IN4MATX 285:
Interactive Technology Studio

Practice: Responsive and Adaptive Design

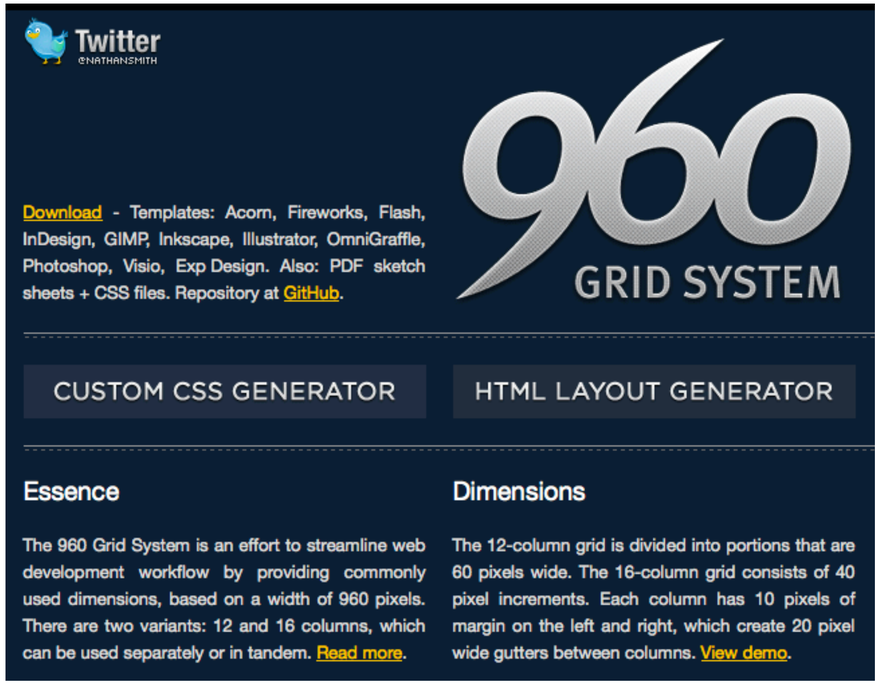
Today’s goals

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

- Explain why we need some form of responsivity
- Define and differentiate responsive and adaptive design
- Implement a grid-based system with Bootstrap

Websites, back in the day

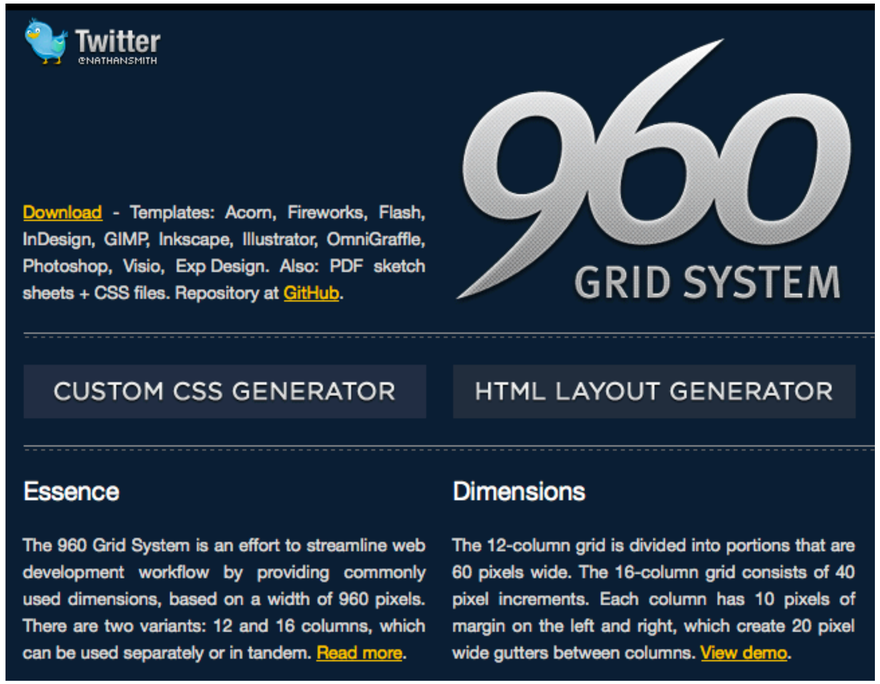
- 960 px wide was pretty common
  - Most screens were 1024x978,
    leave some room for vertical scrollbar
  - Nicely divisible, can create
    even columns



https://960.gs/

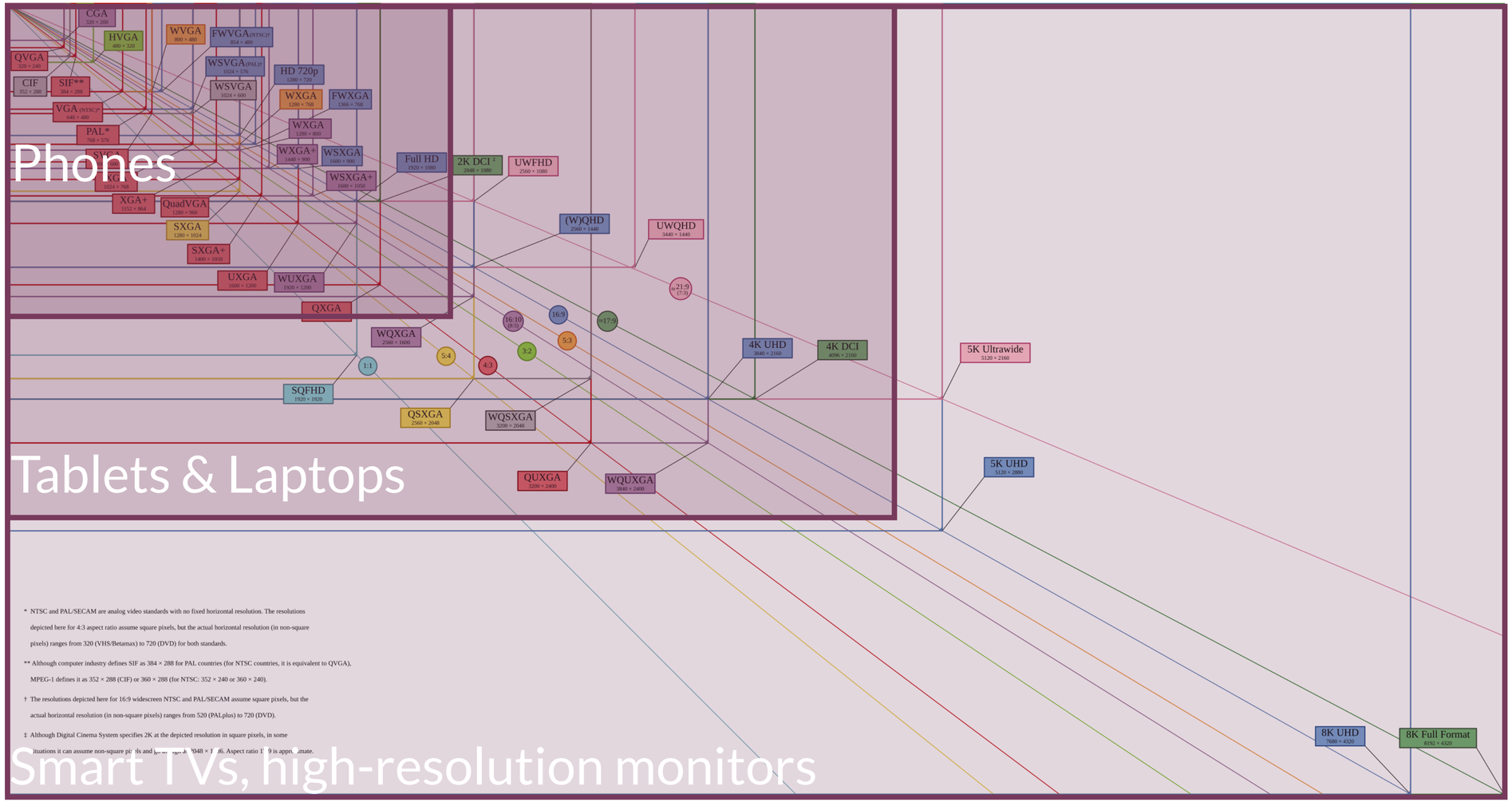
Websites, back in the day

- Because layouts were fixed, one highly-detailed design could go pretty far
- You could (mostly) ensure that everyone would see your page the same way you did



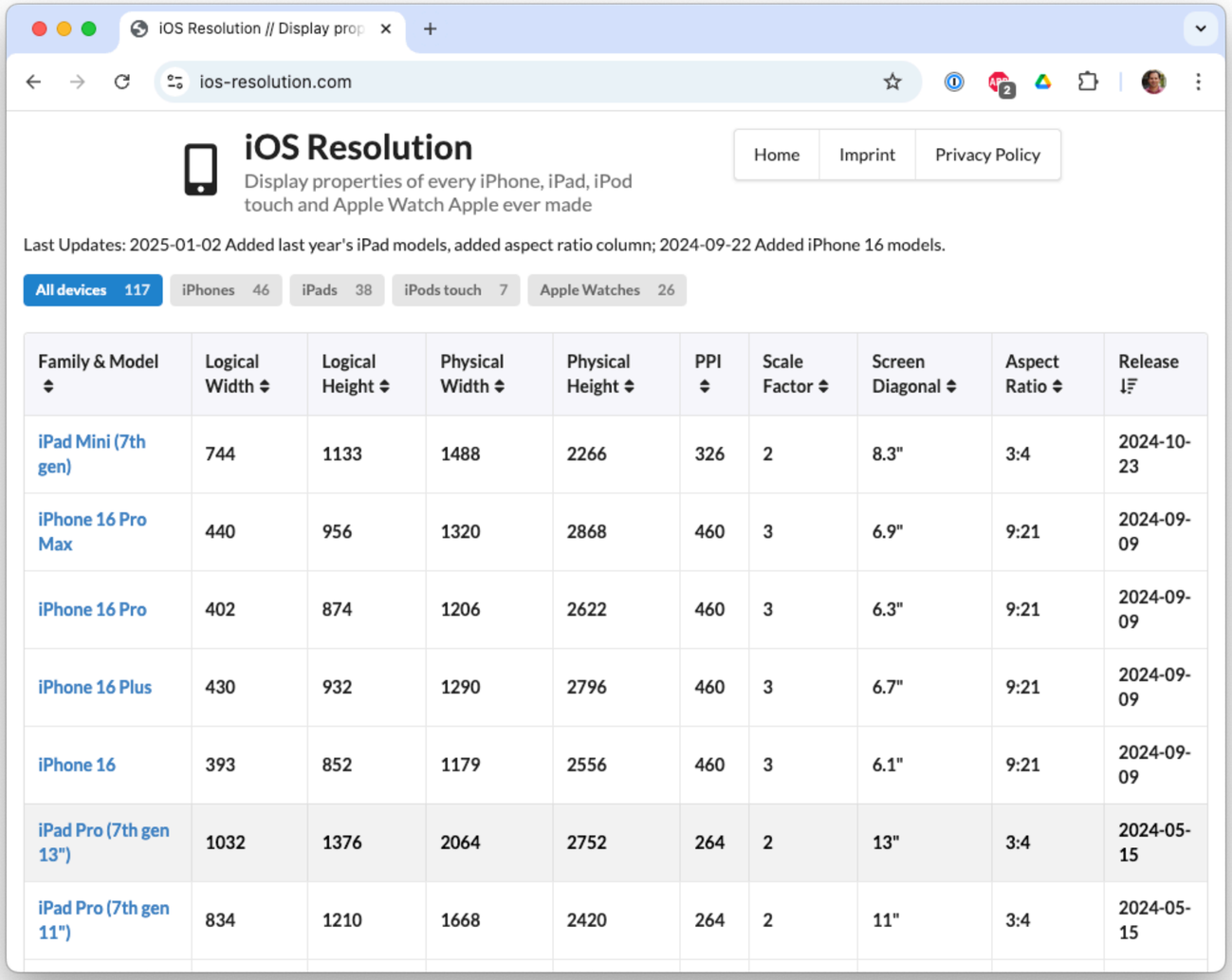
https://960.gs/

Websites today



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

Websites today: just Apple devices!

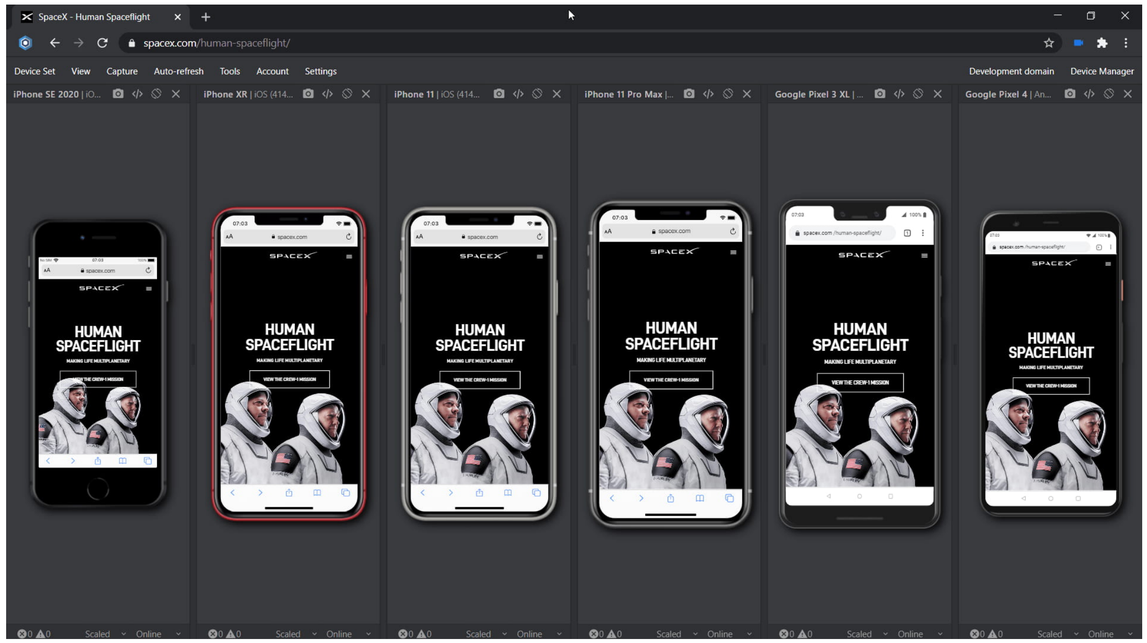


Family & Model	Logical Width	Logical Height	Physical Width	Physical Height	PPI	Scale Factor	Screen Diagonal	Aspect Ratio	Release
iPad Mini (7th gen)	744	1133	1488	2266	326	2	8.3"	3:4	2024-10-23
iPhone 16 Pro Max	440	956	1320	2868	460	3	6.9"	9:21	2024-09-09
iPhone 16 Pro	402	874	1206	2622	460	3	6.3"	9:21	2024-09-09
iPhone 16 Plus	430	932	1290	2796	460	3	6.7"	9:21	2024-09-09
iPhone 16	393	852	1179	2556	460	3	6.1"	9:21	2024-09-09
iPad Pro (7th gen 13")	1032	1376	2064	2752	264	2	13"	3:4	2024-05-15
iPad Pro (7th gen 11")	834	1210	1668	2420	264	2	11"	3:4	2024-05-15

https://www.ios-resolution.com/

Websites today

- A Figma design may not specify precisely how to adjust content for different screen sizes, resolutions, etc.
- When implementing, developers therefore rely on some general strategies for laying out content



So… how do we account for this?

Responsive design or Adaptive design

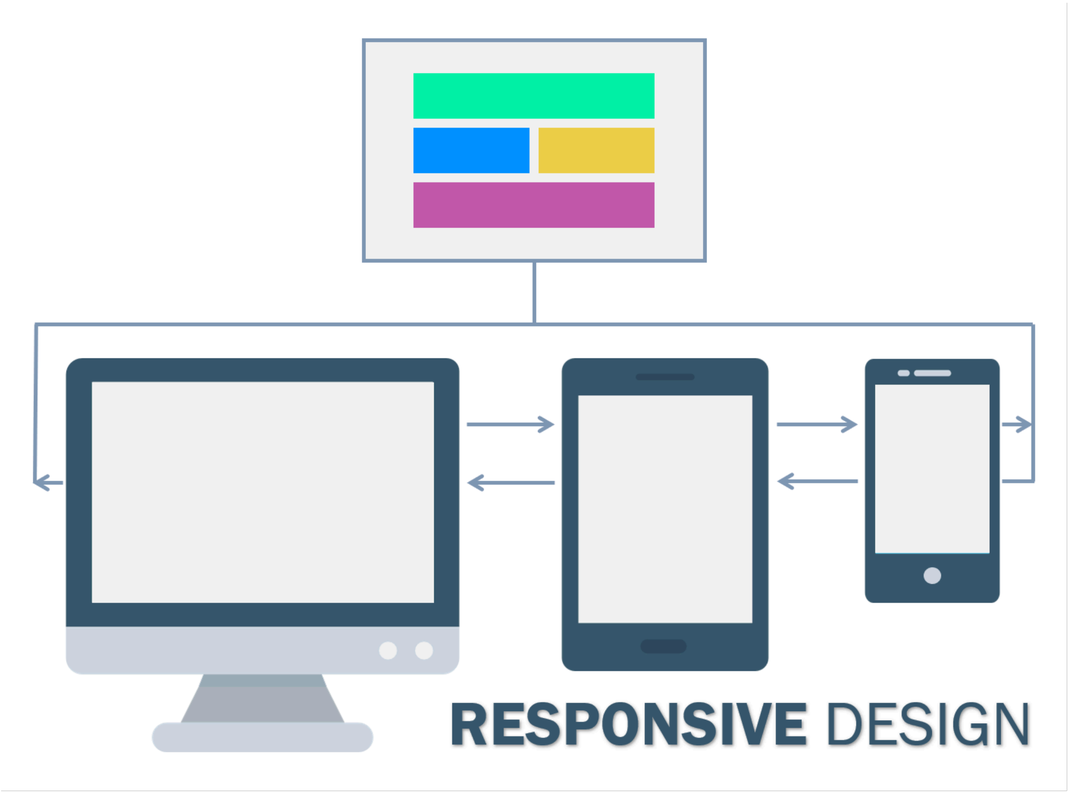
Responsive design

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

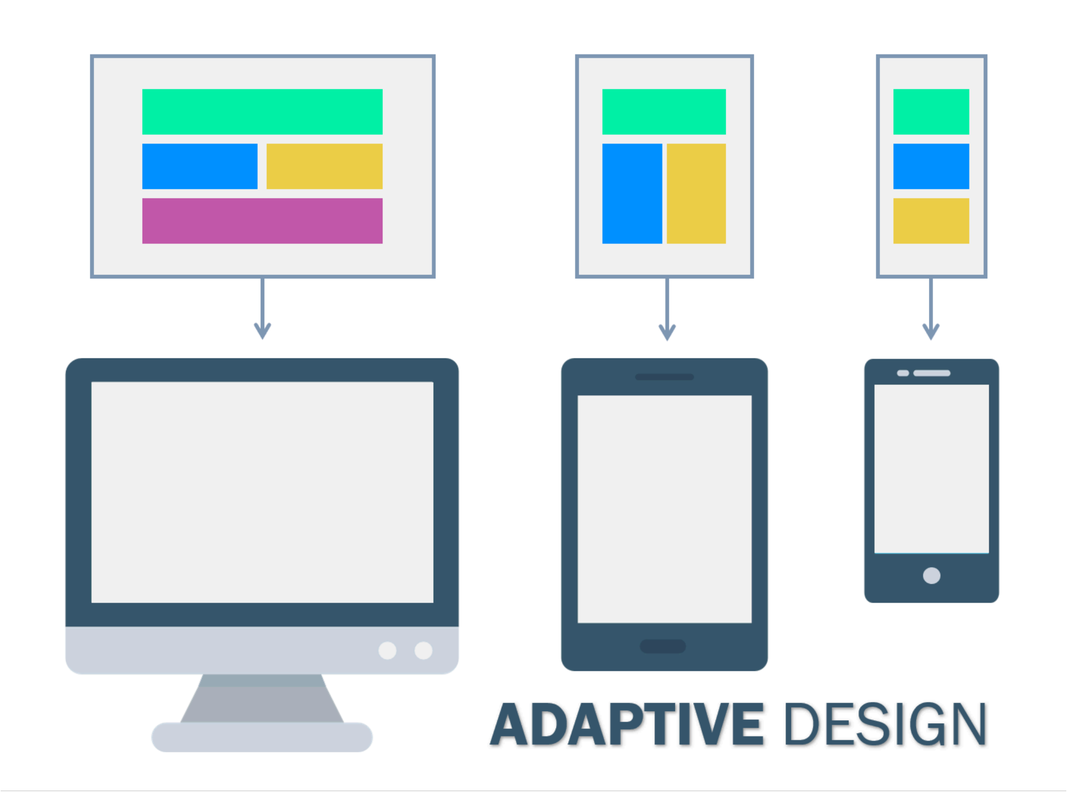
Adaptive design

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

Responsive design



Adaptive design



https://webflow.com/blog/adaptive-vs-responsive-design

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

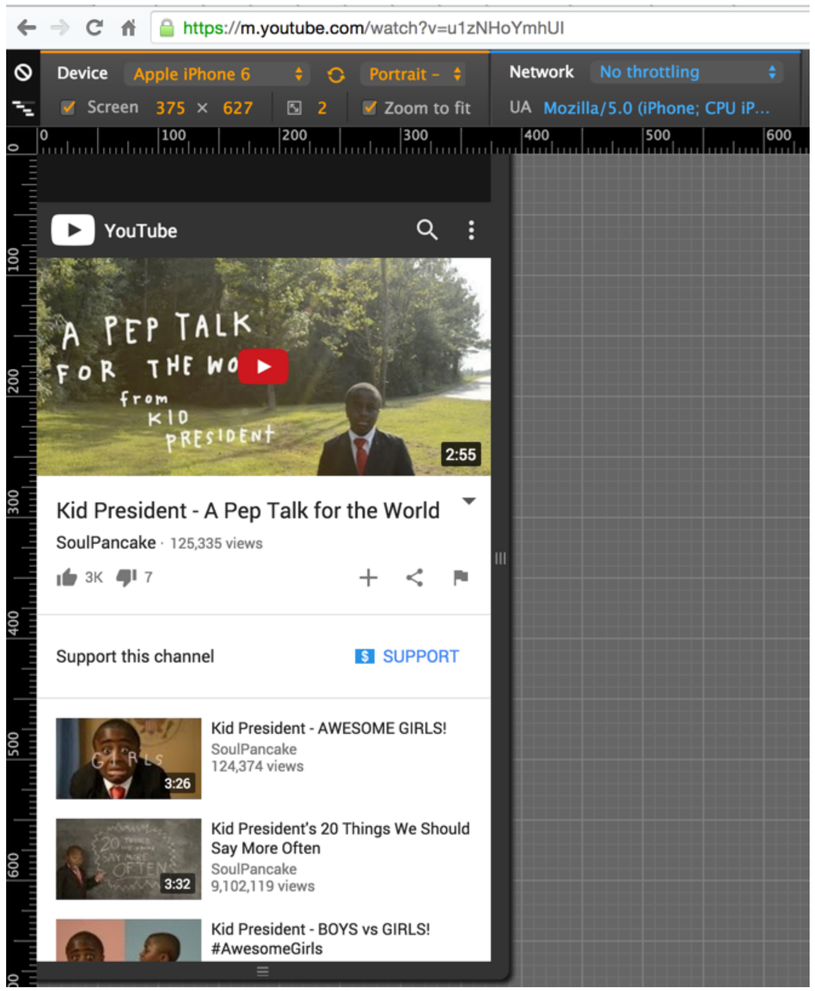
- Much more difficult to maintain separate codebases

- 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



Moving away from adaptive design, 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 (mobile), small (high-res mobile), medium (tablet), large (laptop or desktop), extra large (wide desktop or wall display)

Breakpoints

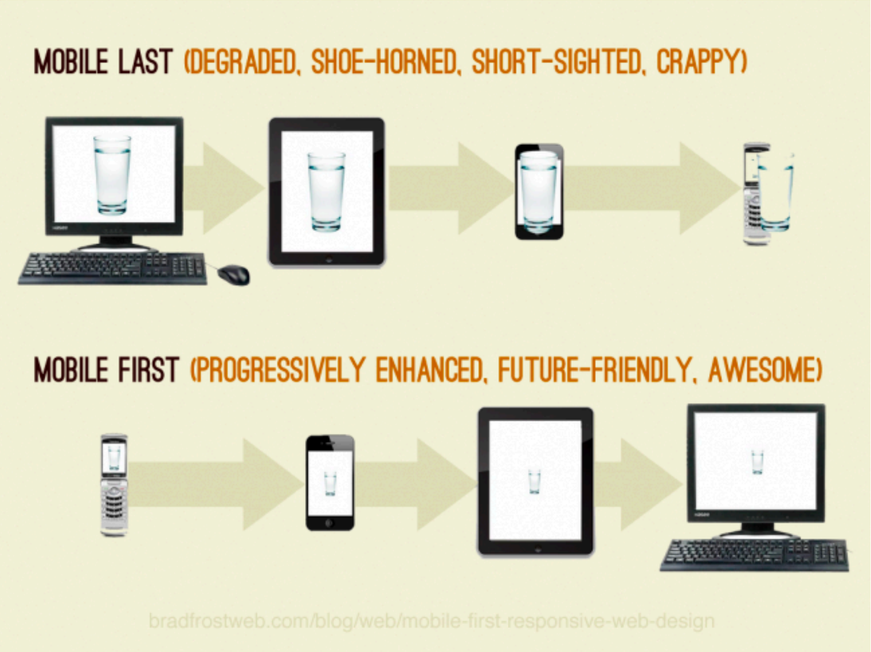
```
@media screen and (max-width: 640px) {
  /* small screens */
}

@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 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`

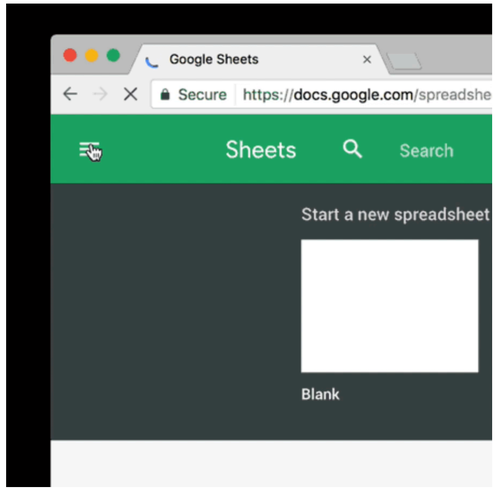
Mobile-first design

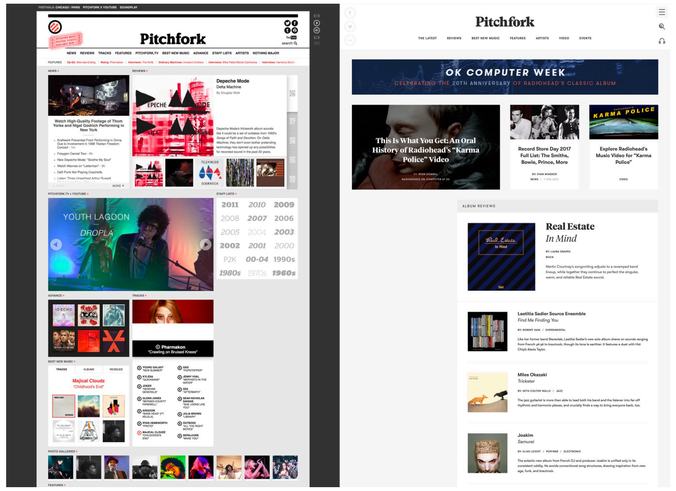
- “Graceful degradation” vs. “progressive enhancement”
- Plan your design for mobile
- Then make your app *better* with more real estate
  - Add more features
  - Make existing features easier to navigate



Mobile-first, not mobile-only

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





https://blog.prototypr.io/mobile-first-desktop-worst-f900909ae9e2

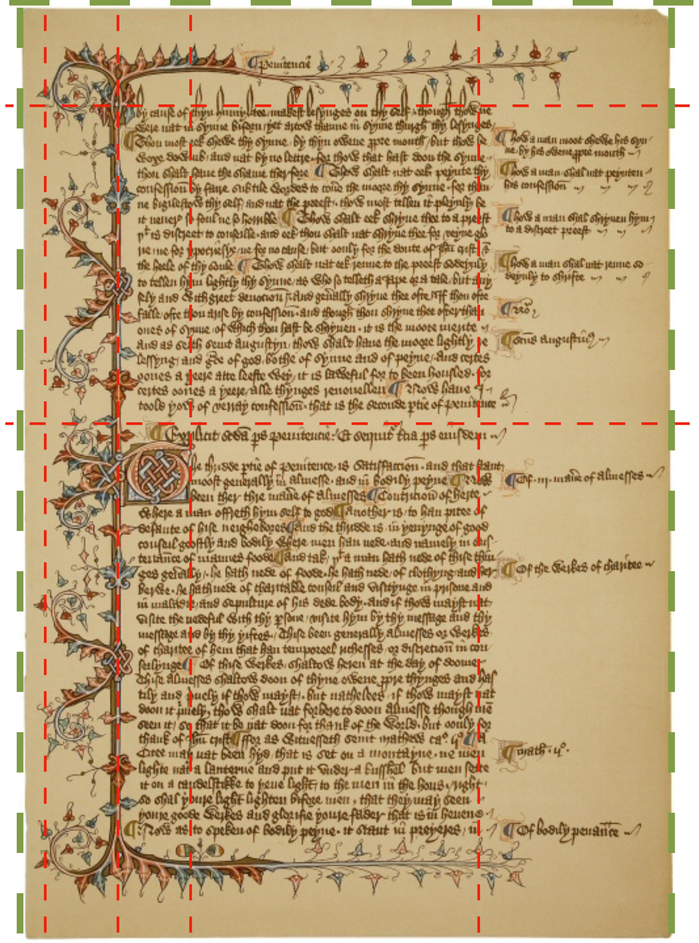
Mobile-first, not mobile-only

- 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

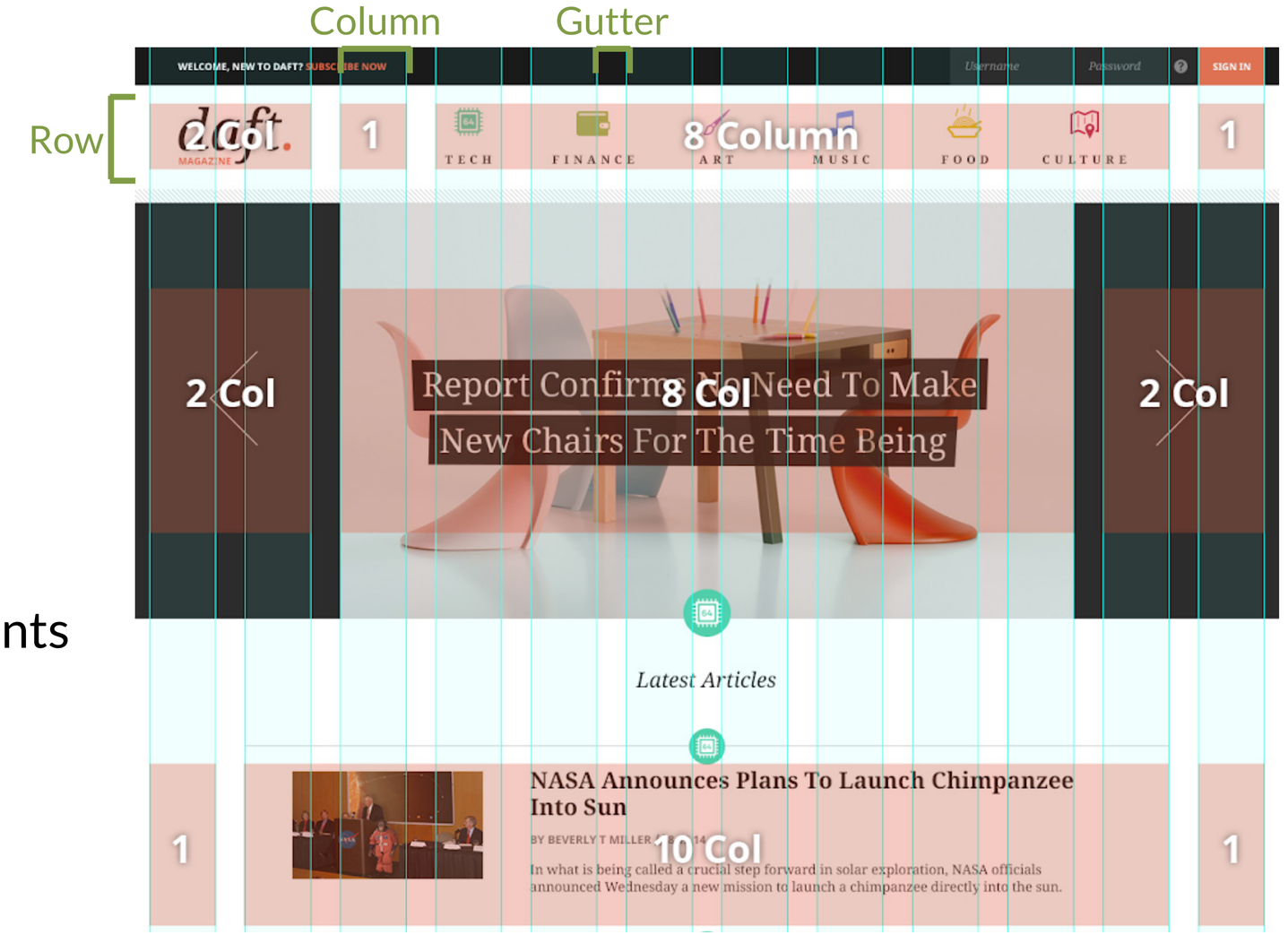
- Established tool for content arrangement
- Gridded content is familiar and easy to follow
- In general, it's good to target fewer lines
  - But breaking that rule is important for creativity and attention-grabbing



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

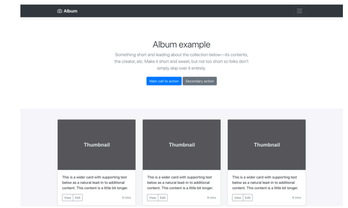
Grid-based layouts

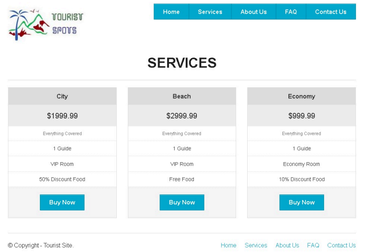
- Rows
- Columns
- Gutters
- Padding/spacing
  - Defined by specific elements



Grid-based frameworks

- Bootstrap (https://getbootstrap.com/)
  - Most popular (used by 20% of all websites!), most extensions
- Foundation (https://get.foundation/)
  - Includes icons, drag&drop editor
- Pure.css (https://pure-css.github.io/)
  - Small file size, 3.5KB





Digging into Bootstrap

Bootstrap

Specifying a viewport

- In page's `head`

- Sets device width and scale level (for zooming)

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

Bootstrap

Designating a container

- All bootstrap content lives in a container

```
<div class="container">
  <!--Bootstrap content-->
</div>
```

- Just a class; anything can be a container

```
<main class="container">
  <!--Bootstrap content-->
</main>
```

Bootstrap

Grid System

- Grid system has 12 columns
  - 12 has a lot of factors (1, 2, 3, 4, 6)
- Content over 12 columns will wrap
  - (3+6+4=13, the 4 will wrap)
- 15px gutter for each
- Classes for `row` and `col-[size]-[number]`

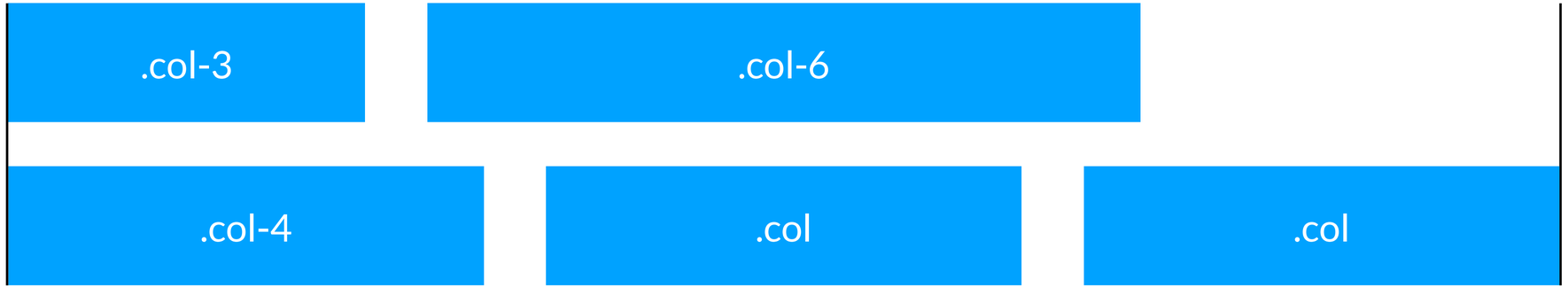
	Extra small <576px	**Small** ≥576px	**Medium** ≥768px	**Large** ≥992px	**Extra large** ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Bootstrap

Grid System

- Within the same row, content will wrap once it goes over 12 columns
    - Size parameter is optional; will divide space proportionally

```
<main class="container">
  <div class="row">
    <div class="col-3">A</div>
    <div class="col-6">B</div>
    <div class="col-4">C</div>
    <div class="col">D</div>
    <div class="col">E</div>
  </div>
</main>
```

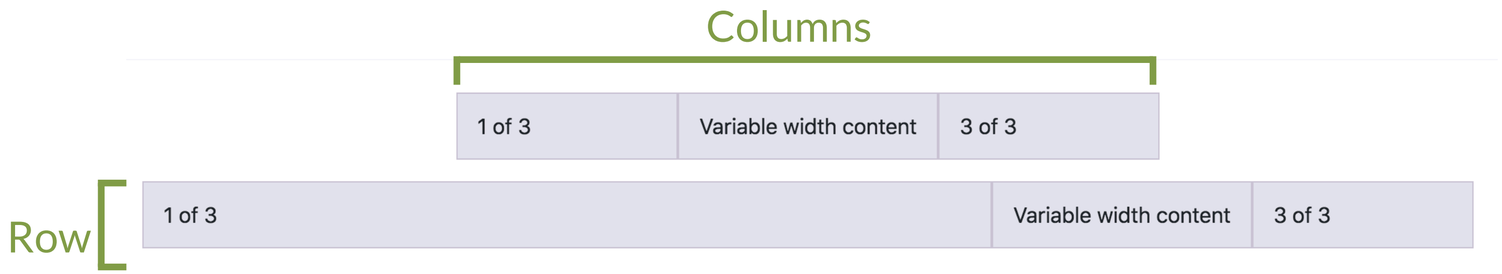


Bootstrap

Grid System

- Rows are block elements, while columns are inline

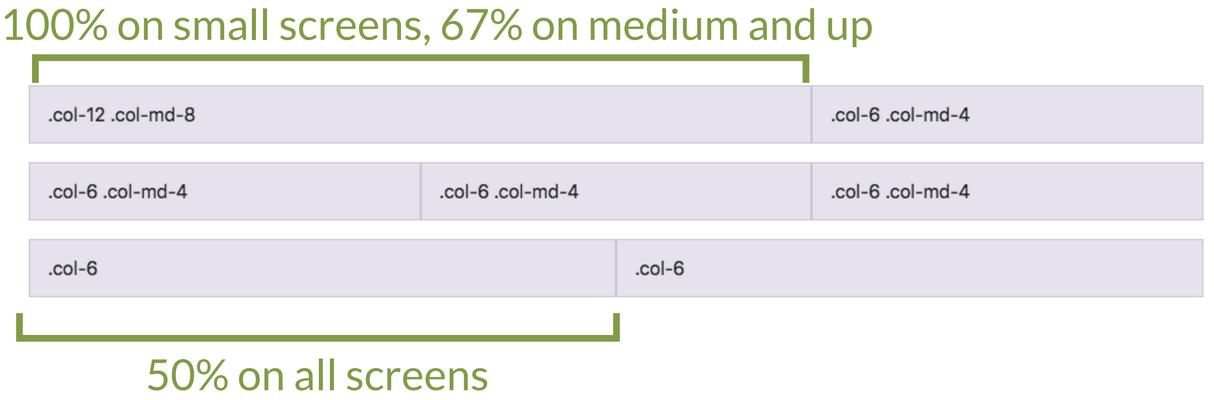
https://getbootstrap.com/docs/5.3/layout/grid/



Bootstrap

Grid System

- `.col` with no size defaults to the smallest (`xs`)
- The largest size listed will cover any larger sizes which are not-listed
- Will default to width 12 when no size is specified



Bootstrap

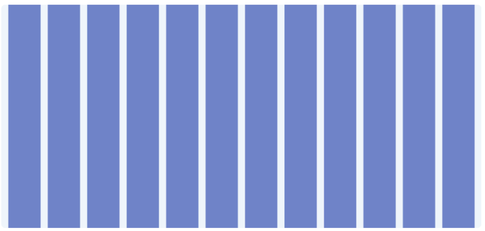
Grid System

- Display property supports applying a display type (like `none` or `block`), can be combined with size properties
- `.d-none` will hide on all sizes
- `.d-md-none` will hide everything `md` and larger
- `.d-md-block.d-none` will hide only on `sm`, `xs`

Grid systems are very common in web design

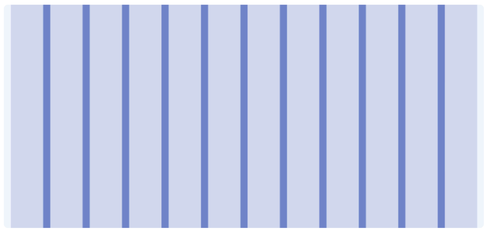
Spectrum grid system

- Adobe spectrum has a grid system which functions similarly
- Installation and use is harder, so we will focus on Bootstrap here



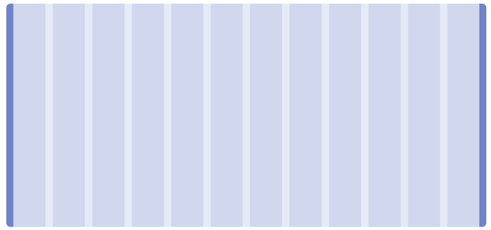
Columns

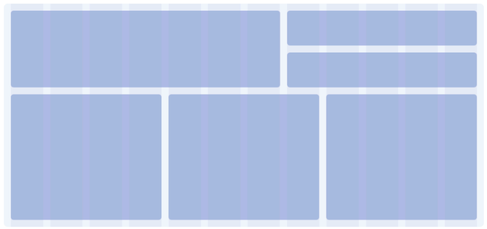
There are 12 columns in the responsive grid system. Column widths change with the size of the grid.



Gutters

Gutters are the gaps between the columns. Gutter widths are fixed values (16 px, 24 px, etc.) based on breakpoints.





Breakpoints and dimensions

Minimum width	Breakpoint	Number of columns	Gutters	Fluid grid width	Fixed grid width
Below 304 px	XXS	12	16 px	304 px, horizontal scroll	304 px, horizontal scroll
304 px	XS	12	16 px	100%	100%
768 px	S	12	24 px	100%	100%
1280 px	M	12	32 px	100%	1280 px, centered
1768 px	L	12	40 px	100%	1280 px, centered
2160 px and above	XL	12	48 px	100%	1280 px, centered

https://spectrum.adobe.com/page/responsive-grid/

Today’s goals

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

- Explain why we need some form of responsivity
- Define and differentiate responsive and adaptive design
- Implement a grid-based system with Bootstrap

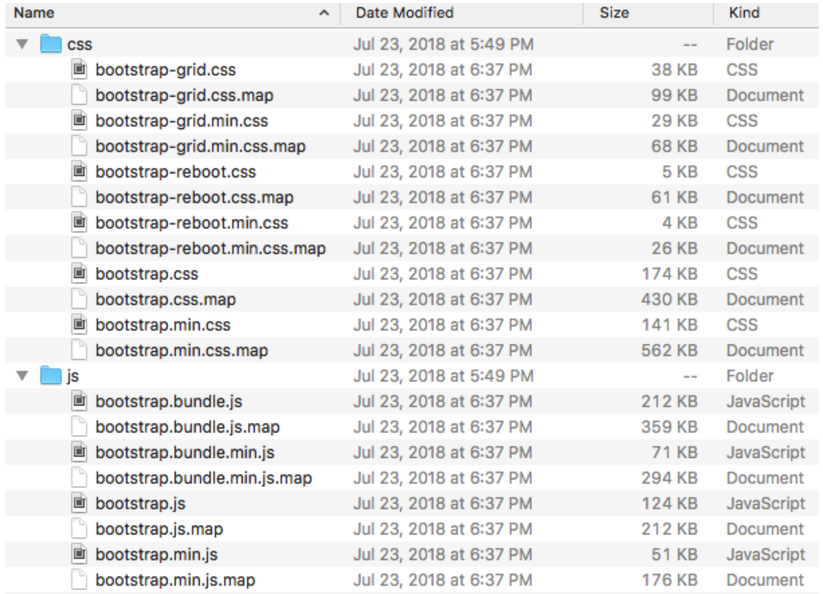
IN4MATX 285:
Interactive Technology Studio

Practice: Responsive and Adaptive Design

Getting Bootstrap

Bootstrap

- Direct download
  - https://getbootstrap.com/docs/5.3/getting-started/download/
- CSS and JavaScript files
- Minified files are compressed, will load faster
- .map files support editing preprocessed files
  - We won't really touch on those in this class
- We'll use bootstrap.min.css for now



Name	Date Modified	Size	Kind
css	Jul 23, 2018 at 5:49 PM	--	Folder
bootstrap-grid.css	Jul 23, 2018 at 6:37 PM	38 KB	CSS
bootstrap-grid.css.map	Jul 23, 2018 at 6:37 PM	99 KB	Document
bootstrap-grid.min.css	Jul 23, 2018 at 6:37 PM	29 KB	CSS
bootstrap-grid.min.css.map	Jul 23, 2018 at 6:37 PM	68 KB	Document
bootstrap-reboot.css	Jul 23, 2018 at 6:37 PM	5 KB	CSS
bootstrap-reboot.css.map	Jul 23, 2018 at 6:37 PM	61 KB	Document
bootstrap-reboot.min.css	Jul 23, 2018 at 6:37 PM	4 KB	CSS
bootstrap-reboot.min.css.map	Jul 23, 2018 at 6:37 PM	26 KB	Document
bootstrap.css	Jul 23, 2018 at 6:37 PM	174 KB	CSS
bootstrap.css.map	Jul 23, 2018 at 6:37 PM	430 KB	Document
bootstrap.min.css	Jul 23, 2018 at 6:37 PM	141 KB	CSS
bootstrap.min.css.map	Jul 23, 2018 at 6:37 PM	562 KB	Document
js	Jul 23, 2018 at 5:49 PM	--	Folder
bootstrap.bundle.js	Jul 23, 2018 at 6:37 PM	212 KB	JavaScript
bootstrap.bundle.js.map	Jul 23, 2018 at 6:37 PM	359 KB	Document
bootstrap.bundle.min.js	Jul 23, 2018 at 6:37 PM	71 KB	JavaScript
bootstrap.bundle.min.js.map	Jul 23, 2018 at 6:37 PM	294 KB	Document
bootstrap.js	Jul 23, 2018 at 6:37 PM	124 KB	JavaScript
bootstrap.js.map	Jul 23, 2018 at 6:37 PM	212 KB	Document
bootstrap.min.js	Jul 23, 2018 at 6:37 PM	51 KB	JavaScript
bootstrap.min.js.map	Jul 23, 2018 at 6:37 PM	176 KB	Document

Bootstrap

- Content Delivery Networks (CDN)
- Browser-side caching reduces burdens of loading files
- Integrity: hashes to ensure the downloaded file matches what's expected
  - Protects against server being compromised
- Crossorigin: some imports require credentials, anonymous requires none

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/
bootstrap.min.css" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
```

Bootstrap

- Load bootstrap

```
<link rel="stylesheet" href="css/bootstrap.min.css">

<link rel="stylesheet" href=“css/override.css">
```