

# Scripting for Artist

MEL, MaxScript, Python & LUA

# Session 1

- MEL, Maxscript, Python and Lua Language descriptions and definitions.
- Data Types:
  - » Int: 1, 3425, -6689 ...
  - » Float: 2.25, 0.0002, 9867234.1, -5.0
  - » Array: a group of (similar) types
    - Vector {3.2, 5.1, -43}
    - Matrix {{4, 2, 1},{8, -3, 5}}
    - String "Mountain View"

# How is scripting different from programming?

- Scripts can be written and executed on the fly without explicit compile and link steps.
- A scripting language is usually interpreted from source code or byte-code.
- Scripts are written for a software environment like Python; the user may not have access to its source code.

# What makes programming difficult to learn?

- Code solution is expressed differently from the way we solve problems.
- Computer cannot infer from context or enter into a clarification dialog.
- Programmers have to break intended tasks into clear, simple, self-containing steps and well-defined computer instructions.

# MEL

## Maya Embedded Language

- MEL™ is Maya scripting language and Maya's GUI is created using MEL.
- MEL is descended from UNIX shell scripting.
- MEL is strongly based on little stand-alone programs with many options that modify their behavior to accomplish anything in Maya.
- MEL can automate and extend everything that can be done using Maya's GUI.
- Maya has thousands of MEL commands to perform various functions.

# Maxscript

- MAXScript is the built-in scripting language for 3ds Max that can control all of its features.
- MAXScript is object-oriented, and mirrors high-level concepts in the 3ds Max GUI.
- Scripts can be packaged within custom utility panel rollouts, giving them a standard 3ds Max UI.
- Scripts can create import/export tools using the built-in file I/O, build procedural controllers that can access the entire state of the scene and build batch-processing tools, such as batch-rendering.

# Python

- Python is an interpreted, object-oriented, high-level programming language.
- Python built in data structures, dynamic typing - binding, and automatic memory management make it very attractive for scripting.
- Python supports modules and packages, which encourages program modularity and code reuse.
- Python in Maya is a direct translation of Maya's command into python functions that is packaged in a Maya library.
- Python scripts in Max is through Blur Studio's Py3dsMax plugin

# Lua

"Lua" (pronounced **LOO-ah**) means "Earth Moon" in Portuguese.

- Lua is an extension programming language designed to support general procedural programming with data description facilities.
- Lua is intended to be used as a powerful, light-weight configuration language.
- Lua is a *dynamically typed language*, which means variables do not have types; only values do.
- Lua only works *embedded* in a host client that can invoke Lua executing functions, write and read Lua variables, and register C functions to be called by Lua code.



# Data Types



[intros.cs.princeton.edu](http://intros.cs.princeton.edu)

# Variables

- Variables are temporary or persistent memory space to store data locally or globally for a later use or processing.
- In general variable names should not start with a number or contain a space
- Variables are passed around, processed by code and outputted in some form or another.

# Integer (int)

In computer science, an integer is a datum of integral data type, a data type which represents some finite subset of the mathematical integers. Integral data types may be of different sizes and may or may not be allowed to contain negative values. Integers are represented in a computer as a group of binary digits. Computer hardware nearly always provides a way to represent a processor register, or memory address as an integer; the set of integer sizes available varies between different types of computer.

[http://en.wikipedia.org/wiki/Integer\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Integer_(computer_science))



Did you understand that?

# int

- In scripting we need to tell the program how much memory space it should reserve for our data variable.
- An **int** is whole number without decimals like 10, 5, 2147483647. It may contain negative value or not. There are several int sizes available that we can select based on our needs.
- We will refer to any real number without a decimal point as an int and we let the program decide what size to use.
- int comparison is more accurate than float

# Int Declaration Sample

- **Maya Mel:**

```
int $red =128; // notice the $ sign before the variable and the ; at the end
```

- **Max Script:** (max script is type-free variables and type-safe)

```
red_int = 128 -- max does not require type symbol before the variable as a good practice  
include the type in the name
```

- **Python:** (python is fully dynamic-type system)

```
red = 128 # python like max does not require the type in a variable  
startCount = int(0) #converting to int
```

- **Lua:** (Lua is a *dynamically typed language* which means that variables do not have types; only values do)

```
red = 128 -- global variable  
local startCount = 0 -- local variable
```

# Float

- The term floating point refers to the fact that the decimal (binary) point can "float"; that is placed anywhere relative to the significant digits of the number.
- The float type can represent numbers as large as  $\pm 3.4\text{E}+38$  and as small as  $1\text{E}-44$ .
- The float type can also represent NaN (Not a Number),  $\pm$  infinity, and  $\pm$  zero.
- Float takes more memory space and is less precise than int.
- Number division is tricky  $3/2 = 1$  if all digits are int while  $3.0/2.0 = 1.5$  if the numbers are float.

# Float Declaration Sample

- **Maya Mel:**  
float \$last\_speed = 200.45; // notice the \$ sign before the variable and the ; at the end
- **Max Script:** (The Number types can be freely intermixed and MAXScript will convert as necessary)  
last\_speed = 200.45 --good practice include the type in the name  
timeOfDay = 0.5 as float --converting to float (or as double)
- **Python:** (python is fully dynamic-type system)  
timeOfDay = float(0.5) # converting to float  
last\_speed = 200.45
- **Lua:** (Lua is a *dynamically typed language* which means that variables do not have types; only values do)  
timeOfDay = 0.5 -- global variable  
local last\_speed = 200.45 -- local variable



# Array

In general array is a series of objects of similar size and type (although they may have different values) and stored contiguously in memory



Array is like the railway cars of the same size and carrying cargo of different amounts. You access the car by their relative position from the first car. We use open and closed bracket with a digit in between to reference array element or to define the size of array. Did you notice that `[]` look like a top view of a railway car.

# Array

- Array is a series of objects.
- Each object in an array is called an array element.
- Arrays can have more than one dimension.
- Array is an object; so when we say **myList** is an array and **yourList = myList** then any change to the element of **yourList** is a change to the element of **myList**. Maya MEL makes a copy of an array at assignment statement.

# For example:

*Python in Maya*

```
someList = list(['kim', 'go']);  
anotherList = someList;  
anotherList[1] = 'stop';  
print anotherList; # will result in  
['kim', 'stop']  
print someList; # will result in  
['kim', 'stop'] # not ['kim', 'go']
```

# Array Declaration Sample

- **Maya Mel:**

```
int $scores[]; //Maya dynamically increase the size of the array as  
needed
```

```
float $testGrades[] = {5.25, 6.6, 1.3}; //an array of 3 floats
```

```
string $grocery_list[] = {"milk", "eggs", "cheese", "bread"};
```

- **Max Script:** (An array is a variable length indexable sequence of values which can be of any type. indexing into the array starts at 1, not at zero)

```
scores = #() --declaring an empty array
```

```
testGrades = #(5.25, 6.6, 1.3)
```

```
grocery_list = #("milk", "eggs", "cheese", "bread")
```

```
oneTestGrade = testGrades[1]; --refer to the first element
```

```
mixedArray = #(10,#(1,2,3),30,"test") --create a mixed type array
```

# Array Declaration Sample

- **Python:**

```
mixdArray = [66.25, 333, 333, 1, 1234.5]
```

#when cut and paset some time single quote get changed to this ' symbol which is not the same as this ' symbol. They are interpreted differently since they have different ascii code

```
My_charArray = array.array('c',['e', 'g', 'g']) # first param id element type
```

```
grocery_list = list(["milk", "eggs", "cheese", "bread"]) #special array of objects
```

- **Lua:** (Tables can be used to hold arrays of information. Table can store mixed data types like numbers, strings, functions, or other tables.)

```
scores = { 1,1,2,3,5,8,13 } # indexing into the array starts at 1, not at zero. scores [0] has the value nil
```

```
testGrades = {}
```

```
testGrades [1]=80.5, testGrades [2]=60.0 , testGrades [3]=20.5
```

```
tableAsDictionary = { apple="green", orange="orange", banana="yellow" }
```

```
tableAsDictionary.apple = "red" #replacing "green" with "red"
```

# Vector

- Vectors are a container with ordered elements following a strict linear sequence.
- Think of vectors as an array of three elements
- In science the word vector means a magnitude and direction while in scripting it is usually a point of x, y & z coordinates with a direction from the origin and through that point.
- Most scripting languages have defined special arithmetic for vectors such as vector addition, multiplication and length.
- Cross Product “ $\times$ ” and Dot Product “ $\cdot$ ” operations often applied to vectors

# Vector Declaration Sample

- **Maya Mel:**

```
vector $myLocation = <<3.0, 7.7, 9.1>>; //use "<<" & ">>" to define vec val
```

- **Max Script:** (vectors have 3 values and describe positions in 3D space. They can also describe locations in RGB color space.)

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

```
my_destination = [100, 30.5, 41.3]
```

```
distance myLocation my_destination -- 91.3123 , distance is a max function
```

- **Python:** (Vector properties are similar to vectors used in science and engineering)

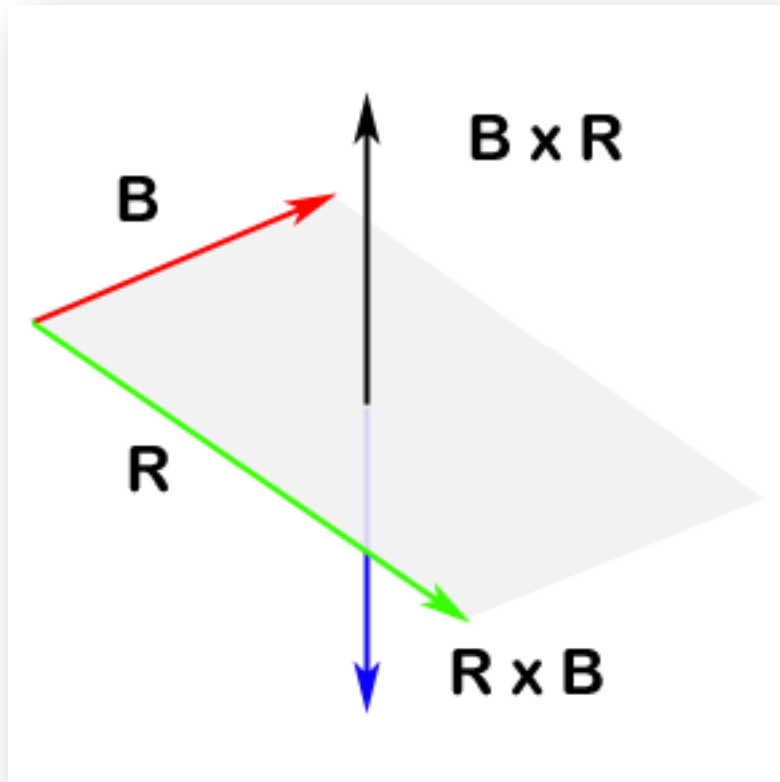
```
k_normal = vector(1.0, 0.5, 0.3)
```

- **Lua:** (Provided that a programmer has set up Lua vector handling in game code and defined vector3 as an array of 3 doubles)

```
myLocation = vector3(10,20,30)
```

```
myLocation.x, myLocation.y, myLocation.z = 100, 200, 300
```

# Cross Product



The cross (vector) product is an operation on 2 vectors in 3D space resulting in a vector which is perpendicular to the plane containing both them.

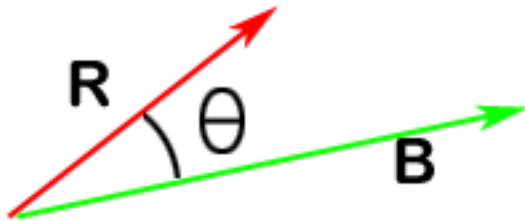
Notice that  $B \times R = -R \times B$

**For example in Maya:**

```
float $v1[3] = {1,2,3};  
float $v2[3] = {4,5,6};  
crossProduct($v1, $v2, 0, 0);  
// Results: (-3, 6, -3) //
```



# Dot Product



$$\mathbf{R} \cdot \mathbf{B} = |\mathbf{R}| |\mathbf{B}| \cos \theta$$

( $\mathbf{R} \cdot \mathbf{B}$ ) is a number that decreases as the angle increases

The dot (scalar) product is an algebraic operation that takes two vectors and returns a single number. It is often used in lighting equations such as getting the dot product of light direction and the surface normal and then multiplying the results into the diffuse color to get the final color.

**For example in Maya:**

```
float $v1[3] = {1,2,3};  
float $v2[3] = {4,5,6};  
dotProduct($v1, $v2, 0);  
// Results: 32 //
```

# Matrix

- A  $m \times n$  matrix is a two-dimensional array with  $m$  rows and  $n$  columns.
- Matrices are often used to store transform data of a 3D object.
- In Maya a matrix is a 2D table of floating point values. Unlike arrays, you must specify the size of a matrix when you create it.
- In Max a Matrix3 class implements a 4x3 3D transformation matrix object which is typically used to hold object coordinate systems and transformations.

# Matrix Declaration Sample

- **Maya Mel:**

```
matrix $m23[2][3]; //a 2D array of 2rows and 3 columns  
$m23[1][0] = 9; //setting the second row first column value to 9
```

- **Max Script:** (When working with object transformation matrix3 values, the rows and their sub-elements cannot be modified directly.)

```
myTransform = $Teapot01.transform
```

```
myTransform.row4 = [10.0,20.0,30.0]
```

```
$Teapot01.transform = myTransform --$Teapot01 identifies a teapot  
objects by pathname
```

# Matrix Declaration Sample

- **Python:** (There are several ways to create a matrix based on the loaded module. Here we use nested list to create a matrix)

```
m23 = [[1.0 2.0,6.6],[ 3.0, 4.0,10.1]]
```

- **Lua:** (One of the main way to represent a matrix in Lua is to use an array of arrays, that is, a table wherein each element is another table)

```
N=2
M=3
mtNM = {}      -- create the matrix
for i=1,N do
  mt[i] = {}   -- create a new row
  for j=1,M do
    mt[i][j] = 0
  end
end
end
```

# String



A string is a sequence of characters implemented as a byte array that stores a sequence of elements, using some character encoding. A string variable may either have a fixed or dynamic size depending on the scripting language. A string is stored as an array containing the characters and terminated with a null character ('\0'). Scripting languages have several string operations that assist in processing its content such as split string, convert all characters to upper or lower case, substitute a character sequence with another sequence and so on...

# String Declaration Sample

- **Maya Mel:**

```
string $oneObject = "mySphere";  
string $selectedObjectsList[] = `ls -sl`; //using a ls function to list selected
```

- **Max Script:**

```
oneObject = "mySphere"  
selectedObjectList = selection as array -- store current selection as a string  
-- array in the named variable
```

- **Python:**

```
oneObject = 'mySphere'  
print len(oneObject) # results in 8
```

- **Lua:** (Strings can be defined using single quotes, double quotes, or double square brackets)

```
obj1 = [[myCube]]  
obj2 = "myCone"  
oneObject = 'mySphere'  
string.len(oneObject) # results in 8
```