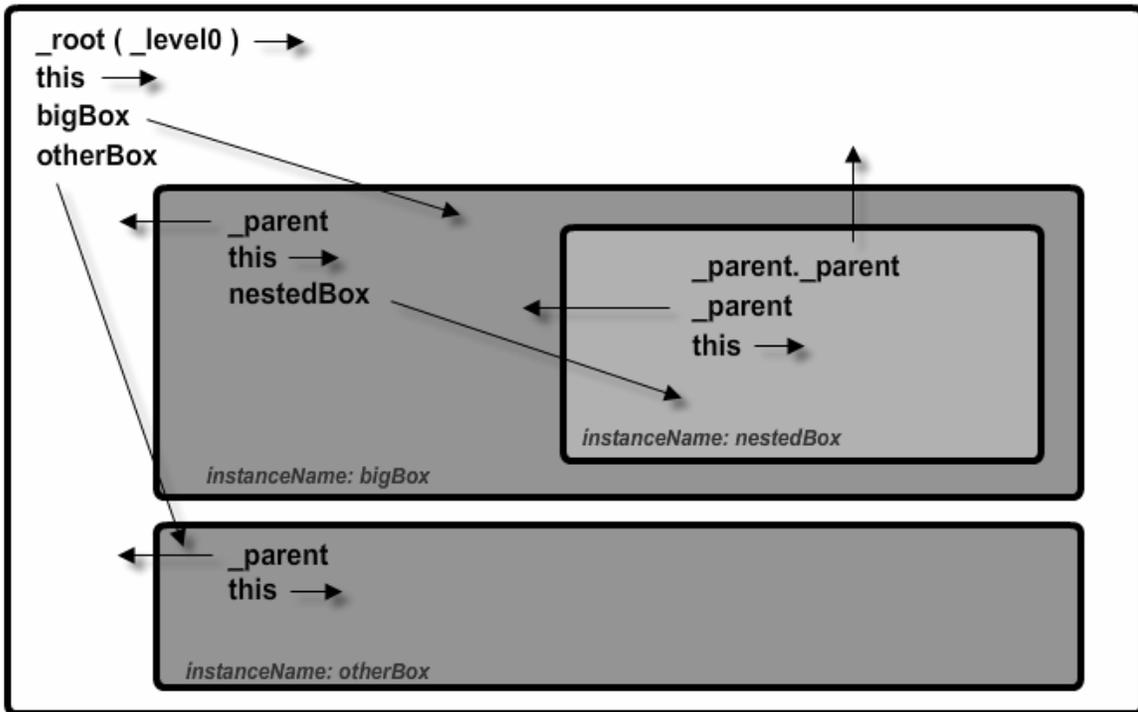
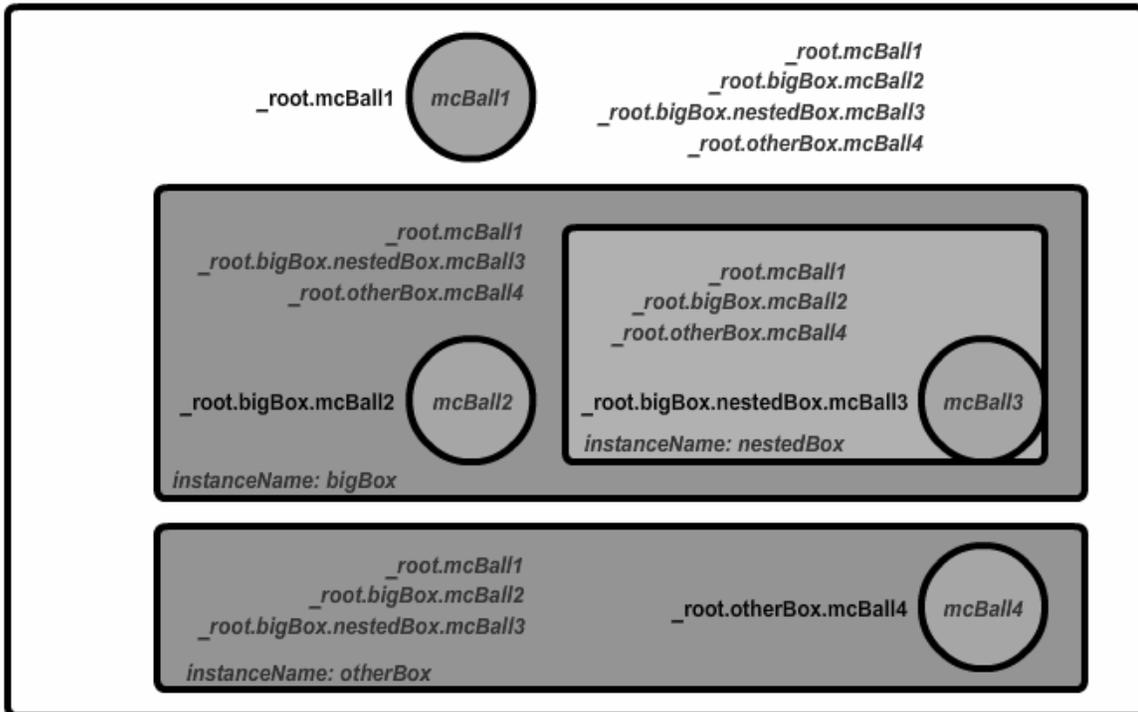


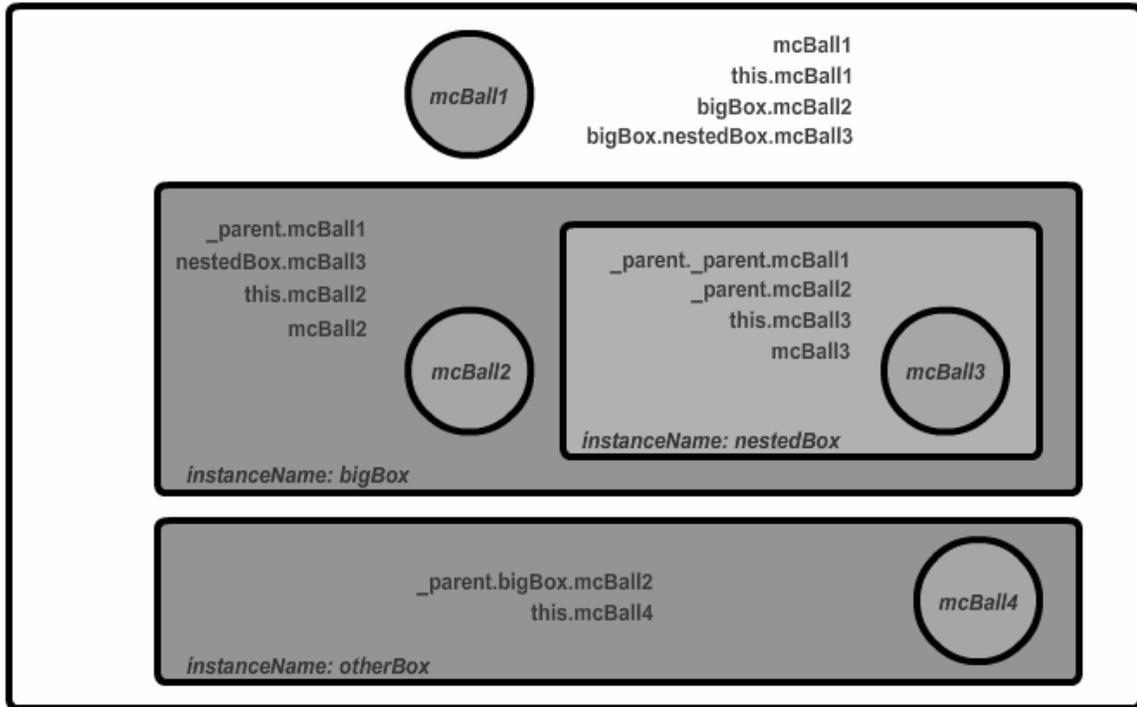
Can you give me something visual on all this?



What does an absolute path look like?



What does a relative path look like? (best practice)



What is the point of referring to objects in other Timelines?

It is a best practice to write your code in as few places as possible. In this course, we emphasize placing all code on the first frame of the top-most layer (when possible). It's important to know how to talk to objects nested inside other MovieClips from that location, ideally using relative paths.

For example, from the actions layer on the main document Timeline, you can change the `_alpha` of `mcBall3` using this code.

```
bigBox.nestedBox.mcBall3._alpha = 70;
```

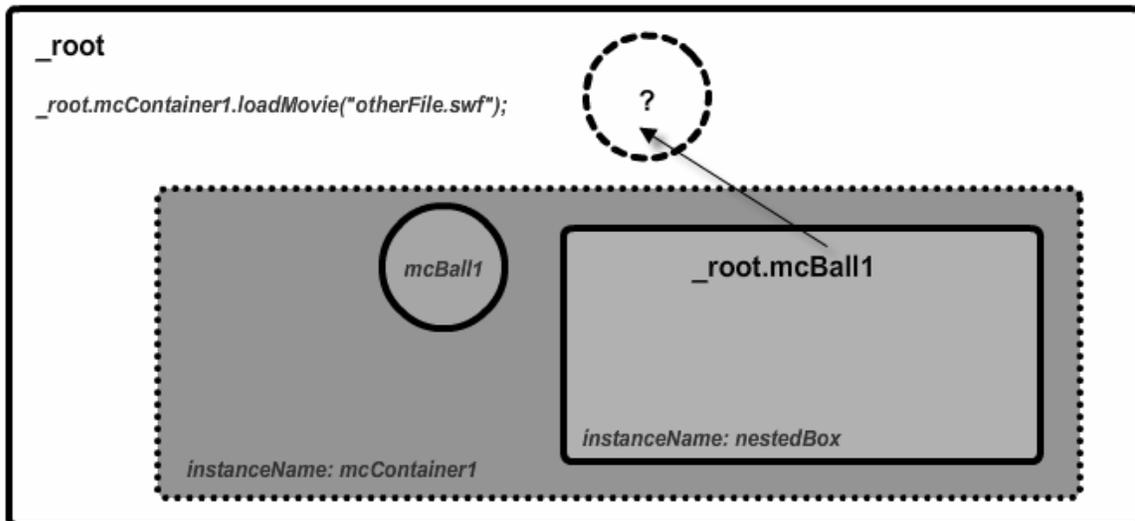
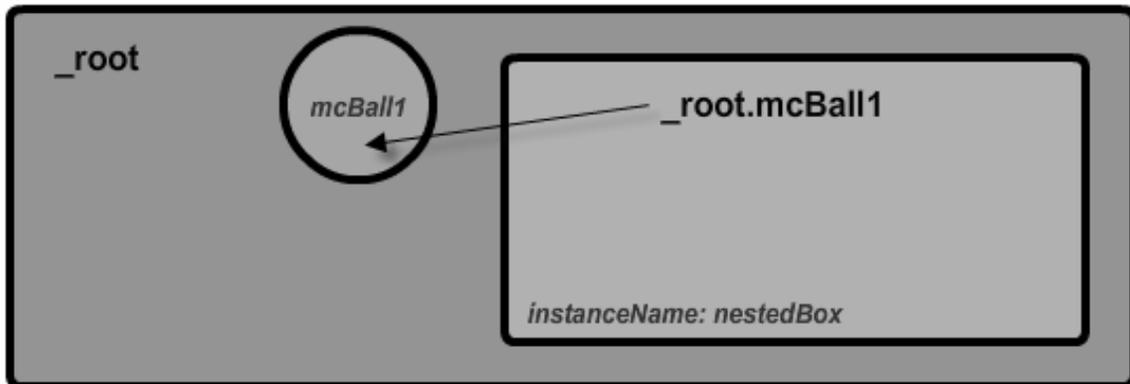
If `nestedBox` had a `TextField` within it, you could write text into it. Or, you could load an external SWF into `mcBall4`.

```
bigBox.nestedBox.txtFirstName.text = "Ginger";  
otherBox.mcBall4.loadMovie("assets/PulseBall.swf");
```

You can reference anything you need, all with easy to find and maintain code centralized in one location on the *main document Timeline*.

Why are relative paths safer than absolute paths?

Relative paths (ones beginning with **_parent**, **this**, or an *instanceName*) are always preferable to *absolute* paths (ones beginning with **_root**, or **_level0**). This is because **_root** always means the *main document Timeline* currently inside the Flash Player. So if your SWF gets loaded into another SWF (using the `loadMovie()` method), the meaning of **_root** changes and suddenly all of your code breaks.



_lockroot property

One approach to avoiding the problems created when using **_root** is to write the following at the top of your code on the *main document Timeline* in your FLA:

```
_lockroot = true;
```

This changes the meaning of **_root** in this document, so that it always refers to (is "locked" to) this particular Timeline where you wrote the statement above, even if the SWF gets loaded into another SWF. Opinions vary widely on whether this is a good idea. The best practice is still to use *relative* pathing, making **_lockroot** unnecessary.