Reflection

Source Code and Single Sourcing

Adam Holley

University of North Texas

Digital Literacies

<?xml version="1.0" encoding="UTF-8"?>

<project2>

<title>Reflection</title>

<subtitle>Source Code and Single Sourcing</subtitle>

<author>

<name>Adam Holley</name>

<school>University of North Texas</school>

<class>Digital Literacies</class>

</author>

</project2>

# <introduction>

At the beginning of this project I didn’t understand how XML or XSL worked and was only slightly familiar with CSS. I’ve tried to learn basic coding in HTML in the past but was never able to reach a sustainable level of understanding. When we were first introduced to XML and XSL as a class, I anticipated a miserable series of classes in which I struggled to understand the new and confusing tools. Over the last 4 weeks however, my knowledge of these technologies has grown to the extent that I was able to create a working XML document that is transformed into a viewable page via XSL (and made aesthetically pleasing via CSS).

I’m a big scary movie fan and given the time of the year, I thought it would be useful to have a table that tells you the information on several scary movies. This table can be used when someone wants to watch a scary movie but is unsure on what they want to watch and doesn’t necessarily know the names of many scary movies. My table consists of almost all “classic” scary movies that are very popular, providing a good starting place for anyone new to the movie genre.

# <tagging\_system\_and\_hierarchy\_rationale>

I chose the tags for my XML document based on logic and simplicity. I tried to make the tags short but understandable so that connecting the document with my XSL file would not be difficult. In Figure 1 below I have provided the format I used for each movie entry and in Figure 2, I give an example of a filled out dataset.

Figure 2 Tag hierarchy (unfilled)

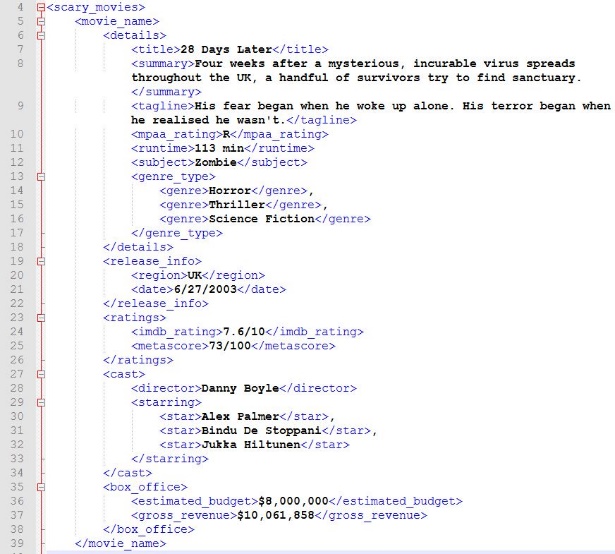


Figure 1 Tag hierarchy (unfilled)

I knew that I wanted to organize my information in a way that separated several sections (title, basic details, release information, box office information, cast and ratings). This seemed not only logical, but also is how I would want to view the data being presented if I was looking at my project from a third-perspective.

I created my hierarchy based on what information would be displayed in what order in my final table and nested any tags where I needed to add multiple data entries. For example, I nested the *director* and *starring* tags under their parent, *cast*, and then added an additional set of children to the *starring* tag so that I could input multiple actor/actress name. Each movie started off with the tag, *movie\_name*, which was followed by the child, *title*, where each movie was actually named. My hierarchy was easy for me to make since I just used logic to decide if any tags needed to be nested inside of another tag.

# <limitations>

I would like for my table to have included more films, possibly categorized by how well known they are. This would make for a more useful table for individuals who are familiar with the scary movie genre and are looking for something a little more obscure. I wish that I had been able to add more polish to the end product by adding additional pages, images, sortable content, etc. I plan to continue studying JavaScript and CSS so that I can implement them more in future work.

# <conclusion>

This project seemed daunting in the beginning, but learning the skills needed to complete the project really wasn’t too difficult. Overall, I found this assignment to be extremely beneficial. I am genuinely proud of myself for finally feeling like I understand something that has eluded my comprehension for such a long time. I plan on continuing to practice and perfect my new skills because I think that they will be beneficial to me in the future.