To: Chris Lindgren

From: Matt Muroya

Subject: DITA topic model design rationale

Date: 12/23/2015

My main topic model is a very short, condensed manual for building a basic responsive web page framework. In researching and examining frameworks for my personal use and for the projects in this class, I found a lot of very complex frameworks and also a lot of barebones, basic frameworks. I think the basic frameworks are the best for learning about what goes into the essential building blocks of websites and some best practices for web coding.

I like the idea of a very barebones framework, and I also like *not* having any code or components that aren’t going to be used for the sake of efficiency and full customization. The framework can be reused for any personal projects that follow similar guidelines. This task topic guides the user through creating his/her own basic framework, allowing for some flexibility and customization. Building a full responsive framework including a CSS grid is a very extensive task, so this task topic pared down a bit and simplified.

Since this project is more of a “mini” topic model, it doesn’t include a lot of the details that a more thorough guide might. Therefore, its practical application is somewhat limited, but I feel it is a good representation of what a more fully-considered guide would include. It is aimed at people with a basic understanding of HTML and CSS and how divs and classes work, but who may not be familiar with media queries and responsive design in particular.

I picked two major components that I thought were fundamental to a responsive framework rather than ornamental: a basic two-column grid (based on standard float grids) and a header text sizing scheme. These two pieces are important to visual page structure and content hierarchy, and would be something that anyone creating a text-best responsive web page would want to style. This has been done before moduarly, so the real value here is learning how it works and being able to create a unique grid and determine your own sizes and breakpoints using media queries.

The topic model contains a concept topic at the front of the structure that explains frameworks in general as they relate to the web. It explains what is included in a framework, how they work, and why a designer might want to use one. This sets the stage for the rest of the document and also describes how/why these might be different from person to person/project to project. Again, this assumes a basic level understanding regarding HTML and CSS (and web documents), but explains the idea of a framework in basic terms.

The first (and primary) task topic is creating a basic responsive CSS grid layout. The grid is very minimal, and includes a header, main content section, sidebar, and footer. The main and sidebar sections represent columns. There is no row class defined, but the container class functions as a row in the case of this task. In order to create this grid and determine breakpoints, the user should have a basic understanding of what a CSS grid is and how it applies to web page design layout. Therefore, I included the “CSS grid” concept topic as a related item to this task in particular. This is not necessarily related to the other task the document describes (the header scheme), so that is why it belongs as a subtopic to this particular task. I also included a reference topic here as well to provide some examples of common device screen sizes and dimensions for the user to determine breakpoints (as directed in steps 5 and 6. These might be helpful when the user is deciding where to add breakpoints and exactly what size they should be.

Using this framework might also require the coder to set the viewport meta tag in the head of the document so that the page renders properly on smaller screens/so that the zoom is set correctly. Therefore, I included a short task topic on how to set this initial viewport width as a related subtopic of setting up the grid. The shortdesc on this topic mentions a particular issue that the user might come across, so the user will know whether or not he/she needs to explore this topic after testing how the document renders.

I also included some conrefs here as well. Like I mentioned before, this full task topic is moderately limited in its practicality, so I implemented some phrase-level reusable content that could potentially be used to increase applicability to different situations. When I specified 568px as an example breakpoint for the media queries in my step examples, it was (sort of) an arbitrary number—It’s the approximate width of an iPhone 5 when viewed horizontally. This means that any device/screen at that width and narrower would collapse the main and sidebar divs in the framework. This is referenced a few more times throughout: in example codeblocks and later in the header task topic as well. They all use the same value for consistencyd so I thought that making this particular number reusable would make it easier for someone manipulating this content later to put in their own value here. So, I tagged it as a phrase-level element with an id of “breakpoint” so when being referenced later to reuse it, it’s a very semantic path: “conref=”t\_css\_float\_grid.css/#float\_grid/breakpoint.” The reference immediately makes sense to the reader and they know where to look for the iteam and what it means. This way, if you wanted to focus on creating for tablets or only focusing on vertical smartphone widths, you could replace this px value with whatever size you wanted (or replace it with ems if you’d rather size that way) and it would update in all of the examples and tasks across the document.

The second major task (third task in total, including the viewport task) is creating a basic header scheme. It includes the use of responsive media queries again to resize the headers based on the width of the device screen for readability. The task includes a conref that points to the breakpoint determined in the grid task to keep everything consistent for the reader with the same number used for the breakpoint. There is also another reference topic included here to provide some standard sizes for different types of headers h1–h4. I included this part of the framework because, while it is still relevant to the visual design and layout of a page, it also has a lot to do with semantic content structure as well, so it addresses a different user need while still fitting within the overall task of creating a responsive framework.

Overall, the task topic I chose to write on is a much larger set of goal than I was able to cover in this mini model (and without a comprehensive XML editor). However, the way I structured it gave it some relevance and practicality, especially with my use of phrase-level reusable content. I wrote shortdescs and topics that explained what each element is, what it does, and why a user might need to do it/why it is important. In the case of the sub-task of setting the viewport meta data, it addressed a particular user need as well that might result from following one of the other task topics. Looking back, if I were to expand on this project, I might even create a separate DITA map for the entire grid section, as it is such a large and complex task outside the scope of my small single-task representation of it. This map would then be included in a master DITA map that contained all of the different components to creating my basic framework.