор скриптов. Создание скрипта, сохранение, загрузка. Выполнение однострочного скрипта. Скрипт – текстовый сценарий, интерпретируемый 3dsMax. последовательное выполнение команд. Использование пробелов и табуляции. Оформление скрипта. Комментарии. Однострочный комментарий, блочный, внутри команды.

```
b = box () -- создание объекта
/*
параметры соданного объекта
ширина, высота, длина

*/
b.width = 10.0
b.height = 20.0
b.length = 30.0
```

Многострочные операторы. Читаемость сценария.  
Sphere radius:17.104 smooth:on segs:200 chop:0 slice:off \  
sliceFrom:0 sliceTo:0 mapcoords:on recenter:off \  
pos:[47.9844,-10.4004,0] isSelected:on

Переменные. Имена переменных. Присвоение значения.  
Типы переменных. Целые. С плавающей точкой. Литеральные.  
Необязательное определение типа. Изменение типа при операции.  
Строковый тип.  
msg = "привет"  
messagebox msg  
Логический тип.

Функции. Вызов функции.  
Независимость имени от регистра.

Сообщение об успешном выполнении команды. Сообщение об ошибке.  
Команда `showproperties b`

Дополнительные свойства. Имя. Положение.

```
b.pos.x
```

Цвет каркаса  
b.wirecolor = [255,0,0]  
b.wirecolor = color 0 255 0

Выделение объекта. Удаление объекта.

Поворот  
a = EulerAngles (random -90 90) (random -90 90) (random -90 90)  
MyPoint.rotation = a as quat

**Задание. Написать сценарий создания транспортного средства (паровозик)**

## Занятие 03

### Функции.

Способы передачи аргументов функции.

```
MyColor = color 122 12 123
```

Аргументы передаются:

В определенном порядке

В любом порядке после ключевого слова

Сначала обязательные без ключевых слов, потом необязательные с ключевыми.

Переменная точки в трехмерном пространстве Point3

```
MyPoint = Point3 10 20 30
```

```
MyPoint = [10,20,30]
```

Независимость от порядка параметров

```
MyPoint = box length: 10 height: 20 width: 30
```

Вызов функции

Функция () – без аргументов

Функция arg1 arg2 – обязательные

Функция Слово:значение слово2:значение

Функция arg1 arg2 Слово:значение слово2:значение

Пример функции

```
MyPoint = box length: 10 height: 20 width: 30
```

```
select MyPoint
```

```
Function ChangeColor obj = (obj.wirecolor = [255,0,0])
```

```
ChangeColor $
```

Задание. Написать функцию для создания десяти предметов случайных размеров и цветов.

## Занятие 04

Возвращение значений

```
function add2 x y = (z=x+y)
print (add2 12 13)
```

Создание функций с ключевыми аргументами

```
Function MyMessage text1:"My Own Text" = (MessageBox text1)
MyMessage text1:"Hello"
```

Возвращение значения функции Return. Плюсы и минусы.

Локальные и глобальные переменные.

```
rad = 10.5
A = sphere radius: rad
B = cylinder radius: rad
```

видимость переменной после объявления в программе.

```
global A
local B
```

Объявление глобальных переменных является нежелательным  
Область действия локальной переменной. Создание блока ()

### Условный Оператор

Оператор IF – THEN

```
if () then ()
if () then () else
```

```
A = sphere radius: (random 5 10) name: "A"
B = sphere radius: (random 5 10) name: "B"
if (A.radius>B.radius) then (messagebox "A") else (messagebox "B")
```

Логическое сравнение >, <, ==, >=, <=

Логические операции И, НЕ, ИЛИ - and, not, or, !=

Возможность разбиения условного оператора на строки.

Задание. Написать скрипт, создающий по случайному параметру и заданному пороговому значению объект нужной геометрии и цвета.

### Циклические конструкции

Оператор For i=1 to 5 do ...

Оператор Do () While i<N

## Массивы

Массив – совокупность , последовательность элементов.

```
MyArray = # ()  
MyArray = #(1,3,56)
```

Присвоение значения

```
My[3] = 12
```

Присоединение элемента

```
append MyArray 2342
```

Массив объектов

```
C = box ()  
A = sphere ()  
B = cylinder ()
```

```
Arr = #(A, B, C)  
d = donut()  
append Arr d
```

Arr.count - число элементов

DeleteItem Arr 2 - удаление элемента массива

Join Arr1 Arr2 – объединение массивов

findItem – возвращает индекс искомого элемента

Пример

```
for i = 1 to 3 do (  
    box ()  
    sphere ()  
)  
A = $box* as array  
join A $sphere*  
for i=1 to A.count do (messagebox A[i].name)
```

Пример

```
NNN = #()  
for i = 1 to 3 do (  
    x = box ()  
    sphere ()  
    NNN [1] = x.name  
    NNN [2] = x.pos  
    --print (NNN[2] as string)  
)  
ZZZ = #()  
for i in objects do (append ZZZ i.name)  
print "FF="+ZZZ as string)
```

## Метасимволы

Использование символа \* в качестве замены символов в именах объектов.

Создание совокупности объектов

```
Coll = $box*
```

Создание массива из совокупности

```
MyArray = Coll as array
```

Если в сцене удалить один из объектов совокупности, а затем выполнить команду

```
MyArray = Coll
```

то совокупность и массив обновятся .

Массив существует независимо от объектов. Обращение к данным в массиве нужно согласовывать с изменениями в составе объектов.

```
for i=1 to 10 do (  
    A = box ()  
    A.pos = [random -100 100, random -100 100, 0]  
  
    )  
  
coll = $box*  
Arrr = coll as array  
  
delete $box005  
  
for i=1 to Arrr.count do (  
    if (not isdeleted Arrr[i])  
    then messagebox Arrr[i].name  
    else messagebox "Deleted"  
  
    )
```

Задание. По данным в массиве о зданиях в «городе», высчитать средний объем здания, здания с недостающим объемом дополнить объектом сверху.

## Строки

Строки – буквенно-цифровые переменные. Используются как сообщения, названия файлов и т.д.

Значение строкового типа при присвоении обязано иметь кавычки.

Операции со строками

Сшивание

```
text1 = "Hello "  
text2 = "World!"  
messagebox (text1+text2)
```

### Поиск строки

Поиск первого экземпляра подстроки в более крупной строке.

```
ttt = "String for search"  
location = findString ttt "ng"  
messagebox (location as string)
```

### Замена строки

```
Text1 = "1234567890"  
Text2 = replace Text1 5 3 "Replace"  
messagebox Text2
```

Преобразование типов строк и чисел.

```
S = sphere ()  
A = S.radius  
messagebox (A as string)
```

```
D = random 20 50  
T = D as string  
messagebox T  
S.radius = T as float
```

### Длина строки

```
<string>.count  
количество символов в строке
```

Конвертация строки в выражение

```
obj = execute ("$" + S.name)
```

### Подстрока

```
s = "Baleroфон"  
ss = substring s 5 3-- returns "rof"  
ss = substring s 5 100-- returns "roфон"
```

### Подмена строки

```
S = sphere ()  
A = S.name  
B = substituteString A "Sphere" ""  
C = B as float  
S.radius = C  
messagebox (C as string)
```

### Пример.

```
selected = $  
for i in selected do (  
  A=i  
  A.name  
  Type1 = "Torus"  
  Type2 = "Box"
```



```
L=findString A.name Type1
case of (
((findString A.name Type1) != undefined): messagebox "Torus"
((findString A.name Type2) != undefined): messagebox "Box"
default:messagebox "Unknown"
)
)
```

Задание.

Выделенные объекты опознать, сложить в цепочку и переименовать по порядку.