

Valuable points from Lesson 9 (and some corrections)

Adobe Flash CS5 Professional Classroom in a Book

You are expected to know how to use/do each of these tasks in Flash CS5, unless otherwise noted below.

1. Do what the instructions say on page 332 (step 2). It's very important that you understand *from the beginning* that there are **four completed SWF files** (note: SWFs, not FLAs) that will be *loaded into* the file you are working on.
2. The two objects you're writing ActionScript for, here, are things you will likely use again and again if you continue working in Flash:
 - a. A Loader object
 - b. A URLRequest object
3. Error on page 334:
 - a. Name your actions layer *actions*; no one names it *actionscript*.
4. Page 334 (step 4): Note that all the other times when you wrote a **listener** like this one, you were writing a listener for a *button*. (In the baby monitor analogy, the *button* was the baby who cries.) This is the same, except that here the "baby" is a *movie clip*, not a button.
5. We use a **URLRequest** object to get external data files. It is also used when a button opens a Web page (pp. 334–335).
6. The author's movie clips are kind of weird. I don't understand why he put the **white plus signs** on them—I couldn't figure out any reason for them being there. But they do indicate each one of the four movie clips that he has arranged in his "pages" layer on the timeline. You need to give each of the four movie clips an *instance name*, as he instructs.
7. Page 336: **Please do** save and test your movie at this point (step 7) and recognize that you have scripted **ONE** of the four movie clips on the Stage to listen and then activate the Loader, which (via the function named "page1content") loads in the external SWF named *page1.swf*.
8. Now you will simply *do the same thing* for the *other* three movies clips. Copy and paste to save yourself unnecessary typing (and typos).
 - a. Make sure you give each movie clip on the Stage the correct (and unique) *instance name*.
 - b. Make sure to change *the function called by the listener* as well as *the function name* on the function itself.

- c. (Hey, a month ago you didn't know what any of this stuff meant!)
9. Page 339: After you have loaded an external SWF, usually you want to write a script to *unload* it. Think of `removeChild(myLoader)` as the opposite of `addChild(myLoader)`. And *why* (you may ask) is the word "child" used? Let's keep it simple and simply say that in object-oriented programming, objects have relationships, and one way these relationships are configured is parent-child. *Objects on the Stage are considered "children" of something else.* And if we do not include `addChild(myLoader)` when necessary, then the loaded content will never become visible.
- 10.** Another thing to note about the scripting on page 339: Adding that new listener and its function are what *makes it possible to go back* to the "contents page" of this package. *Without* the function named "unloadcontent," you'd be **stuck in the loaded SWF forever.**
11. Page 340 has a good explanation of itself. (I'm shocked!)
12. Pages 341–346: Creating masks is very cool, and very useful. Please follow these steps, but recognize that you could load external SWFs *without using any masks.* In other words, they are two entirely separate things. (In my fishbowl example of loaded SWFs, there are no masks!)
13. Page 342, steps 3–4: Last week I taught you to just *right-click the layer* and tell it to be a Mask. To me that is easier, less work, than the author's way.
14. Page 343, step 6: Usually you *do not need to tell a masked layer that it is masked.* It gets indented (under the mask layer) automatically. You can also drag other layers under the mask (so that they indent). You can drag out masked layers to *unmask* them. You can play around with this. Any *indented* layer under a mask layer will be affected by the mask.
15. Page 344, step 4: CONVERT TO SYMBOL FIRST!!! I don't know what's wrong with your book author. He's sloppy.
16. Page 436, step 12: Everyone except this guy would name that layer *actions.*
17. He neglects to tell you that the layer **lock/unlock** and the layer **hide/show** (red X) options are very, very useful *when working with masks* in the editing environment. Play around and you will see why.

More about masks:

<http://flashjournalism.com/CS4examples/masks/masks.html>