Week 6: Accessibility and Responsive Design

INFSCI 2560
Web Technologies & Standards

Housekeeping

- Rename your Glitch projects to be descriptive and include your Pitt username. (e.g. *toe6-activity4*)
- Exam 1 October 14.
- Discussion boards & Exam Study Guide will be posted this week.
- Assignment 1 due by 11:59p - *typo on question, refer to criteria #5, not criteria #6*
- Plagiarism
  - DO NOT copy any code from anywhere
  - Code that comes from other sources is discouraged and should not be needed
  - This includes code which you have adapted from another source but which is essentially someone else's work.
  - Failure to do so will result in a 0 for the assignment and may result in an F for the course.

Updated Plagiarism Policy

- **First offense:** ZERO on plagiarized assignment and/or up to 10 percentage points off student's final grade (up to teacher's discretion).
  - So for example, if the student earns a 94% overall, their grade will be reduced to 89%.
  - Submission to the academic integrity board.
- **Second offense:** Receive a failing grade for the course.

What does it mean for the web to be accessible?

Accessibility

- The UN Convention on the Rights of Persons with Disabilities recognizes access to information and communications technologies, including the Web, as **a basic human right**
- Accessibility supports social inclusion for people with disabilities as well as others , such as older people , people in rural areas, and people in developing countries.
- Accessibility overlaps with other best practices such as mobile web design , device independence, multi-modal interaction, usability, design for older users , and search engine optimization (SEO) .

https://www.w3.org/WAI/WCAG21/Understanding/

- **Perceivable**: Users must receive information and user interface components in ways that they can perceive, such as providing text alternatives for graphical and other content with no text.
- **Operable**: User interface components and navigation must be operable. An example is making all functionality available from a keyboard.
- **Understandable**: The information on the user interface must be understandable. The user should be able to figure out how to use the interface easily; think ease of setting the language, a clear focus element on each page, and navigation consistency.
- **Robust**: Content must be robust enough so a wide variety of user agents, including assistive technology, can interpret it.

Accessibility Categories

Vision - low/limited vision as well as complete blindness.

Hearing - partial or complete auditory loss.

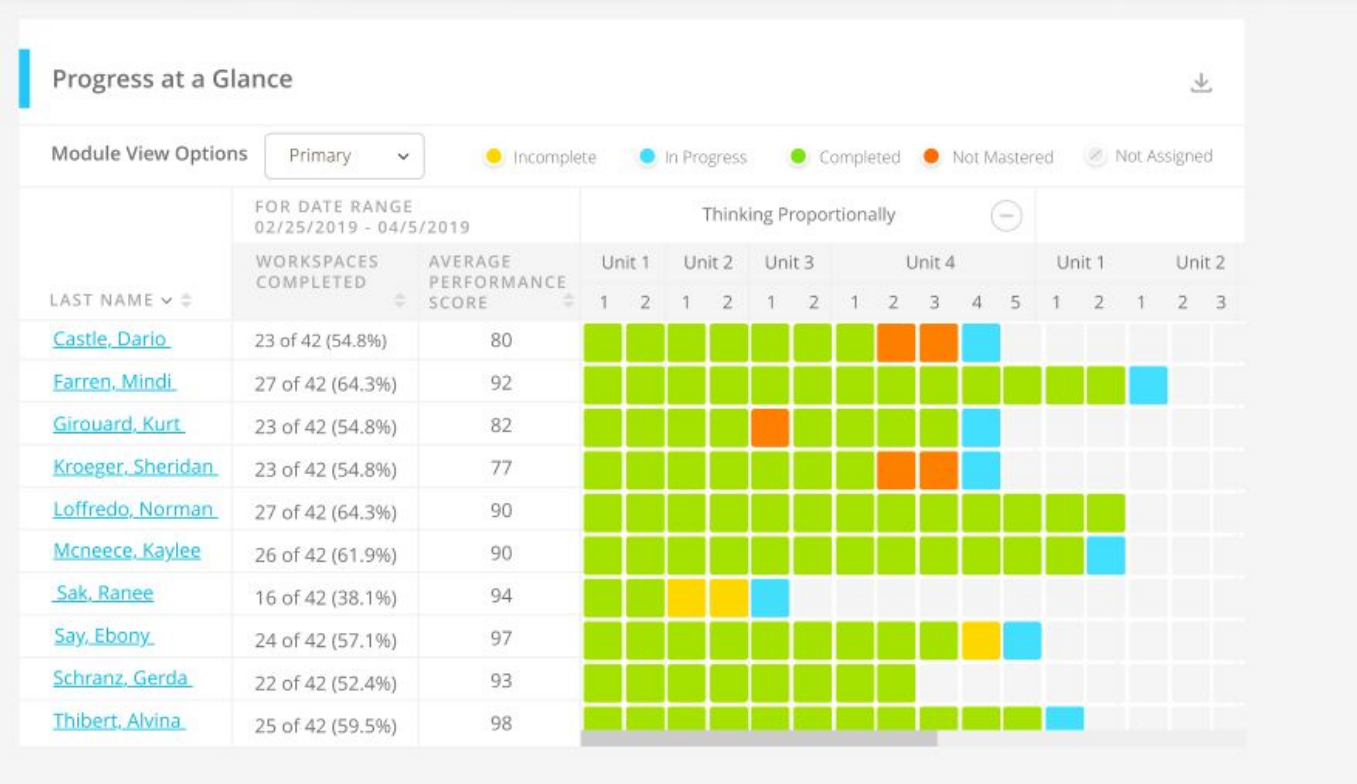
Motor or Dexterity - difficult to use mouse or touch screen.

Cognitive - ADHD, dyslexia and autism (to name a few). These can be

Vision

- **Goal**: Create a website accessible to a screen reader.
- Create logical flow of HTML elements
- HTML headers should provide semantic descriptors of content sections.
- Provide good alternate text and captions
- Don’t use color to communicate information
- Colorsafe.co - accessible color palettes.

Bad design decision



Progress at a Glance

Module View Options: Primary

Incomplete · In Progress · Completed · Not Mastered · Not Assigned

FOR DATE RANGE 02/25/2019 - 04/5/2019

Thinking Proportionally

LAST NAME	WORKSPACES COMPLETED	AVERAGE PERFORMANCE SCORE
Castle, Dario	23 of 42 (54.8%)	80
Farren, Mindi	27 of 42 (64.3%)	92
Girouard, Kurt	23 of 42 (54.8%)	82
Kroeger, Sheridan	23 of 42 (54.8%)	77
Loffredo, Norman	27 of 42 (64.3%)	90
Mcneece, Kaylee	26 of 42 (61.9%)	90
Sak, Ranee	16 of 42 (38.1%)	94
Say, Ebony	24 of 42 (57.1%)	97
Schranz, Gerda	22 of 42 (52.4%)	93
Thibert, Alvina	25 of 42 (59.5%)	98

Unit 1 (1 2) · Unit 2 (1 2) · Unit 3 (1 2) · Unit 4 (1 2 3 4 5) · Unit 1 (1 2) · Unit 2 (1 2 3)

Page Title

- **Good Examples:**
  - Acme Web Solutions home page
  - About Acme Web Solutions
  - Contact Acme Web Solutions
  - History of Acme Web Solutions
- **Bad Examples:**
  - Welcome to home page of Acme Web Solutions, Inc.
  - Acme Web Solutions, Inc. | About Us
  - Acme Web Solutions, Inc. | Contact Us
  - Acme Web Solutions, Inc. | History

Color Contrast Ratio

- The contrast between text and background affects how anyone reads text, but can have a more significant effect on people with visual impairments.
- High contrast, like white text on a black background, may be required by some
- Check out the Web AIM Color contrast checker for help and generate a score for your website.

https://accessibility.colostate.edu/media/sites/128/2017/09/text-on-page-diff-contrasts-01.png

Hearing

Goal: make certain nothing is hidden from those who cannot hear audio.

- Create videos with good captions
- Do not use audio alerts
- Use visual alerts

Motor

Goal: Site should be usable to keyboard only users or those with debilitating diseases or with a paralyzed arm.

- Provide multiple ways to navigate the site
- Try testing without a mouse, keyboard or use non-dominant hand

Cognitive

- Use simple fonts
- Use high contrast
- Do not put images behind text (distracting for those with dyslexia and ADD/ADHD and may also render text unreadable)
- Use meaningful icons and images
- Limit the use of sounds and animations
- Create a consistent user experience


Designing for users on the
autistic spectrum
Do...
Don't...
use simple colours
use bright contrasting colours
write in plain English
Do this.
use figures of speech and idioms
use simple sentences and bullets
create a wall of text
make buttons descriptive
Attach files
make buttons vague and unpredictable
Click here!
build simple and consistent layouts
build complex and cluttered layouts


Designing for users of
screen readers
Do...
Don't...
describe images and provide transcripts for video
<alt>
only show information in an image or video
follow a linear, logical layout
spread content all over a page
structure content using HTML5
<h1>
<nav>
<label>
rely on text size and placement for structure
36pt, bold
Header
build for keyboard use only
force mouse or screen use
write descriptive links and headings
Contact us
write uninformative links and headings
Click here


Designing for users with
low vision
Do...
Don't...
use good colour contrasts and a readable font size
Aa
use low colour contrasts and small font size
Aa
publish all information on web pages
HTML
bury information in downloads
use a combination of colour, shapes and text
Start
only use colour to convey meaning
200% magnification
follow a linear, logical layout
200% magnification
spread content all over a page
put buttons and notifications in context
Submit
separate actions from their context
Submit

Source

Designing for users with physical or motor disabilities

Do…

- make large clickable actions
- give form fields space
- design for keyboard or speech only use
- design with mobile and touchscreen in mind
- provide shortcuts

Don't…

- demand precision
- bunch interactions together
- make dynamic content that requires a lot of mouse movement
- have short time out windows
- tire users with lots of typing and scrolling

Designing for users who are Deaf or hard of hearing

Do…

- write in plain English
- use subtitles or provide transcripts for videos
- use a linear, logical layout
- break up content with sub-headings, images and videos
- let users request an interpreter for appointments

Do this.

Don't…

- use complicated words or figures of speech
- put content in audio or video only
- make complex layouts and menus
- make users read long blocks of content
- don't make telephone the only means of contact with users

Designing for users with dyslexia

Do…

- use images and diagrams to support text
- align text to the left and keep a consistent layout
- consider producing materials in other formats (for example, audio or video)
- keep content short, clear and simple
- let users change the contrast between background and text

Don't…

- use large blocks of heavy text
- underline words, use italics or write in capitals
- force users to remember things from previous pages - give reminders and prompts
- rely on accurate spelling - use autocorrect or provide suggestions
- put too much information in one place

[Source]

Accessibility In Practice

- Keep accessibility in mind at the start of your design process
  - It can be difficult to add afterwards
- Accessibility is a guiding principle
  - You strive for it, but never fully achieve it
- There are tools you can use to check the accessibility of a web page
- Even if a page passes these checks, it doesn't mean the page will be 100% accessible
- These simple checks are not comprehensive, but they are a place to start

Resources

- Google Accessibility: https://developers.google.com/web/fundamentals/accessibility
- Examples & Articles (READ THESE)
  - https://blog.prototypr.io/common-accessibility-problems-good-and-bad-examples-in-modern-websites-a13efb7256ad
  - https://www.w3.org/WAI/demos/bad/
  - https://www.dbswebsite.com/blog/ada-compliance-starts-with-accessible-design/

Break (5 minutes)

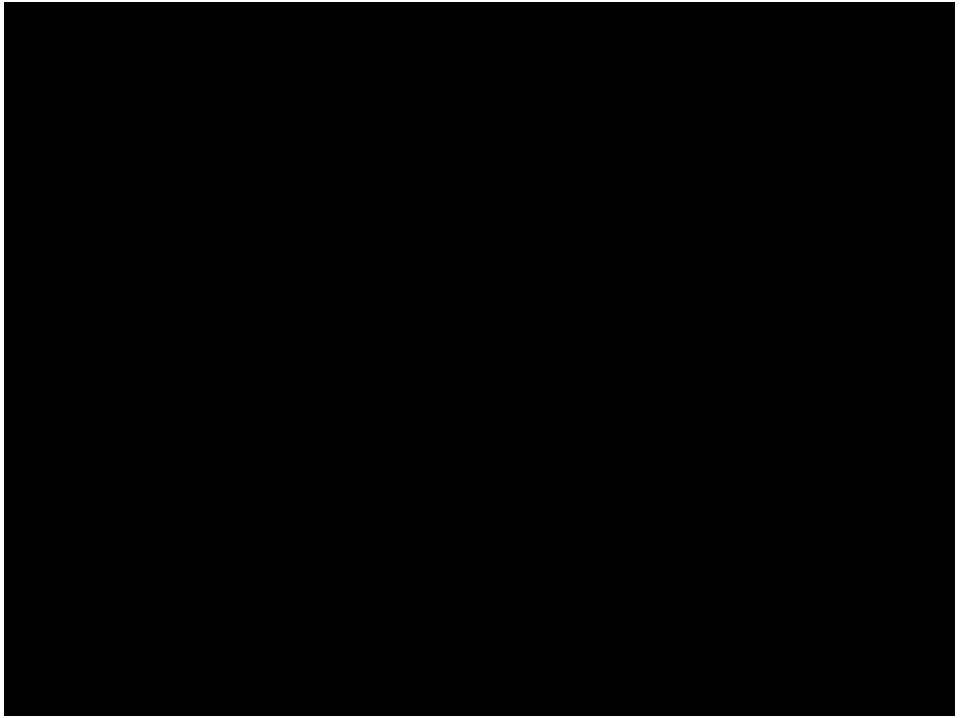
Why responsive design?

- Responsive Design is a set of techniques for structuring HTML and CSS so web pages are readable on multiple devices
  - Don’t have to make separate websites for different devices
- The page design responds to the needs and capabilities of the user’s device
  - Does this with just HTML & CSS, no JavaScript

How NOT to create a responsive webpage

This is how a proper responsive web page should behave

- Content restructures instead of shrinking
- The design and layout of the page responds to the size of the browser window



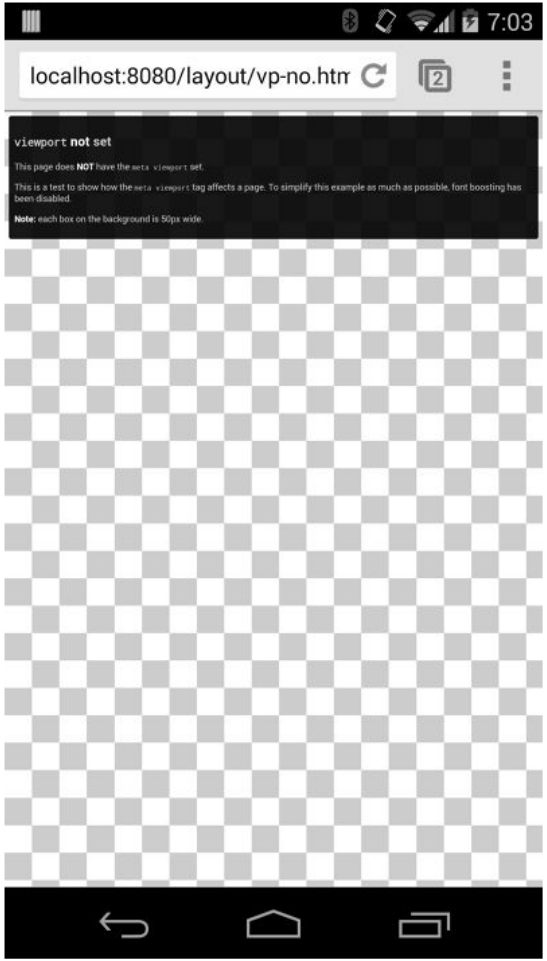
Source Google Responsive Web Design Overview

The <viewport> element

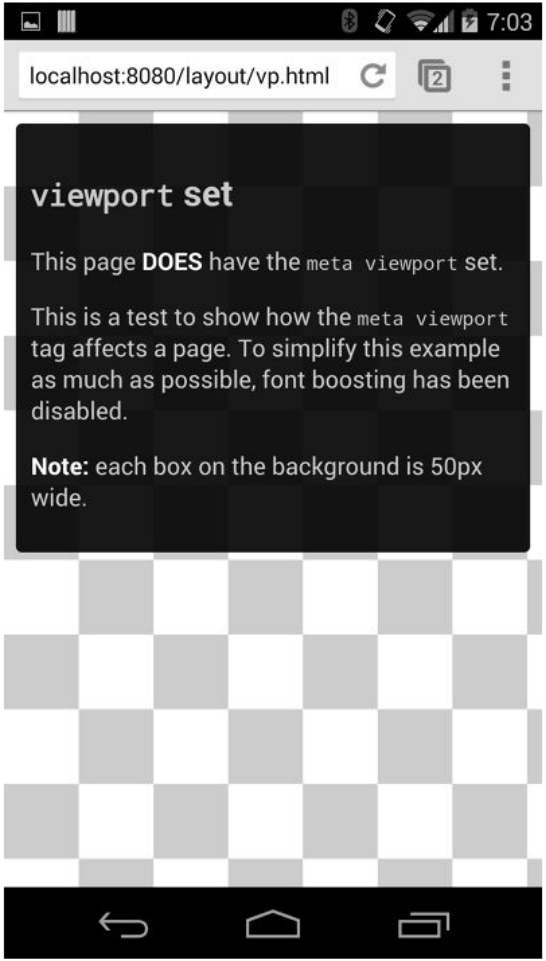
- The first step in a responsive design is to set the **viewport**
- The viewport is the browser window area that displays the page
  - Different sizes depending on devices
- HTML 5 introduces a way for designers to control how the browser scales content

```
<metaname="viewport"content="width=device-width,
initial-scale=1.0">
```

- The `width=device-width` attribute tells the page to match the width of the screen in device-independent pixels
- The `initial-scale=1.0` tells the browser what zoom level to set when the page is loaded
- Do not use fixed width elements



Page without a viewport set



Page with a viewport set

Responsive Images

- Responsive Design can enable images to scale nicely
- If you set width=100% then the image will sometimes scale too big, making a pixely image.
- Use max-width=100%, then the image will be scaled down, but never beyond its size.

Media Queries

- Sometimes you need to make substantive changes to the structure of the page
- More styles on individual elements, you need to adjust how everything fits together
- *Media queries* let you apply different CSS rules based upon the viewport size
- When you use a media query to cause changes you are defining *breakpoints*, the specific places that cause a trigger in the CSS rules for that media query
    a. Specified in pixels

```
@media (query) {

        /* rules for when the query matches */

          }
```

Media Queries

- Possible media query parameters
  - **min-width** - applies when viewport width is greater than specified value
  - **min-height** - applies when viewport height is greater than specified value
  - **max-width** - applies when viewport width is less than specified value
  - **max-height** - applies when viewport height is less than specified value
  - **orientation** - Two possible values portrait for when height greater than or equal to width or landscape for when width greater than height

```
@media (max-width: 600px) {
h1.rgb {
  color: red;
}
@media (min-width: 600px) and (max-width: 700px) {
  h1.rgb {
    color: green;
}
@media (min-width: 700px) {
  h1.rgb {
    color: blue;
}
```

W3Schools Example

Try it! (10 minutes)

The **HTML `<source>` element** specifies multiple media resources for the `<picture>`, the `<audio>` element, or the `<video>` element.

1. Get 2 images
2. Use the <source> element to add a media query at 500px.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/source

Free Stock Photos: (don't spend a lot of time looking up pictures)

https://www.pexels.com/ OR https://unsplash.com/

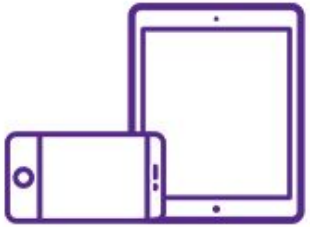
Breakpoints

- When you use media queries to trigger different layouts you are specifying a set of *breakpoints*.
- Breakpoints define a set of *ranges* where the layout will remain the same
- The most common breakpoint ranges you will need to design for are:
  - Phones
  - Tablets in portrait mode
  - Tablets in landscape mode
  - Desktop browser windows
  - Extra-wide desktop browser windows

Common Breakpoints

0-480

Smaller smartphones



481-768

Tablets & larger smartphones

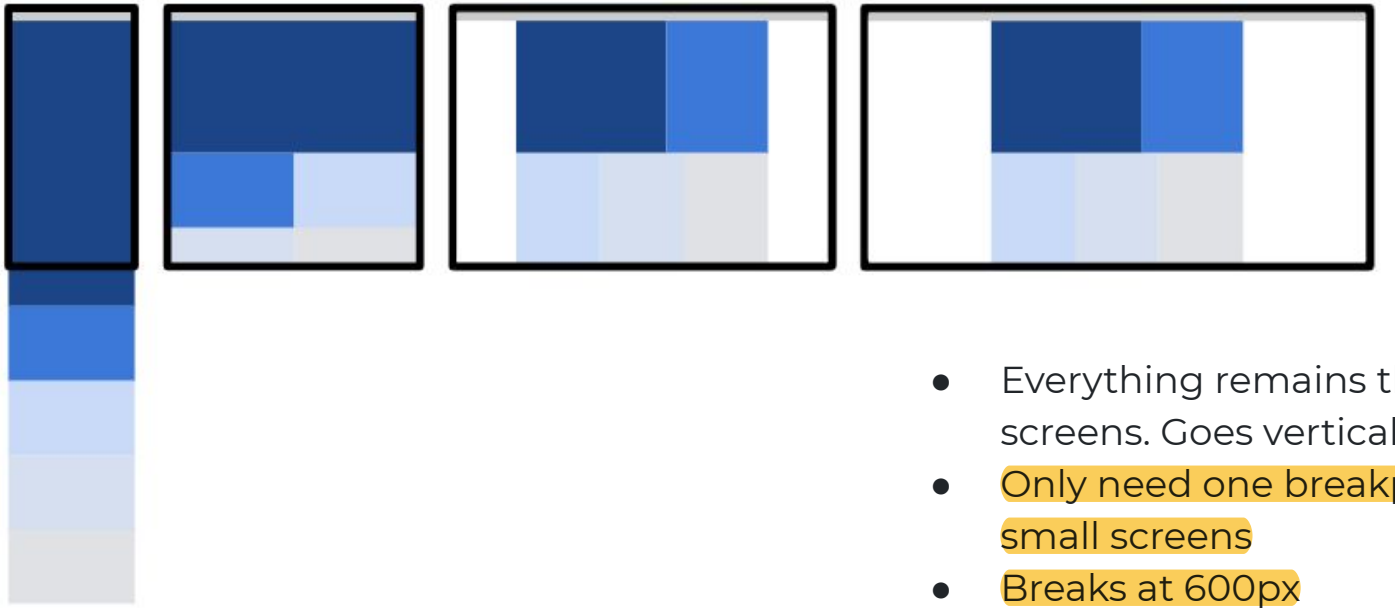
769-1279

Laptops, larger tablets in landscape, and small desktops

1280+

Larger desktops and monitors

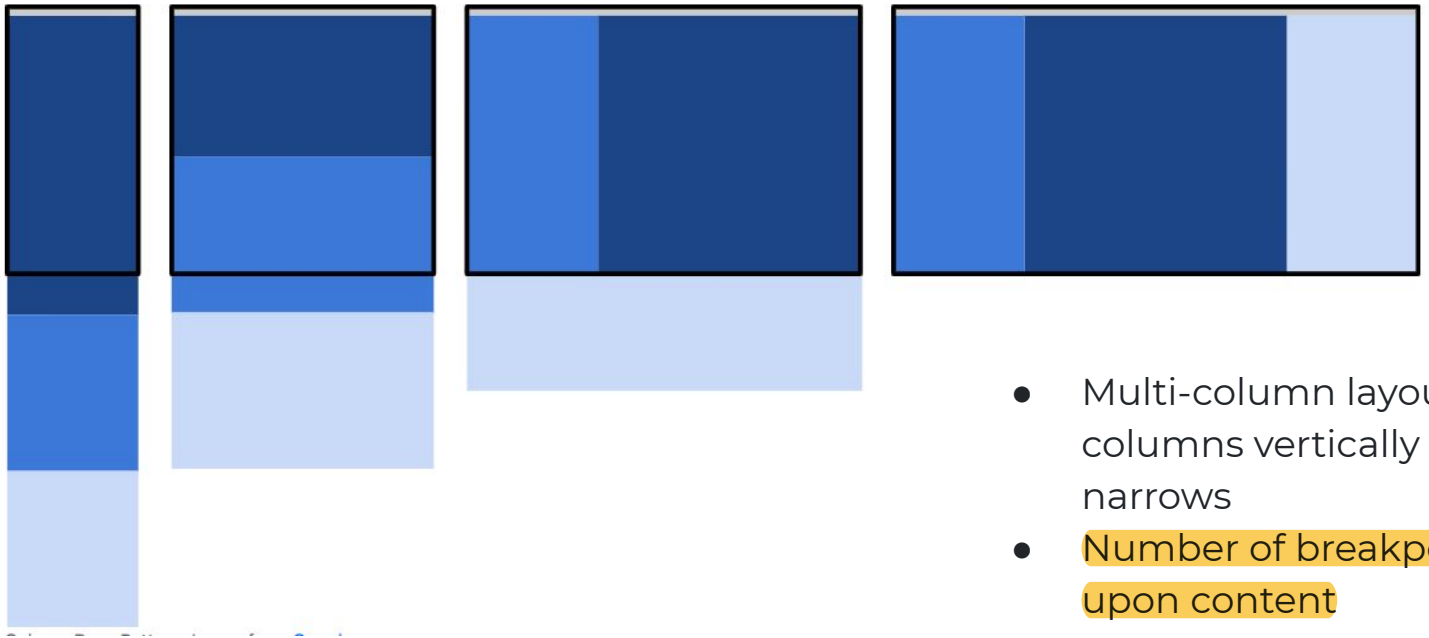
Responsive Design Patterns : Mostly Fluid



- Everything remains the same on large screens. Goes vertical on small screens
- Only need one breakpoint for large and small screens
- Breaks at 600px

Image from Google

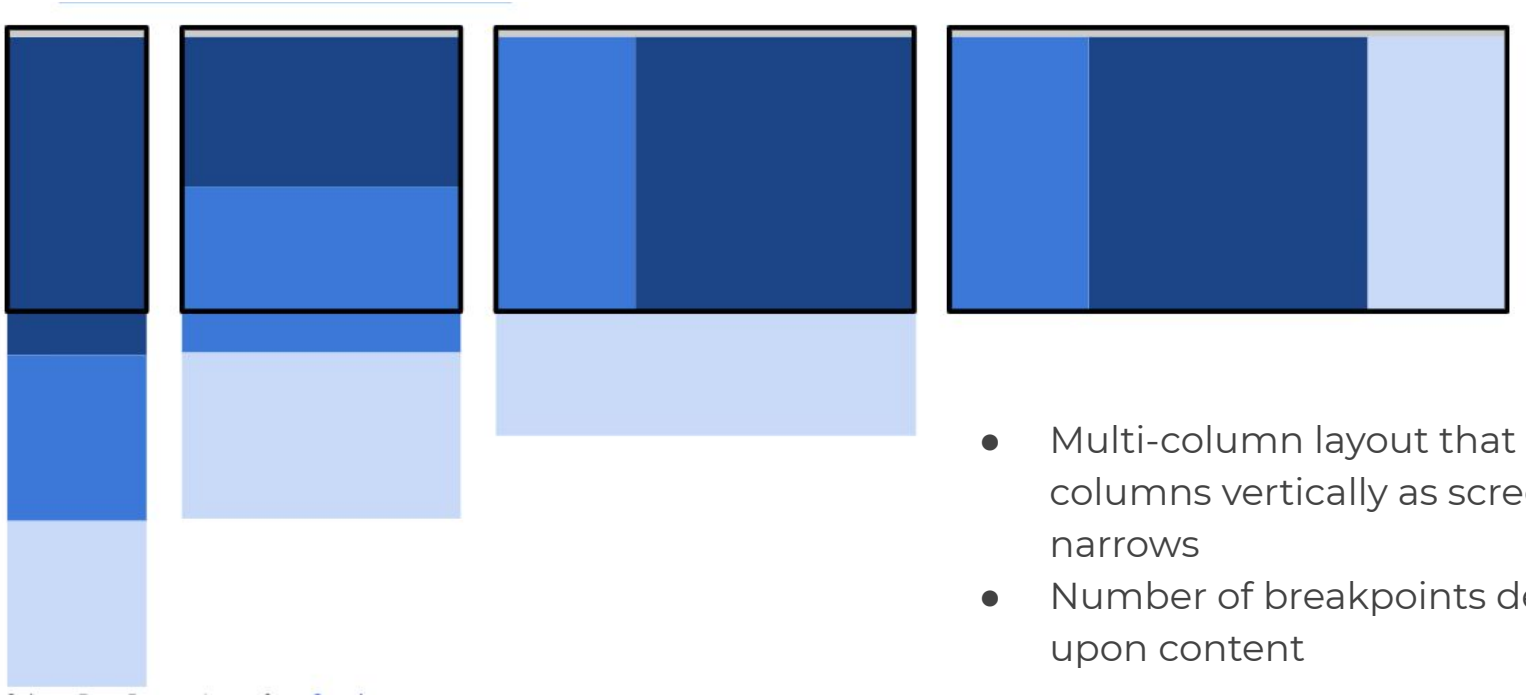
Responsive Design Patterns: Column Drop



- Multi-column layout that stacks columns vertically as screen size narrows
- Number of breakpoints depends upon content

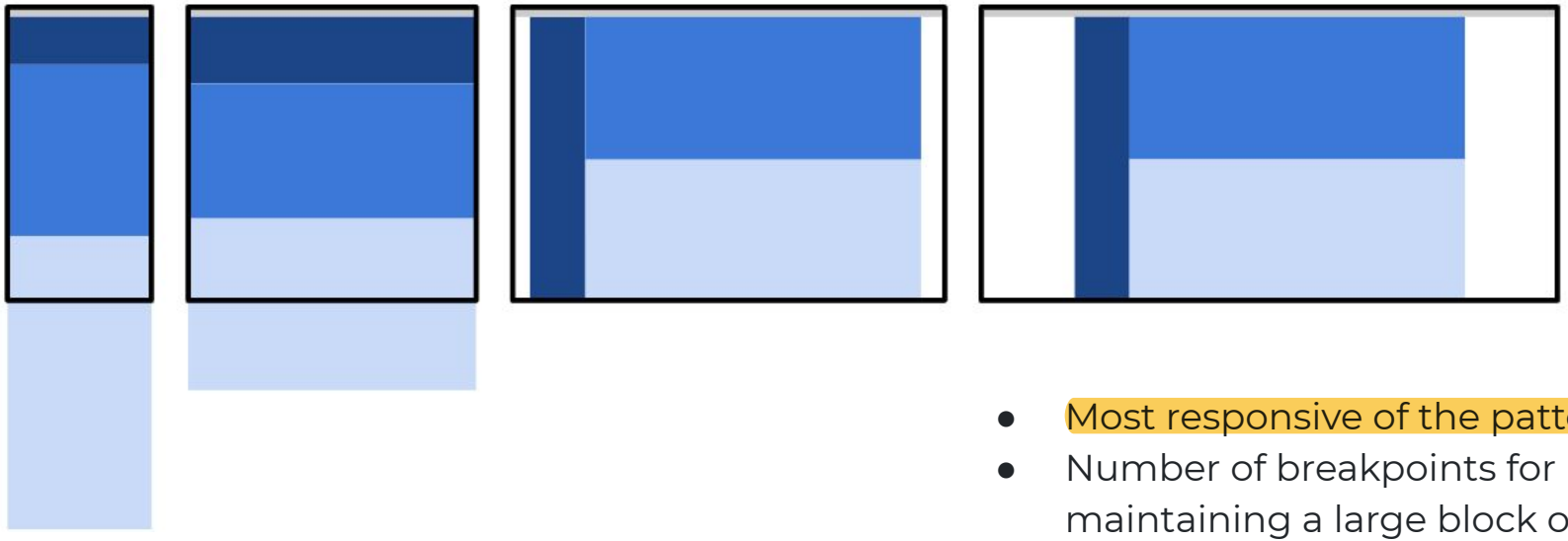
Image from Google

Responsive Design Patterns: Column Drop



- Multi-column layout that stacks columns vertically as screen size narrows
- Number of breakpoints depends upon content

Responsive Design Pattern: Layout Shifter



- Most responsive of the patterns
- Number of breakpoints for maintaining a large block of content and two stacked blocks of content.

Other two: Tiny-weak; Off-canvas

Mobile First Design

Design to the smallest screen and work your way up.

- With progressive enhancement is mobile first design. The smallest designs will have the essential features.
- Graceful degradation includes all of the complexities up front, then strips them away for smaller devices.

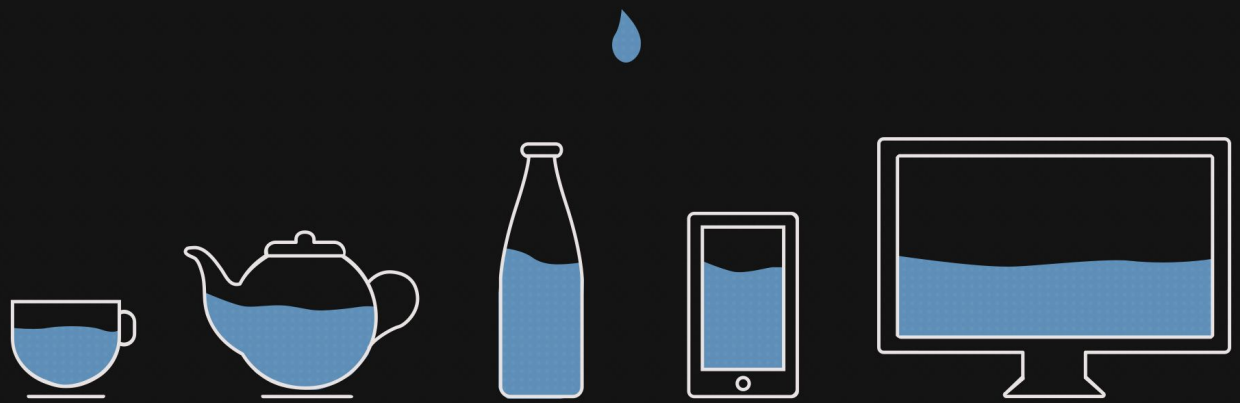
In 2018, 52.2% of all internet traffic happened on a mobile device.

Source: https://www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/

Responsive Design In Summary

- Don’t use fixed width elements
  - use width:100% not width:320px
- Content should not rely on a particular viewport width to look good
  - Horizontal scrolling is BAD
- Use CSS Media Queries to apply different styling
  - Apply different styles based on the device
  - Use min-width and min-device-width
  - Use relative sizing
- Always design for Mobile First and design the site to grow vs. design to shrink

CONTENT IS LIKE WATER



"You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot."

Josh Clark *(originally Bruce Lee)* - Seven deadly mobile myths

Illustration by Stéphanie Walter

Additional Resources

- WebKit CSS extensions
- https://developer.mozilla.org/en-US/docs/Web/CSS/WebKit_Extensions
- CSS media queries
- https://www.w3schools.com/css/css_rwd_mediaqueries.asp

Activity 5 - Accessibility & Responsive Design

https://infsci-2560-edmonds.glitch.me/week/six.html