

IN4MATX 133: User Interface Software

Inclusive/Responsive Design

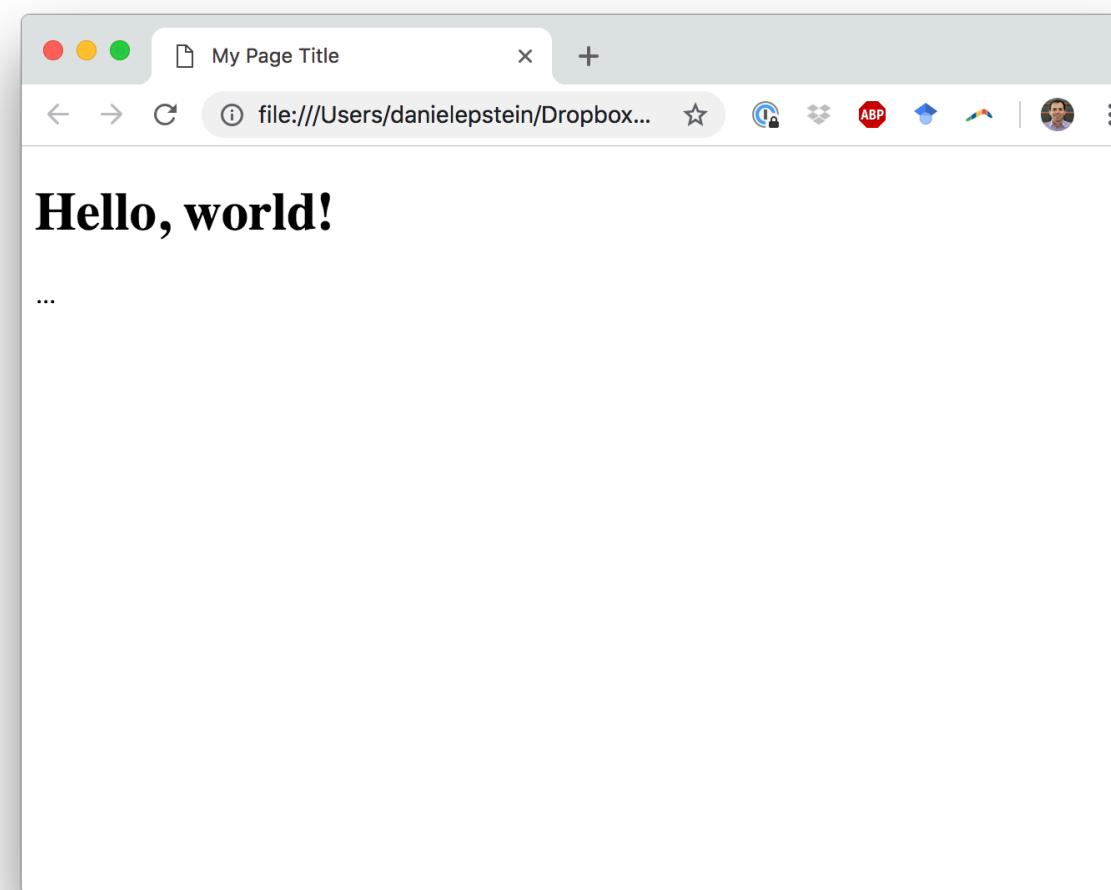
Today's goals

By the end of today, you should be able to...

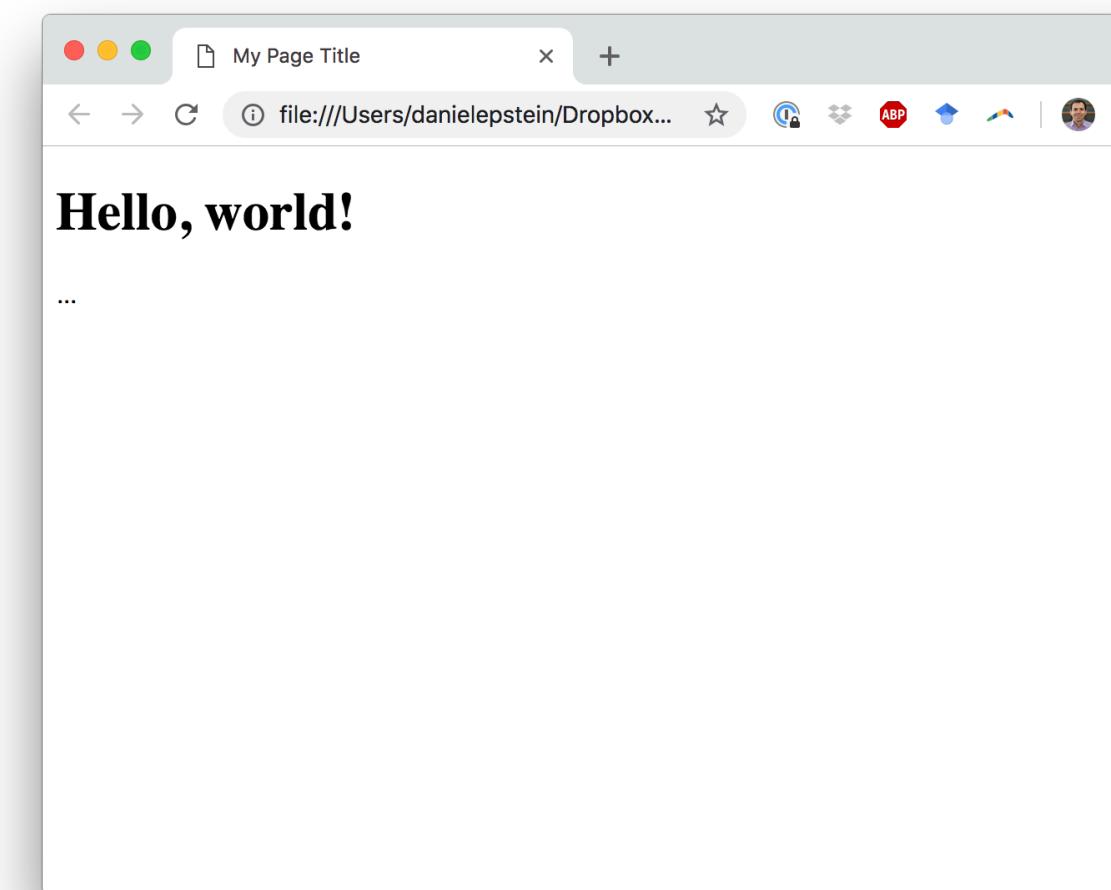
- Explain the importance of accessible and semantically meaningful markup
- Generate markup which meets accessibility standards
- Describe how responsive and adaptive design differ and when you might prefer one or the other
- Explain the advantages and disadvantages of a mobile-first design

HTML structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="author" content="your name">
  <title>My Page Title</title>
</head>
<body>
  <h1>Hello, world!</h1>
  ...
</body>
</html>
```



```
<html>
<head>
  <title>My Page Title</title>
</head>
<body>
  <h1>Hello, world!</h1>
  <p>...</p>
```



Why does HTML structure matter?

Taking a step back:
Web access is important

How do you use the web?

**“The power of the Web is in its universality.
Access by everyone regardless of disability is an essential aspect.”**

*–Tim Berners-Lee, inventor of the World Wide Web and 2016 Turing award winner
<https://www.w3.org/WAI/fundamentals/accessibility-intro/>*

All sorts of people
will use the webpage you create

Common Disabilities that Affect Technology Use

- Vision
 - Blind, low vision, colorblind
- Auditory impairments
 - Deaf, hard of hearing
- Motor impairments
 - Arthritis, cerebral palsy, tremors, paralysis
- Cognitive impairments
 - Autism, dyslexia, language barriers
- Much more

The technology exists...



...but does the software support it?

How do we support easy navigation with a screen reader?

How do we support easy navigation with a screen reader?

Add semantic meaning to tags

Semantic (landmark) elements

ARIA roles—the “old” way

- Give non-semantic elements (like <div>s) a `role` attribute to provide semantic meaning

```
<div role="main">
```

```
<div role="navigation">
```

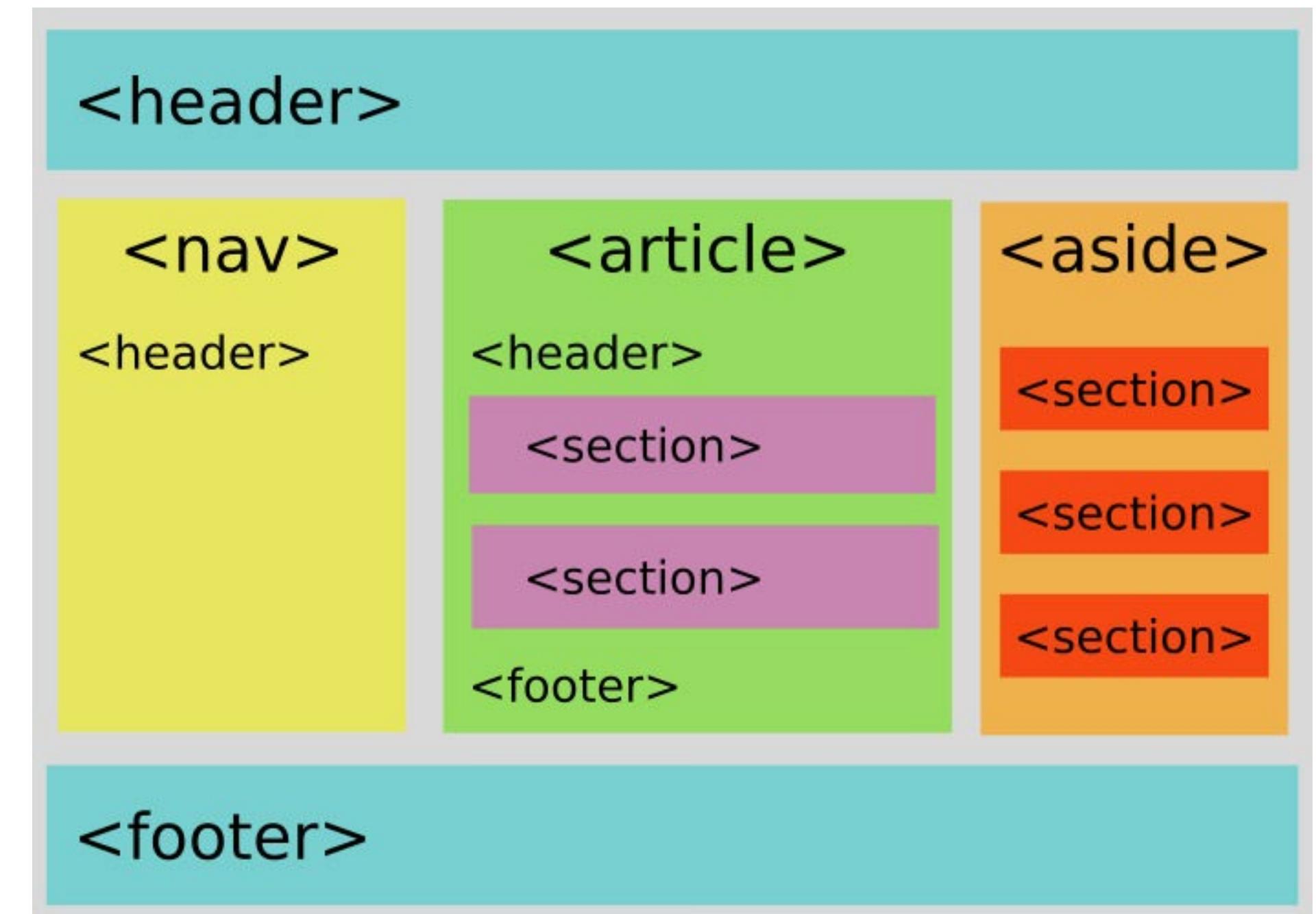
```
<div role="form">
```

- <https://www.w3.org/TR/wai-aria-practices/examples/landmarks/HTML5.html>

Semantic (landmark) elements

HTML5 tags—the “new” way

- Dedicated semantic tags
- [https://www.w3schools.com/html/html5 semantic elements.asp](https://www.w3schools.com/html/html5_semantic_elements.asp)



A few other accessibility examples

- “alt” attributes in images
- “aria-label” attributes to describe non-visual elements (like buttons)

```
<button aria-label="Close">X</button>
```

Alt text guidelines

1. Always include an alt attribute, even if it's empty
 2. Describe the information, not the picture
 3. “Active” images and images which contain information require descriptive alt text
 4. Decorative images should have empty alt text
 5. Be succinct, avoid being redundant with text
- <https://webaim.org/techniques/alttext/>
 - <https://www.abilitynet.org.uk/blog/five-golden-rules-compliant-alt-text>

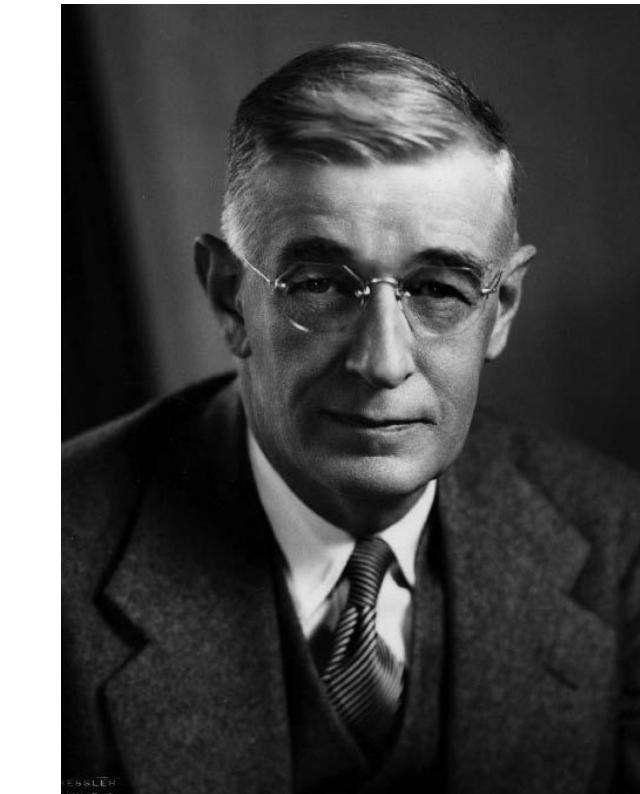


Icons in Google Docs

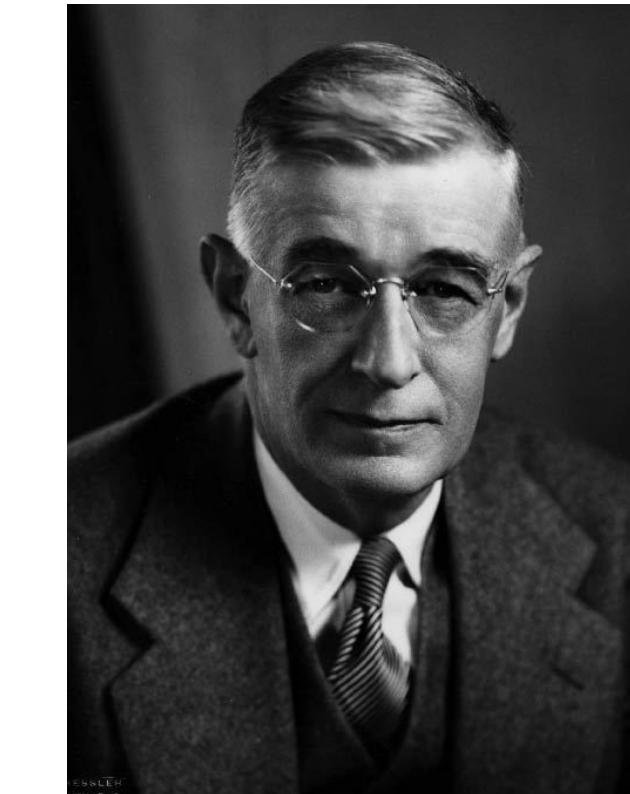


Cover photos on Twitter/Facebook

Which alt text would best describe this image?



Which alt text would best describe this image?



- ``
- ``
- ``
- ``
- ``

Accessibility validators

- WAVE <http://wave.webaim.org/>
- ACheker <https://achecker.ca/checker/index.php>
- Both over-report problems, requires you to think through whether something is actually an accessibility issue
- Can try on your own with a screen reader
 - VoiceOver (Mac, under Settings -> Accessibility)
 - NVDA (Windows, requires download from <https://www.nvaccess.org/>)

Accessibility Resource

<https://it.studentlife.uci.edu/accessibility/basics/>

The screenshot shows a Firefox browser window with the URL <https://it.studentlife.uci.edu/accessibility/basics/index.php> in the address bar. The page title is "Accessibility Basics" and the sub-page title is "Accessibility Basics Home". The main content includes a "Welcome" section, a "How to Navigate the Slide Deck" section, and a sidebar titled "Is this a training?". The footer features the UCI logo and links to Accessibility Basics and IT STUDENTLIFE.UCI.EDU.

Accessibility Basics

Accessibility Basics Home

Welcome

Welcome! This website is intended to be a quick and easy reference for users interested in learning and practicing **digital accessibility**. If you encounter any accessibility issues while using the site, you may report them or request assistance from our [accessibility statement and help page](#).

How to Navigate the Slide Deck

Use the page navigation buttons at the bottom of the page to advance to the next slide in the presentation. This will help ensure that you proceed in the recommended order. Once you are familiar with the "Concepts" and "5 Basic Skills" material, you may use the primary navigation menu options at the top of the screen to review material or explore additional information.

Is this a training?

Although the content is designed for learning accessibility concepts and skills, this is not a formal training and there is no signup/registration process or records of completion. The site is intended for introductory learning with easy access for return visits.

For formal trainings, please see our [Trainings](#) page to learn more about some of the options we have available at the UCI campus.

NEXT ►

Slide 1 of 23

UCI

Accessibility Basics

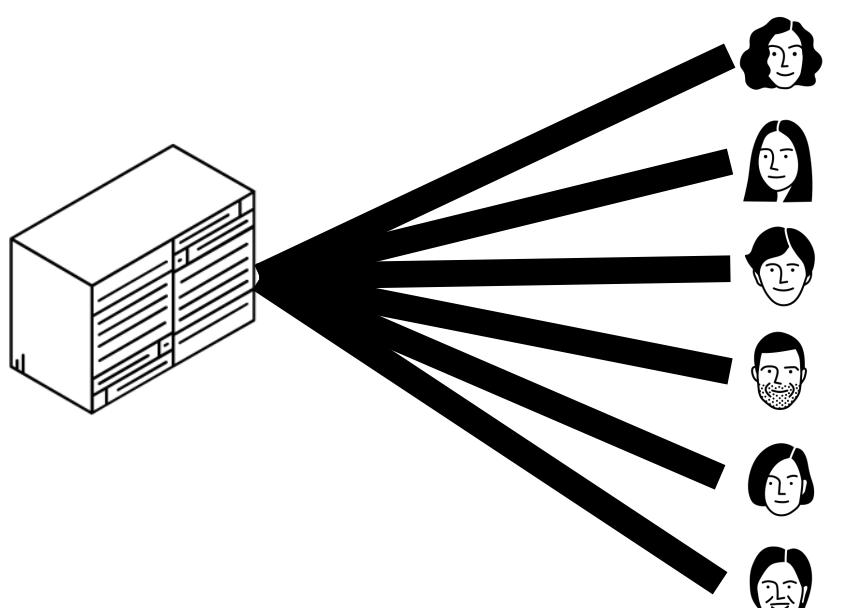
IT STUDENTLIFE.UCI.EDU

Wrap-up:
Inclusive design
is better for everyone

Three waves of computing



Mainframe
computing



“Many to one”



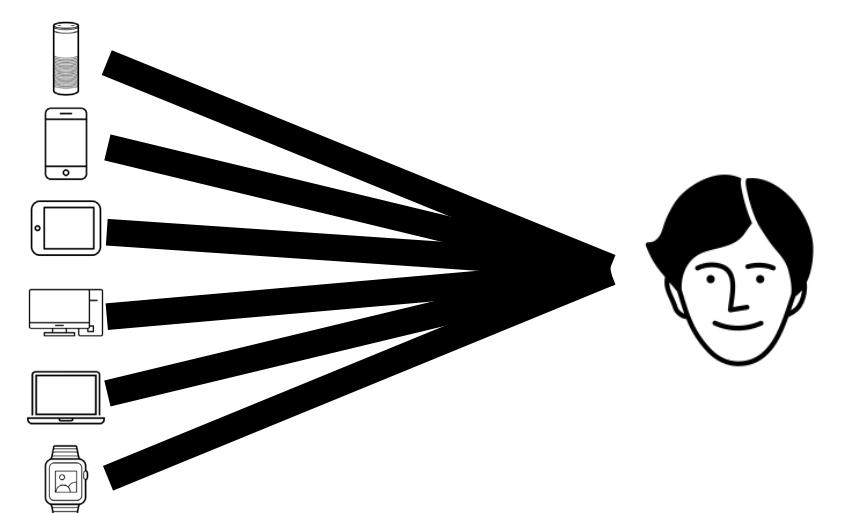
Personal
computing



“One to one”



Ubiquitous
computing



“One to many”

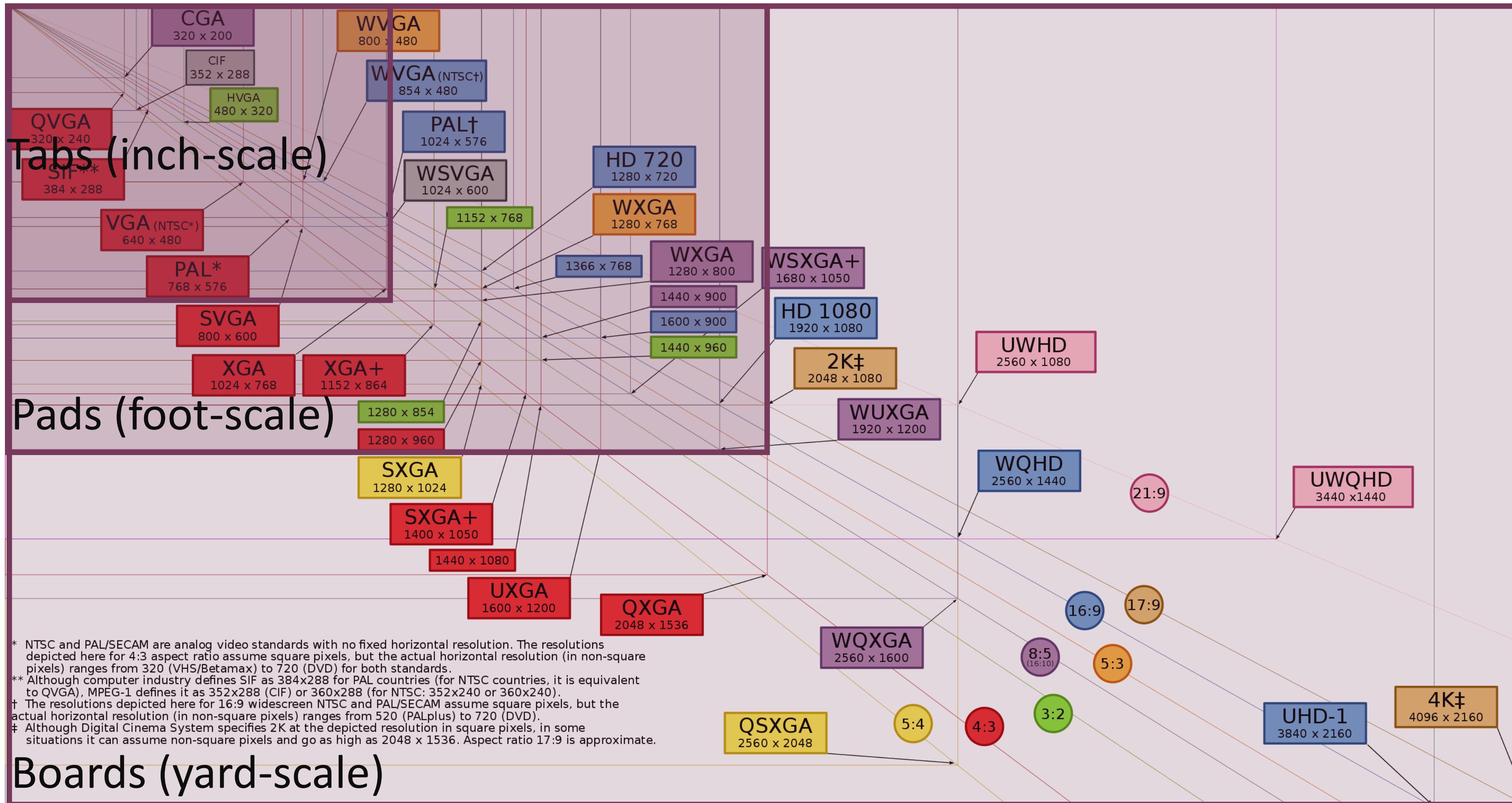
Websites in the personal computing era

- 960 px wide was pretty common
 - Most screens were 1024x978, leave some room for vertical scrollbar
 - Nicely divisible, can create even columns



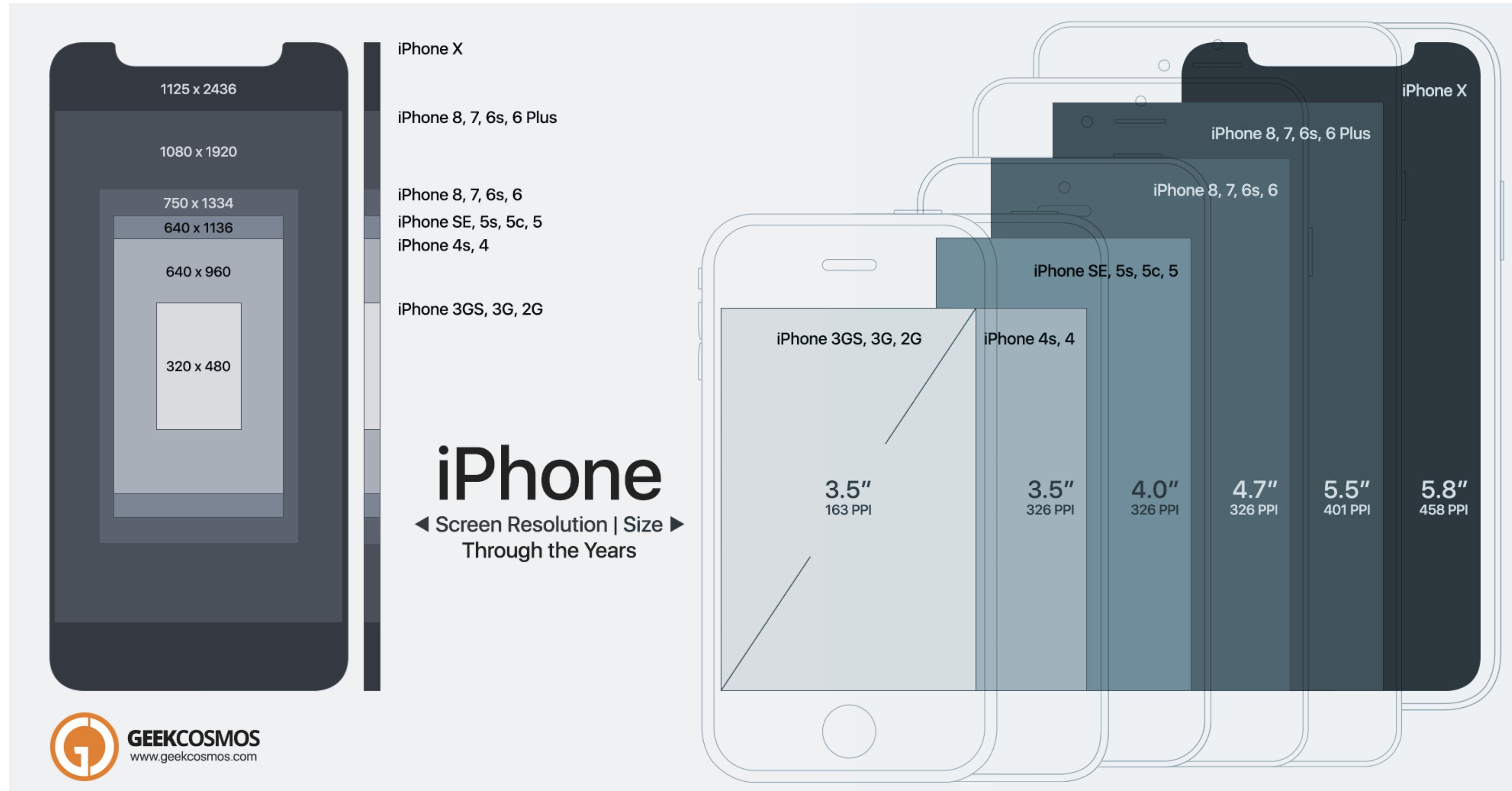
<https://960.gs/>

Websites today: ubiquitous computing



https://en.wikipedia.org/wiki/Display_resolution

Websites today: just the iPhone!



So... how do we account for this?

Responsive design or Adaptive design

Responsive vs Adaptive

Responsive design

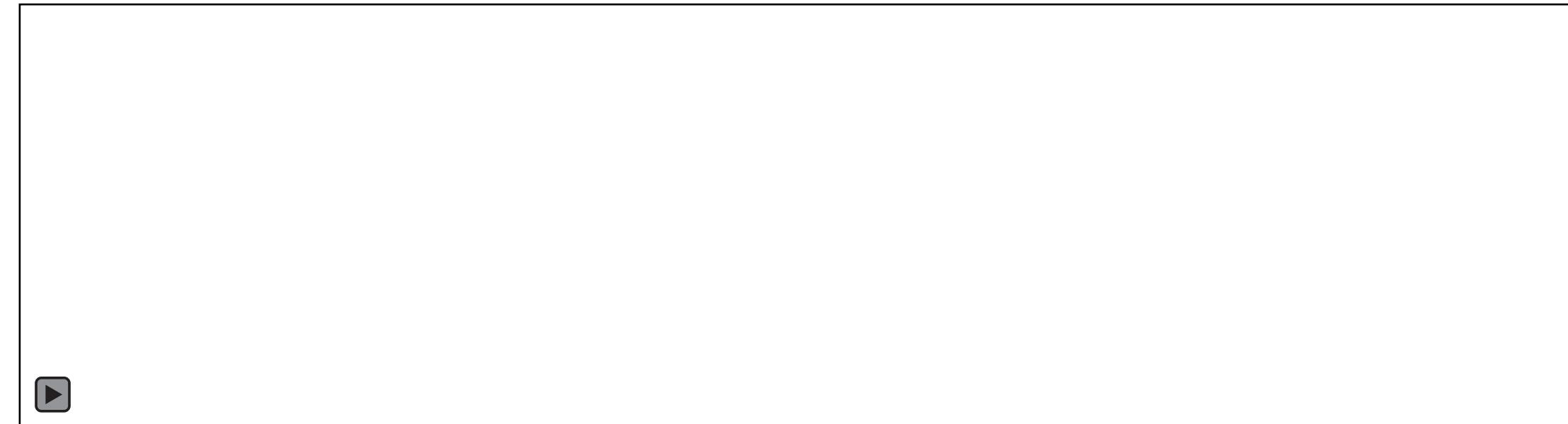
- Develop one set of HTML and CSS which changes layout depending on screen sizes

Adaptive design

- Develop and maintain multiple sets of code, change layout depending on device type and screen size

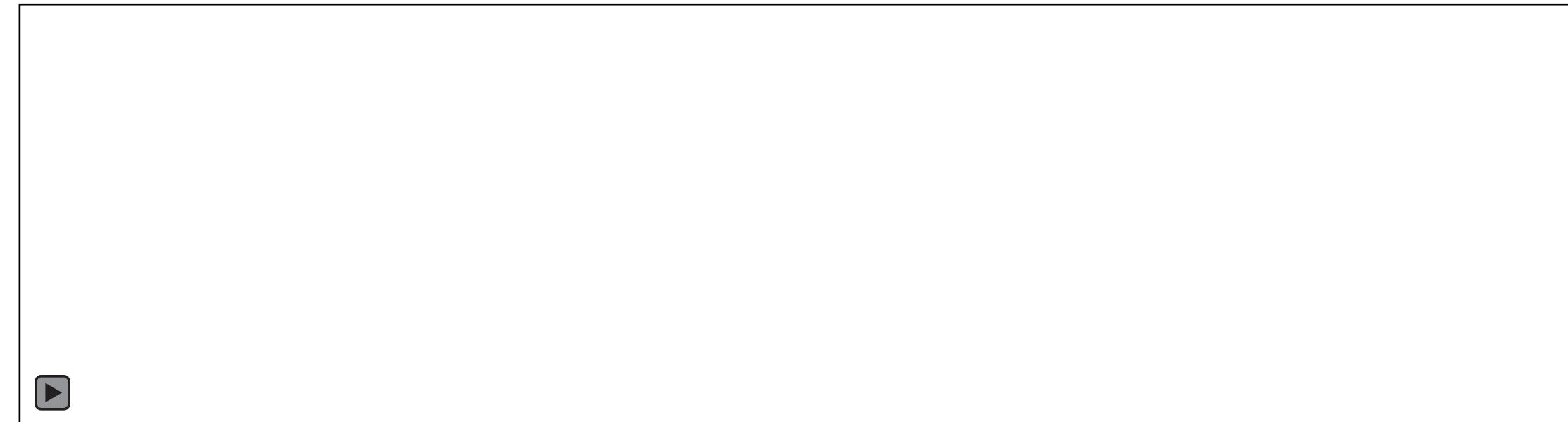
Responsive or Adaptive?

- A Top is responsive, bottom is adaptive
- B Top is adaptive, bottom is responsive
- C Both are responsive
- D Both are adaptive
- E These are neither responsive nor adaptive



Responsive or Adaptive?

- A Top is responsive, bottom is adaptive
- B Top is adaptive, bottom is responsive
- C Both are responsive
- D Both are adaptive
- E These are neither responsive nor adaptive



Responsive vs Adaptive

Responsive design

- + Easier to maintain one code base, future-proof
- Worse performance; requires downloading entire stylesheet
- Emphasis on making it “look right” rather than creating an experience

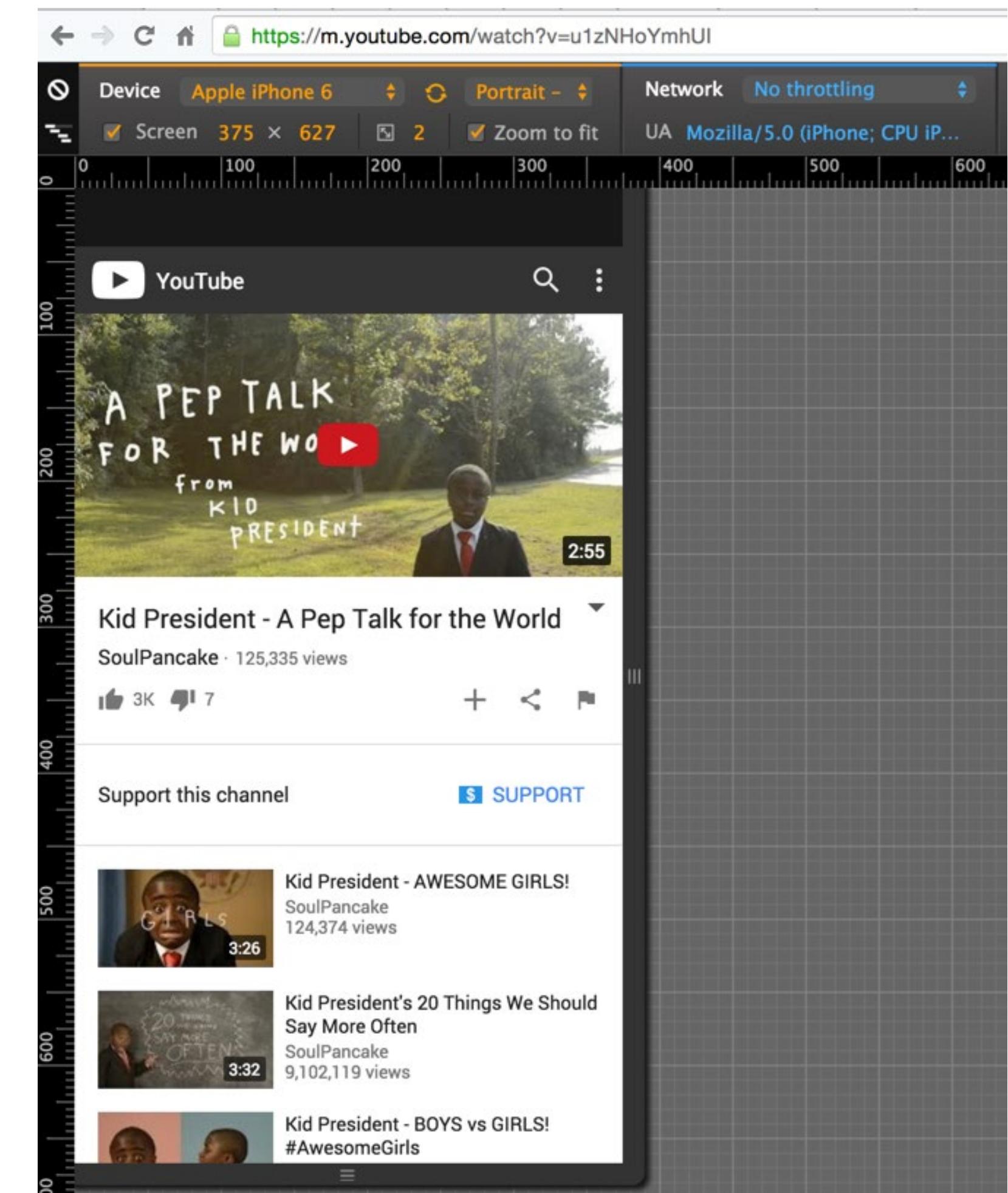
Adaptive design

- + Can cater experience to a device’s capabilities and performance
- Much more difficult to maintain separate codebases
- Limits development to a few key capabilities because you have to implement for everything

Most pages are responsive,
but sometimes it's crucial
to create the best experience

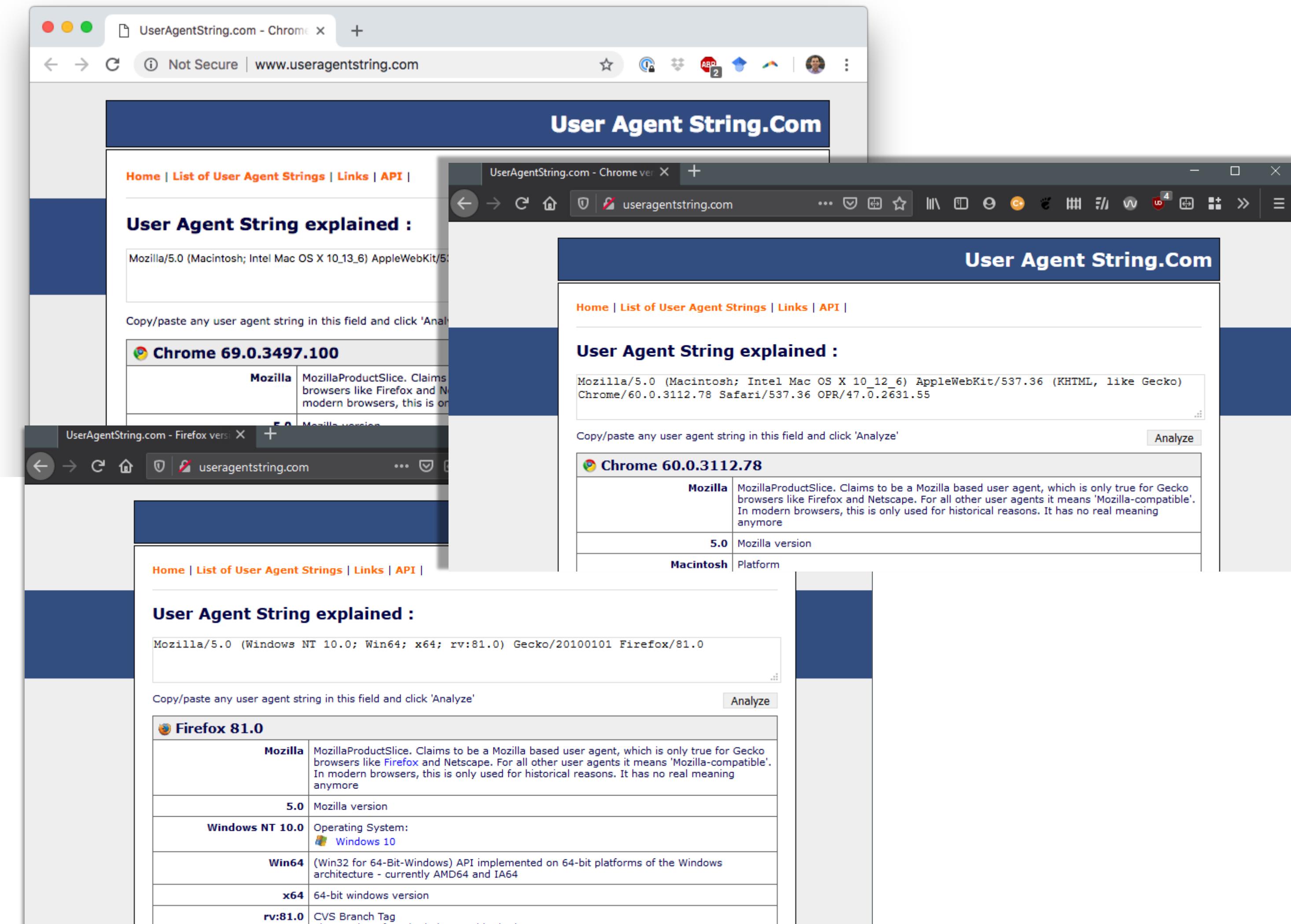
Adaptive design

- Video = a lot to load
 - Why send a higher resolution than the screen can render?
 - Why use up your own bandwidth?
 - Laggy videos mean unhappy users
- Google can afford the development burden



Adaptive design

- User agent string accessible via JavaScript
 - `navigator.userAgent`
- There's usually a better way
 - Do you care about the browser or operating system?
Or is resolution sufficient?
 - Can be spoofed or incorrect



https://developer.mozilla.org/en-US/docs/Web/HTTP/Browser_detection_using_the_user_agent

Adaptive design

- Media queries in CSS

```
/* CSS */  
@media screen and (device-width: 375px) and (device-height: 667px)  
and (-webkit-device-pixel-ratio: 2) {  
    /* iPhone 8-specific CSS */  
}
```

- Load appropriate external stylesheet

```
<!--HTML-->  
<head>  
    <link rel="stylesheet" media="screen and (device-width: 375px)  
        and (device-height: 667px) and (-webkit-device-pixel-ratio: 2)" href="iPhone8.css">  
</head>
```

Media query syntax

- @media
- screen, print, speech, all
- min-width, max-width
- orientation, -webkit-min-device-pixel-ratio
- Many, many more

https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

Transitioning to responsive design

Breakpoints

- The point at which your design “breaks” and is no longer visually appealing or usable
- Designs vary, but most have 3-5 breakpoints
 - extra small (old mobile), small (mobile), medium (tablet), large (laptop or desktop), extra large (wide desktop or wall display)
 - Again, somewhat similar to Weiser’s three types of computers

Breakpoints

```
@media screen and (max-width: 640px) {  
    /* small screens */  
}
```

```
@media screen and (min-width: 640px) and (max-width:  
1024px) {  
    /* medium screens */  
}
```

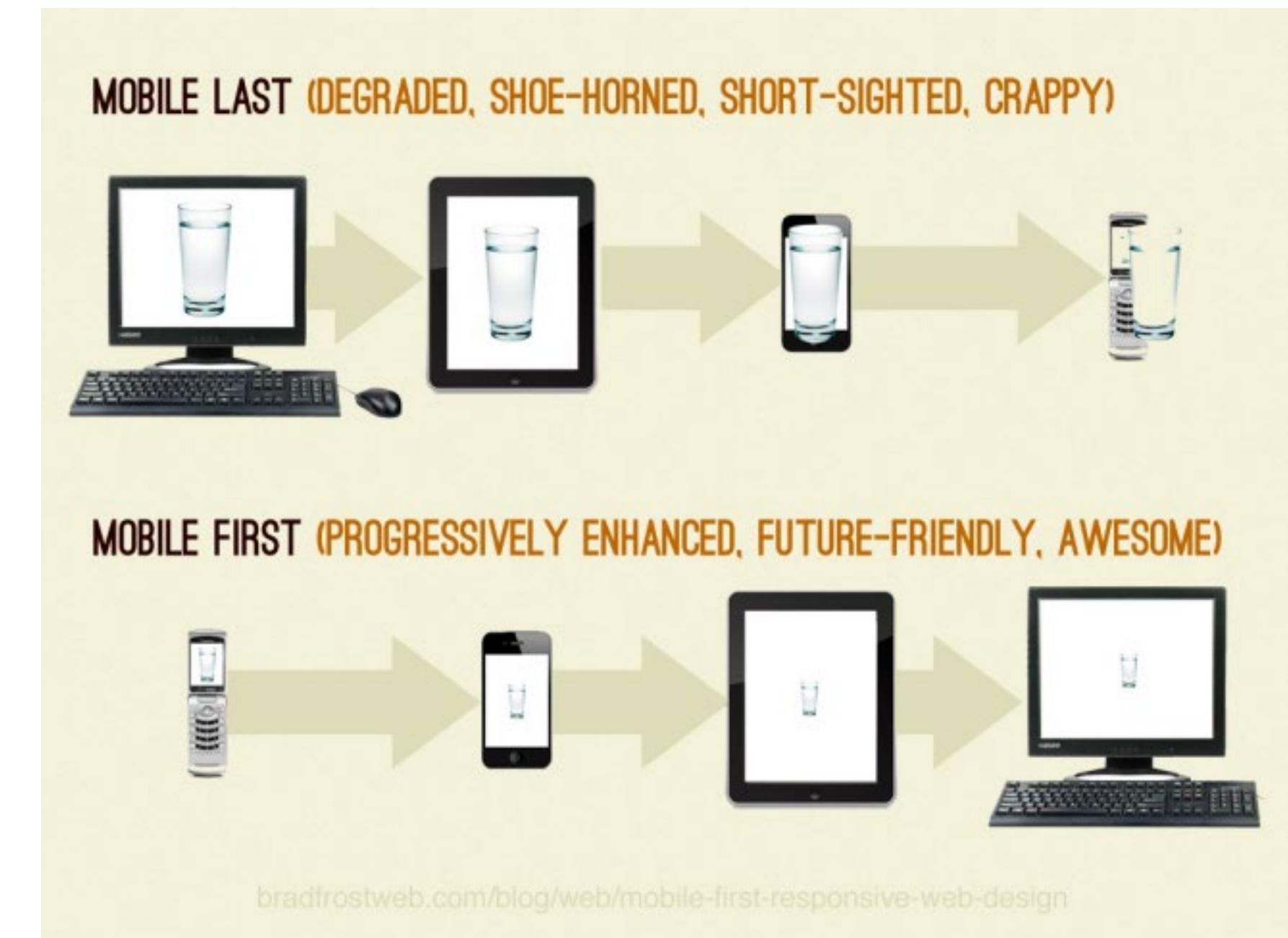
```
@media screen and (min-width: 1024px) {  
    /* large screens */  
}
```

Responsive design

- Fluid grids
 - Lay out content in columns whose widths can vary
 - Bootstrap (and other CSS toolkits) helps with this; more on that in a bit
- Flexible images
 - Let image size change based on screen layout
 - Put images in containers which will scale appropriately
 - Set `width: 100%, max-width: 100%, height: auto`

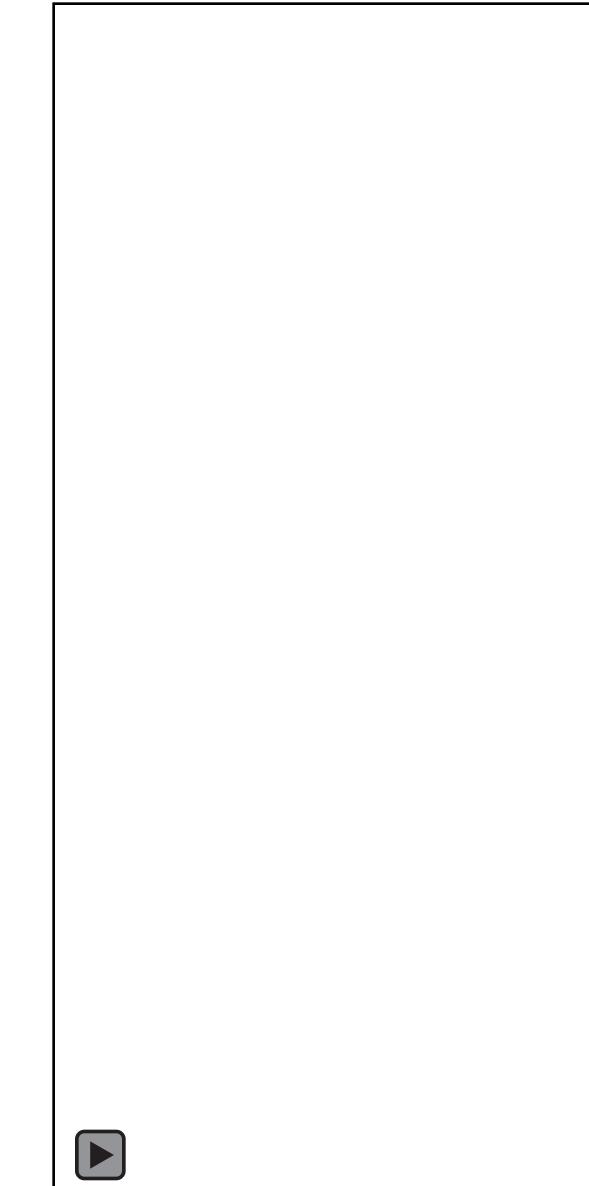
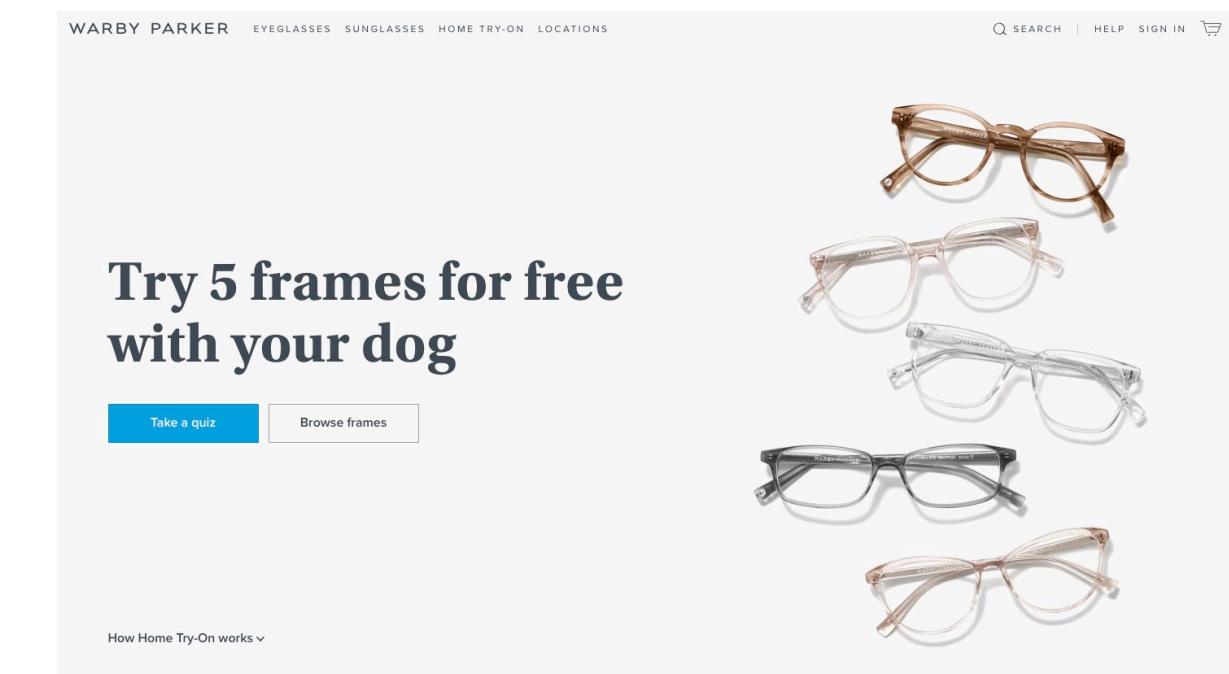
Mobile-first design

- “Graceful degradation” vs. “progressive enhancement”
- Plan your design for mobile
- Then make your app *better* with more real estate
 - Add more features
 - Make existing features easier to navigate



A few tips for mobile design

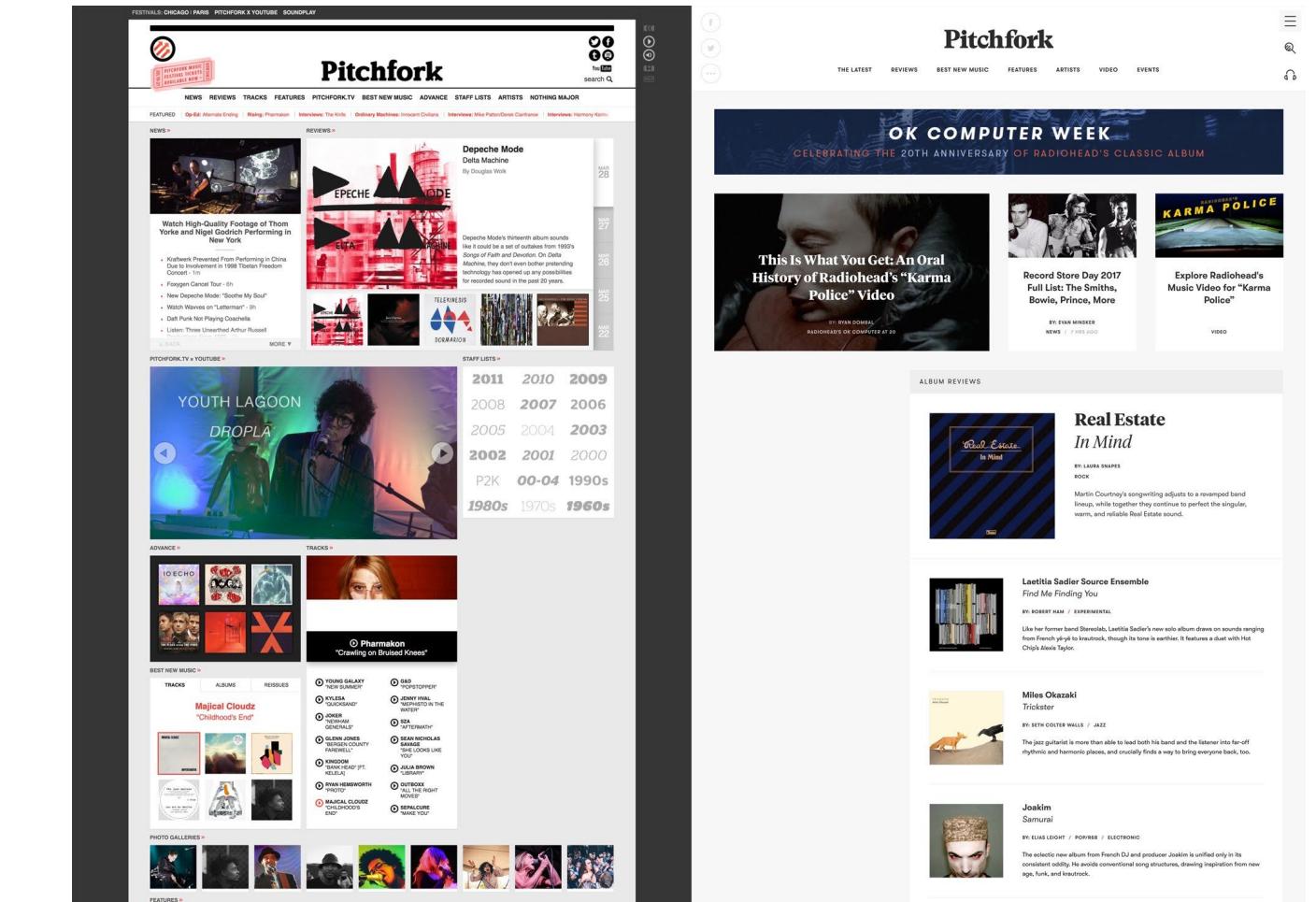
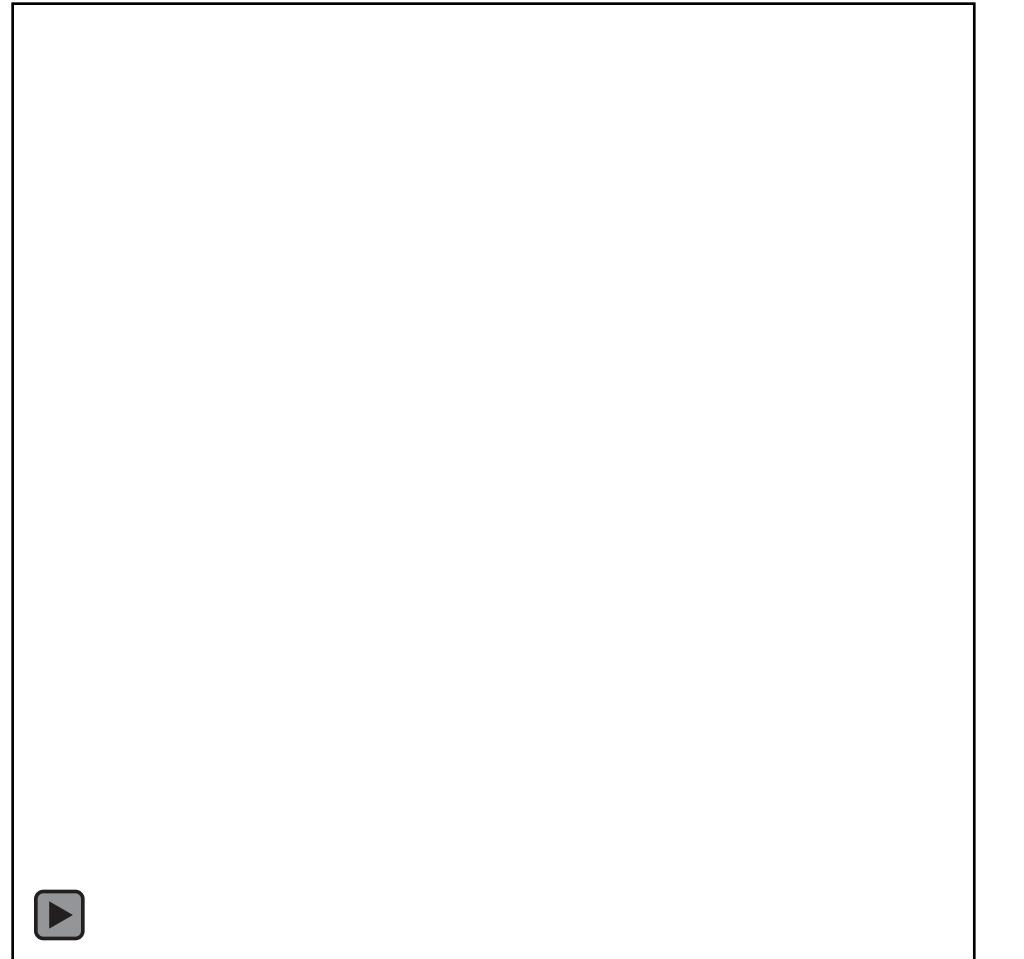
- Show the same content, organize it appropriately
- Stack content vertically
- Show navigation on demand
- Larger touch targets



<https://www.bluefountainmedia.com/blog/desktop-vs-mobile-three-key-website-design-differences>

Mobile-first, not mobile-only

- Copying mobile UI to desktop creates inefficiencies
 - Extra clicks to navigate
 - Underutilized real estate



<https://blog.prototypio.io/mobile-first-desktop-worst-f900909ae9e2>

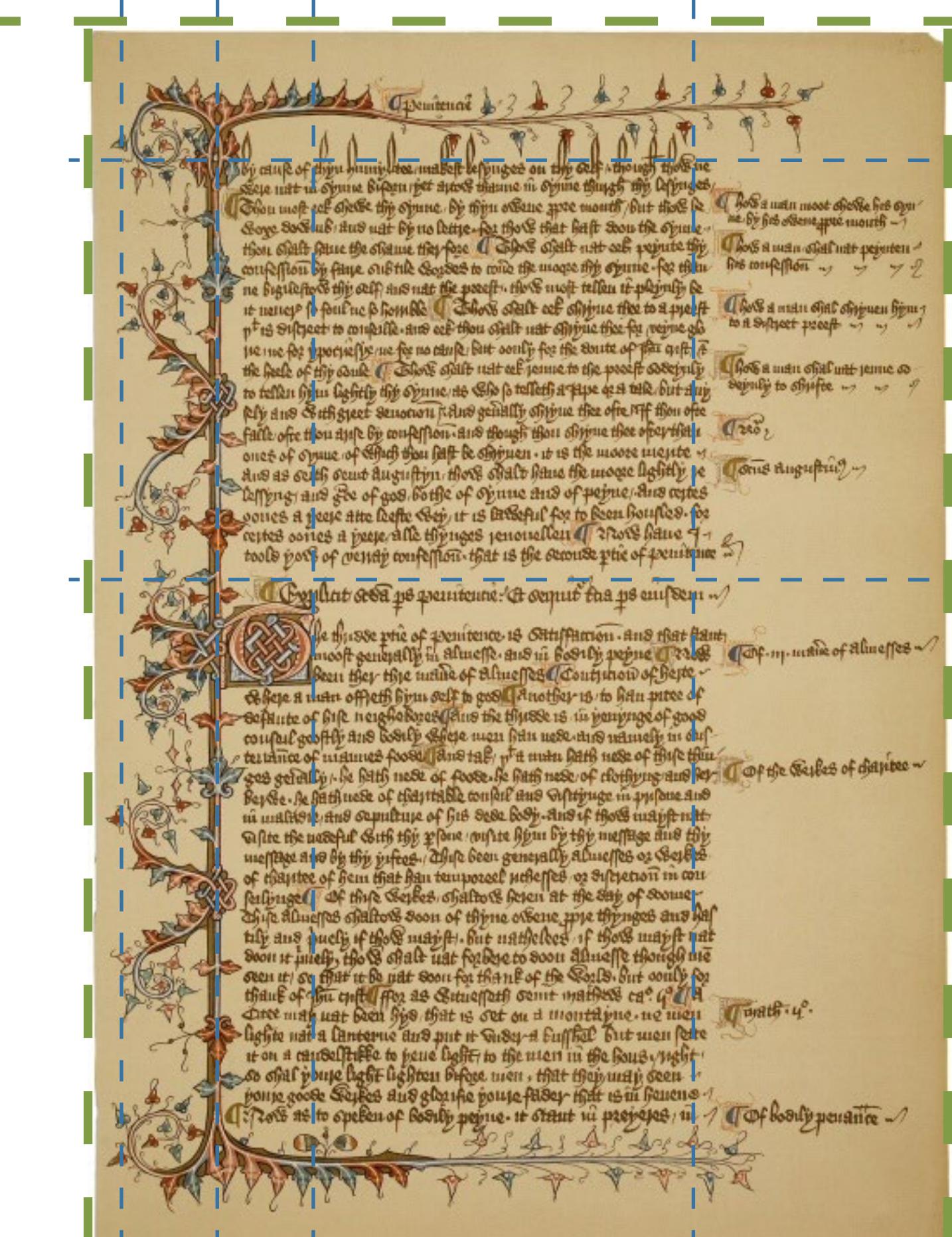
Mobile-first, not mobile-only

- Plan your design for mobile
- But consider how the experience should change on desktop, etc.
- Go beyond making everything bigger
 - *Enhance* your design

Grid-based layouts

Grid-based layouts

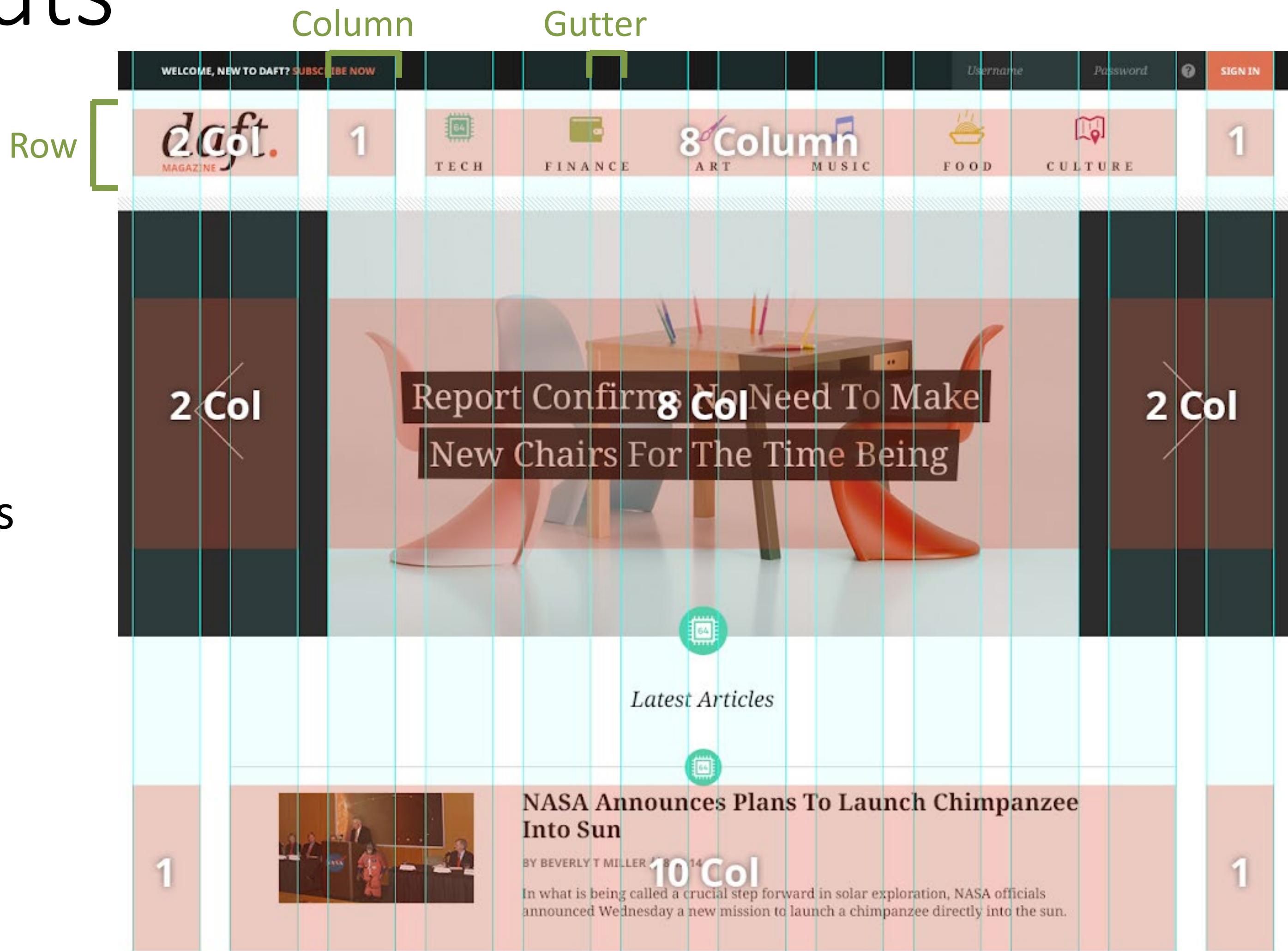
- Established tool for content arrangement
- Gridded content is familiar and easy to follow
- In general, it's good to target fewer lines
 - But breaking that rule is important for creativity and attention-grabbing



<http://printingcode.runemadsen.com/lecture-grid/>

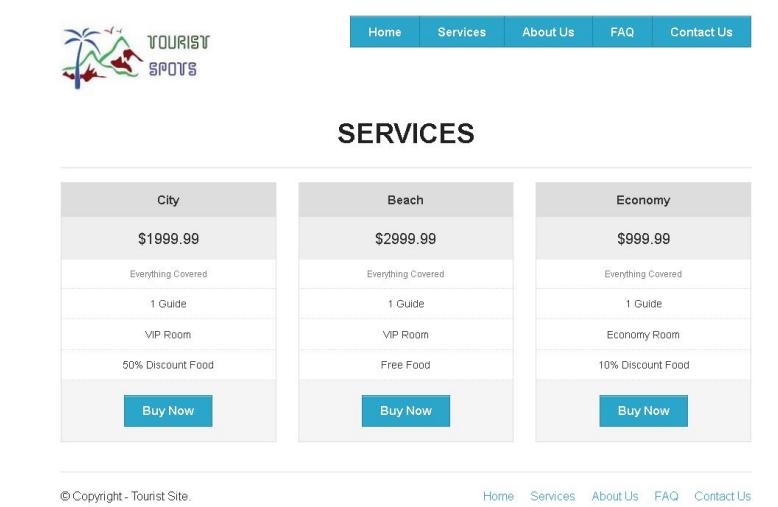
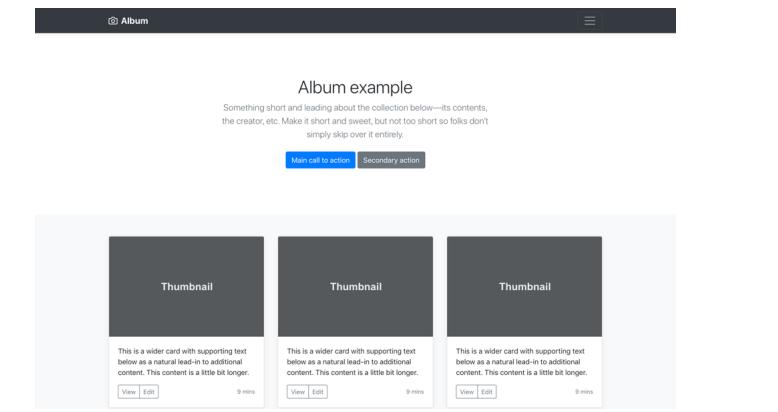
Grid-based layouts

- Rows
- Columns
- Gutters
- Padding/spacing
 - Defined by specific elements



Grid-based frameworks

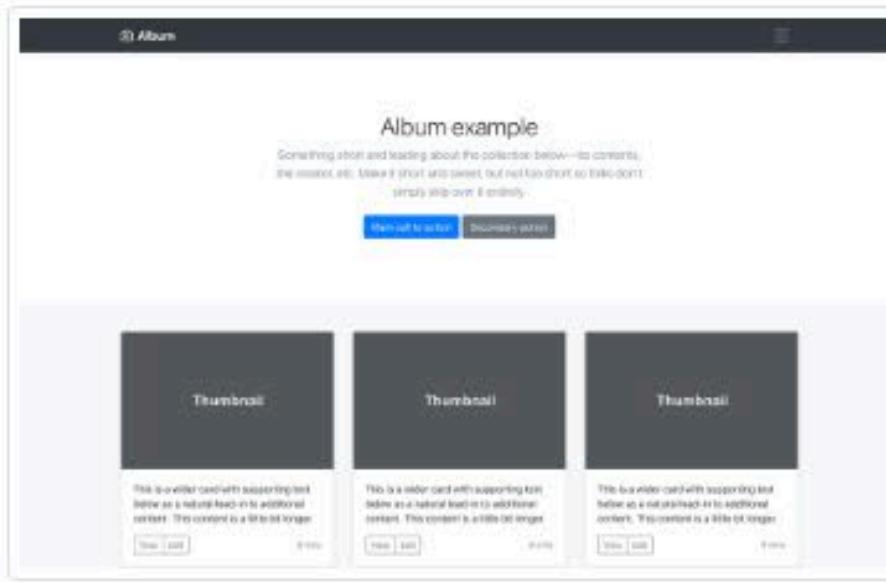
- Bootstrap (<https://getbootstrap.com/>)
 - Most popular, most extensions
- Foundation (<https://foundation.zurb.com/>)
 - Includes icons, drag&drop editor
- Pure.css (<https://purecss.io/>)
 - Small file size, 3.8KB
- Basscss (<https://basscss.com/>)
 - Even smaller, 3.39KB
 - Low-level (closer to raw CSS)



Grid frameworks
make development easier.
What are the downsides?

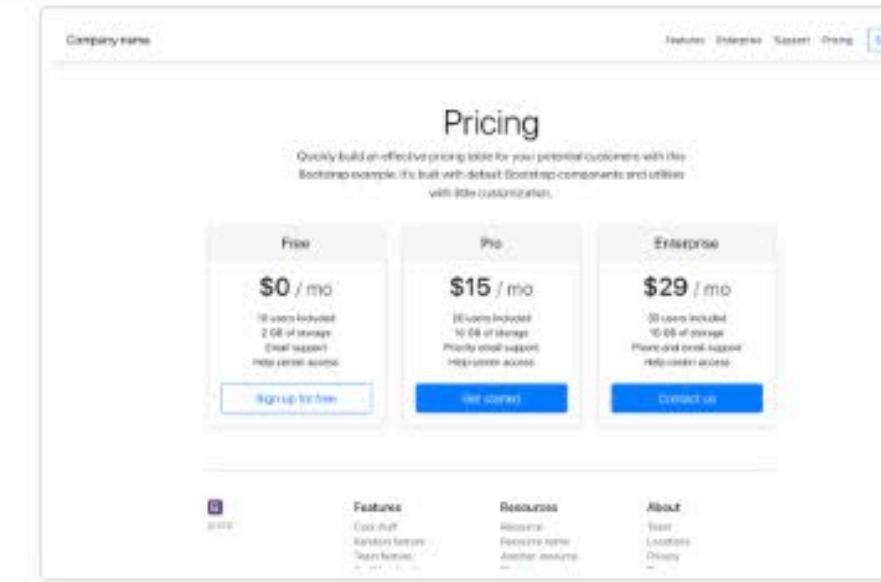
Opposition to Grid-based frameworks

Can lead to similar-looking webpages



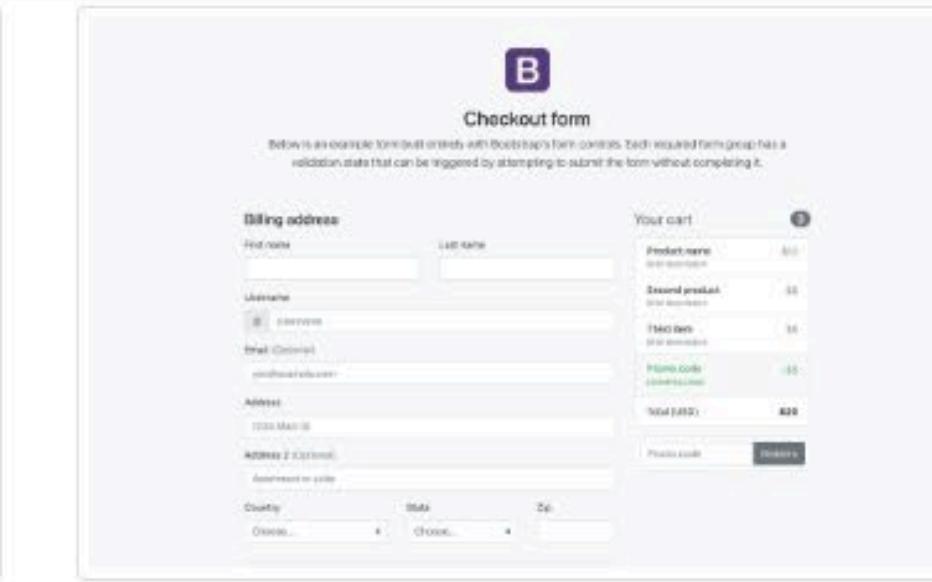
Album

Simple one-page template for photo galleries, portfolios, and more.



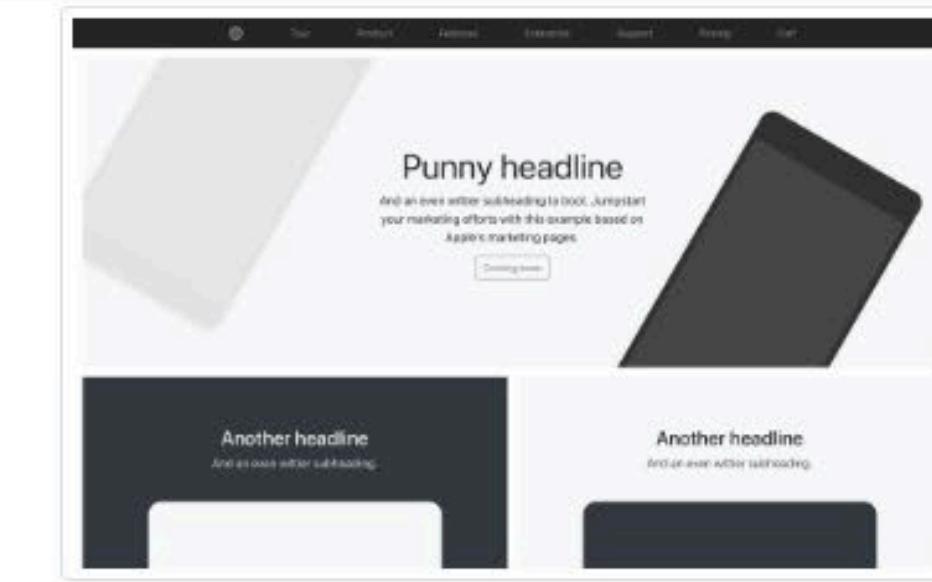
Pricing

Example pricing page built with Cards and featuring a custom header and footer.



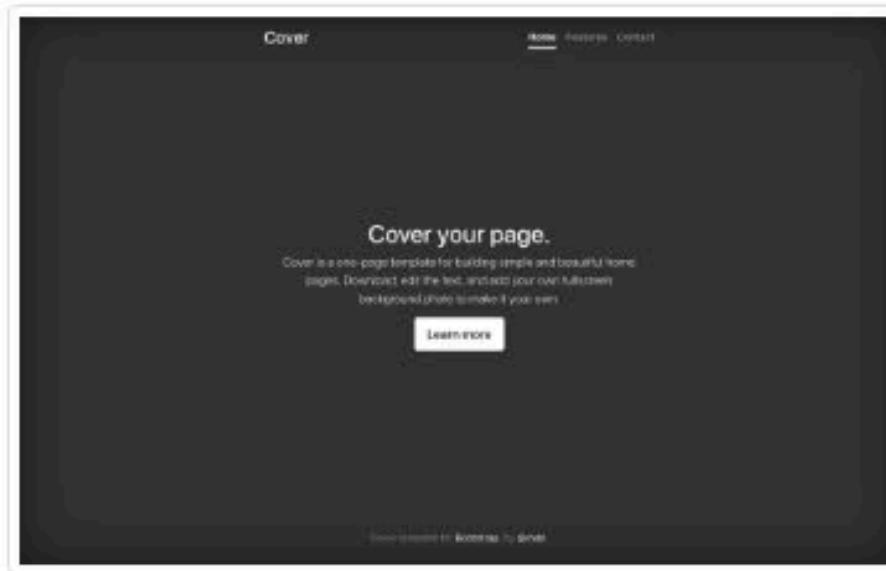
Checkout

Custom checkout form showing our form components and their validation features.



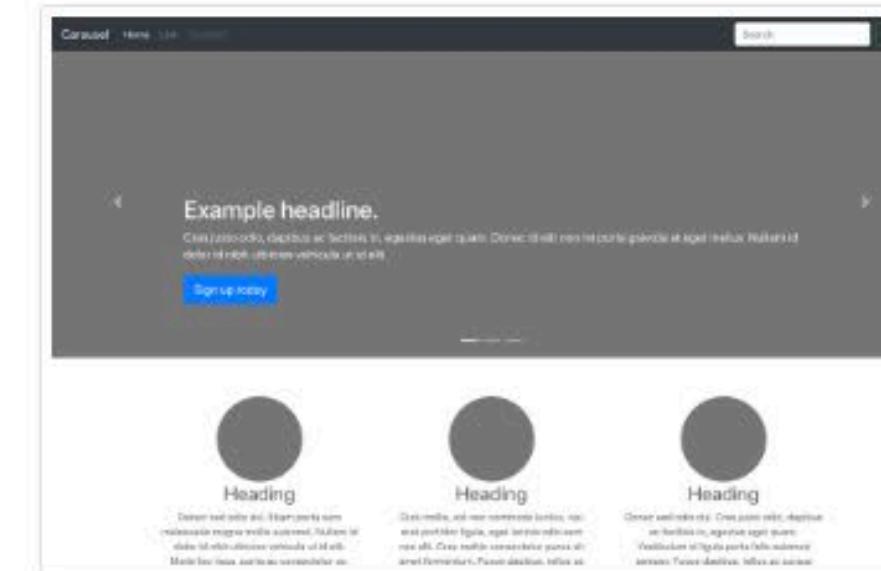
Product

Lean product-focused marketing page with extensive grid and image work.



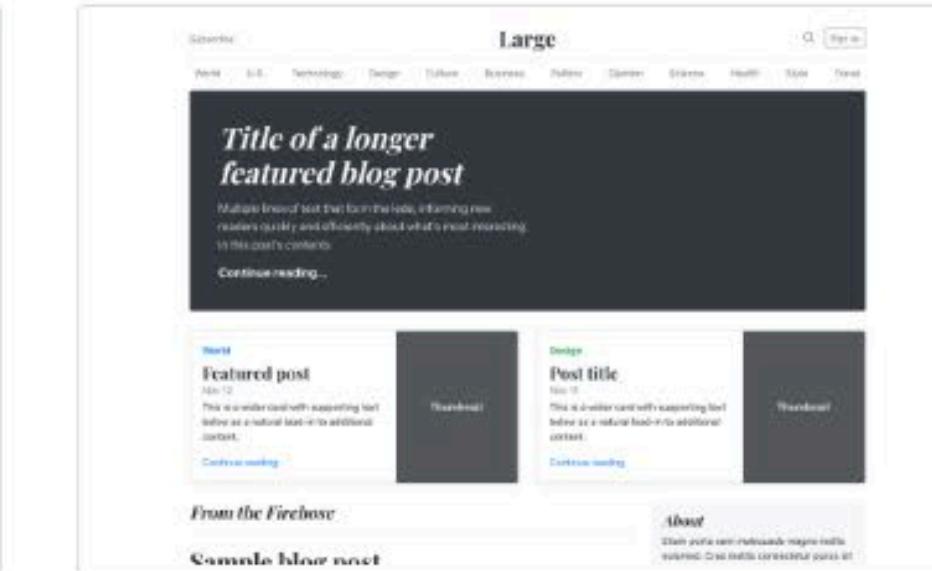
Cover

A one-page template for building simple and beautiful home pages.



Carousel

Customize the navbar and carousel, then add some new components.



Blog

Magazine like blog template with header, navigation, featured content.



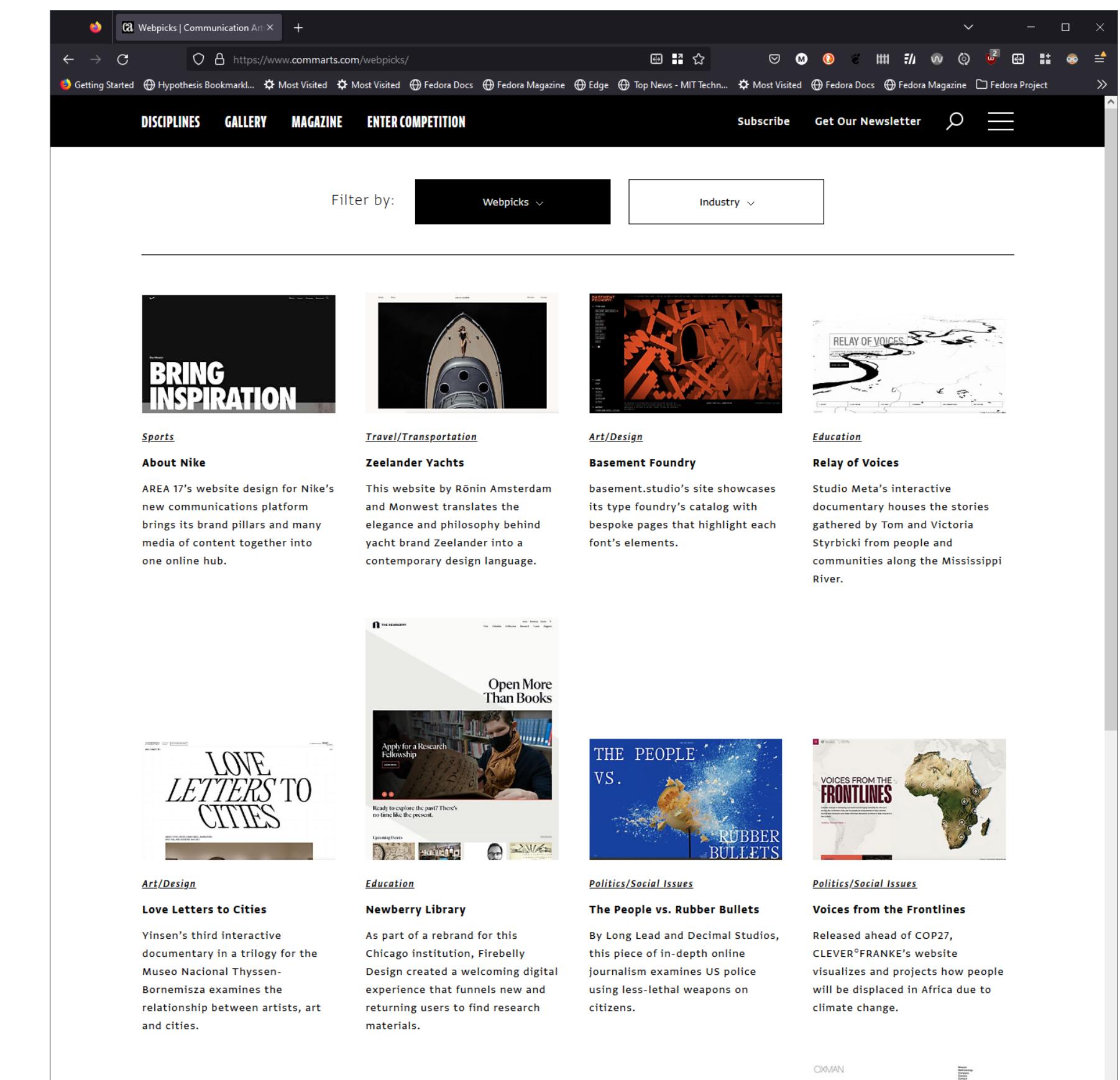
Dashboard

Basic admin dashboard shell with fixed sidebar and nav bar.

Opposition to Grid-based frameworks

For inspiration take a look at **Communication Arts Webpicks**

<https://www.commarts.com/webpicks/>



Opposition to Grid-based frameworks

Can stifle creativity

Themes built by or reviewed by Bootstrap's creators.

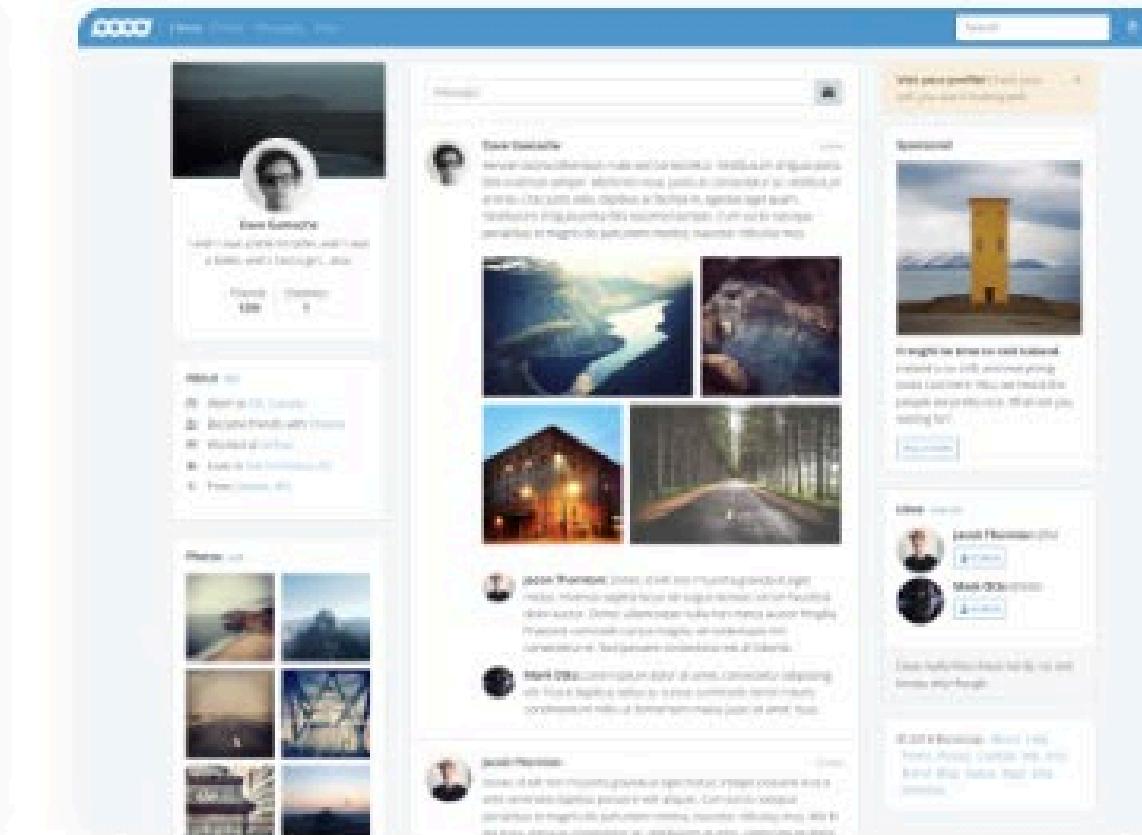
Why our themes?

Built by Bootstrap Team

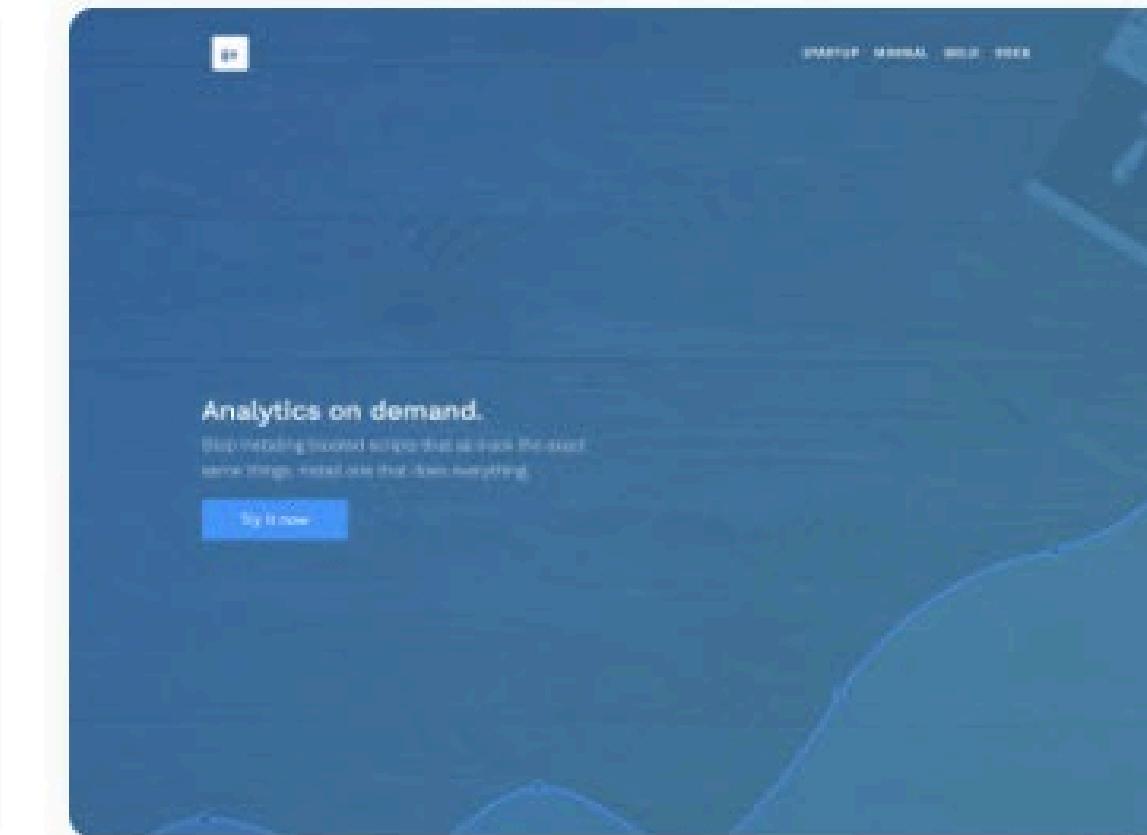
Component-based frameworks designed, built, and supported by the Bootstrap Team.



Dashboard
Admin & Dashboard



Application
Application

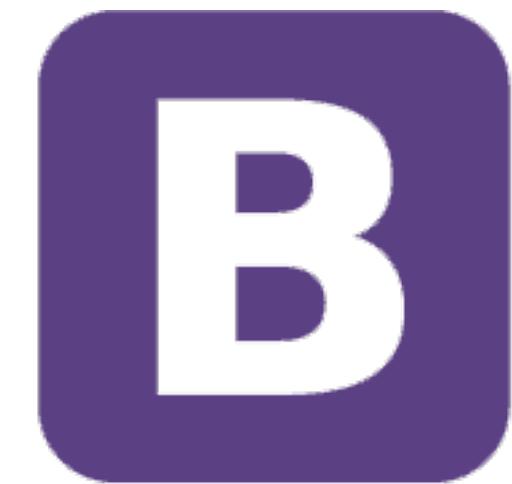


Marketing
Landing & Corporate

Friday Remote Class

Demo some responsive layouts

Dig into Bootstrap



Bootstrap

Today's goals

By the end of today, you should be able to...

- Explain the importance of accessible and semantically meaningful markup
- Generate markup which meets accessibility standards
- Describe how responsive and adaptive design differ and when you might prefer one or the other
- Explain the advantages and disadvantages of a mobile-first design