

ALPHA PIXEL

- PYTHON CHEATSHEET -

Accessing An Object

To access an object with python in C4D you need to use either the `GetActiveObject()` or `GetActiveObjects()` functions. The main difference being that the first will grab a single object while the second will grab multiple objects by storing them in a list.

We need to store the object or objects we are accessing in a variable. For our purposes we will use set our variable name to "obj" to stand for object. So our code to access the object we want to select would be:

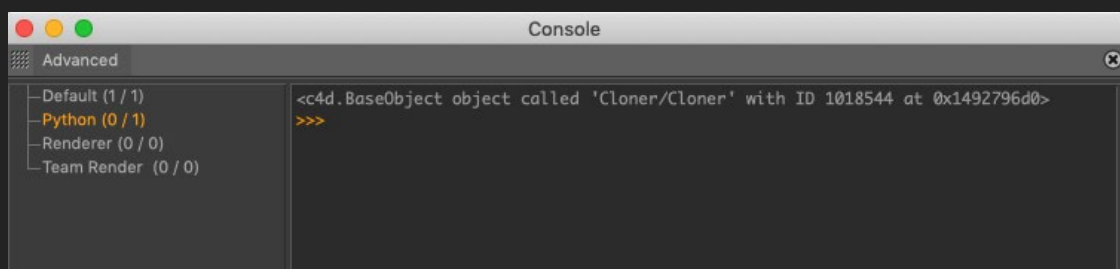
```
obj = doc.GetActiveObject()
```

You can test this code out by printing the contents of your variable(the object) to the console. Go to the Script menu and open the console so you can view the results. Then back in the script managet simply type:

```
print obj
```

If you had an object selected you will see it's information pop up in the console. Otherwise if you have nothing selected or more then one object selected you will get either an error or a message saying "None".

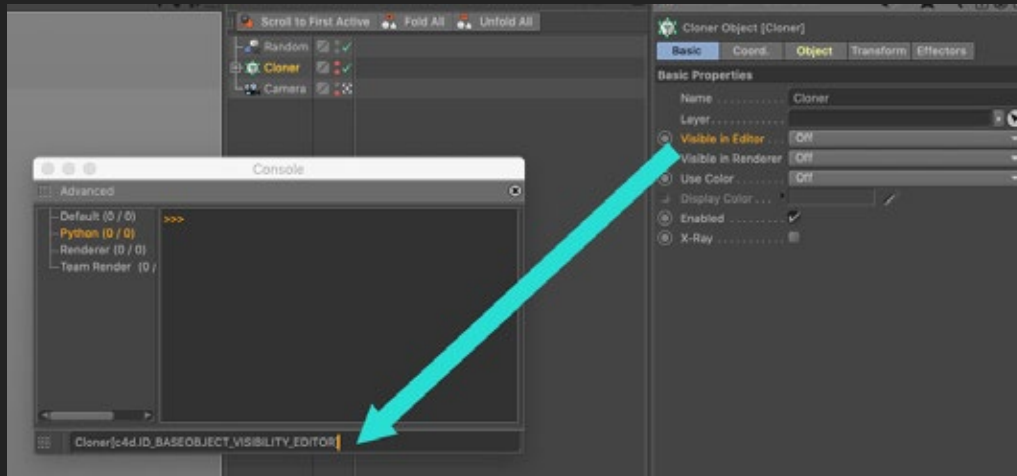
```
1 import c4d
2 from c4d import gui
3 # Welcome to the world of Python
4
5 # Main function
6 def main():
7     obj = doc.GetActiveObject()
8
9     print obj
10
11
12
13 # Execute main()
14 if __name__ == '__main__':
15     main()
```



If your console looks similar to this then you have properly accessed your object!

Accessing An Objects Attributes

To access an objects attributes you need to know the internal code for that attribute. Luckily that is very easy to find out via dragging any attribute into the input line of the console!



From there we just need to grab the code inside the brackets [] including the brackets themselves and paste it into our script manager. Then we just need to tell the code which object this parameter will be modifying. So we put the variable we created earlier in front of the first bracket. Then all that's left is to put an equals sign at the end and set the value of the attribute. For example:

obj[c4d.ID_BASEOBJECT_VISIBILITY_EDITOR] = 2

Just be sure you know what value your attribute has so you know how to set it. Here are a few types of values and their translation to python:

Integer = 1, 2, 3, 4, 5, or any whole number

Float = any number that isn't a whole number. ex. 1.45, .35, 1.00

Checkboxes = 0(off) and 1(on)

Type = anything with type. ex. "hello world" "Cinema 4D" "12345"

Before you hit "Execute" in the script manager make sure you follow your code up with:

c4d.EventAdd()

This will tell Cinema 4D that something has changed and to refresh the system. So your object will be modified. Otherwise you won't see a change until you do something another operation or even click somewhere in the interface.

```
6 def main():
7     obj = doc.GetActiveObject()
8
9     obj[c4d.ID_BASEOBJECT_VISIBILITY_EDITOR] = 2
10
11     c4d.EventAdd()
12
13
```

Accessing Multiple Objects

To access multiple objects you will need to first change your `GetActiveObjects` function to be `doc.GetActiveObjects(1)`. The key difference is the “s” after the word “object”. This also changes the output of the function from an object to a **list** of objects even if there is only one object selected. The “1” in the parenthesis is a flag which tells the `GetActiveObjects` function how to handle the selection. You can find all of the flags in the C4D Python Developers doc online but it is important to note that there does need to be a number inside the parenthesis for the function to work otherwise you will get an error. Now try printing the results of the “obj” variable and see how it is different. Just make sure you have some objects selected! You’ll notice all of the objects show up in brackets now `[]`, that is a clear indication within python that our objects are in a list.

If you want to access an individual object from the list you just need to put a bracket after your variable that holds the list and put a number inside the brackets. Note: the first object in the list is 0 not 1! Here is an example:

```
obj[3][c4d.ID_BASEOBJECT_VISIBILITY_EDITOR] = 2
```

With that code while you wrote 3 in the brackets you are actually accessing the 4th object because 0 is the first object. You will need to get used to this way of thinking within python and several other coding languages.

Now to modify all of the objects at once we need to utilize the very powerful **FOR** loop in python which will loop through all objects in our list. I will show you the code to make the FOR loop work and then explain it:

```
for x in obj:  
    x[c4d.ID_BASEOBJECT_VISIBILITY_EDITOR] = 2
```

We write the word “for” then we write in a temporary variable “x” that is just going to represent all of the objects we are going to run through in our list of objects represented by the “obj” variable we created earlier. Each time this code finishes it will replace the temporary “x” variable we created with the next objects. So anything we do to “x” is done to ALL objects. Here is what the final code looks like:

```
6 def main():  
7     obj = doc.GetActiveObjects(1)  
8  
9     for x in obj:  
10         x[c4d.ID_BASEOBJECT_VISIBILITY_EDITOR] = 2  
11  
12     c4d.EventAdd()  
13  
14
```

Congratulations! You just modified multiple objects in the click of a button!