

Basics of MAXScript

In this tutorial you will learn the fundamentals of MAXScripting through three different scripts that will help you be more efficient when working in 3ds Max. Let's start with the tools we are going to use to create our MAXScripts: the MAXScript Editor and the MAXScript Listener.

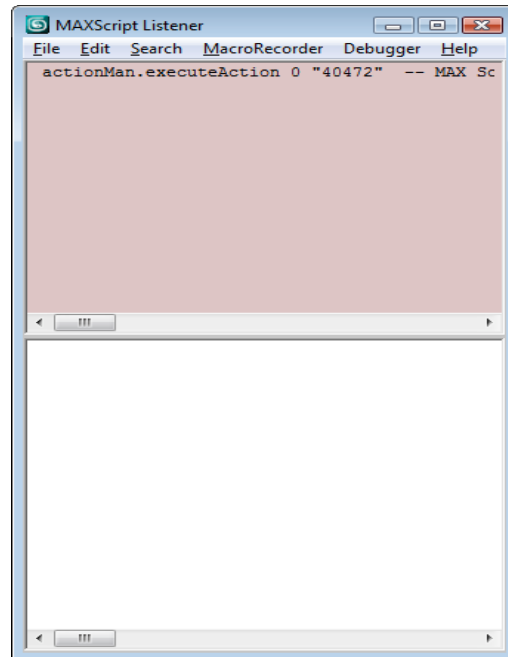


Figure 1 – MAXScript Listener. Use the top window to type scripts to evaluate, the bottom window also works to type scripts but it's used to display results

The MAXScript Editor can be opened by going to MAXScript Menu > MAXScript Editor.

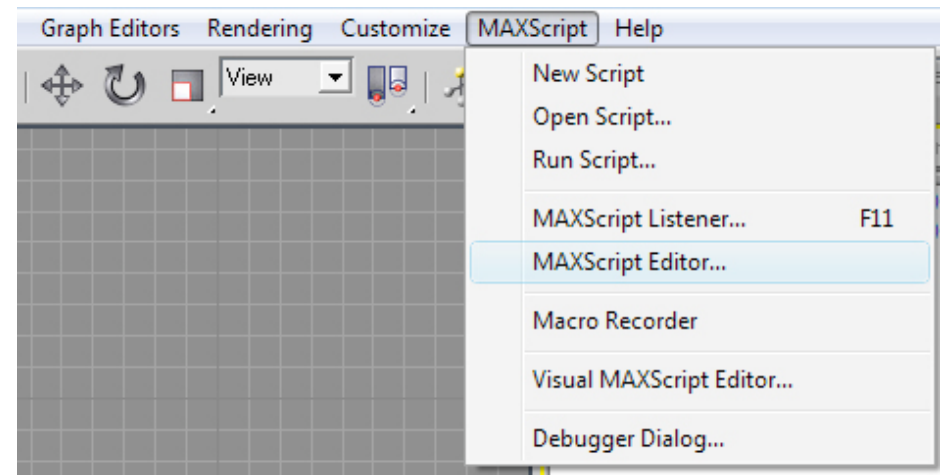


Figure 2 – The MAXScript Editor can be opened through the MAXScript Menu

This will start the MAXScript Editor with a new empty tab, or with the last scripts you had open the last time you closed Max. You will use this to create your scripts. It gives you the power to test your script while you are writing it and gives you visual clues on your syntax and how your code is structured.

```

1  -----
2  -- CD_AddLightMap.ms
3  -- created 12.03.08
4  -- by Carlos Dominguez
5  -- DESCRIPTION: Add a new UV Channel for LightMap (flatter UVs)
6  -- Simple script to cover basics of
7  -- manipulating the Stack and Modifiers
8  -----
9
10
11
12 rollout rAddLightMap "Create Light Map"
13 (
14   spinner spinnerMapChannel "Map Channel " pos:[10,10] range:[1,5,2] fieldWidth:30 type:#i
15   spinner spinnerSpacing "UV Spacing " pos:[10,35] range:[0.01,0.1,0.02] fieldWidth:40 scale:
16   button buttonAddLightMap "Add Light Map" pos:[30,65] align:#center tooltip:"Add new Light
17
18   on buttonAddLightMap pressed do
19   (
20     MapChannel = spinnerMapChannel.value
21     UVSpacing = spinnerSpacing.value
22
23     global selMeshes = selection as array
24     global sel = selMeshes.count
25
26     for i = 1 to sel do
27     (
28       select selMeshes[i]
29       selObj = selection [1]
30       AddLightMap selObj MapChannel UVSpacing
31
32       clearSelection()
33     )
34   )
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
259
```

Other useful tools to develop MAXScripts are the MAXScript Reference Guide (from Main Window go to Help > MAXScript Reference, or Help menu from Script Editor or Listener) and the internet. Search for pieces of scripts or code others have created and try to understand how the script works. Learn how to break down the code and then build your scripts from this knowledge. Some helpful sites are the Autodesk AREA Network, ScriptSpot.com and Highend3d.com

Basic Elements of Scripting

In the last section, you learned about the tools you will use to create your scripts. Now we are going to jump right into scripting, writing your first snippet of code, and running your first MAXScript.

First Thing... Label Your Script

You will need to get into the habit of commenting as much as you can. This means you insert lines in your code that are not evaluated (executed as part of the code) but just have comments of what you are doing or defining at that point. MAXScript uses two dashes (--) as a comment tag. So, you will use this to create a label for your script. Once you come up with something you like, you can just copy/paste it into all your scripts. This is how you will start your script.

Your First Script... GreyShader

Open a new window in the MAXScript Editor by going to File > New. You will get a new empty tab where you will start creating your script. Now type in your label as a comment like this:

```
1.
2.  -- CD_GreyShader.ms
3.  -- created 12.03.08
4.  -- by Carlos Dominguez
5.  -- DESCRIPTION: Simple script to cover basics of: variables, loops, functions
6.  --
7.  -----
8.
```

NOTE:

Every time I show you example code, I will frame it in a table and use an Italic font so you can differentiate between code and the written tutorial. I will also use the color highlight syntax used in the MAXScript Editor.

Let's break these lines of code:

- Line 1 and 7 are just used to create a frame for the label. Just like the other 5 lines, you start the line with two dashes (--) so it's a comment line.
- Line 2 is what you will name your script when saved. We will save this script in a moment and that is the name you will use to save your script.

NOTE:

The name starts with CD, my initials (Carlos Dominguez), so use your initials at this point. If this was a script for High Moon Studios you would start the name with HMS. When you name script, it's a good practice to use Camel Case. Camel Case is when you type multiple words together and you always make the first letter of each word Upper Case. So instead of typing Grey shader you will type GreyShader. You will use the underscore (_) when you want to separate different parts of the name, like who is the owner of the script, then the name of the script, and finally the version of the script (i.e.: CD_ScriptName_1)

- Line 3 notes the creation date
- Line 4 is the author of the script. Sometimes the author is different than the owner so you will want to always include this.
- Line 5 is the description of the script so anyone can open the script and know what it is for and what it can do.

Save your Script by going to File > Save

Remember to use the same name as you put in your label (CD_GreyShader.ms for mine) and pick either the default directory, or make a directory where you will keep all your scripts.

Declaring Variables

The next thing you want to do is declare your variables. Variables are like boxes that hold information for you until you need it, and they can be Local (it only exists inside that function) or Global (exists everywhere in Max until you close the script). This is the Scope of the Variable. You need to give variables a name that makes sense, a name that can tell you what the content is. It is always good practice to use Camel Case format and always start the name with a lower case letter.

The information inside the variable can be of different types, and based on the function or execution you may need a specific type. Some of the data types are:

- **String:** any text, including numbers and symbols
- **Integer:** holds numbers with no decimals (not 1.5 or 2.756, just 1 or 2, etc.)
- **Float:** numbers with decimals (i.e. 2.756, 3.1416, etc.)
- **Boolean:** just holds *True* (or 1) or *False* (or 0)

To declare a variable all you have to do is type your variable name, the equal sign (=), and the value you want. For example:

```
myVar = "Hello"
myNumber = 10
```

Notice you have to use quotes (") to tell MAXScript that is a String.

Sometimes you will want to change the variable type, and we can do this by defining a new variable that takes the value as a different type.

For example, if I have a number 6 that I want to convert to the string "6", all you have to do is:

In the MAXScript Listener, type in the top window:

```
myNumber = 6
```

This will declare your Integer variable. When you hit Enter, you will notice the value of your variable show in the bottom window.

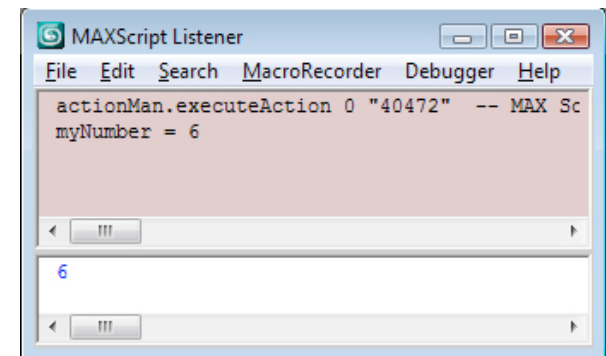


Figure 5 – Declaring your first variable

Now, let's change the type by typing in the top window:

```
myString = myNumber as string
```

When you hit Enter noticed the value of myString is now "6".

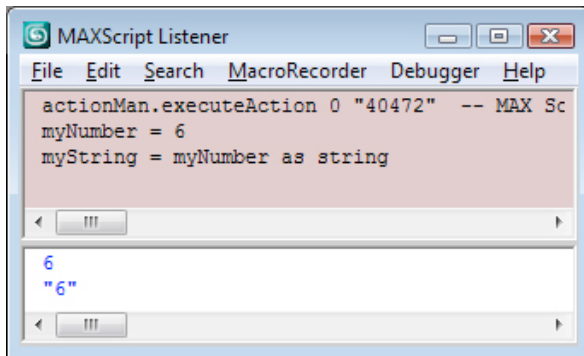


Figure 6 – Converting variable types

Now we have two (2) variables with different values. You can practice by assigning a different value to your already declared variable myNumber. Just like before type myNumber = and enter a number you want to store there.

If you want to print, meaning you want to see what the value is of a variable on the bottom window, just type in the top window of the MAXScript Listener the command print followed by the name of the variable and hit enter like this:

```
print myNumber
```

You will see the value show up on the bottom window.

Now, go back to the MAXScript Editor and declare a variable that will define what you have selected. To trap the value of a selection you want to use selection. So this is what your code will look like:



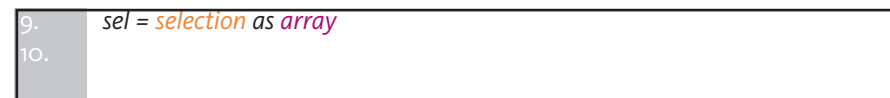
The name of my variable is sel and its default value (the value when I declared it) is selection. This means my variable now holds the name of the object or objects I have selected in my scene.

Arrays

When you have a selection, usually you want to do something with that selection. You want to use it as a list of things that you are going to change in some way. If we want to work with the objects in our selection, we need to convert this selection into an Array.

The definition of an array is a group of indexed elements. Why indexed? Is like having a list where you put each object into a numbered space, so we can later call space number 2, and we get the object in that space.

To convert our selection to an array you will use the same syntax as converting an Integer to a string but use "as array" instead. So update your line 9 like this:



Say you want to test what you have written so far, you want to evaluate your code. You can do this by clicking in the MAXScript Editor... Tools > Evaluate All (or Ctrl+E) and this will run your code in the current scene. You will notice that you can also just evaluate a line or just a selection of your code by selecting it and then clicking on Tools > Evaluate Line / Selection (or Shift+Enter or the num pad Enter).

Create some meshes in your scene, select them and Evaluate your code. Say if you created 2 boxes and select them, once you evaluate your code you will get a print out on the bottom window of the MAXScript Listener that shows: `#$Box:Box02 @ [-1177.232544,-277.807922,0.000000], $Box:Box01 @ [-645.178467,578.898438,0.000000]`

Now you know how to evaluate your code and test as you write.

Conditions

When you write a script, you can make your script dynamic by using conditions. This means based on the answer (result or return) to a question (condition), your script will execute different commands or functions. Now we get into the interesting parts of coding!

One of the most important is the **IF** statement. **If** your name is Robert, **then** your Nickname is Rob, **else** (or if your name is something else) your Nickname is Guy. That is how you create an IF statement, and how you can build conditions into your script.

What happens when you evaluate your code, but nothing is selected? If you look at the MAXScript Listener you get `#()`. What happens when you print the value of your variable `sel`?

Remember how we print the value of a variable? You will type in the top window of the MAXScript Listener (or in the MAXScript Editor and just evaluate that line):

```
print sel
```

The result, which we can see in the bottom window of the MAXScript Listener is:

```
OK
```

What does this mean? Well, it's telling you Ok, I can print the value of `sel` because that variable is defined (it has been declared), but it's empty, so I will not return anything.

To build a good script, and you should look for this, we are going to check and make sure the user has something selected. If they don't we can warn them to select something. This is how you can do that:

```
11. if sel.count !=0 then
12. (
13. some code we run here
14. )
15. else
16. (
17. some other code to warn them to select something
18. )
```

Breaking down the code you can see that Line 13 handles the condition: If the selection count is not zero, then do something, else do other code. Let's stop and point out two things here.

First, now that our variable `sel` contains the value for the selection array, `sel` inherits all the properties that an array will have. One of those properties is how many elements are in the array. Access this property you use the following syntax:

```
object.property
```

The dot gives you access to the property, like establishing a relation between a child and a parent. And so we can find what the count or number is of elements of the array. In this example we compare it is NOT equal. These are the symbols you will use to compare two values:

- Equal =
- Not Equal !=
- Greater Than >
- Greater or Equal >=
- Less Than <
- Less or Equal <=

The second thing to notice is the indentation. When you write code you need to get in the habit of being clean and adding lots of comments so that you or someone else can come back to that code later and understand what each section does. Remember, MAXScript evaluates each line at a time. If you need to execute multiple lines of code when your condition is true, then use parenthesis “()” to indicate the beginning and end of a block or section of code which needs to happen in a specific case.

The last thing we need to do is add the code we want to execute when our conditions apply. If our selection count is NOT zero, that happens when we have something selected, then lets *print* “Hello!” or some other message. If that condition does not apply then *print* “You need to have something selected”. Now update your code like this:

```
11. if sel.count != 0 then
12. (
13.   print "Hello!"
14. )
15. else
16. (
17.   print "You need to have something selected"
18. )
```

Evaluate your script (in the MAXScript Editor hit Ctrl+E) and see the results in the MAXScript Listener.

You will want to test if you are getting the proper count, select multiple objects and try to print out how many objects are in the array. What do you think would be the proper syntax for that line? If *print* “Hello” is used to print the word “Hello”, then to print a variable or a value you would use the variable name and no quotes like this:

```
print sel
```

But that will give you the whole array with objects. If all you want is that count... I think you guessed it! Let’s update Line 15 so you can print how many objects are in the array.

```
13.   print sel.count
```

Now, select some objects and evaluate your code. Test it a couple of times with different selections and see how it works. What happens when you have nothing selected?

Loops

Sometimes you want to repeat the same operation on a series of objects, maybe on an array of objects like a selection. For this case, you will want to use Loops. Here you will learn how to use the most common one: the For Loop.

Like conditions, the for loop will compare to see if a condition is true, and if so it will go into a block of code. It will continue to repeat running that block of code until the condition is false or something “breaks” the loop.

The basic syntax for loop is:

```
for obj in sel do
(
    do some stuff
)
```

Breaking up the code you can see that for each Object (*obj*) in the array (*sel*) do some code.

What if you want to loop a specific number of times? Then you can write:

```
for i = 1 to 10 do
(
    print i
)
```

Now you are temporarily declaring *i* as a number that starts at 1 and will increase 1 every time the loop is repeated up to ten times. Once it reaches 10 it will run the block of code and exit the for loop.

Back to your script, add a for loop that will print each of the objects in your selection. This means we want to add the loop after we confirm the `sel.count` is not zero. It would look something like this:

```
11. if sel.count !=0 then  
12. (  
13.     for obj in sel do  
14.     (  
15.         print obj  
16.     )  
17. )  
18. else  
19. (  
20.     print "You need to have something selected"  
21. )
```

Lines 15 to 18 are very straight forward: you are temporarily declaring a variable (`obj`) which will hold each of your array's elements (your selection) one by one as you go through and print the value of that variable.

Again, select (or not) some objects in your scene and evaluate your code. You will see your objects printing line by line in the bottom window of the MAXScript Listener. When the script is finished an OK will print to indicate the script has finished running with no errors.

As you can see, it's key to test your script as you write it, section by section, to make sure all the pieces are executing correctly. You want to catch the syntax errors and the bugs when they happen, early in the process. If you wait until the end to test, you may find yourself debugging hundreds of lines of code with no real idea where the problem is or where to begin. Keep it organized and contained.

Now you have learned the basic structures of scripting. With these tools you should be able to start learning how to change object properties, create tools for enhancing workflow, and work with file systems and interact with other applications.

In the next section you will learn how to manipulate object properties, using functions and making user friendly scripts. If you are ready for this, here is a little taste of manipulating the properties of meshes you have selected.

Let's add a few lines to your script so your code can change any selected mesh into a grey shaded object.

Since you are going to change the Wireframe color of your mesh and the Meshes color (diffuse color), declare variables for those next to where you declared the selection variable. It should look something like this:

```
wireColor = [0,0,0] as color      -- for WireColor  
objColor = [150,150,150] as color -- for Object Color
```

Remember that color is defined by a vector (3 numbers) Red, Green, and Blue, or R, G, B, with values from 0 to 255. Next, you need to update Line 15 so instead of printing the object it will change the properties of the mesh. You will change the object's wire color by assigning the value of your new variable `wireColor` to it. Then you will add a material to your objects, and that material will assign the diffused color you defined in your `objColor` variable.

Your whole script should look something like this:

```

1. -----
2. -- CD_GreyShader.ms
3. -- created 12.03.08
4. -- by Carlos Dominguez
5. -- DESCRIPTION: Simple script to cover basics of: variables, loops, functions
6. --
7. -----
8. wireColor = [0,0,0] as color           -- for WireColor
9. objColor = [150,150,150] as color     -- for Object Color
10.
11. sel = selection as array              -- for Selection
12.
13. -- Check if there is something selected
14. if sel.count != 0 then
15. (
16.     for obj in sel do
17.     (
18.         -- change wirecolor to black
19.         obj.wirecolor = color wireColor.r wireColor.g wireColor.b
20.
21.         -- change material color and apply it to the mesh
22.         meditMaterials[24].Diffuse = color objColor.r objColor.g objColor.b
23.         obj.material = meditMaterials[24]
24.     )
25. )
26. else
27. (
28.     print "You need to have something selected"
29. )
30.

```

Manipulating Objects in Your Scene

In the last section you learned the basics of scripting, and at the end got a taste of making a script that yields results in the scene. In this section you will learn about functions and making your scripts work as a part of the working pipeline of Max.

Functions

When you are writing scripts, you need to make sure you are efficient and not repeating commands or steps when not necessary. You practiced this when using a for loop to repeat changing the color of each selected object. Now you are going to learn how to use functions to make your code more flexible and efficient.

A function is a set of commands, almost like a mini-script inside a script, which will run every time you call this function. Functions usually receive values to use inside as variables, also called parameters. These parameters are local variables that only exist while the function is running. Once it finishes executing, the parameters no longer exist. Some functions not only do something but also return a value (or values) that can be used later in the script.

In MAXScript, functions are defined with the following syntax:

```

fn functionName parameters =
(
    Function code
)

```

Where functionName is, the name of your function and parameters are the name of the variables you are going to use. If there is more than one parameter you need to separate them using spaces like:

```
parameter1 parameter2 parameter3 etc.
```

Now, let's convert your main code into a function we can use when we need to. All you have to do is wrap your IF statement (include the selection variable) into your function like this:

```

8. blk = [0,0,0]
9. grey = [150,150,150]
10.
11. fn setColor wColor oColor =
12. (
13.     sel = selection as array           -- for Selection
14.
15.     wireColor = wColor as color
16.     objColor = oColor as color
17.
18.     -- Check if there is something selected
19.     if sel.count != 0 then
20.     (
21.         for obj in sel do
22.         (
23.             -- change wirecolor to black
24.             obj.wirecolor = color wireColor.r wireColor.g wireColor.b
25.
26.             -- change material color and apply it to the mesh
27.             meditMaterials[24].Diffuse = color objColor.r objColor.g
28.             objColor.b
29.             obj.material = meditMaterials[24]
30.         )
31.     )
32.     else
33.     (
34.         print "You need to have something selected"
35.     )
36. )

```

If you break down the code, you can see our function is called setColor and has two (2) parameters: wColor and oColor for the wire and mesh colors respectively. Also you are redefining the variables wireColor and objColor inside our function using the parameters as colors. This allows you to pass any vector as wColor or oColor and your function recognizes it as a color. If you don't do this inside your function, you will get an error because the vector won't be recognized as a color.

How do you run your new function? Well functions are great because once you define them (evaluate the function) you can call them anytime you want as long as Max is open. Once you close Max, all functions are removed from memory, and you need to run them next time you open Max if you want to use them. Here is how you name a function:

```
functionName parameter
```

From this example, to run your function you need to evaluate your function first. Evaluate your whole script, including the two (2) variables (blk and grey). Now just type your function into the MAXScript Listener top window and hit Enter:

```
setColor blk grey
```

or

```
setColor red blue
```

If you had something selected, your mesh (or meshes) turned black and grey or red and blue. How come the second time you ran the function you could just type red and blue and not have to define those variables? You may have guessed that those are reserved words in Max. They are pre-defined variables that Max already has defined as the red color and the blue color. In fact you can use black and gray, which are other pre-defined colors. You can always tell which words are reserved in the MAXScript Editor by the color highlight.

UI: the User Interface

You learned to add comments to your script so other scripters or programmers (or even you after a year) can figure out how it works. By the same token, when you are creating scripts and tools you always want to make sure they are easy to use for other people. Here is where the User Interface or UI comes into play.

The UI is an integral part of creating proper tools whether you are scripting for yourself or for others. In MAXScript you create UIs by using floaters or windows which contain rollouts. There are different types of rollouts you can use depending on what you need. In the following figure you can see some of the options, which you can find under the Help menu by searching for Rollout > User-Interface Controls and then select the Rollout User-Interface Control Types. You can click on the link of the control you want to get a description, syntax, and an example for the specific control you want.

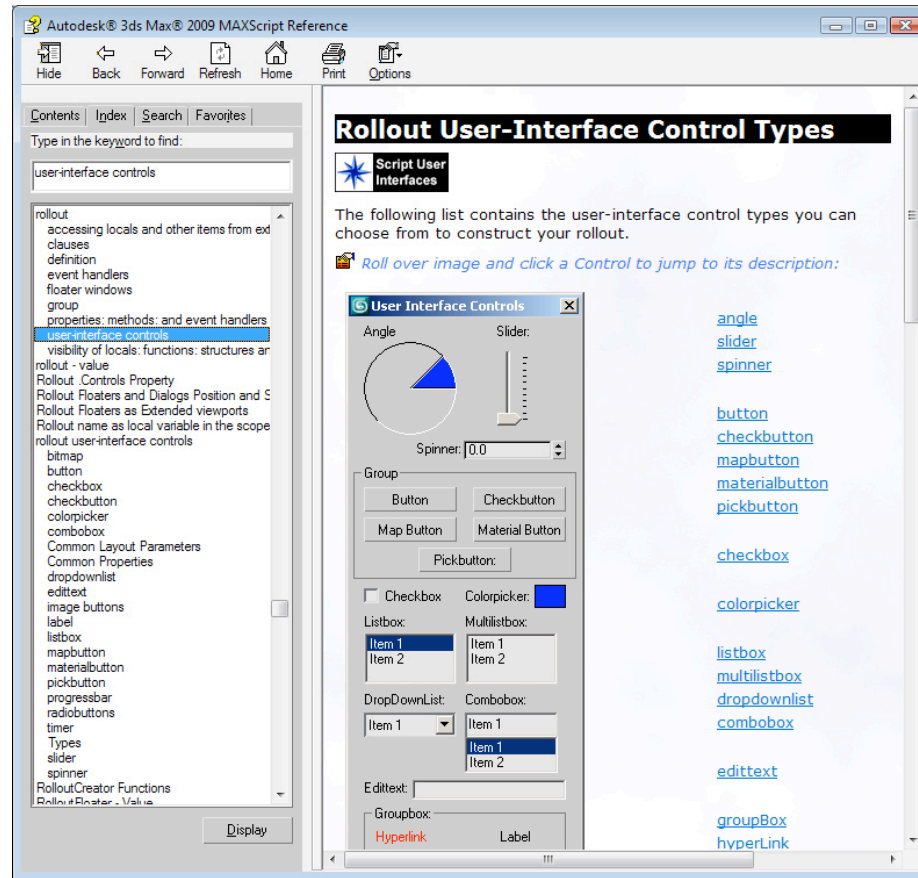


Figure 7 – Rollout Control Types

To create a UI we need a rollout, which has all your buttons and controls that will call your functions in your script. Then you will create a floater (or window) where you can add these rollouts to. The basic code for a rollout looks like this:

```
rollout rMyRollout "My First Rollout"  
(  
    button bMyButton "My Button" tooltip "my Tip of what this button does"  
  
    on bMyButton pressed do  
    (  
        messagebox "You pushed My Button"  
    )  
)
```

The first line defines your rollout with the name `rMyRollout` (I like to start UI elements with an initial of what they are: `b` for button, `r` for rollout, `f` for floater, etc) and the label `"My First Rollout"`. The label is used as a display text on your rollout. Looking into the rollout code block, you defined a button called `bMyButton` with the label `"My Button"`. When you hover on top of the button it will show a tooltip of `"my Tip of what this button does"`. Next you will establish what to do when that button is pressed: a message box will pop up and say `"You pushed My Button"`. This "on button pressed do" is also called an event handler.

You can test this by creating a new script (Ctrl+N) in the MAXScript Editor. Add the rollout code, and then after your last parenthesis you will create a dialog window to show your button, like this:

```
createDialog rMyRollout 150 40
```

The last two numbers are the dimensions of that dialog window (width of `150`, height of `40`). Evaluate your code (Ctrl+E) and check out your UI.

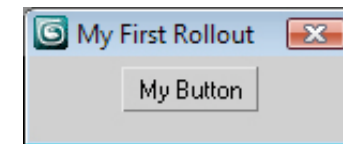


Figure 8 - My first UI

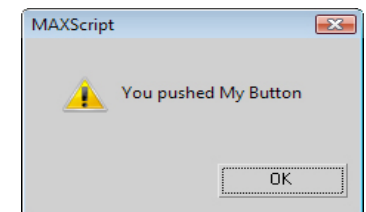


Figure 9 - Message Box

For your Grey Shader script you are going to use two rollout controls: button and colorpicker. Back in your script, after the function definition, you will add the code for the rollout, like this:

```

35.
36. rollout rGreyShader "Grey Wired Mesh"
37. (
38.     button bSelObjects "Selected" tooltip: "Apply grey material with color wire to selected objects"
39.     colorpicker cMeshColor "Mesh color:" color:[150,150,150] title: "Choose a color for the Mesh"
40.
41.     on bSelObjects pressed do
42.     (
43.         setColor black cMeshColor.color
44.     )
45.
46.     on cMeshColor changed newColor do
47.     (
48.         setColor black newColor
49.     )
50. )
51.

```

You declared a rollout that has a button and a color picker. By now you understand most of this code, so let's focus on the color picker and the event handlers for both controls.

Line 39 defines a **colorpicker** named **cMeshColor** that you are using to define the color you will give the mesh. The label is **"Mesh Color"** and you are assigning a default color of grey (RGB = 150,150,150). Lastly, the **title** is the label that will be used for the pop up window that you use to pick the color.

Next you have defined event handlers for the button and the color picker (**Lines 41 to 49**).

When the button is pressed you call your **setColor** function with parameters **black** for the wire color and **cMeshColor.color** for the mesh color. What is **cMeshColor.color**? You guessed it! It is the current swatch color in your color picker.

Let's go to **Line 46** and see what is happening on the event handler for the color picker. When the color is changed in the color picker, the variable **newColor** will hold that value (the RGB color) and the code block will be executed. Now on **Line 48** you are calling your function to run with the same parameters... except instead of using **cMeshColor.color** you are using **newColor**. We have defined our rollout and the event handler for it. Let's create our floater so we can see it in action.

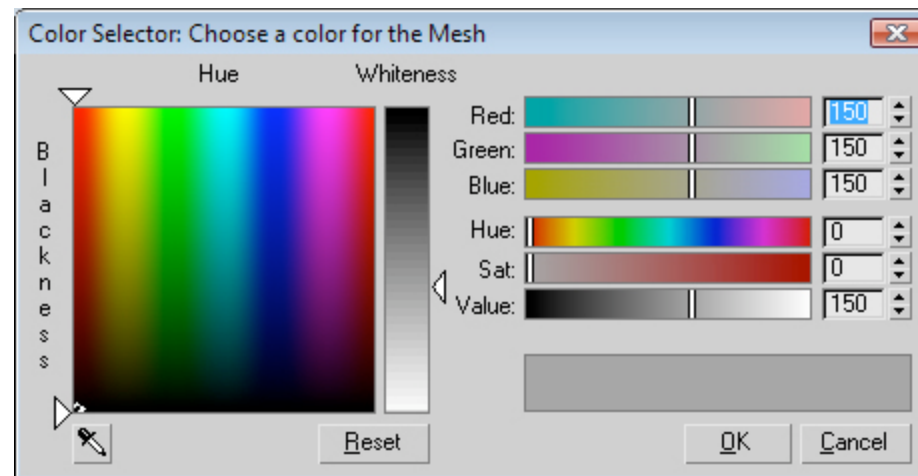


Figure 10 - Color Picker

Unlike the dialog box, a floater allows you to add multiple floaters and gives you more control over the look of it. The standard syntax for creating a floater is:

```
fMyFloater = newRolloutFloater "My floater label" 160 90 10 150
```

Your new floater will have a label (title) of "my n", will have dimensions of 160 by 90 and it will be positioned 10 pixels right and 150 pixels down from the top left corner of the screen. To add a rollout to the floater you will use:

```
addRollout rMyRollout fMyFloater
```

Let me show you how this looks in your script.

```
52.
53. if fGSFloater != undefined then CloseRolloutFloater fGreyShaderFloater
54. fGSFloater = newRolloutFloater "Grey Shade Mesh" 160 90 10 150
55. addRollout rGreyShader fGSFloater
56.
```

You have already learned what is happening on **Line 54** and **55**, but let me tell you about **Line 53**. Remember how it is always good practice to check for certain things to happen? One of them is to make sure your floater does not exist in your scene, or if it does then you want to close that floater. This allows you to create your floater without problems, and avoids having multiple windows open with the same thing.

Congratulations, you have written your first MAXScript tool!

Evaluate your code and see it in action.



Figure 11 - Grey Shader UI

The last thing I want to show you is how to make this script easily accessible through the toolbar. It's all about ease of use, so to do this you are going to add your script as a button on Max. All you have to do is select your code (Ctrl + A) and left click and drag it into your toolbar.



Figure 12 - Script button

Next time you start Max, that button will still be there and you can just click it to call your script.

In this tutorial you have learned the basics of scripting; declaring variables, and how to use them in your script. You now understand the importance of clean, efficient coding and the role loops and functions play. You created your first tool by adding a UI to your script to call your functions. Now you know how to use MAXScript to create tools that enhance workflow and make art creating pipelines more efficient.