Jennifer Soh

Mobile Applications: Design

Responsive Design Fundamentals

The video, Responsive Design Fundamentals, discusses the fundamental techniques for responsive design. These techniques are fluid grids for layouts, media queries to adapt content to specific displays, and flexible media that respond to display types. In addition to utilizing these techniques, a developer needs to rethink content strategies in consideration for display types, varying screen resolutions and screen density and other inherent challenges of trying to accommodate such a wide range of viewports. Developers will also consider how the information is accessed, speed, data usage, and usability problems that may be encountered on smaller screens and touch sensitive devices. They can also consider utilizing other features on the phone such as GPS, touch events, accelometers, and messaging.

Designing for Multiple Screen Densities

When planning a design for responsive sites, it’s important to consider multiple screen densities. While older models of the same phone might have a low screen density, newer models might have higher screen densities. Screen densities also vary from device to device. This is important because it influences how images and page will be sized on a screen and whether or not a high resolution will be needed to accommodate a high density screen. By using media queries and defining the size of an image with device size pixel ratios, coupled with a range of choices you can make regarding the use high density and low density images, image file types, drawing in CSS, using icon fonts, etc., you can accommodate a wide range of devices. However, this process can become quite tedious if and frustrating if you decide to create a graphic-heavy site for mobile screens. It is, therefore, recommended that the developer consider eliminating graphic images that aren’t absolutely necessary.

Understanding Media Queries

Media queries are an extension of media types. They allow designers to extend media declarations to include a number of media properties that media types don’t cover, such as screen size, orientation, device width, device height, color and more. Media query expressions contain media features that are evaluated to know when to apply types. With media queries, you can cover a range of screen sizes rather than try to define different classes for every device screen size the site will be displayed on. Using minimum width and maximum width styles, you can change how the site will appear at different breakpoints in your site, and refine at several other points in between.

Using Fluid Grids

Fluid layouts allow content to adapt by resizing and flexing within a certain range. Grid systems create fluid, properly structured layouts. They are “a number of columns and gutters that are then used by the designer to establish widths of page elements and regions”. These columns and base widths are usually related to a font size or ratio and many times it’s a base value that divides well into the width. The developer can then use the grid to map out coordinates of all the elements on the page.

Making Images Responsive

There are a few options for making images responsive, though none of them are prefect. One of the options available would be to replace fixed image size with a percentage. This can be use with both flash and other videos. While this might seem like an easy solution, it has its limits. The image will not resize relative to the rest of the page. You would also be relying on a larger image with a higher resolution that might take more time to download. Another option would be to use media queries and swap out images. The downside to this method is that it can only be used on images that don’t need to be in the html and can be placed in the CSS sheet. Another downside to this method is that all of the resources would have to be downloaded for any Android users with older devices.

One of the better options available would be to use a JavaScript program that can display the images based on the device that it’s being displayed on. The downside to this method would be that JavaScript parsing takes time. In addition to that, JavaScript programs will prefetch images and store them in the cache before they’re needed so that it will be available instantly. This leads to downloading assets that might not be needed. While there are some solutions being developed that might make responsive images less challenging, for now, those solutions have yet to emerge. For now, it’s important to takes those challenges into consideration when designing for mobile devices.

Responsive Forms and Improving Site Peformance

When designing a site, handling how forms are going to be displaying on mobile devices should be one of the top considerations. Designs should focus on usability, how it flows, how it’s stacked when condensed into a smaller screen, how it’s labeled and how easy it is to use in a touch screen environment. The user should be able to clearly understand what input is expected in each field and the flow of the form should have an intuitive flow to it. It’s important to make the mark-up clean so it’s easier to convert to mobile and use structural elements such as lists and fields.

In general, the mark-up should always be clean and lean for better site performance; which also applies to scripts and CSS. Any unnecessary mark-up should be cleaned out and the number of external script and style sheets reduced as much as possible to improve loading time and efficiency by reducing http requests. Eliminate unnecessary images and try to draw simple images and icons in CSS. Embedding images directly into stylesheets can also improve efficiency.

Responsive Design Strategies

To start planning your mobile design, start by taking an inventory of the content. Anticipate the problems that will be encountered when the content shifts from desktop to mobile or mobile to desktop. Develop some goals and emphasize important content. Create a hierarchy for the content by importance and plan how it will be implemented.

The next step would be to create a mobile wireframe and a desktop wireframe and expand simultaneously. Starting with a mobile wireframe helps whittle out what might be less important and pull together the main points of the site, then it might be easier to build outwards on that structure. The rest of the design can be built within a browser so that the design can be tested for its breakpoints and tweakpoints, which leads to the third step, finding breakpoints and tweakpoints.

When establishing breakpoints and tweakpoints, try to keep realistic and design towards screen sizes, starting from smallest range and moving out towards the biggest. Breakpoints will reveal themselves naturally when the navigation menu starts to fall apart or when the sections start to break apart. Those breakpoints are where the major changes will likely happen, and the tweakpoints can refine the transitions. Testing sites should not be limited to just resizing a browser. The design can be tested in several desktop applications as well as through some free online resources and mobile emulators.

The fourth step would be device-specific functionality and how it will be, or can be, utilized, such as geo-location, instant messaging, location detection, motion detection, compass, camera, microphone, and touch sensitivity. Users want an integrated and rich user experience. Try to consider how users will interact with the site, what their expectations are, and how that functionality and their expectations will change across different devices.

The fifth step would be to consider the layout, especially navigation and images which should be planned out very early, along with consistent coding standard.

The last step would be to consider whether or not everything can be created in one site or if it might be necessary to create the site in a separate mobile application for mobile devices.

Creating Flexible HTML

In order to create a responsive site, the page has to be built around responsive functions so that html and styles can work together. When designing, consider how the content flows. Keep the code clean and establish standards for each content area. Start to modularize content by thinking of them as discrete sections of code. Use scripting but sparingly. It’s also important to remember that some CSS styles are not compatible with responsive design.

Conclusion

Designing responsive websites can be challenging as there are many factors to consider. Without a plan, it can be difficult to pull together a coherent structure that can degrade gracefully into a small window view. Trying to incorporate new additions or add in forgotten functions or features that are later discovered to be crucial could easily result in a lot of wasted time. Usability suffers when the layout doesn’t flow well or isn’t intuitive. The tools and methods for creating a responsive site exist and can be implemented well with good planning and a thorough understanding of goals, priorities, and a strategy for usability.