Maya API Classes – Day 3

Autodesk Maya Python API Training

Kristine Middlemiss, *Senior Developer Consultant*
Autodesk Developer Network (ADN)

Autodesk

Day 2 Questions:

1. Knowing which compiler is important, but where do we find that out on our Maya version and platform?

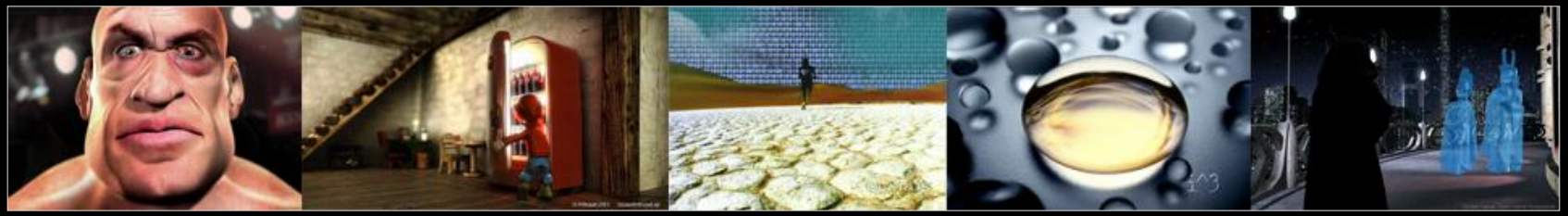
2. Is it possible to have the self-command presentation you talked about earlier?

3. Did the API documentation change format or style, in 2012, from 2011?

Home Work Review

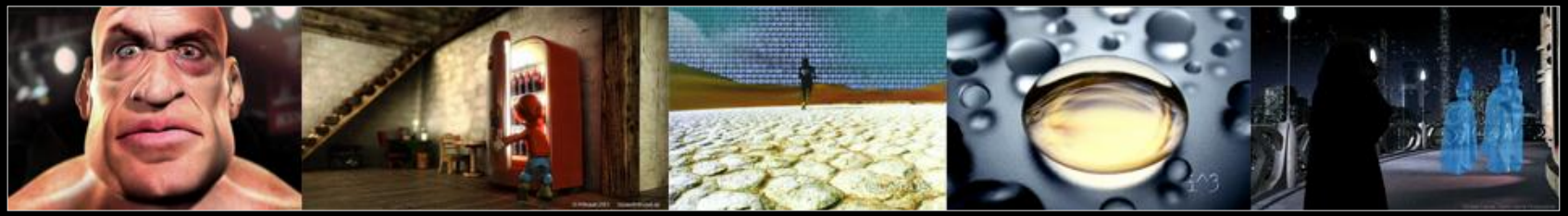
Agenda

- Maya Libraries
- Proxy Classes
- Mobject & Function Set Classes
- Iterator Classes
- Wrapper Classes



Autodesk®

Maya Libraries



Autodesk®

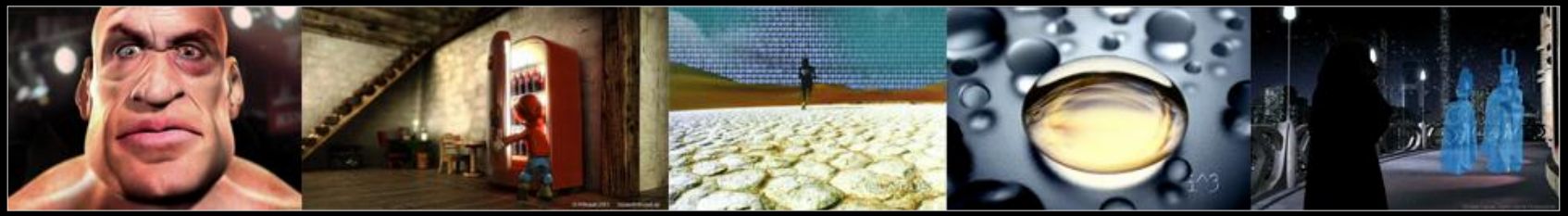
Maya Libraries (7 in total)

1. **OpenMaya:** fundamental classes for defining nodes and commands and for assembling them into a plug-in
2. **OpenMayaUI:** classes necessary for creating new user interface elements such as manipulators, contexts, and locators
3. **OpenMayaAnim:** classes for animation, including deformers and inverse kinematics.
4. **OpenMayaFX:** classes for Autodesk® Dynamics™
5. **OpenMayaRender:** classes for performing rendering
6. **OpenMayaMPx:** classes for proxy objects, no C++
7. **OpenMayaCloth:** classes for working with nCloth objects, no C++

Class Categories

Naming Convention	Logical Grouping	Examples
MPx***	Proxy	MPxCommand MPxNode
MFn***	Function set	MFnAttribute, MFnDependency Node
MIt***	Iterator	MItDependencyN odes, MItMeshEdge
M***	Wrapper et. al	MObject, MPoint, M3dView

Proxy Classes



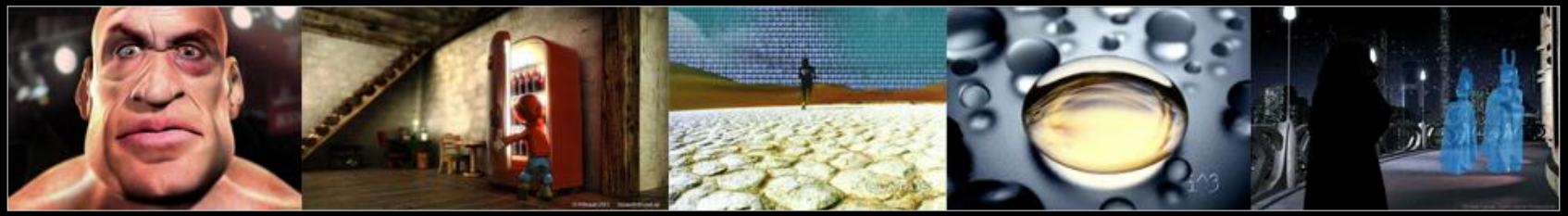
Autodesk®

Proxy Classes

- Proxy Object classes serve as base classes for your custom extensions.
- Proxy Classes begin with “MPx”
- Proxy Objects allow you to extend the Maya architecture through the creation of new Maya constructs (nodes, commands, etc.).
- The most prevalent proxy classes include:
  - MPxCommand
  - MPxNode

Function Set Classes

Looking at MObject



Autodesk®

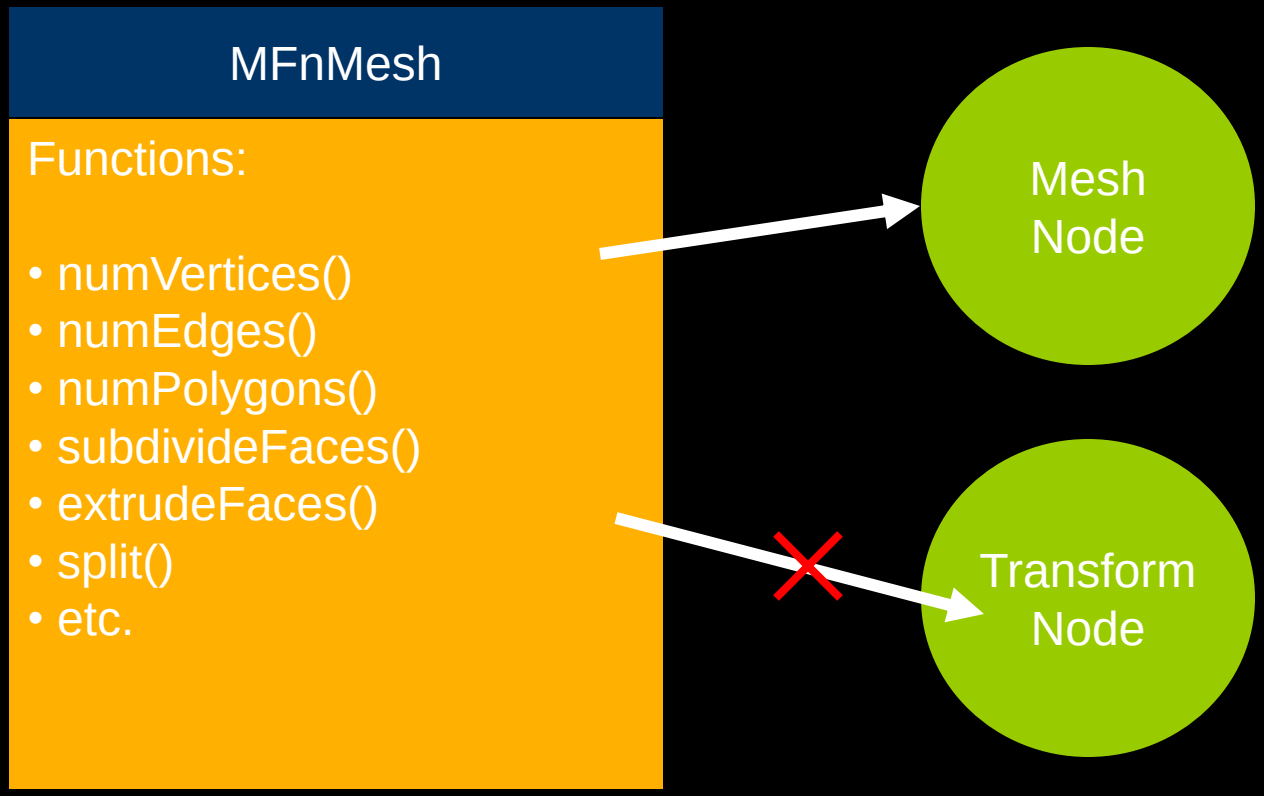
Function Set Classes & MObject

- Separate data with functionality

Classical OOP VS. Maya Approach

Function Set Classes

- Function Sets are classes that provide type specific APIs to the corresponding type(s) of MObjects



Function Set Classes

- Objects and function sets are always used together.

- They are separate which easily establishes ownership:
  - objects are always owned by Maya
  - function sets are always owned by you.

Function Set Classes

- MFn::Type enumeration is used throughout the API to indicate item types

  MFn::Type MFnBase::type()

- Once a function set is initialized to an MObject, you can call methods to query or set values on the object using

  ```
  myMeshFn = OpenMaya.MFnMesh ( myMeshObj ) # Opt 1: set through constructor
  myMeshFn.setObject( myMeshObj2 ) # Opt 2: if the function set already exists you can use it
  ```

  - Opt 2: This is more efficient than creating and destroying function sets each time you need one.

- Some of the commonly used function sets:
  - MFnDependencyNode, MFnDagNode and MFnAttribute

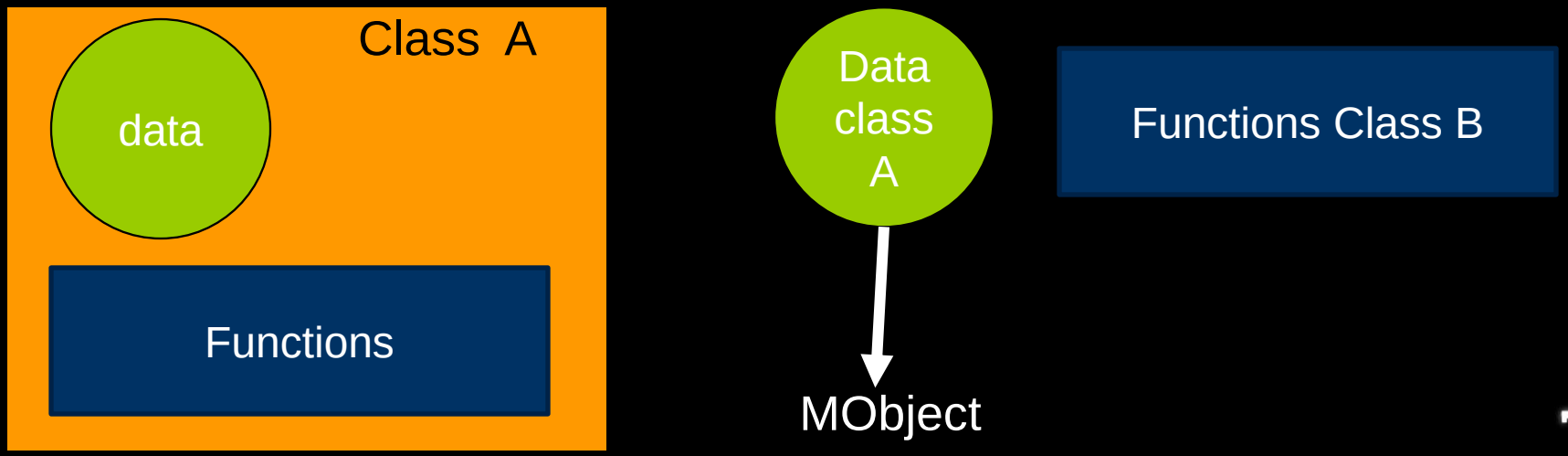
Function Set Classes

- MFnDependencyNode
  - Base class providing fundamental operators for all dependency nodes
  - Contains methods to query the name of a node, locate an attribute and parse connections
- MFnDagNode
  - Derived from MFnDependencyNode
  - Provides methods to query or modify parent/child relationships in the DAG
- MFnAttribute
  - Base class for Maya DG attributes
  - Offers methods to create an attribute or query/set properties of an attribute on a node

Function Set Classes & MObject

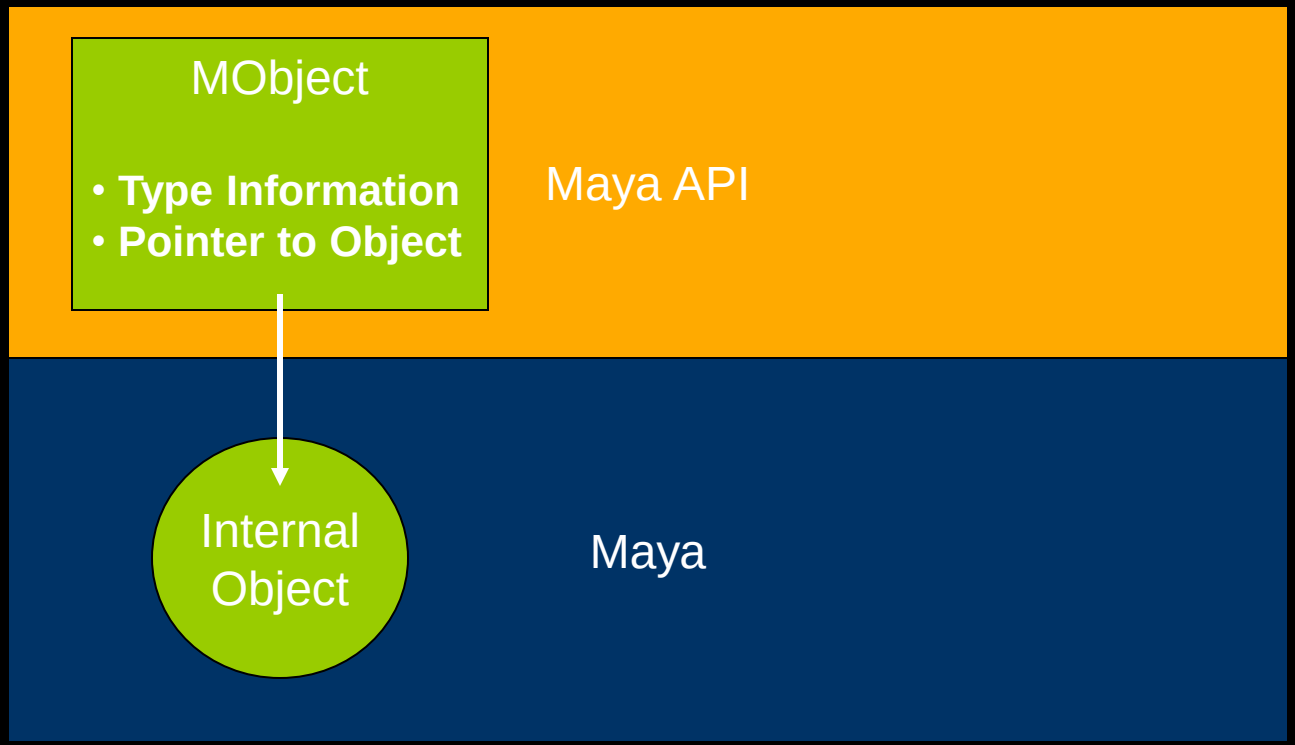
- Separate data with functionality

Classical OOP VS. Maya API Approach



MObject

- MObject is the fundamental data type that represents an object in Maya.



MObject

- Objects owned by Maya are accessed via an MObject
- MObject class only has two method to get:
  - (void *)
  - (type information)
- MObject's are handles to Maya internal objects
- Maya objects are created and destroyed by Maya

MObject (con'd)

- Use function sets class to operate on MObjects
- Used extensively. Every plug-in uses it.
- This class guarantees that Maya owns the data
- MObject resides in the maya.OpenMaya module
- It can be created using

```
mObj = maya.OpenMaya.MObject()
```

MObject & MFn::Type

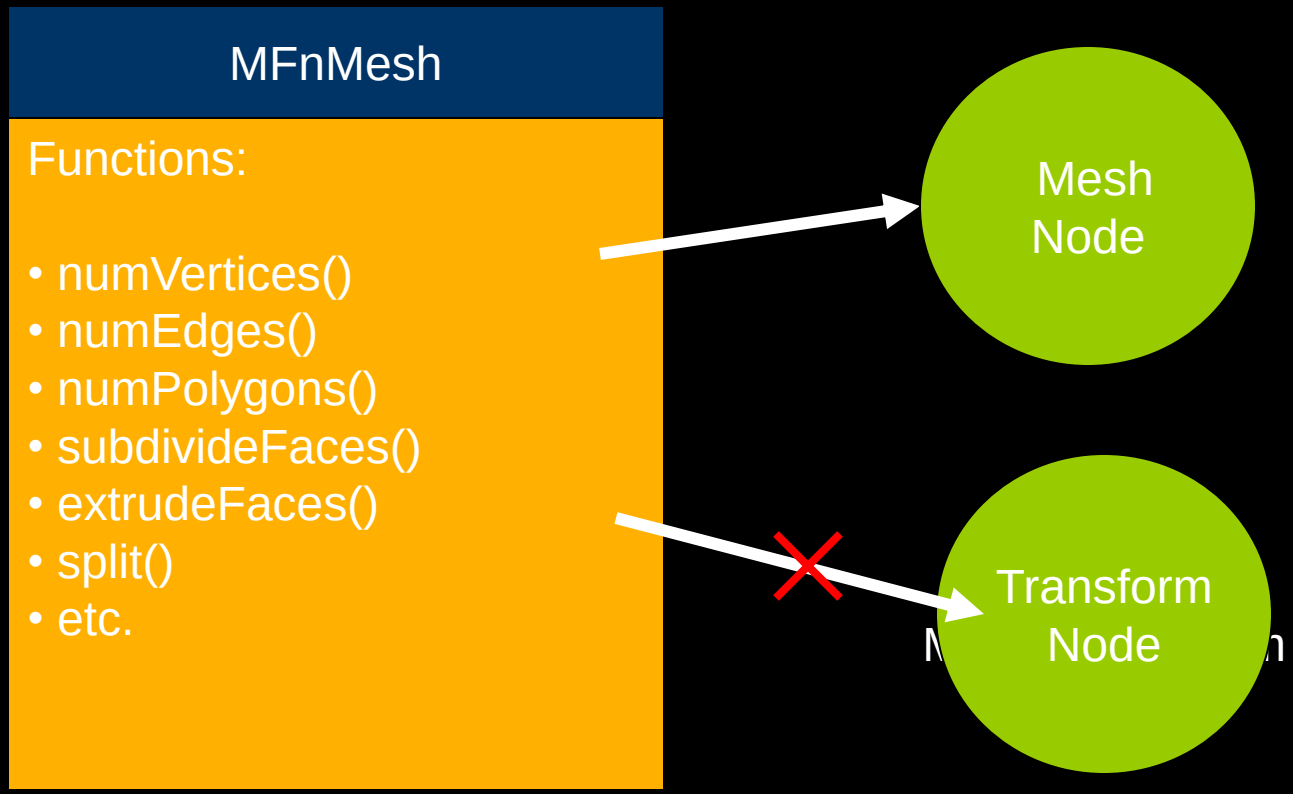
- Each MObject carries a type field.

```
MFn::Type MObject::apiType()
```

- This type comes from an enumerated list of all node types internal to Maya.

- For a comprehensive list of all Maya node types, see MFn.h.

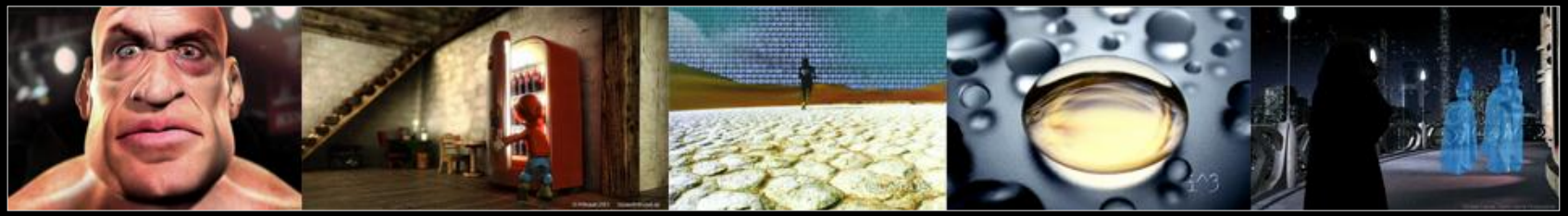
MObject and MFn::Type



MObject

- As a pointer to internal objects:
  - MObjects are not guaranteed to be valid between calls to your plug-in.
  - It is strongly recommended that you do not hang onto an MObject between calls to your plug-in. Use it as soon as it's created.
  - MObjectHandle can be used to test the validity of an MObject.

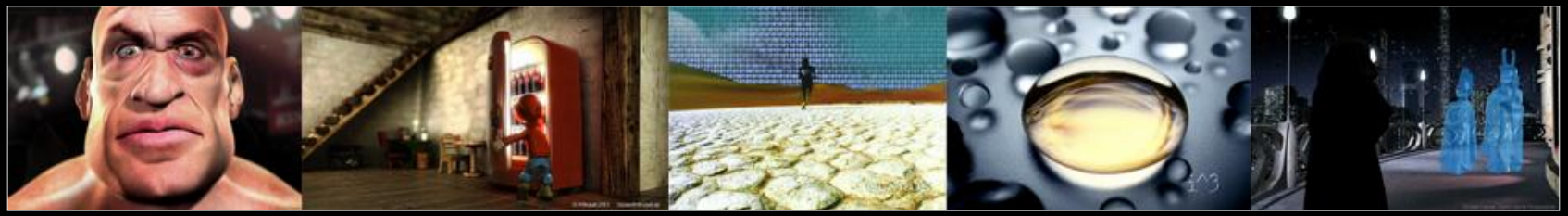
Iterator Classes



Iterator Classes

- Used to loop over elements of the same type
- Iterators start with “MIt”
- Some common iterators are
  - MItDag
  - MItDependencyGraph
  - MItMeshEdge
  - MItMeshVertex
  - MItMeshPolygon
  - MItSurfaceCV

Wrapper Classes



Wrapper Classes

- Wrapper class exist for simple classes (such as MPoint, MVector, etc…)
- Fully implemented Python classes

Autodesk