**Responsive Design**

**resets, layouts**

## overview

Responsive Design -- design that responds to the size/orientation of the user platform/device

Increased use of mobile computing devices

Interactions also need to work in a touch-only environment

Examples:

:hover effects and mouse events (over/out/enter/leave/click...) will not work on mobile devices

mobile devices use touch and swipe events

## mobile friendly vs. mobile first

both: reduce the percentage of users who leave your website prematurely

mobile friendly

- satisfy different device use cases for your website or web application (desktop/phone/tablet)

- Provide an optimized experience for their device of choice

mobile-first

- the mobile version done first by design

- design is modified to accommodate screen size increases

Example Use Cases

example: "additional information" section

- information overload on a mobile device

- show/hide page elements based on the screen width (hide on mobile devices)

display: none;

example: horizontal displays

- display section content vertically on mobile devices

- reduce left/right margins to 0 on mobile devices

margin: 20px 0;

## viewport meta tag

add to <head> section of html page

ensures proper rendering and touch zooming

<meta name="viewport" content="width=device-width, initial-scale=1">

## media queries

Enable different design specifications for different screen widths

Syntax -- media type followed by one or more expressions

Breakpoints -- used to match screen size to proper css code

Media types -- used to identify display

options: all, screen, print, tv, braille, 3d-glasses (new)

default media type: screen

## media query syntax examples

breakpoints should be generalized rather than targeted at specific devices

device categories: large screen, desktop, laptop, tablet, mobile

tablets and mobile devices can have landscape/portrait parameters

max-width and min-width are used to set the boundaries of a media query

/\* Large desktop \*/

@media all and (min-width: 1200px) { ... }

/\* Medium devices (laptops/desktops, 980px to 1199px) \*/

@media all and (min-width: 980px) and (max-width: 1199px) { ... }

/\* tablet (portrait) to tablet (landscape) and desktop \*/

@media all and (min-width: 768px) and (max-width: 979px) { ... }

/\* phone (landscape) to tablet (portrait) \*/

@media all and (min-width: 480px) and (max-width: 767px) { ... }

/\* phone (portrait) to phone (landscape) \*/

@media all and (min-width: 320px) and (max-width: 479px) { ... }

/\* phone (portrait) \*/

@media all and (max-width: 319px) { ... }

generalized breakpoints

@media (min-width: 768px) { ... } // mobile

@media (min-width: 992px) { ... } // tablet

@media (min-width: 1200px) { ... } // laptop

## design techniques

layout

float

floated elements will naturally collapse to vertical as screen width shrinks

floating elements technique is increasingly discouraged due to better alternatives (e.g. flex)

clearfix (see https://stackoverflow.com/questions/8554043/what-is-a-clearfix)

used in float layouts where elements are floated horizontally

fixes the zero-height container problem for floated elements

causes an element to automatically clear floats from its child elements

display

inline displays favor natural collapsing to vertical

display: inline;

display: inline-block;

table-cell display

forces non-table elements to behave like <td> elements

display: table-cell;

fixed widths vs percentage widths

fixed width -- will not change width based on the size of the page

width: 100px;

percentage width -- allows text blocks to resize with the page

width: 80%;

max-width -- content will expand responsively up to max-width value (more control than percentage)

max-width: 100px;

mobile styles

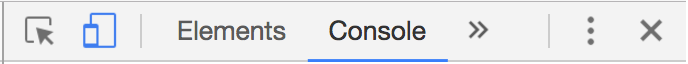
avoid CSS3 shadows, gradients, transforms, and complex animations

use show/hide techniques to temporarily display menus

testing responsive design

activate Chrome developer tools window (Option Command J)

click Device Toolbar toggle:



set width and height of test window

## reading

http://alistapart.com/article/responsive-web-design/