Week 5: Accessibility and Responsive Design

INFSCI 2560
Web Technologies & Standards

Housekeeping

- On activities you should rename your Glitch projects to be descriptive and include your Pitt username. (e.g. *toe6-activity4*)
- Extended to Monday at 11.59 PM weekly
- Quizzes are still due Friday at 11.59 PM!
- Plagiarism
  - DO NOT copy any code without citation
  - Code that comes from other sources is discouraged and should not be needed
  - Failure to do so will result in a 0 for the assignment and may result in an F 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) .

Web Accessibility

- Definition: the practice of making web content and applications accessible to people with disabilities
- Importance of web accessibility in today's society
- Legal framework: Americans with Disabilities Act (ADA) and Section 508 of the Rehabilitation Act

Who Benefits from Web Accessibility

- People with disabilities: visual, auditory, motor, cognitive, and neurological
- Older adults: age-related changes in vision, hearing, and dexterity
- People with temporary disabilities: broken limbs, repetitive strain injuries
- People with diverse abilities: people with different cultural and linguistic backgrounds

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.

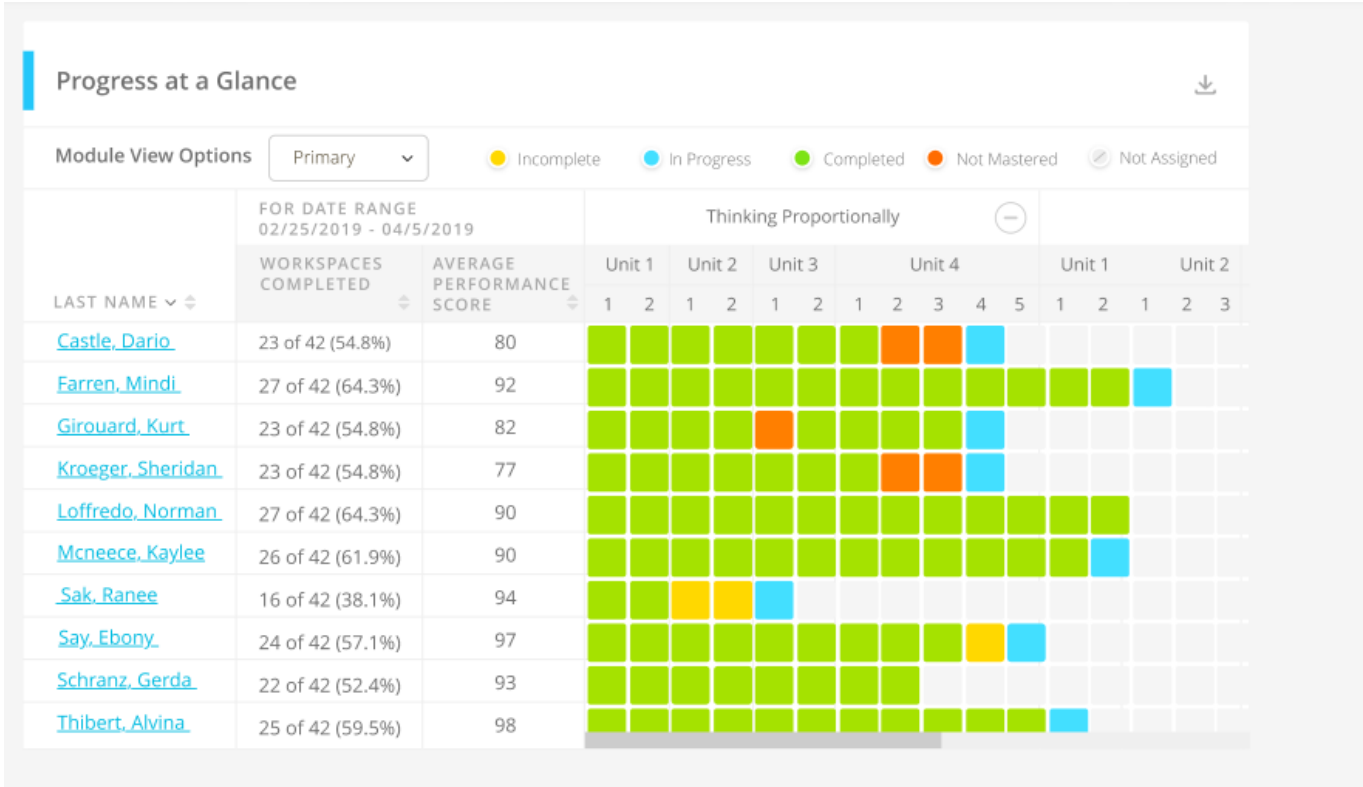
Motor or Dexterity - difficult to use mouse or touch screen.

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

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



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

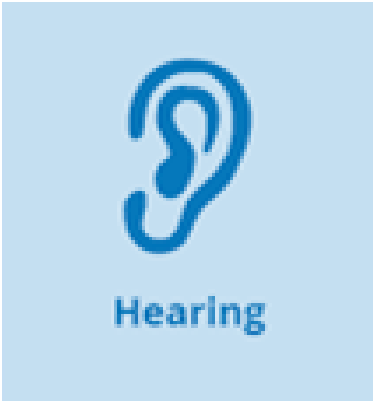
- The contrast between text and background affects how anyone reads text, but can have a more significant effect on people with visual impairments.
- High contrast, like white text on a black background, may be required by some
- Check out the Web AIM Color contrast checker for help and generate a score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



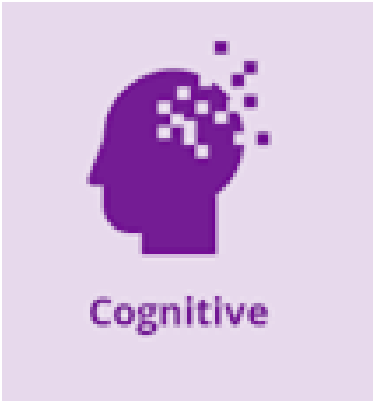
Motor

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



https://accessibility.blog.gov.uk/2016/09/02/dos-and-donts-on-designing-for-accessibility/

https://accessibility.blog.gov.uk/2016/09/02/dos-and-donts-on-designing-for-accessibility/

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

Keyboard Navigation

- Keyboard navigation should be possible without requiring specific timings for individual keystrokes
- The tab order should be logical and follow a meaningful sequence
- The user should be able to access all the interactive elements using the keyboard
- The user should be able to interact with the interface using keyboard-only techniques, such as using the tab key to navigate between links and form controls

Alt Text and Description

- Alt text is a short description of an image, used to provide context to users who cannot see the image
- Alt text should be concise and meaningful, and describe the purpose of the image
- Description should be provided for non-textual elements, such as graphs and charts, to provide accessibility to users who are unable to see the visual information

Audio and Video Content

- Audio and video content should be accompanied by captions and transcripts
- The captions should accurately reflect the speech and sound effects in the content
- The captions should be synchronized with the audio or video
- Audio content should have a text transcript available

Page Structure and Navigation

- Pages should have a clear and consistent structure, with headings and subheadings used to organize content
- Navigation should be consistent across the site, with clear labeling of links and buttons
- Navigation should be easily accessible, with a skip navigation link provided to bypass repeated navigation elements
- Search functionality should be provided, allowing users to quickly find specific content

Form Design and Interactivity

- Forms should be designed to be accessible, with clear labeling and instructions provided
- Error messages should be clear and concise, and should explain what needs to be corrected
- Required fields should be clearly marked, and error messages should be displayed when a required field is not filled in
- Forms should be accessible using keyboard-only techniques, with appropriate focus states and tab order

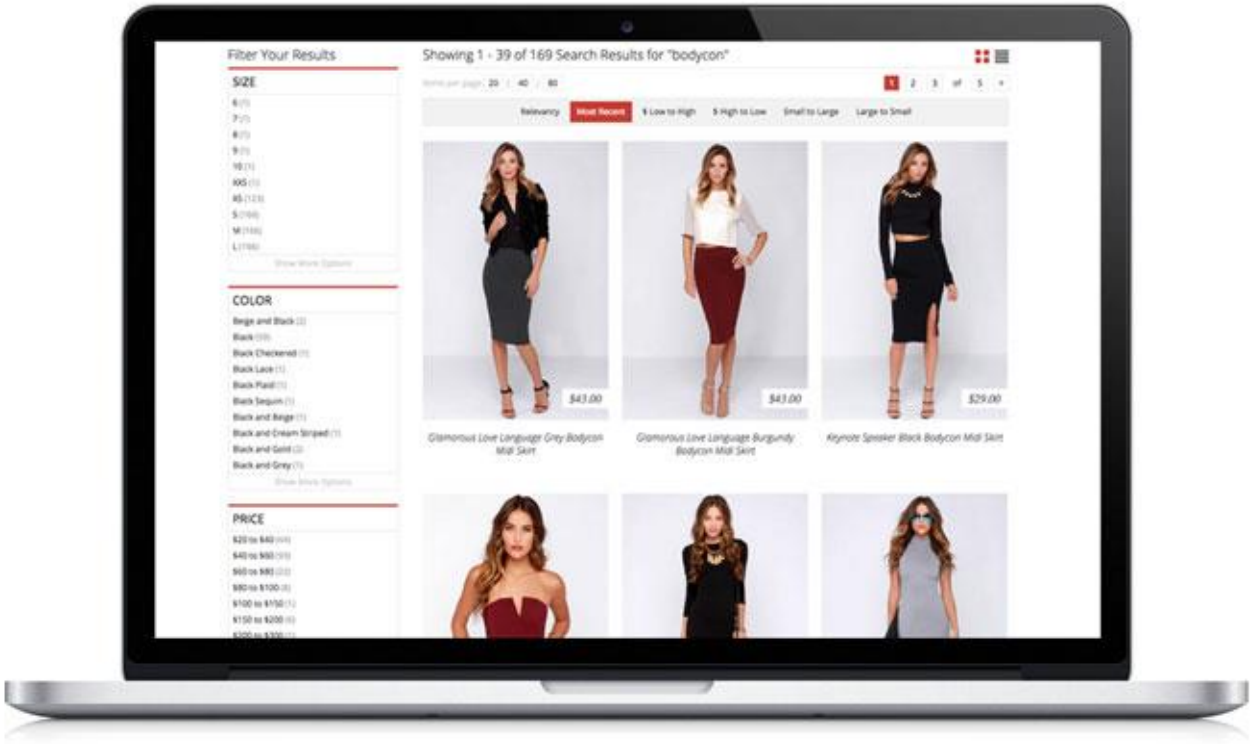
Resources

- 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/
- Guidelines:
  - **Web Content Accessibility Guidelines (WCAG)**
  - **User Agent Accessibility Guidelines (UAAG)**
  - **Authoring Tool Accessibility Guidelines (ATAG)**

Break

Why responsive design?

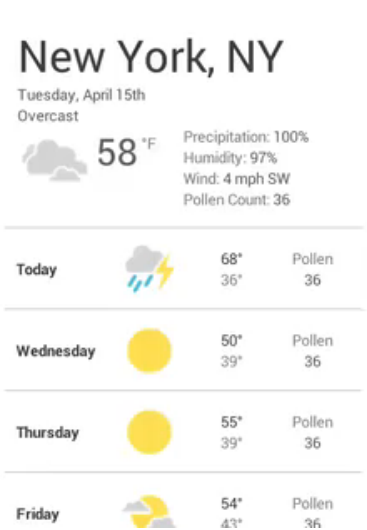
- Responsive Design is a set of techniques for structuring HTML and CSS so web pages are readable on multiple devices
  - Don’t have to make separate websites for different devices
- The page design responds to the needs and capabilities of the user’s device
  - Does this with just HTML & CSS, no JavaScript



https://searchspring.com/blog/responsive-sites-gone-bad/

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



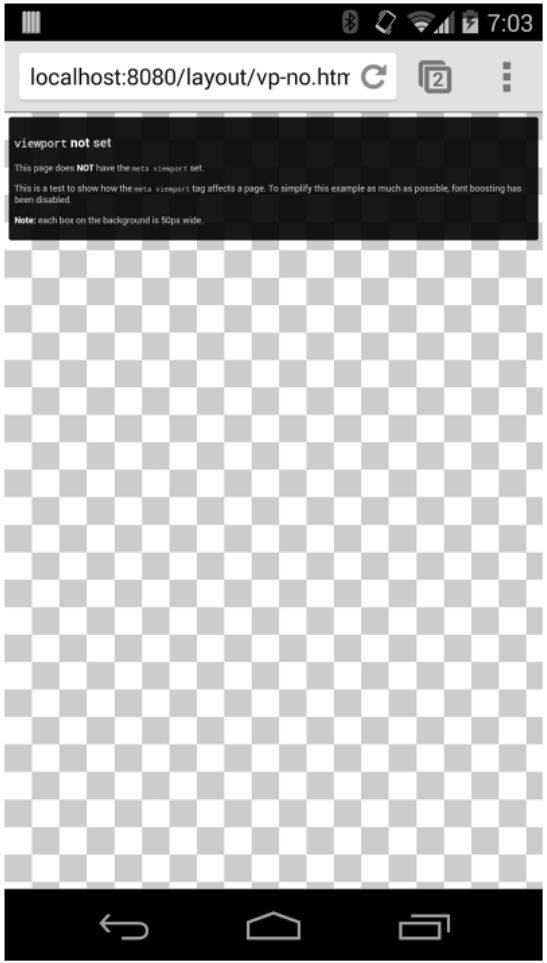
Source Google Responsive Web Design Overview

The <viewport> element

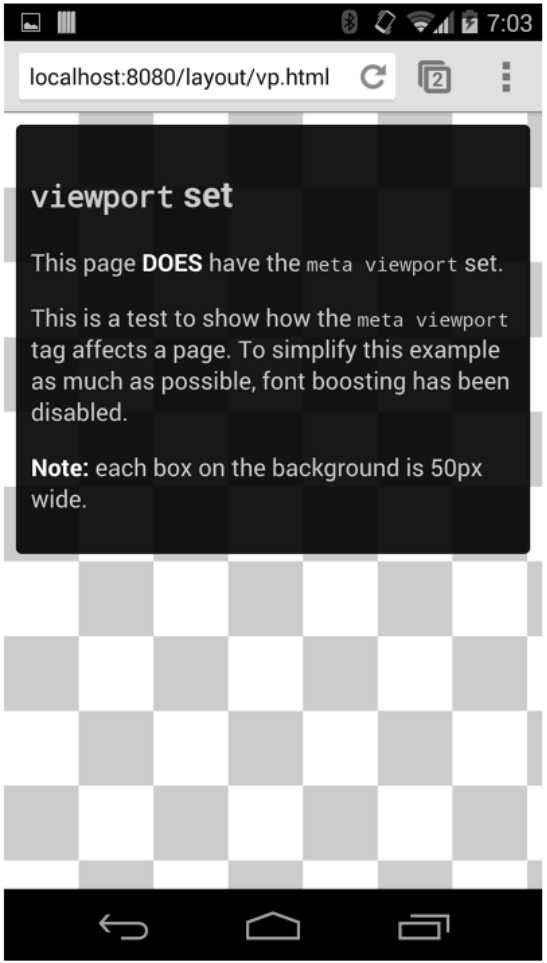
- The first step in a responsive design is to set the **viewport**
- The viewport is the browser window area that displays the page
  - Different sizes depending on devices
- HTML 5 introduces a way for designers to control how the browser scales content

```
<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)

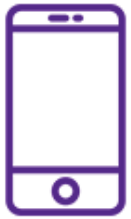
The **HTML `<source>` element** specifies multiple media resources for a `<picture>` element.

1. Get 3 GIF URLs from https://giphy.com
   a. Click "Copy Link" then "GIF Link"
2. Visit https://codepen.io/pen/
3. Create a <picture> element that encapsulates **three** <source> elements with `media` attributes that change the pic on resize
   a. Hint: MDN documentation

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

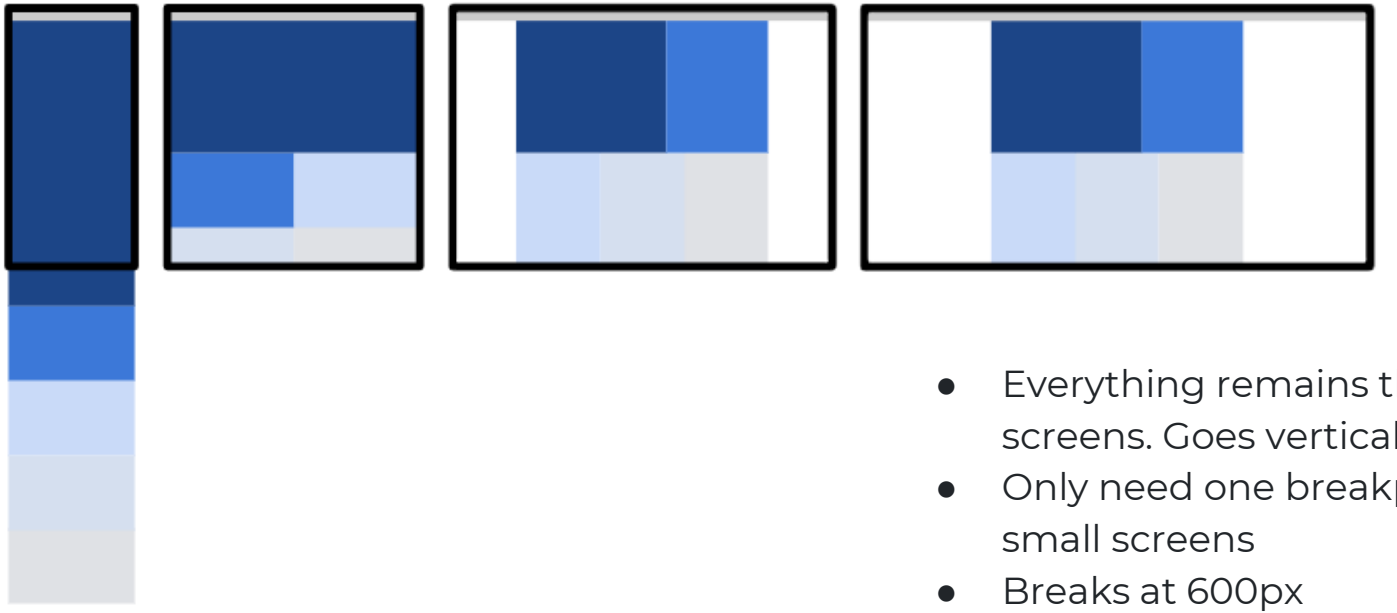
769-1279

Laptops, larger tablets in landscape, and small desktops

1280+

Larger desktops and monitors

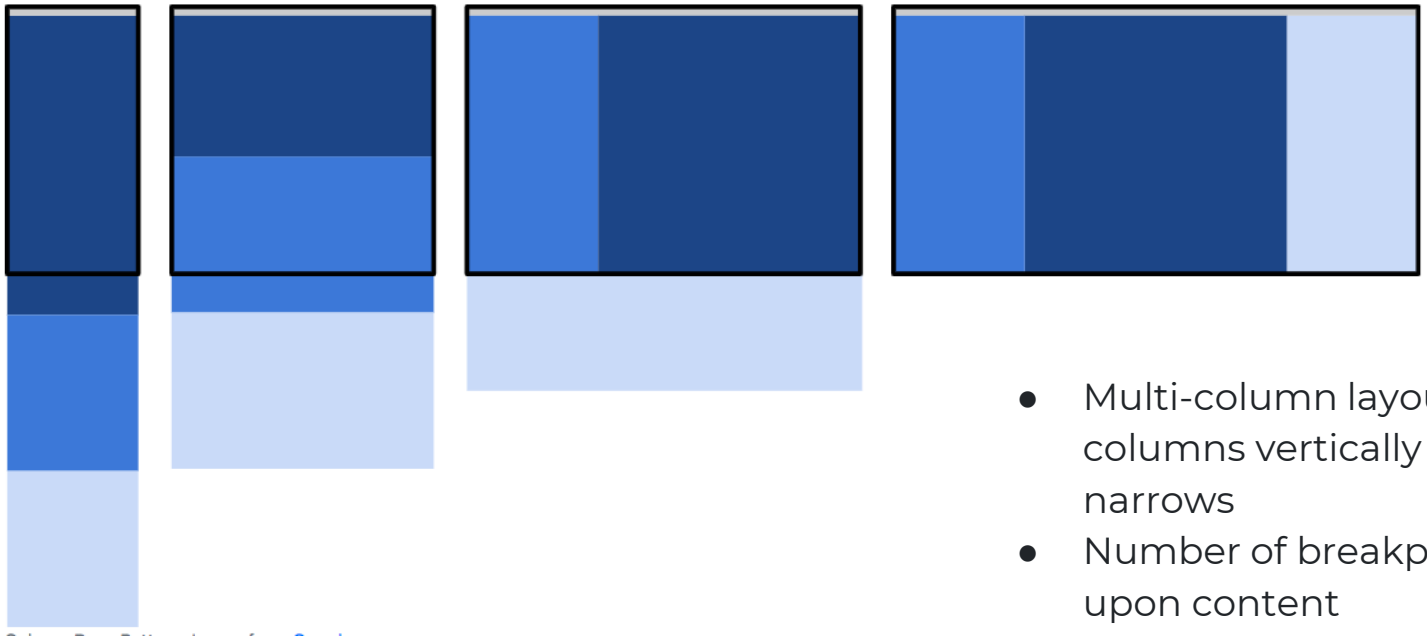
Responsive Design Patterns : Mostly Fluid



- 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

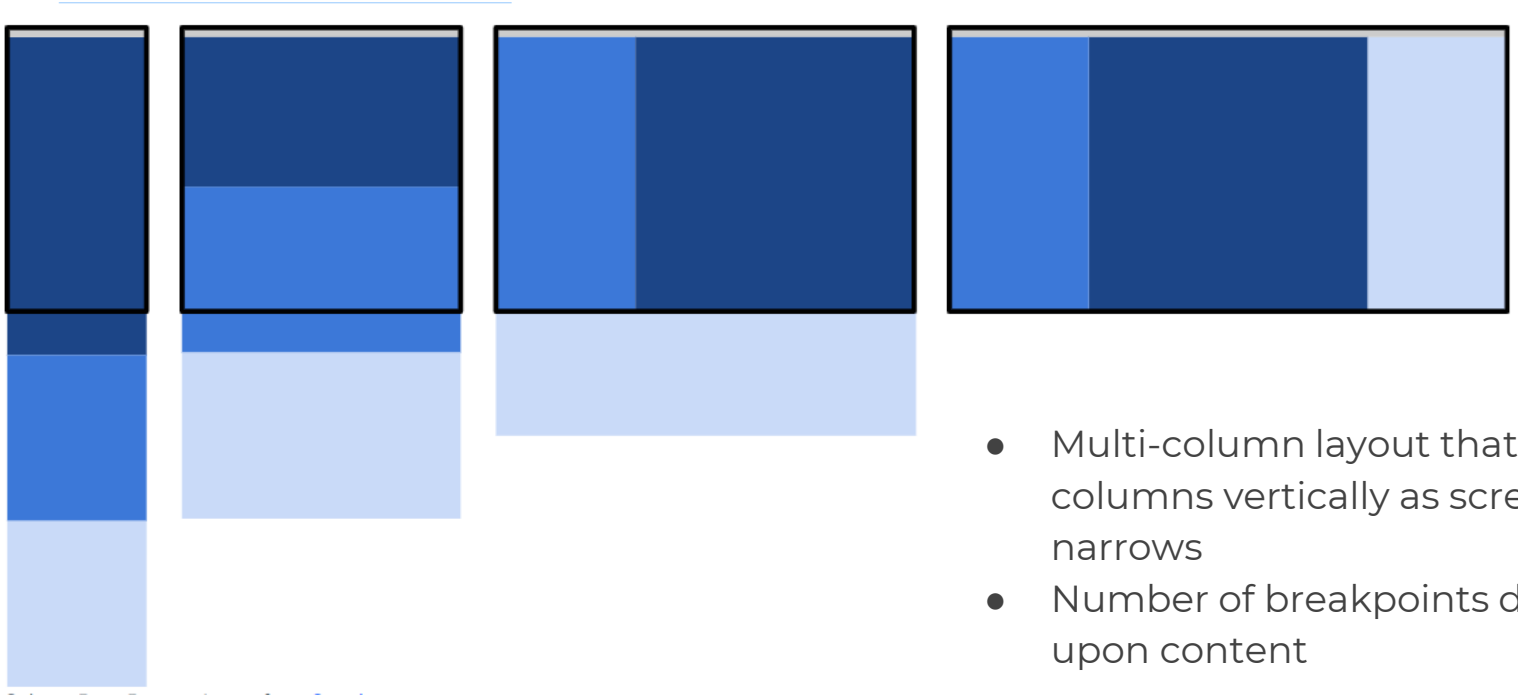
Responsive Design Patterns: Column Drop



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

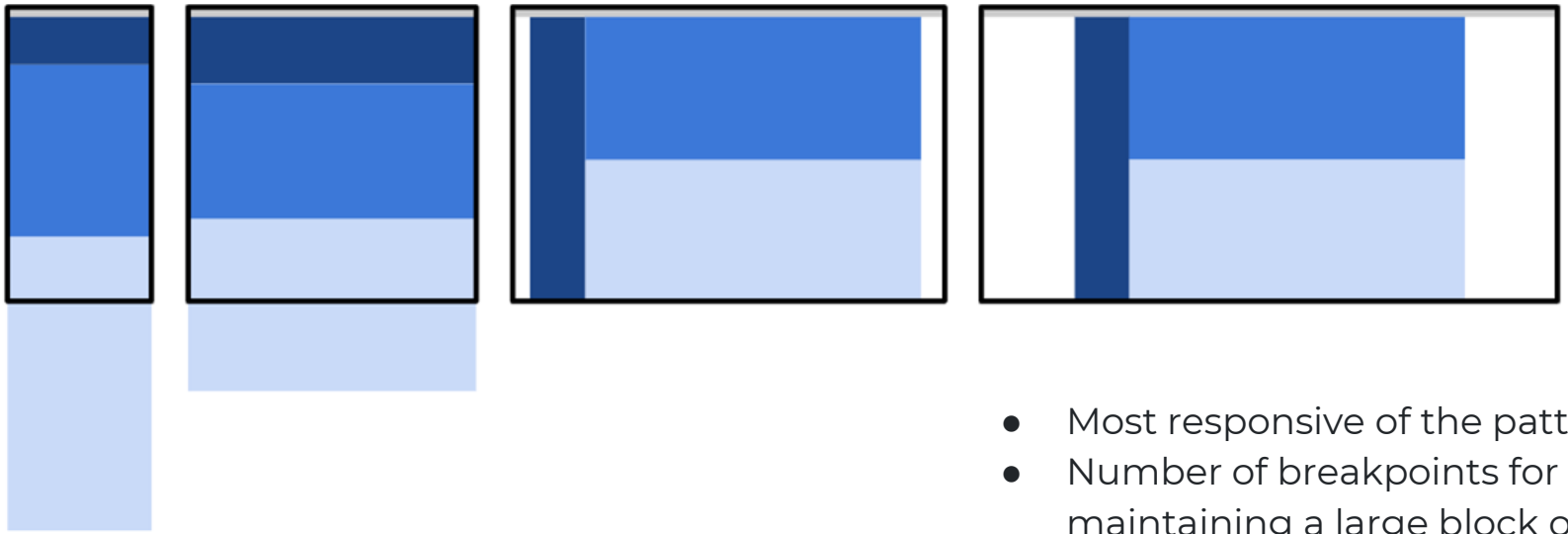
Image from Google

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.

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.

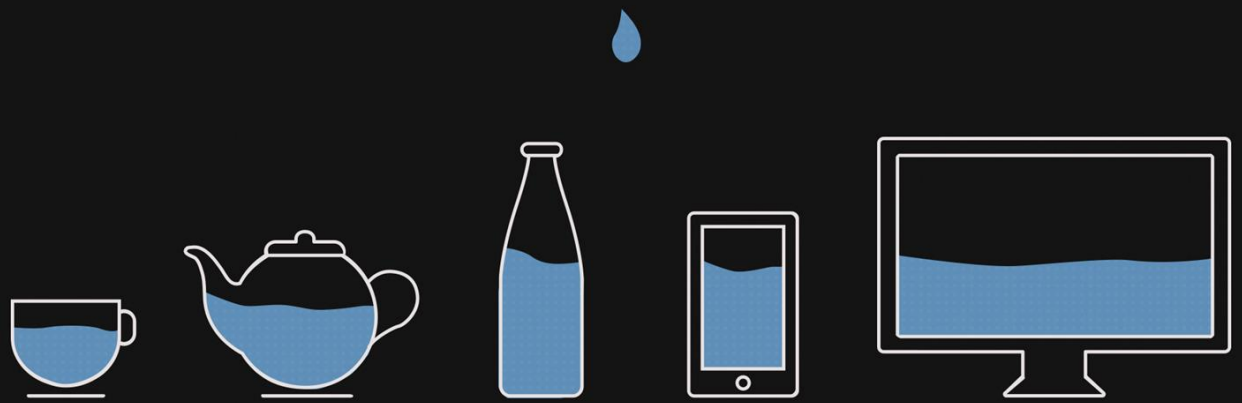
In 2022, 58.99% of all internet traffic happened on a mobile device.

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

- Remix your Recipe site from Activity 2
  - Rename to <pitt username>-activity5
- Make it more accessible and responsive
  - Document your changes in the `readme.md`
- See instructions on Canvas
  - Activity 5 - Accessibility & Responsive Design
- Submit a the project page for your remixed site
  - http://glitch.com/~<pitt username>-activity5