# IN4MATX 133: User Interface Software

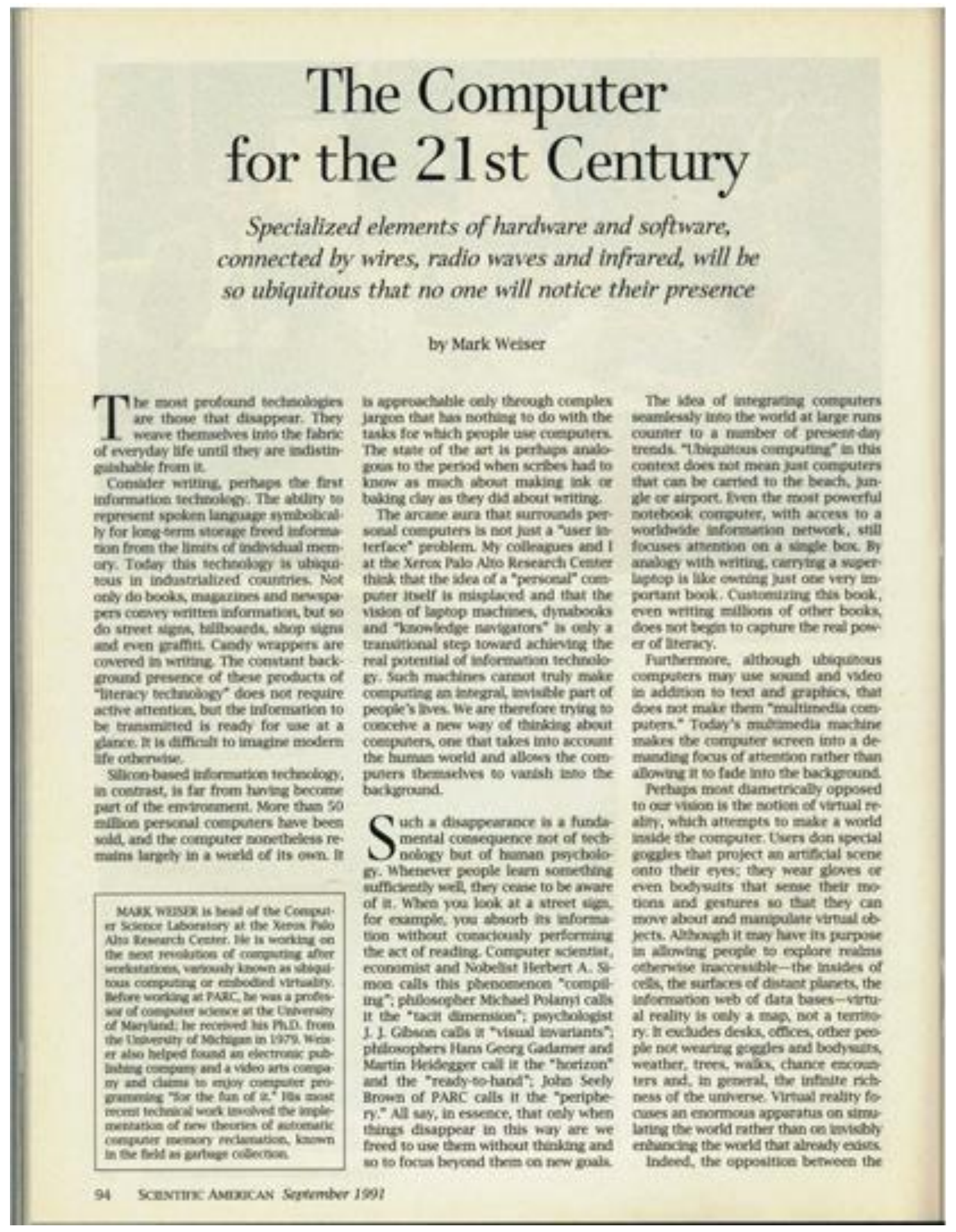
Lecture 4:
Responsive Design & Bootstrap

## Today’s goals

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

- 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
- Utilize media queries to create responsive layouts
- Develop grid-based layouts using Bootstrap

# Recall the three waves of computing…



The Computer for the 21st Century

*Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence*

by Mark Weiser

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

Consider writing, perhaps the first information technology. The ability to represent spoken language symbolically for long-term storage freed information from the limits of individual memory. Today this technology is ubiquitous in industrialized countries. Not only do books, magazines and newspapers convey written information, but so do street signs, billboards, shop signs and even graffiti. Candy wrappers are covered in writing. The constant background presence of these products of "literacy technology" does not require active attention, but the information to be transmitted is ready for use at a glance. It is difficult to imagine modern life otherwise.

Silicon-based information technology, in contrast, is far from having become part of the environment. More than 50 million personal computers have been sold, and the computer nonetheless remains largely in a world of its own. It is approachable only through complex jargon that has nothing to do with the tasks for which people use computers. The state of the art is perhaps analogous to the period when scribes had to know as much about making ink or baking clay as they did about writing.

The arcane aura that surrounds personal computers is not just a "user interface" problem. My colleagues and I at the Xerox Palo Alto Research Center think that the idea of a "personal" computer itself is misplaced and that the vision of laptop machines, dynabooks and "knowledge navigators" is only a transitional step toward achieving the real potential of information technology. Such machines cannot truly make computing an integral, invisible part of people's lives. We are therefore trying to conceive a new way of thinking about computers, one that takes into account the human world and allows the computers themselves to vanish into the background.

Such a disappearance is a fundamental consequence not of technology but of human psychology. Whenever people learn something sufficiently well, they cease to be aware of it. When you look at a street sign, for example, you absorb its information without consciously performing the act of reading. Computer scientist, economist and Nobelist Herbert A. Simon calls this phenomenon "compiling"; philosopher Michael Polanyi calls it the "tacit dimension"; psychologist J. J. Gibson calls it "visual invariants"; philosophers Hans Georg Gadamer and Martin Heidegger call it the "horizon" and the "ready-to-hand"; John Seely Brown of PARC calls it the "periphery." All say, in essence, that only when things disappear in this way are we freed to use them without thinking and so to focus beyond them on new goals.

The idea of integrating computers seamlessly into the world at large runs counter to a number of present-day trends. "Ubiquitous computing" in this context does not mean just computers that can be carried to the beach, jungle or airport. Even the most powerful notebook computer, with access to a worldwide information network, still focuses attention on a single box. By analogy with writing, carrying a super-laptop is like owning just one very important book. Customizing this book, even writing millions of other books, does not begin to capture the real power of literacy.

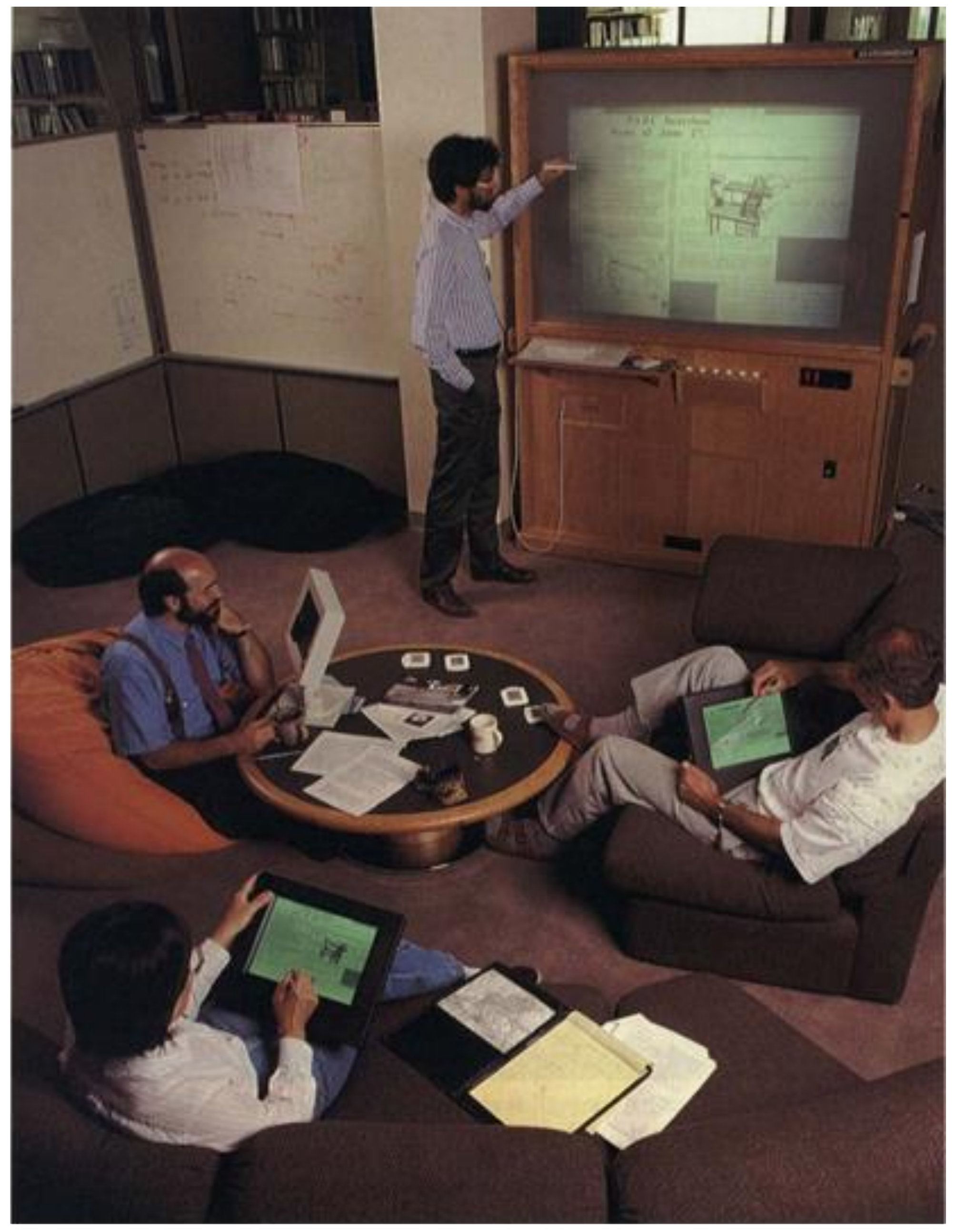
Furthermore, although ubiquitous computers may use sound and video in addition to text and graphics, that does not make them "multimedia computers." Today's multimedia machine makes the computer screen into a demanding focus of attention rather than allowing it to fade into the background.

Perhaps most diametrically opposed to our vision is the notion of virtual reality, which attempts to make a world inside the computer. Users don special goggles that project an artificial scene onto their eyes; they wear gloves or even bodysuits that sense their motions and gestures so that they can move about and manipulate virtual objects. Although it may have its purpose in allowing people to explore realms otherwise inaccessible—the insides of cells, the surfaces of distant planets, the information web of data bases—virtual reality is only a map, not a territory. It excludes desks, offices, other people not wearing goggles and bodysuits, weather, trees, walks, chance encounters and, in general, the infinite richness of the universe. Virtual reality focuses an enormous apparatus on simulating the world rather than on invisibly enhancing the world that already exists.

Indeed, the opposition between the

MARK WEISER is head of the Computer Science Laboratory at the Xerox Palo Alto Research Center. He is working on the next revolution of computing after workstations, variously known as ubiquitous computing or embodied virtuality. Before working at PARC, he was a professor of computer science at the University of Maryland; he received his Ph.D. from the University of Michigan in 1979. Weiser also helped found an electronic publishing company and a video arts company and claims to enjoy computer programming "for the fun of it." His most recent technical work involved the implementation of new theories of automatic computer memory reclamation, known in the field as garbage collection.

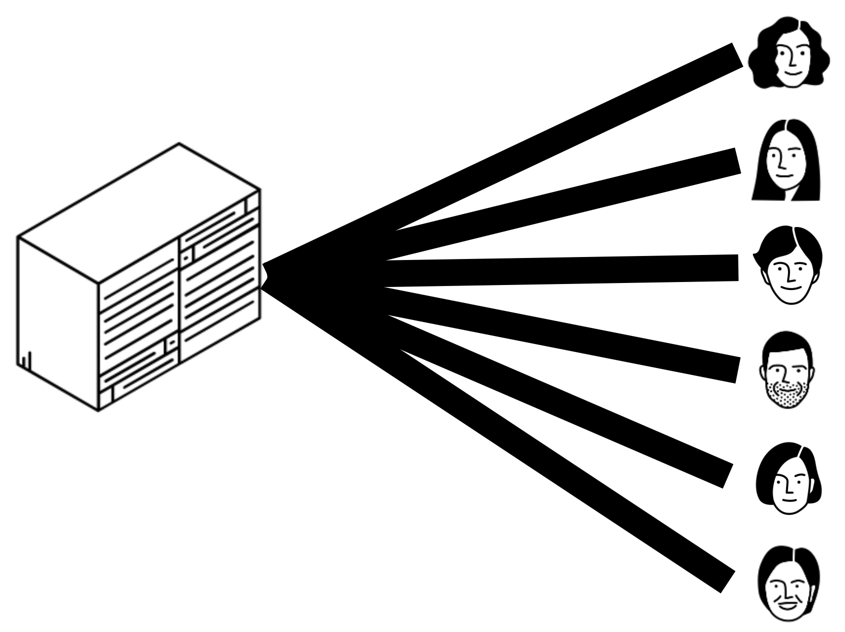
94 SCIENTIFIC AMERICAN *September 1991*



## Three waves of computing

1

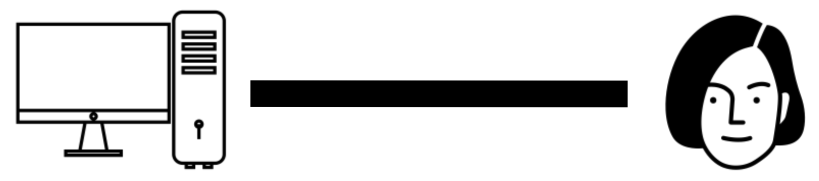
Mainframe computing



"Many to one"

2

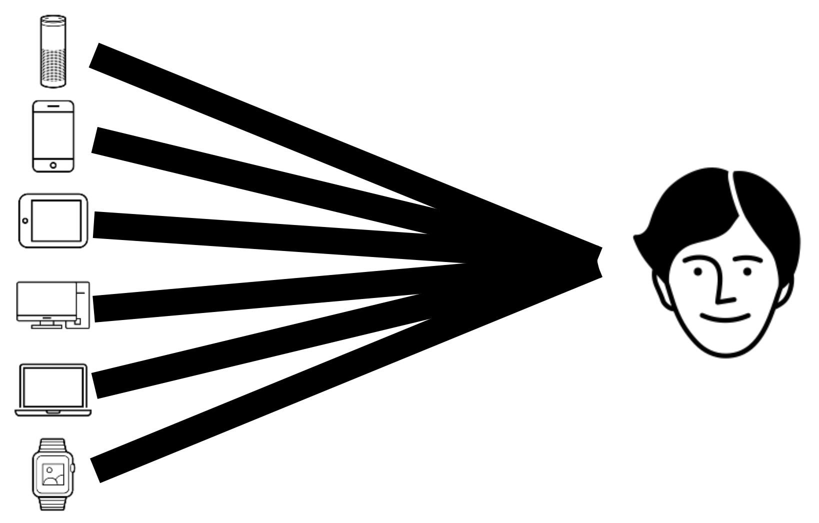
Personal computing



"One to one"

3

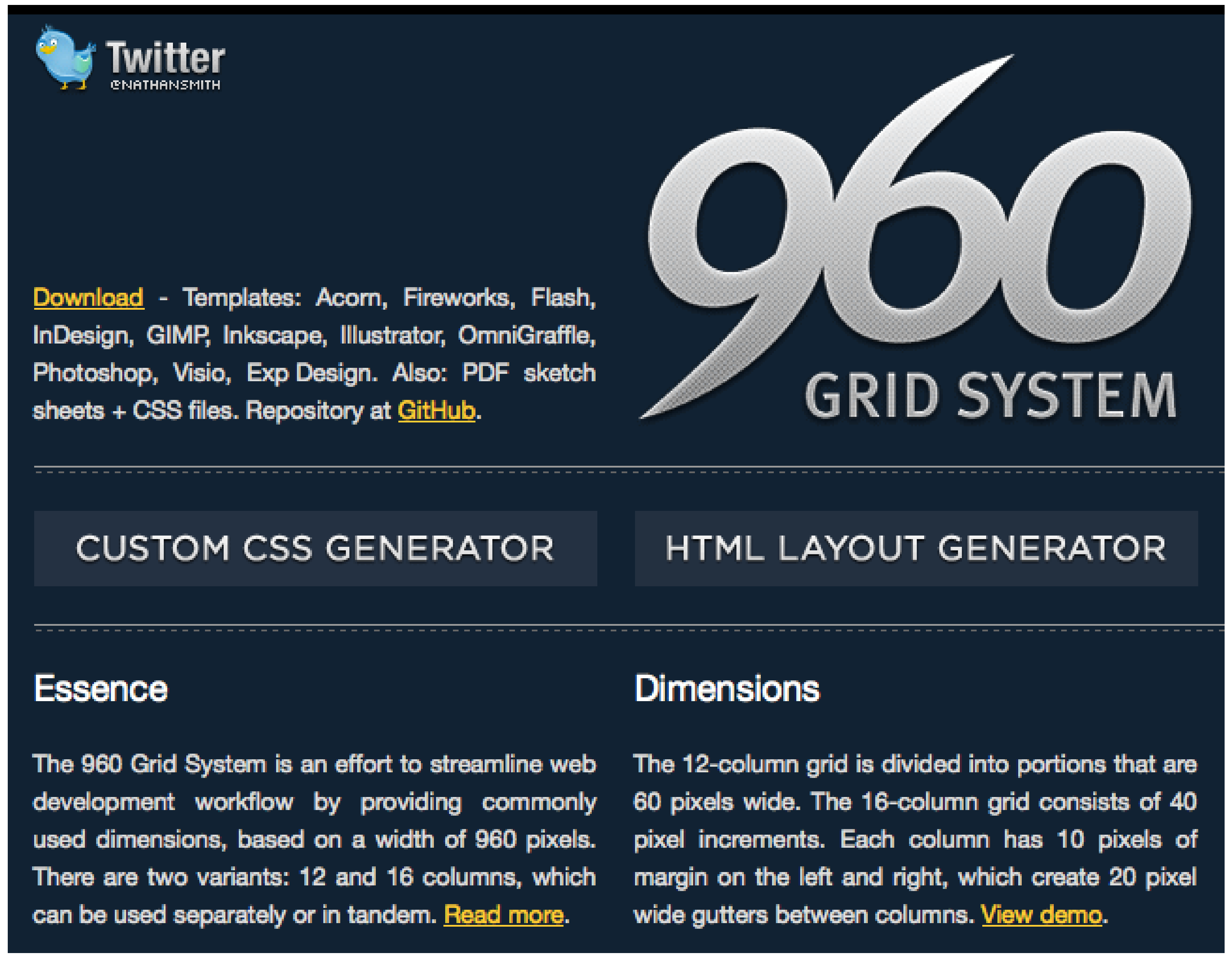
Ubiquitous computing



"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



Download - Templates: Acorn, Fireworks, Flash, InDesign, GIMP, Inkscape, Illustrator, OmniGraffle, Photoshop, Visio, Exp Design. Also: PDF sketch sheets + CSS files. Repository at GitHub.

**Essence**

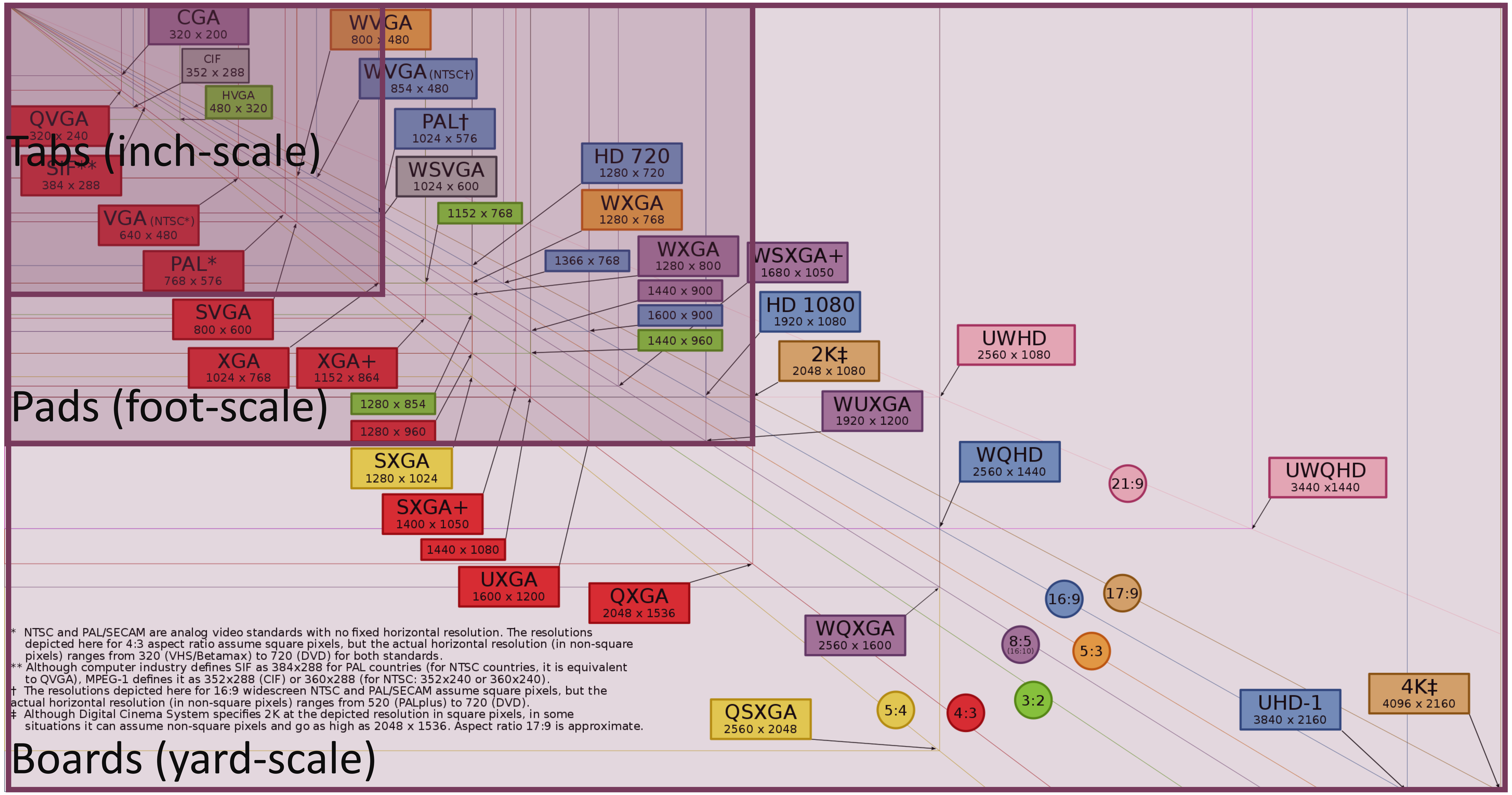
The 960 Grid System is an effort to streamline web development workflow by providing commonly used dimensions, based on a width of 960 pixels. There are two variants: 12 and 16 columns, which can be used separately or in tandem. Read more.

**Dimensions**

The 12-column grid is divided into portions that are 60 pixels wide. The 16-column grid consists of 40 pixel increments. Each column has 10 pixels of margin on the left and right, which create 20 pixel wide gutters between columns. View demo.

https://960.gs/

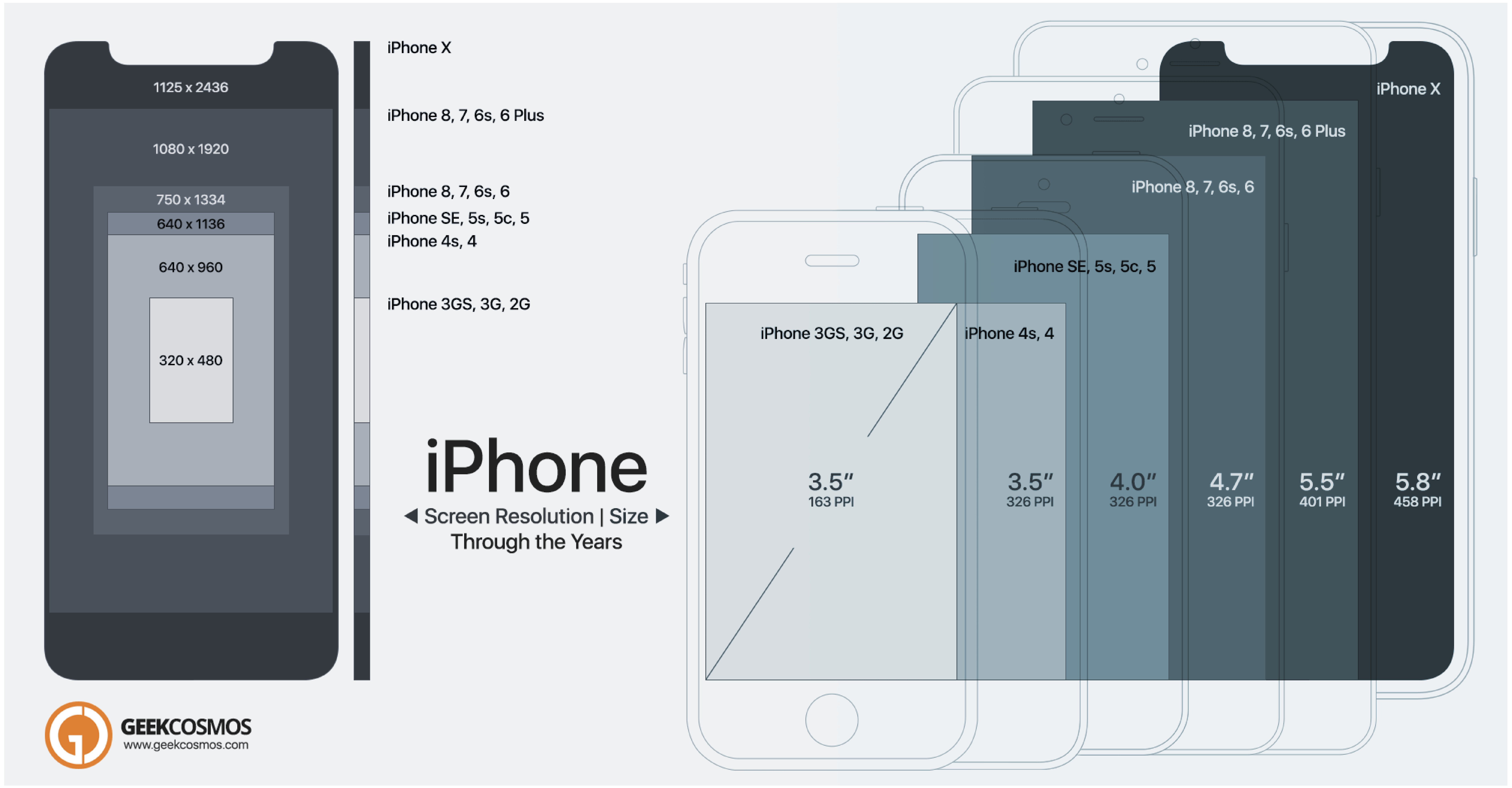
## Websites today: ubiquitous computing



* NTSC and PAL/SECAM are analog video standards with no fixed horizontal resolution. The resolutions depicted here for 4:3 aspect ratio assume square pixels, but the actual horizontal resolution (in non-square pixels) ranges from 320 (VHS/Betamax) to 720 (DVD) for both standards.
** Although computer industry defines SIF as 384x288 for PAL countries (for NTSC countries, it is equivalent to QVGA), MPEG-1 defines it as 352x288 (CIF) or 360x288 (for NTSC: 352x240 or 360x240).
† The resolutions depicted here for 16:9 widescreen NTSC and PAL/SECAM assume square pixels, but the actual horizontal resolution (in non-square pixels) ranges from 520 (PALplus) to 720 (DVD).
‡ Although Digital Cinema System specifies 2K at the depicted resolution in square pixels, in some situations it can assume non-square pixels and go as high as 2048 x 1536. Aspect ratio 17:9 is approximate.

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

https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/

## 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

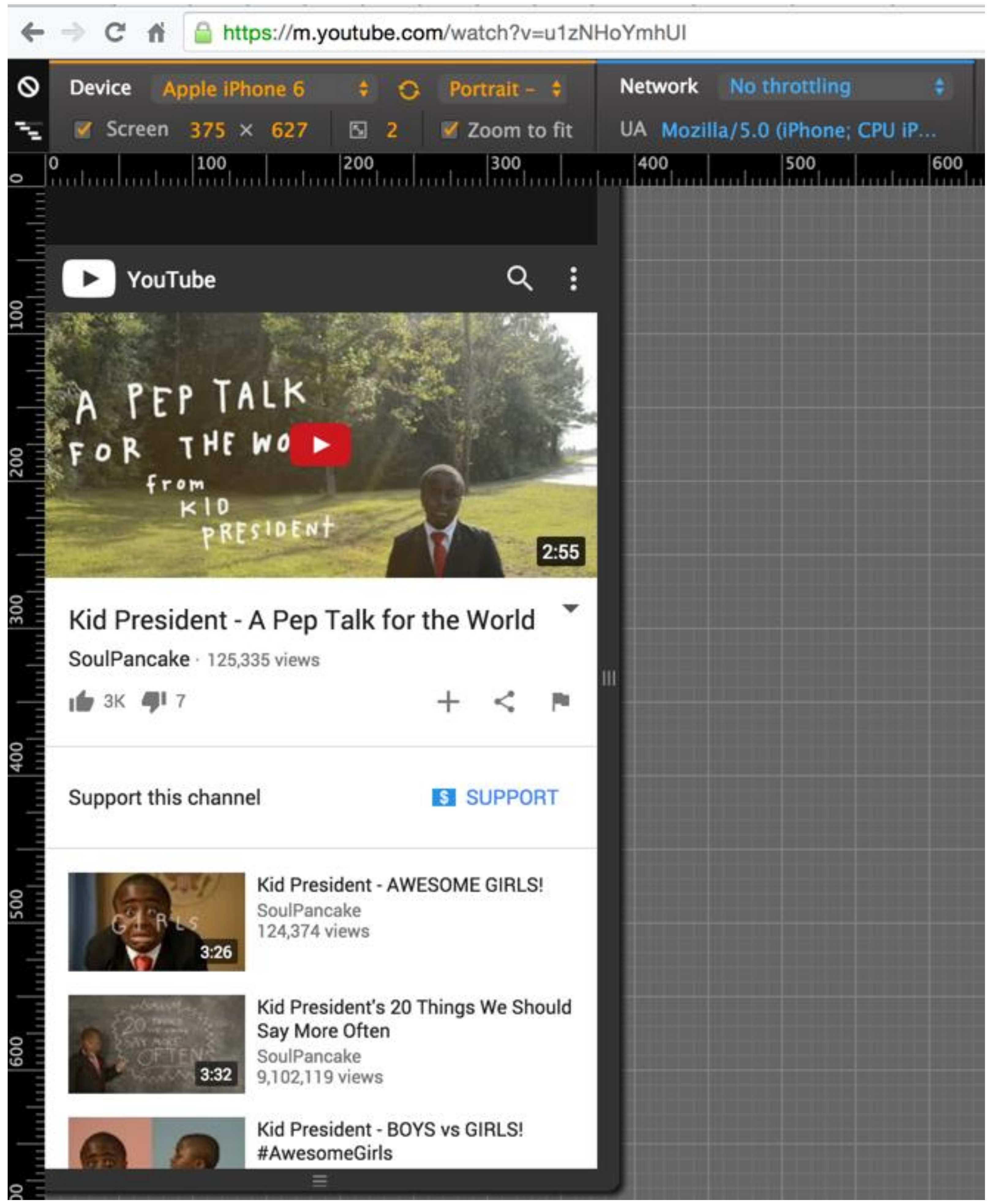
**-** Much more difficult to maintain separate codebases

**-** 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

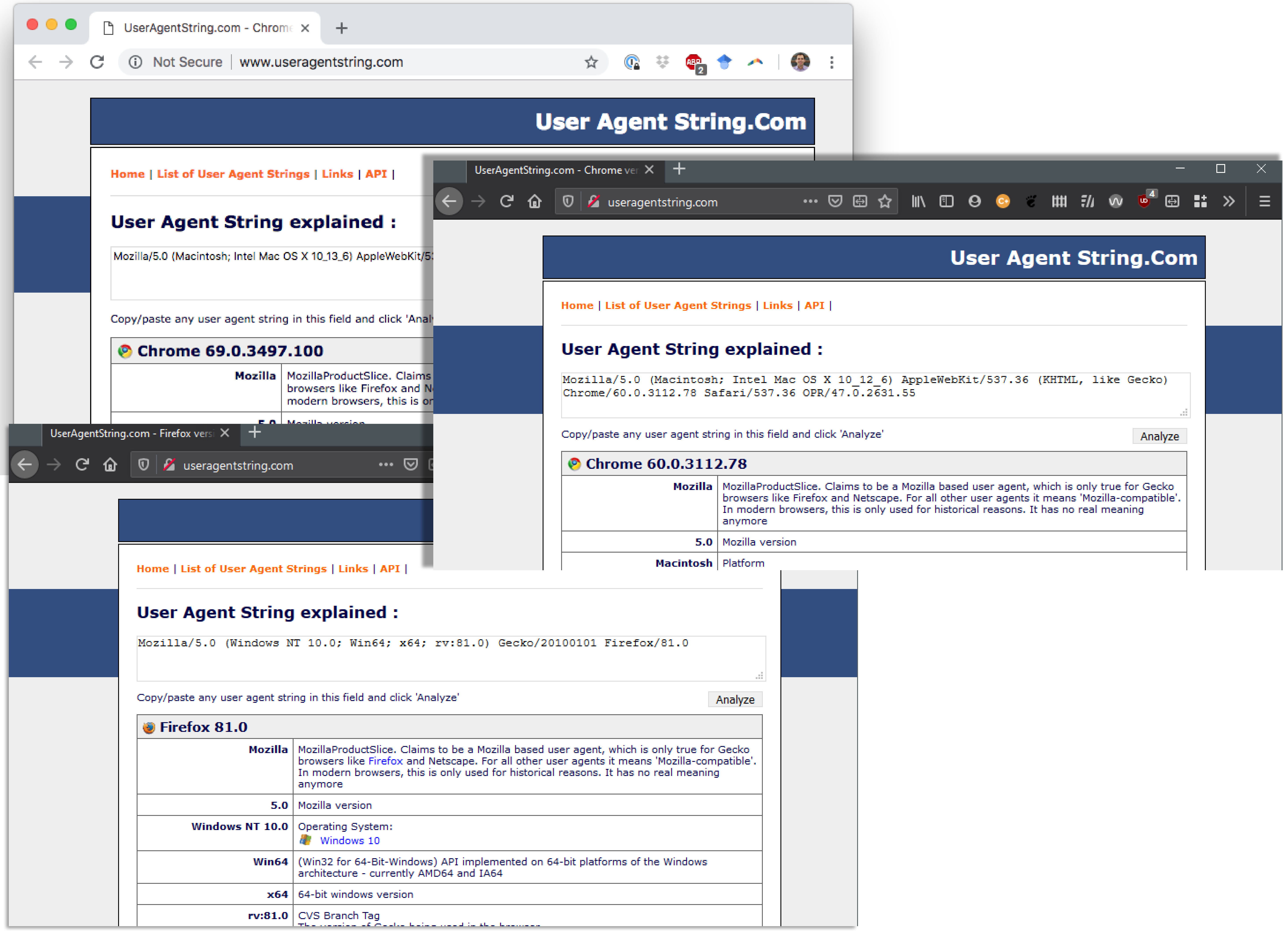
# Adaptive design

- 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


https://m.youtube.com/watch?v=u1zNHoYmhUI
Device Apple iPhone 6
Portrait
Network No throttling
Screen 375 × 627
2
Zoom to fit
UA Mozilla/5.0 (iPhone; CPU iP...
YouTube
A PEP TALK FOR THE WO
from KID PRESIDENT
2:55
Kid President - A Pep Talk for the World
SoulPancake · 125,335 views
3K
7
Support this channel
SUPPORT
Kid President - AWESOME GIRLS!
SoulPancake
124,374 views
3:26
Kid President's 20 Things We Should Say More Often
SoulPancake
9,102,119 views
3:32
Kid President - BOYS vs GIRLS! #AwesomeGirls

# 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



UserAgentString.com - Chrome
Not Secure | www.useragentstring.com

**User Agent String.Com**

Home | List of User Agent Strings | Links | API |

**User Agent String explained :**

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/5
```

Copy/paste any user agent string in this field and click 'Anal

**Chrome 69.0.3497.100**

| | |
|---|---|
| **Mozilla** | MozillaProductSlice. Claims browsers like Firefox and N modern browsers, this is o |
| **5.0** | Mozilla version |

UserAgentString.com - Chrome ve
useragentstring.com

**User Agent String.Com**

Home | List of User Agent Strings | Links | API |

**User Agent String explained :**

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/60.0.3112.78 Safari/537.36 OPR/47.0.2631.55
```

Copy/paste any user agent string in this field and click 'Analyze' Analyze

**Chrome 60.0.3112.78**

| | |
|---|---|
| **Mozilla** | MozillaProductSlice. Claims to be a Mozilla based user agent, which is only true for Gecko browsers like Firefox and Netscape. For all other user agents it means 'Mozilla-compatible'. In modern browsers, this is only used for historical reasons. It has no real meaning anymore |
| **5.0** | Mozilla version |
| **Macintosh** | Platform |

UserAgentString.com - Firefox vers
useragentstring.com

Home | List of User Agent Strings | Links | API |

**User Agent String explained :**

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
```

Copy/paste any user agent string in this field and click 'Analyze' Analyze

**Firefox 81.0**

| | |
|---|---|
| **Mozilla** | MozillaProductSlice. Claims to be a Mozilla based user agent, which is only true for Gecko browsers like Firefox and Netscape. For all other user agents it means 'Mozilla-compatible'. In modern browsers, this is only used for historical reasons. It has no real meaning anymore |
| **5.0** | Mozilla version |
| **Windows NT 10.0** | Operating System: Windows 10 |
| **Win64** | (Win32 for 64-Bit-Windows) API implemented on 64-bit platforms of the Windows architecture - currently AMD64 and IA64 |
| **x64** | 64-bit windows version |
| **rv:81.0** | CVS Branch Tag |

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

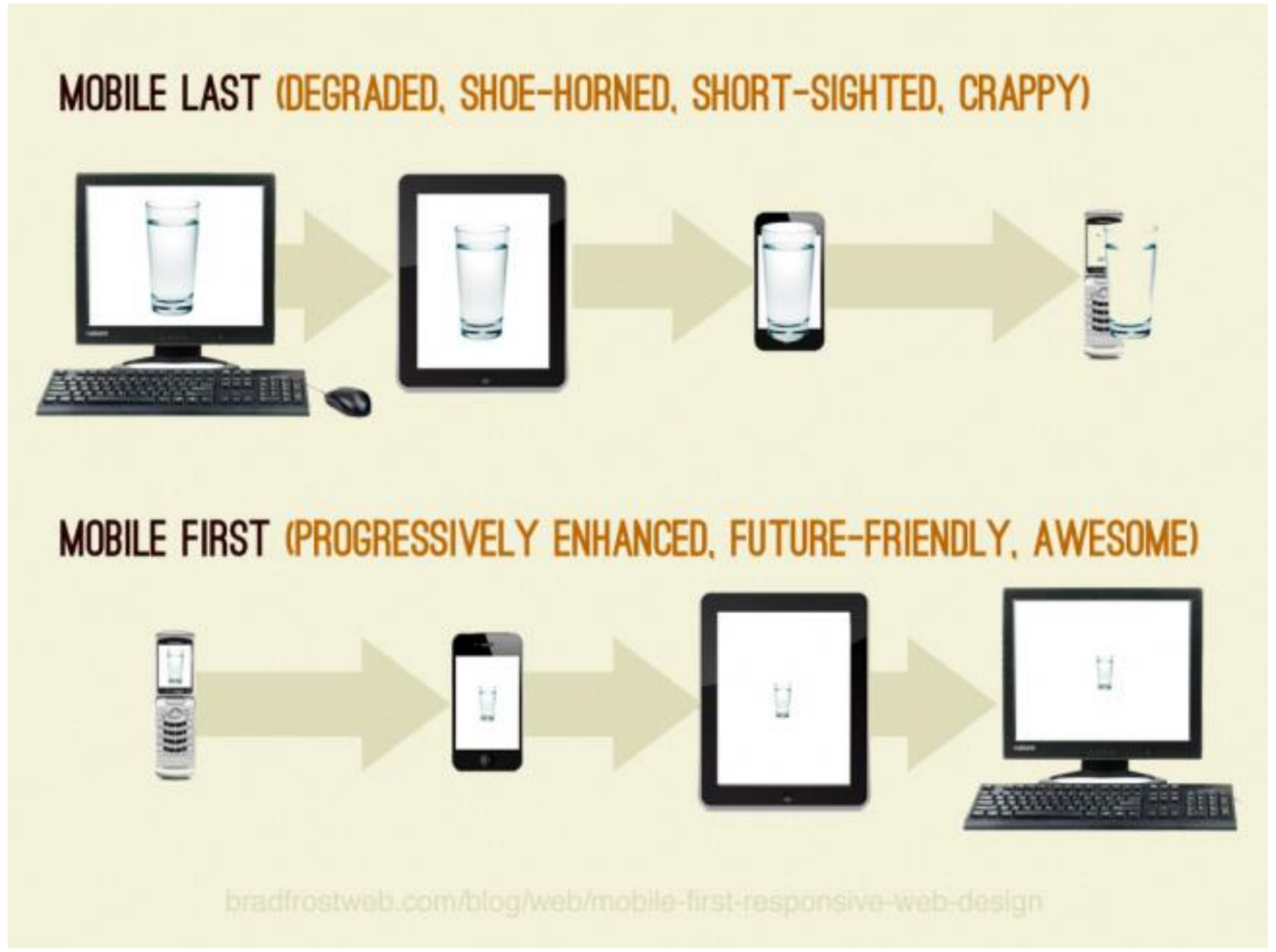
```
@media screen and (max-width: 640px) {
  /* small screens */
}

@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`

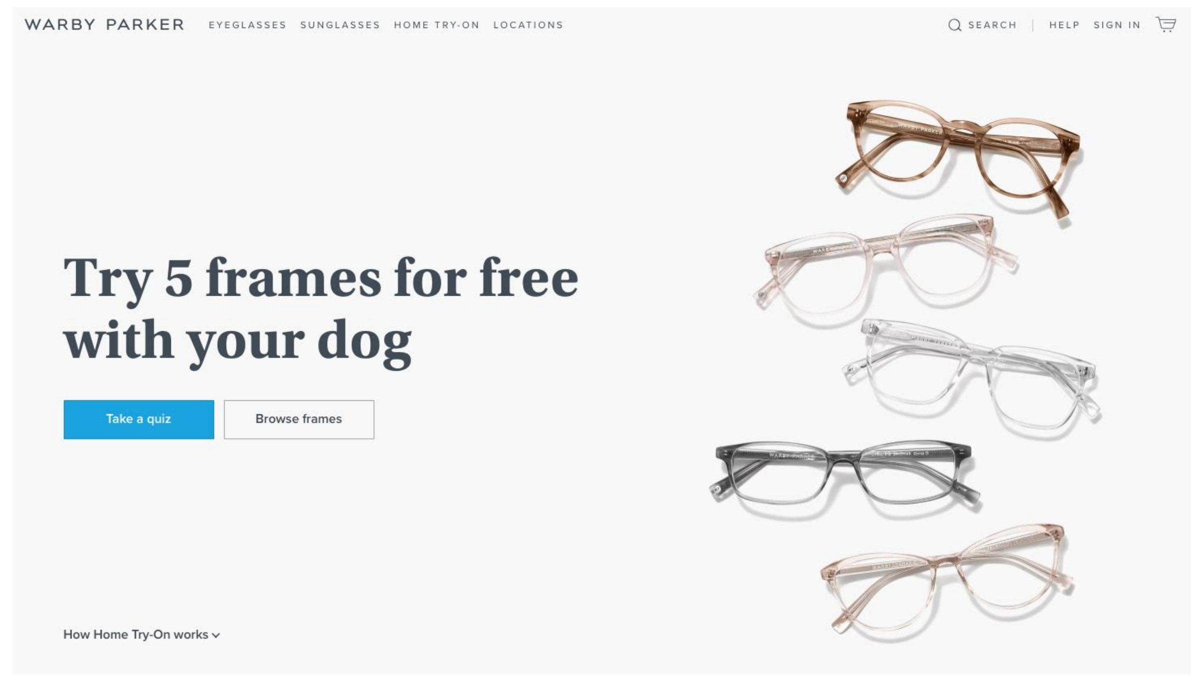
# Mobile-first design

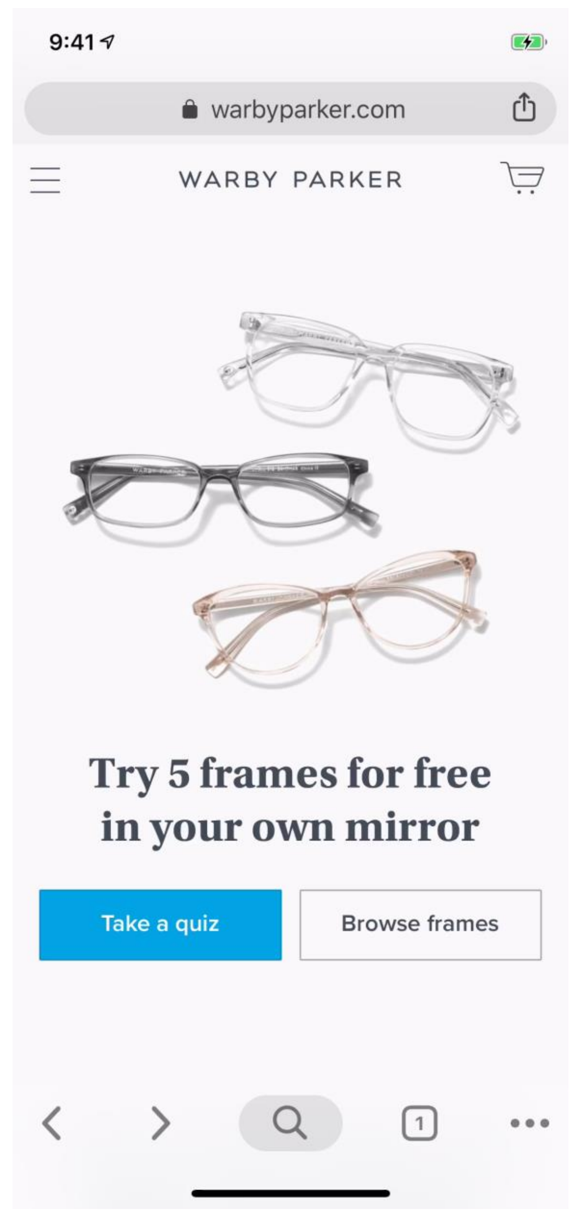
- “Graceful degradation” vs. “progressive enhancement”
- Plan your design for mobile
- Then make your app *better* with more real estate
  - Add more features
  - Make existing features easier to navigate


MOBILE LAST (DEGRADED, SHOE-HORNED, SHORT-SIGHTED, CRAPPY)
MOBILE FIRST (PROGRESSIVELY ENHANCED, FUTURE-FRIENDLY, AWESOME)
bradfrostweb.com/blog/web/mobile-first-responsive-web-design

# A few tips for mobile design

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

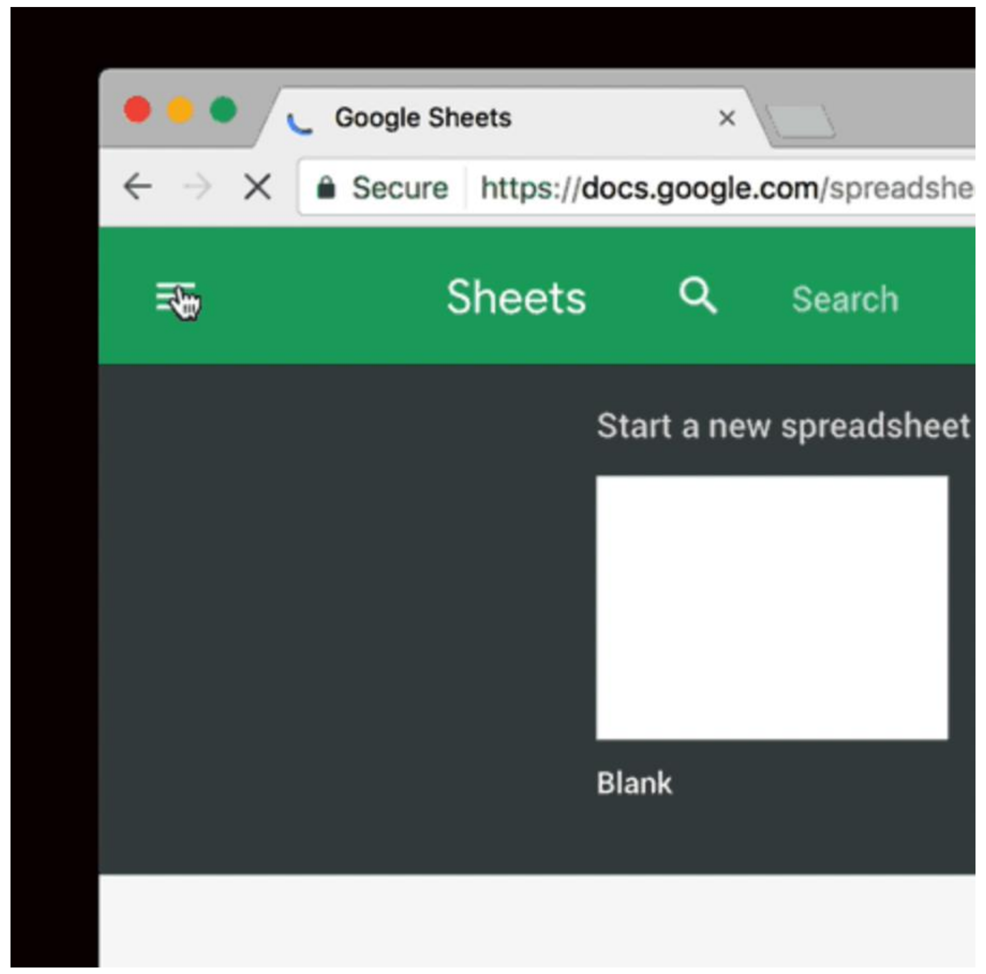


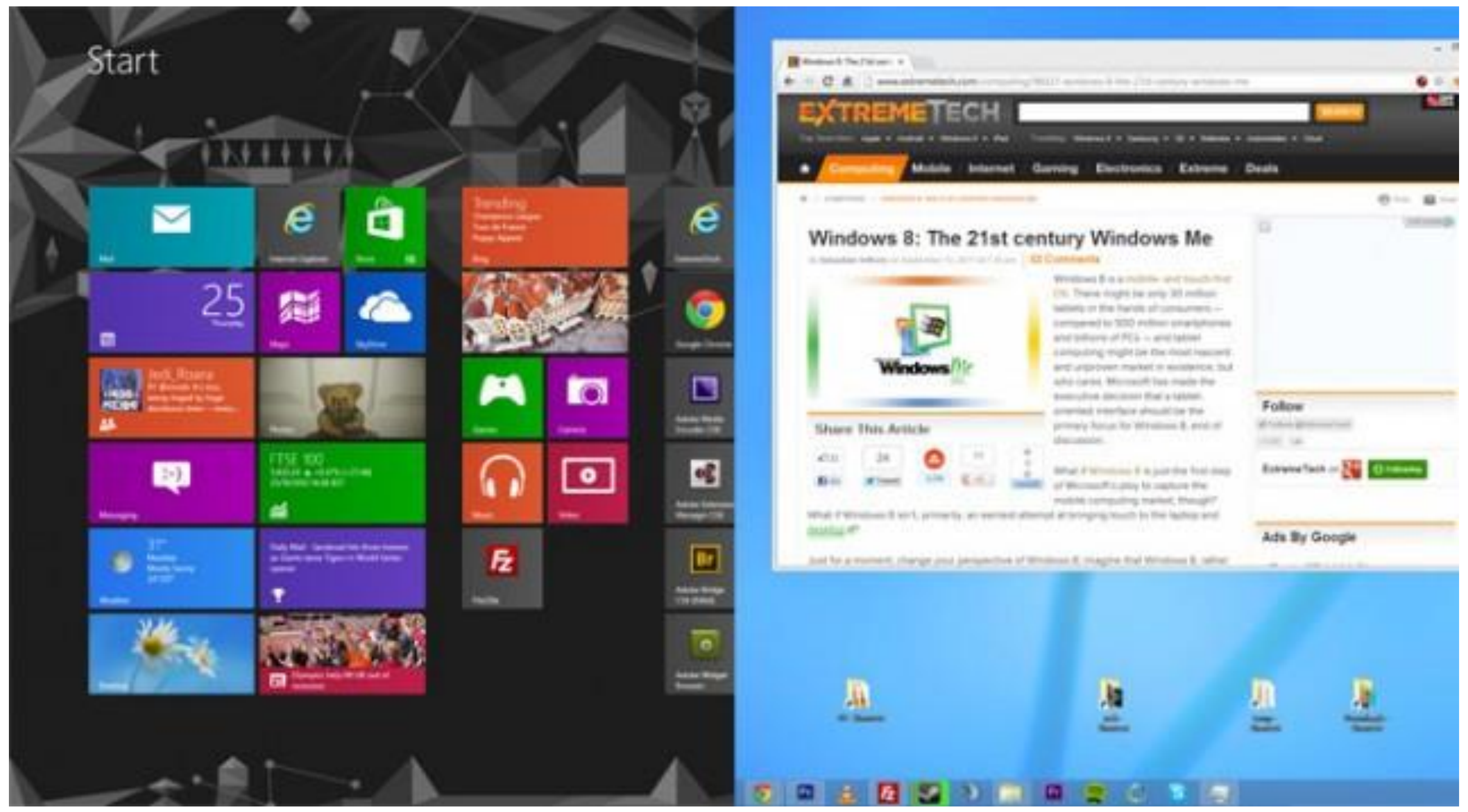


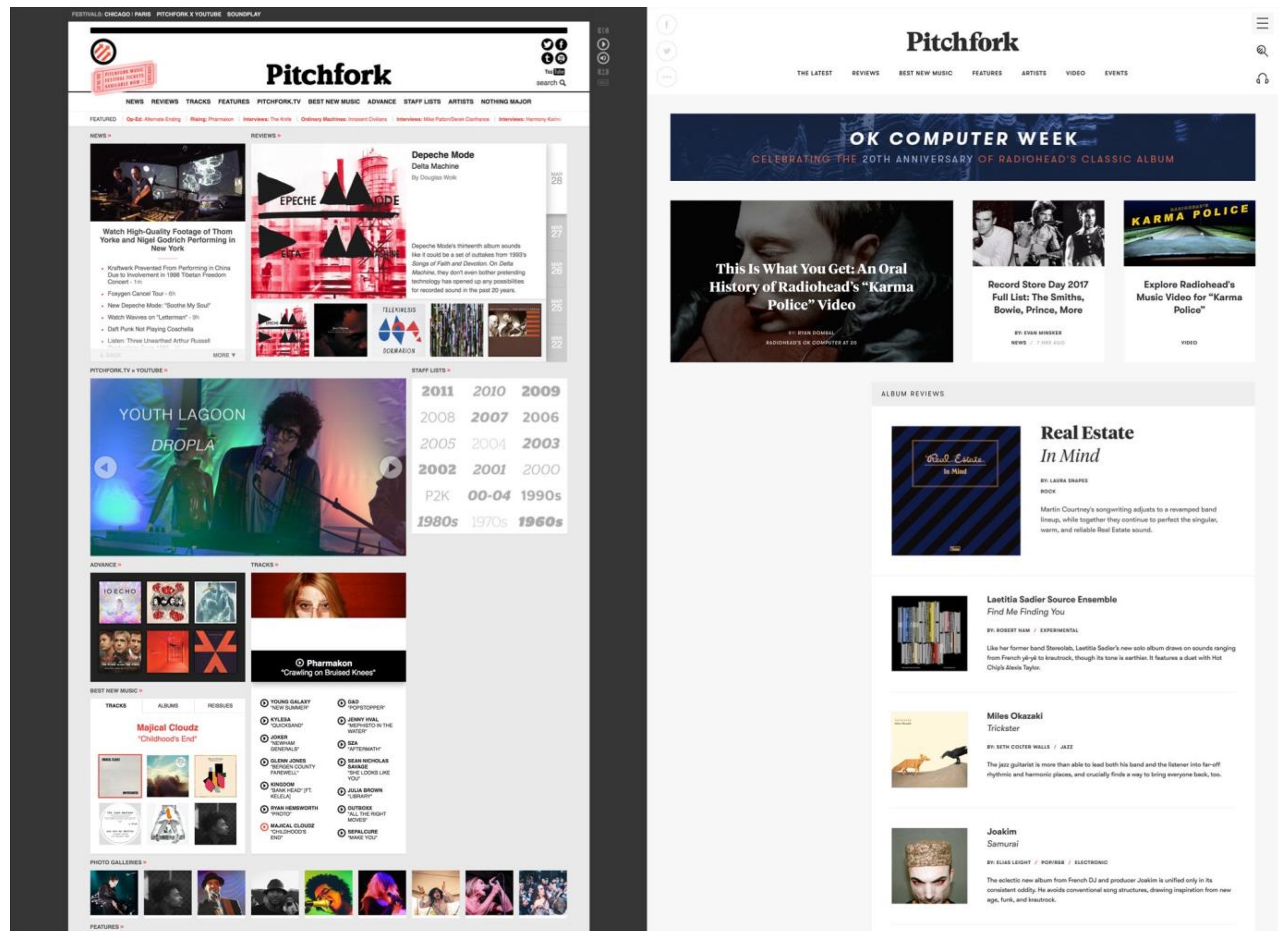
https://www.bluefountainmedia.com/blog/desktop-vs-mobile-three-key-website-design-differences

# Mobile-first, not mobile-only

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







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

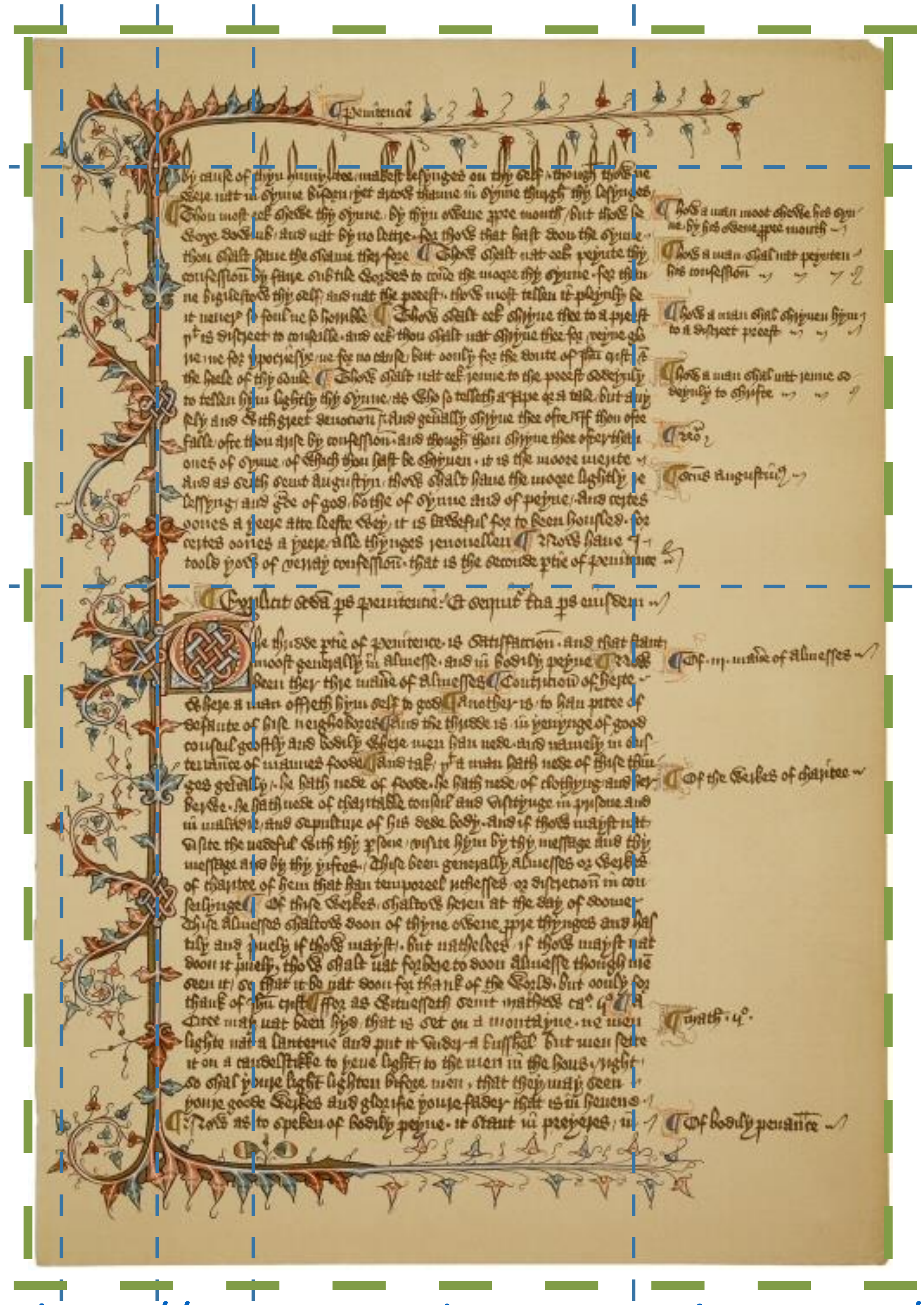
## Mobile-first, not mobile-only

- 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

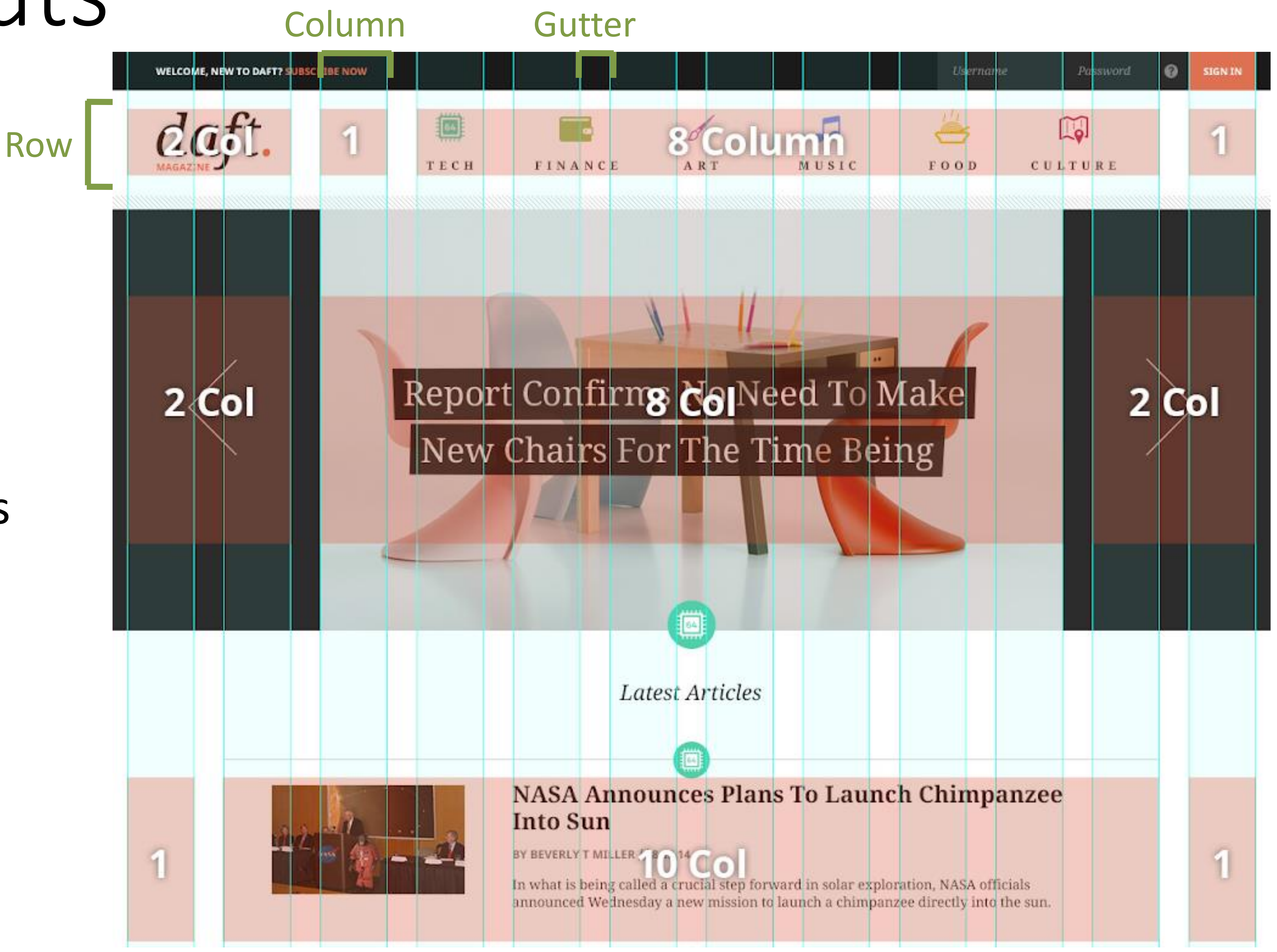
- Established tool
  for content arrangement
- Gridded content is familiar
  and easy to follow
- In general, it's good to target
  fewer lines
  - But breaking that rule is important
    for creativity and attention-grabbing



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

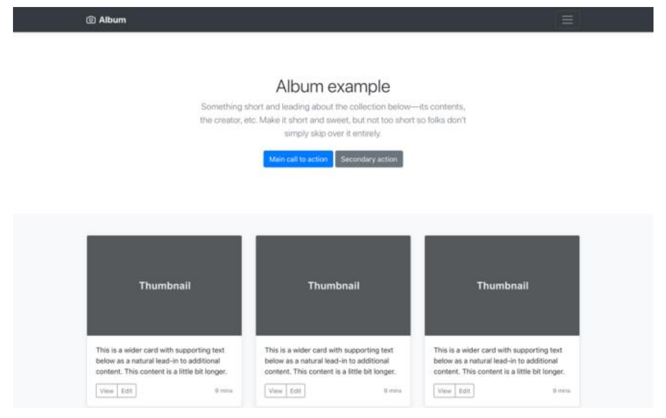
## Grid-based layouts

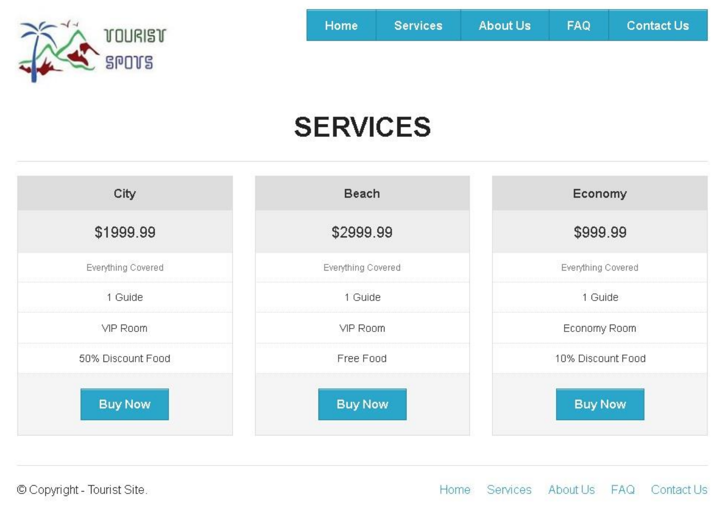
- Rows
- Columns
- Gutters
- Padding/spacing
  - Defined by specific elements


Column
Gutter
Row
WELCOME, NEW TO DAFT? SUBSCRIBE NOW
Username
Password
SIGN IN
daft. MAGAZINE
2 Col
1
TECH
FINANCE
8 Column
ART
MUSIC
FOOD
CULTURE
1
2 Col
Report Confirms No Need To Make New Chairs For The Time Being
8 Col
2 Col
Latest Articles
1
NASA Announces Plans To Launch Chimpanzee Into Sun
BY BEVERLY T MILLER
10 Col
In what is being called a crucial step forward in solar exploration, NASA officials announced Wednesday a new mission to launch a chimpanzee directly into the sun.
1

# 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/v7/)
  - Even smaller, 3.39KB
  - Low-level (closer to raw CSS)





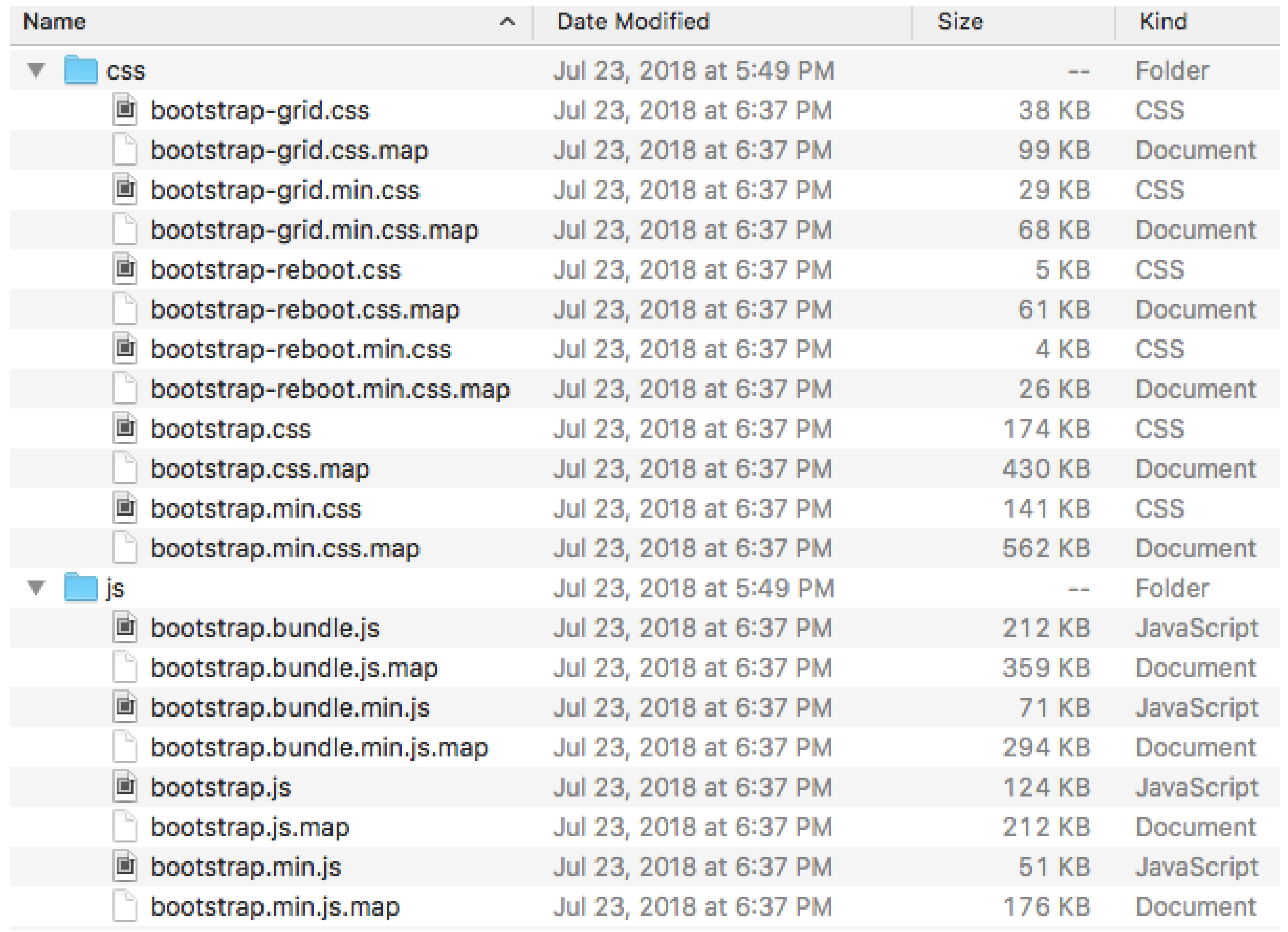
A set of small, responsive CSS modules that you can use in every web project.

## Digging into Bootstrap

Bootstrap

# Bootstrap

- Direct download
  - http://getbootstrap.com/docs/4.5/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

- Load bootstrap

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

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

# 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://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
```

# 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>
```

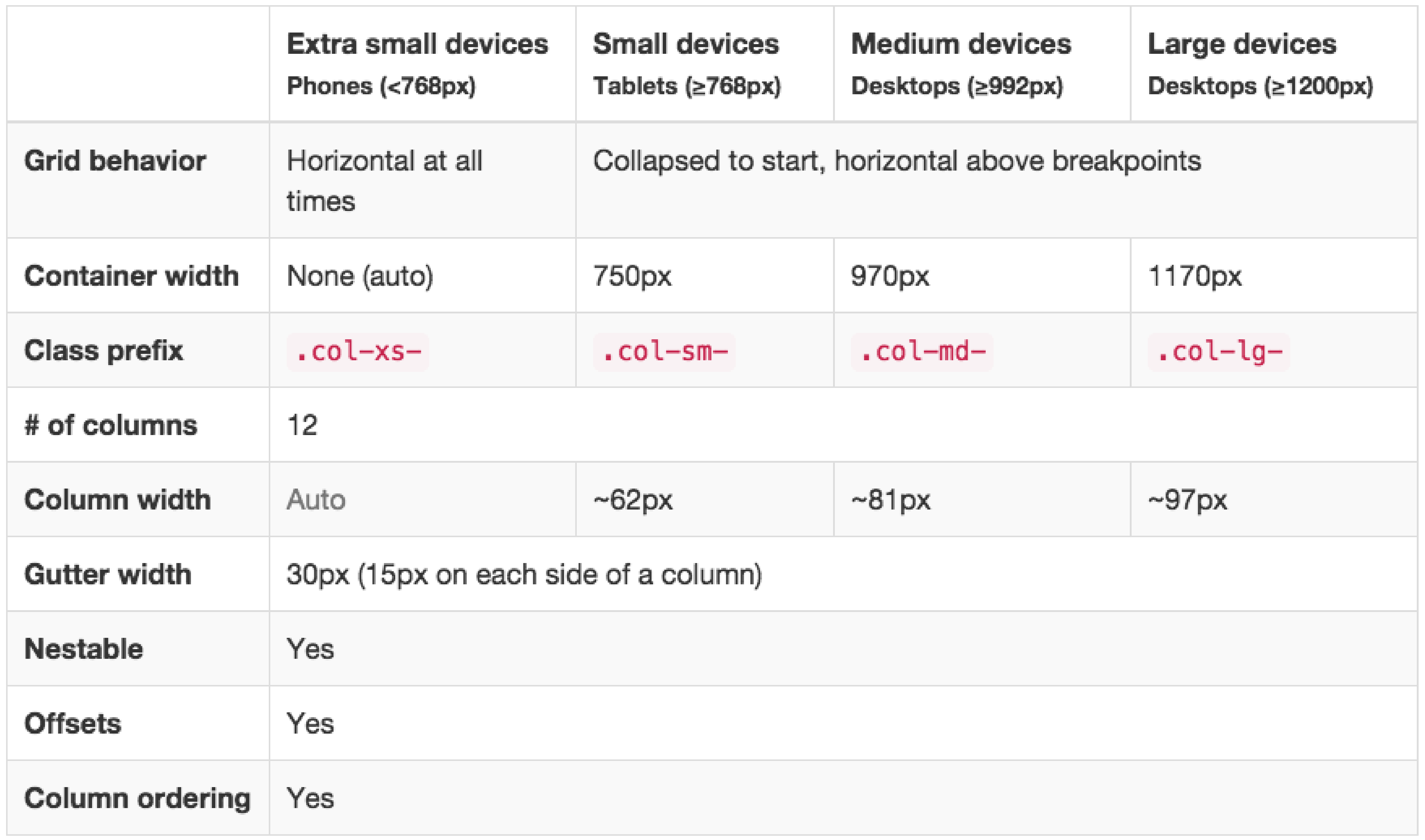
- Just a class; anything can be a container

```
<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]`



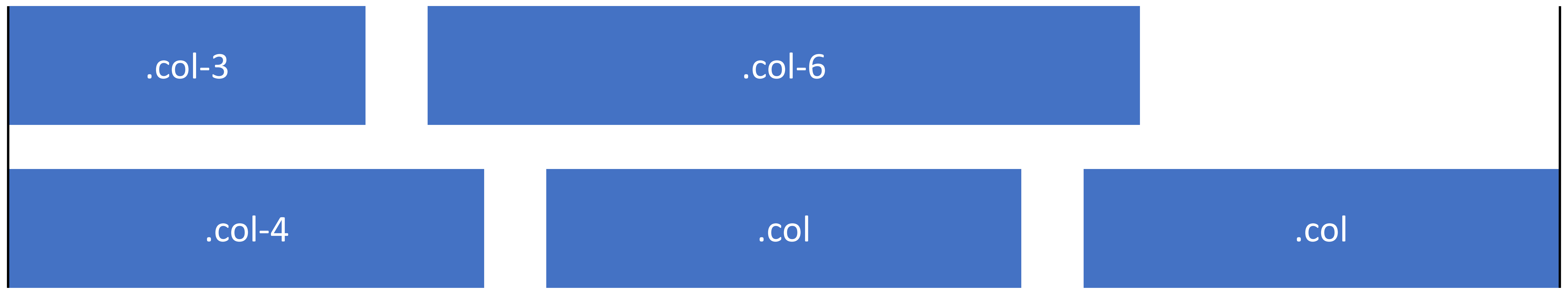
| | **Extra small devices** Phones (<768px) | **Small devices** Tablets (≥768px) | **Medium devices** Desktops (≥992px) | **Large devices** Desktops (≥1200px) |
|---|---|---|---|---|
| **Grid behavior** | Horizontal at all times | Collapsed to start, horizontal above breakpoints | | |
| **Container width** | None (auto) | 750px | 970px | 1170px |
| **Class prefix** | `.col-xs-` | `.col-sm-` | `.col-md-` | `.col-lg-` |
| **# of columns** | 12 | | | |
| **Column width** | Auto | ~62px | ~81px | ~97px |
| **Gutter width** | 30px (15px on each side of a column) | | | |
| **Nestable** | Yes | | | |
| **Offsets** | 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>
```

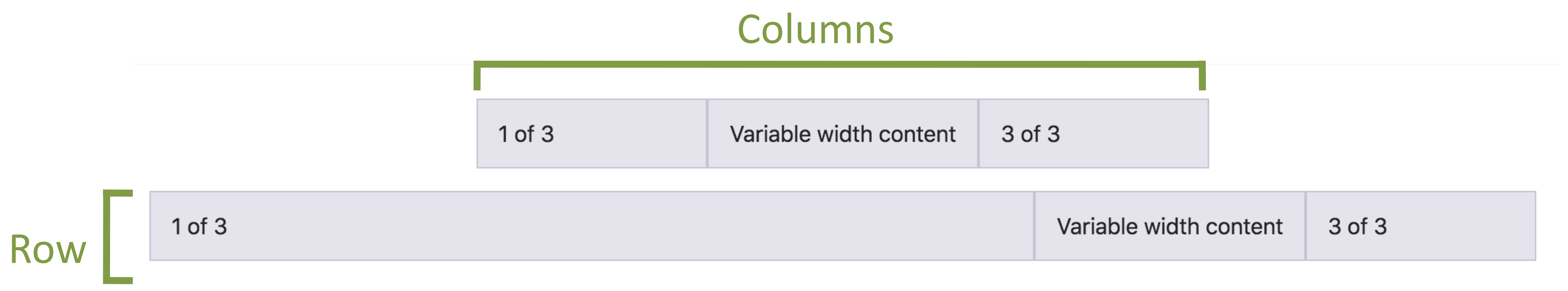


# Bootstrap

## Grid System

- Rows are block elements, while columns are inline

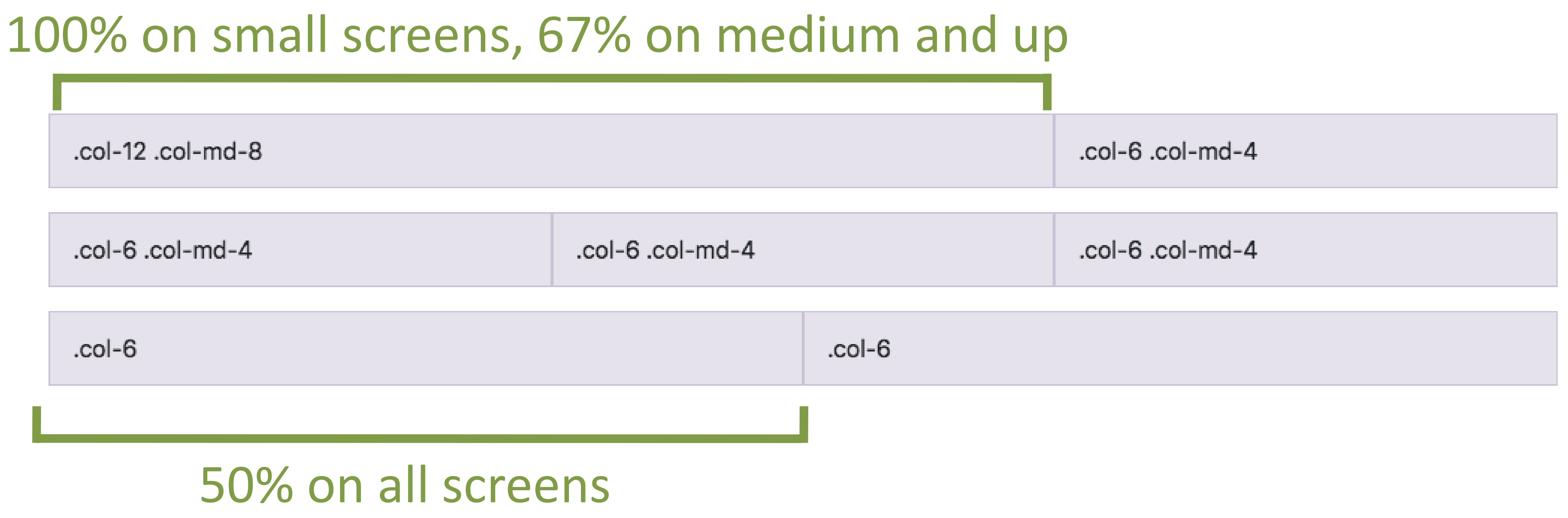
https://getbootstrap.com/docs/4.1/layout/grid/



## 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



## 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 */
}
```

## Bootstrap

### Media queries

```
/* Extra small devices (phones, less than 768px) */
@include media-breakpoint-up(xs) { ... }

/* Small devices (tablets, 768px and up) */
@include media-breakpoint-up(sm) { ... }

/* Medium devices (desktops, 992px and up) */
@include media-breakpoint-up(md) { ... }

/* Large devices (large desktops, 1200px and up) */
@include media-breakpoint-up(lg) { ... }
```

- Variables are Sass mixins, we'll discuss those later in the quarter

# Bootstrap

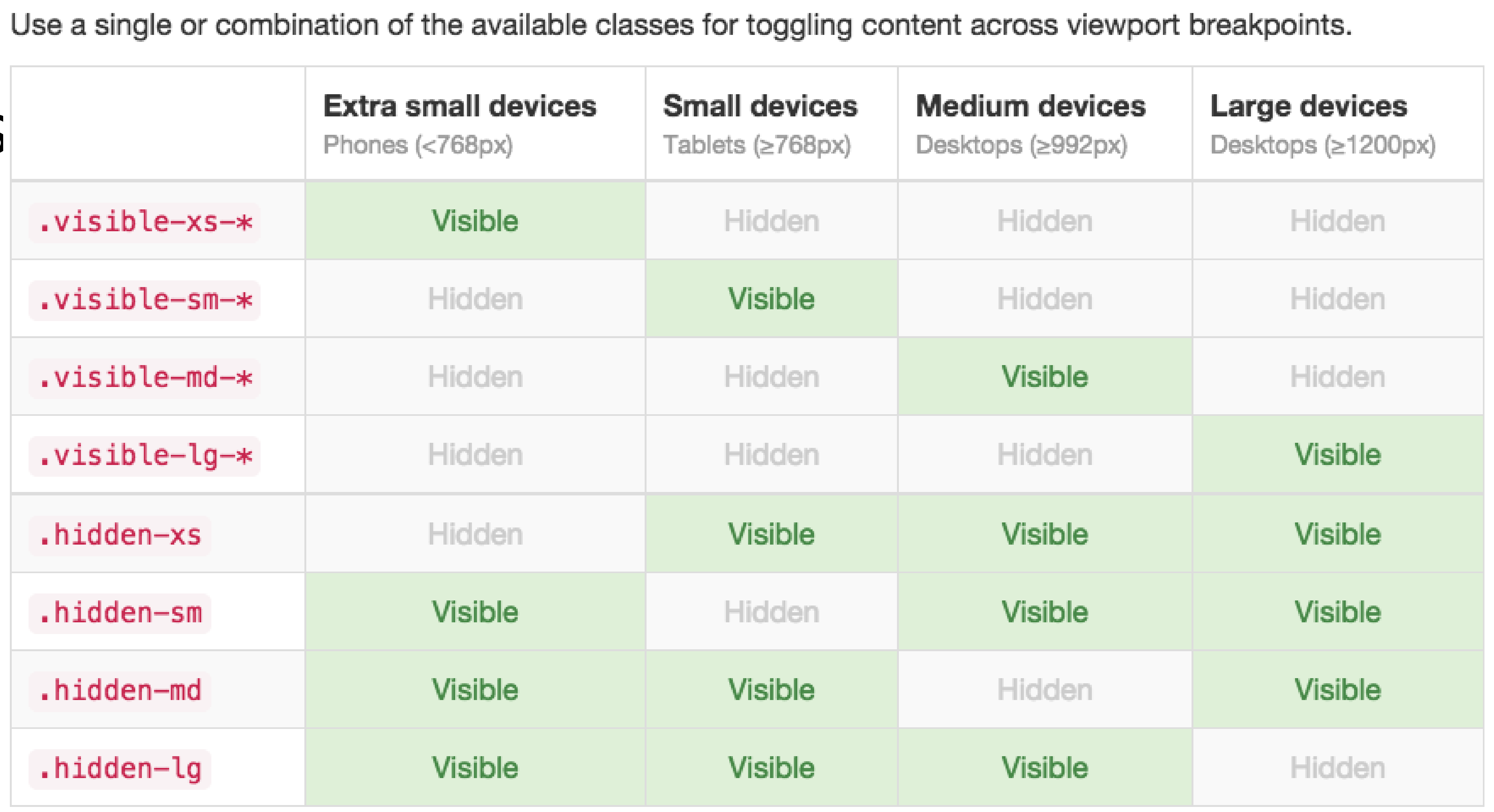
## Media queries

```
// Example usage:
@include media-breakpoint-up(sm) {
  .some-class {
    display: block;
  }
}
```

# Bootstrap

## Hiding and showing

- There are some helpful classes for showing and hiding content acros breakpoints


Use a single or combination of the available classes for toggling content across viewport breakpoints.

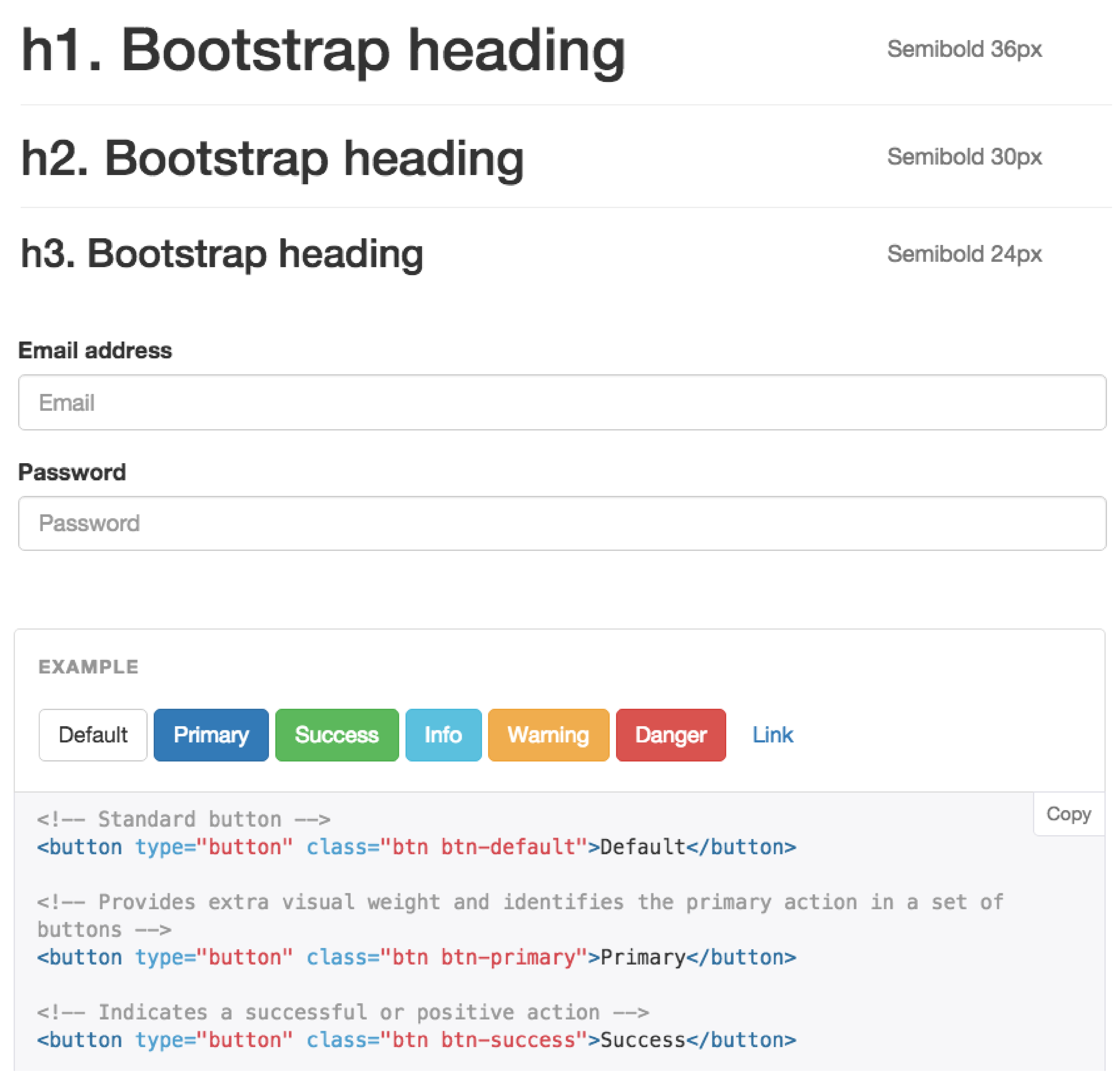
| | Extra small devices Phones (<768px) | Small devices Tablets (≥768px) | Medium devices Desktops (≥992px) | Large devices Desktops (≥1200px) |
|---|---|---|---|---|
| `.visible-xs-*` | Visible | Hidden | Hidden | Hidden |
| `.visible-sm-*` | Hidden | Visible | Hidden | Hidden |
| `.visible-md-*` | Hidden | Hidden | Visible | Hidden |
| `.visible-lg-*` | Hidden | Hidden | Hidden | Visible |
| `.hidden-xs` | Hidden | Visible | Visible | Visible |
| `.hidden-sm` | Visible | Hidden | Visible | Visible |
| `.hidden-md` | Visible | Visible | Hidden | Visible |
| `.hidden-lg` | Visible | Visible | Visible | Hidden |

http://getbootstrap.com/css

# Bootstrap

## Default styling

- Bootstrap will change a lot of styles for you
- There are other custom styles involving various suffixes



h1. Bootstrap heading — Semibold 36px

h2. Bootstrap heading — Semibold 30px

h3. Bootstrap heading — Semibold 24px

Email address

Email

Password

Password

EXAMPLE

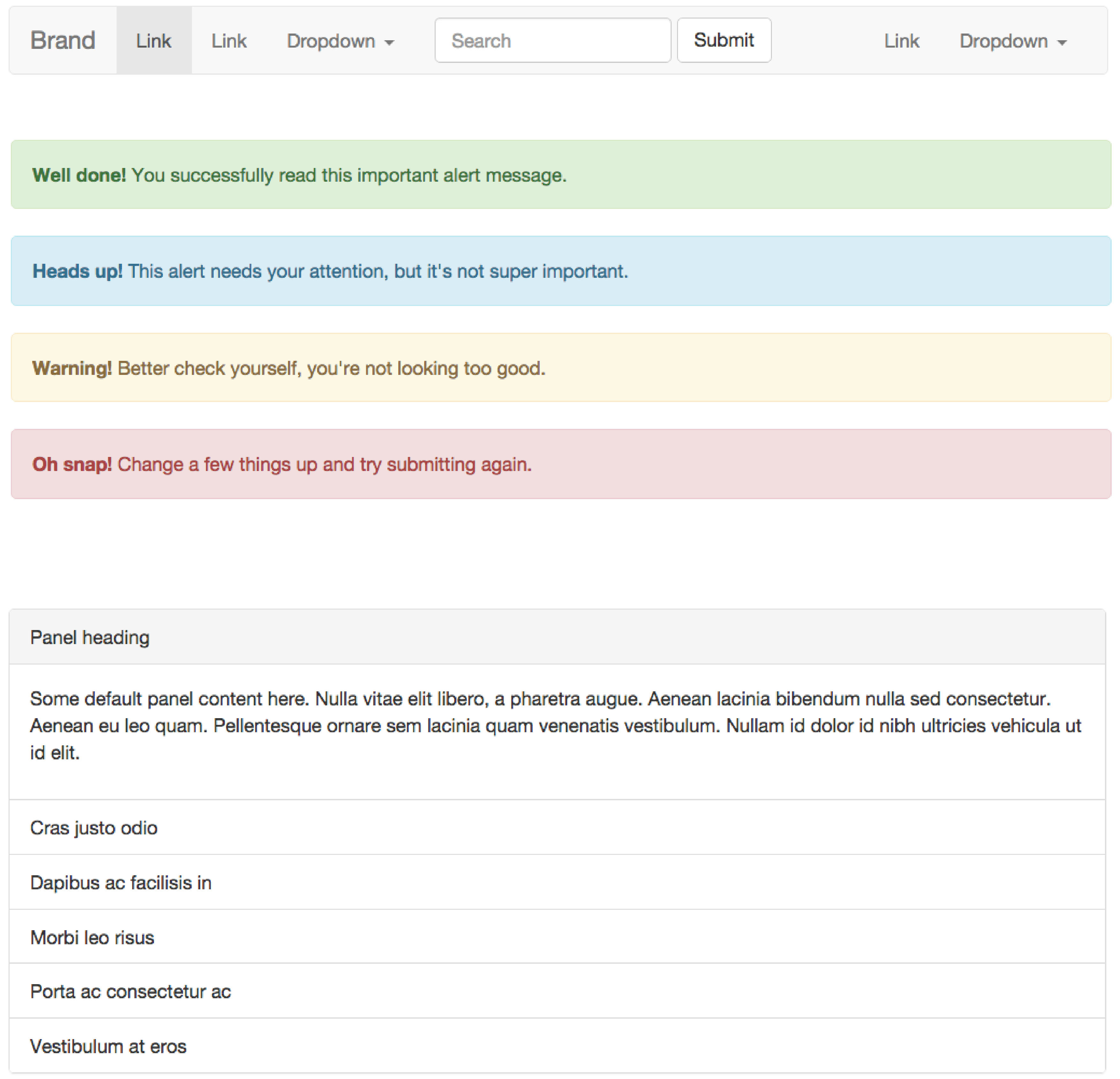
```
<!-- Standard button -->
<button type="button" class="btn btn-default">Default</button>

<!-- Provides extra visual weight and identifies the primary action in a set of
buttons -->
<button type="button" class="btn btn-primary">Primary</button>

<!-- Indicates a successful or positive action -->
<button type="button" class="btn btn-success">Success</button>
```

http://getbootstrap.com/css

# Bootstrap

## Components

- Components are elements pre-arranged into common patterns
- Makes making navigation bars, dropdowns, alerts, etc. simpler
- Some require JavaScript



**Well done!** You successfully read this important alert message.

**Heads up!** This alert needs your attention, but it's not super important.

**Warning!** Better check yourself, you're not looking too good.

**Oh snap!** Change a few things up and try submitting again.

Panel heading

Some default panel content here. Nulla vitae elit libero, a pharetra augue. Aenean lacinia bibendum nulla sed consectetur. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum. Nullam id dolor id nibh ultricies vehicula ut id elit.

Cras justo odio

Dapibus ac facilisis in

Morbi leo risus

Porta ac consectetur ac

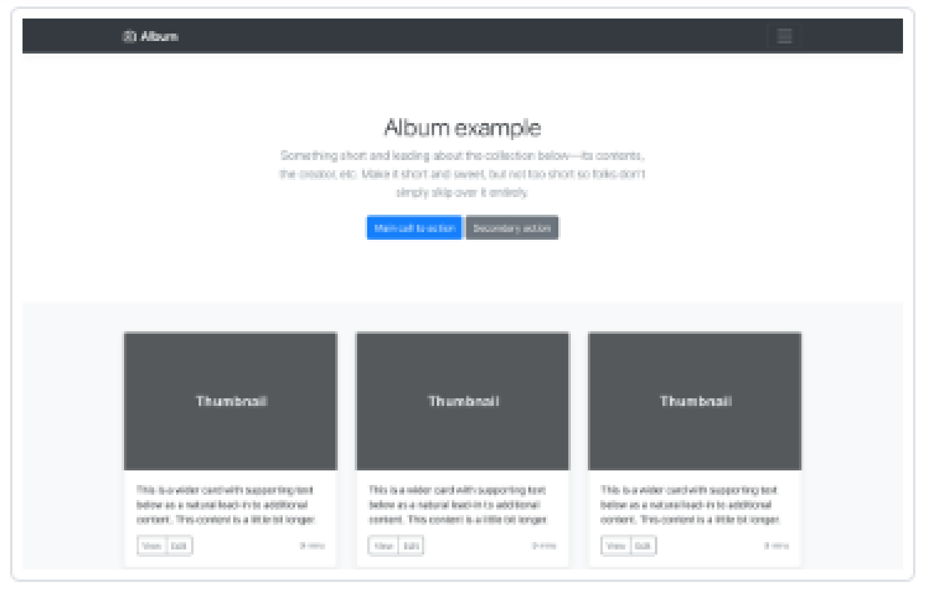
Vestibulum at eros

http://getbootstrap.com/css

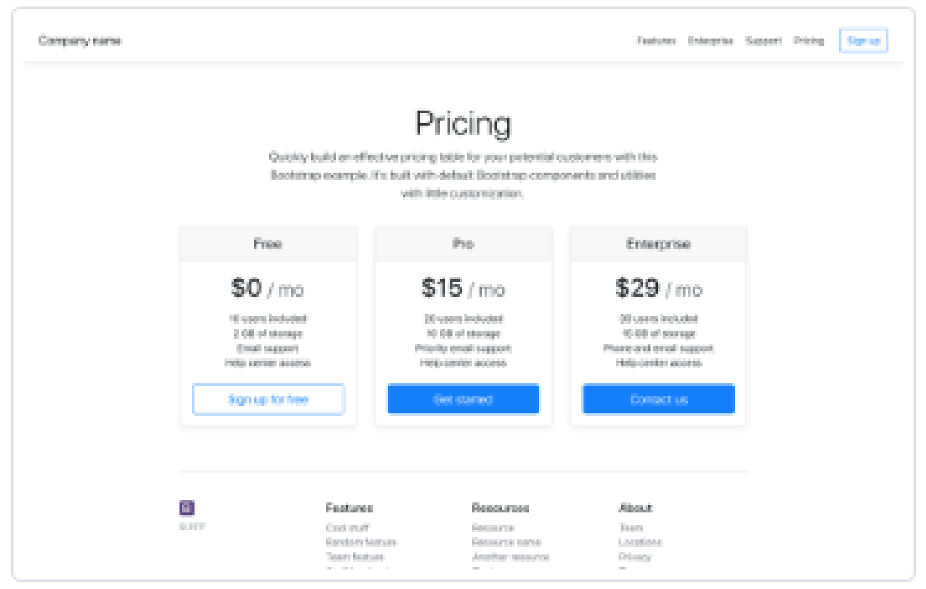
## Grid frameworks make development easier. What are the downsides?

# Opposition to Grid-based frameworks

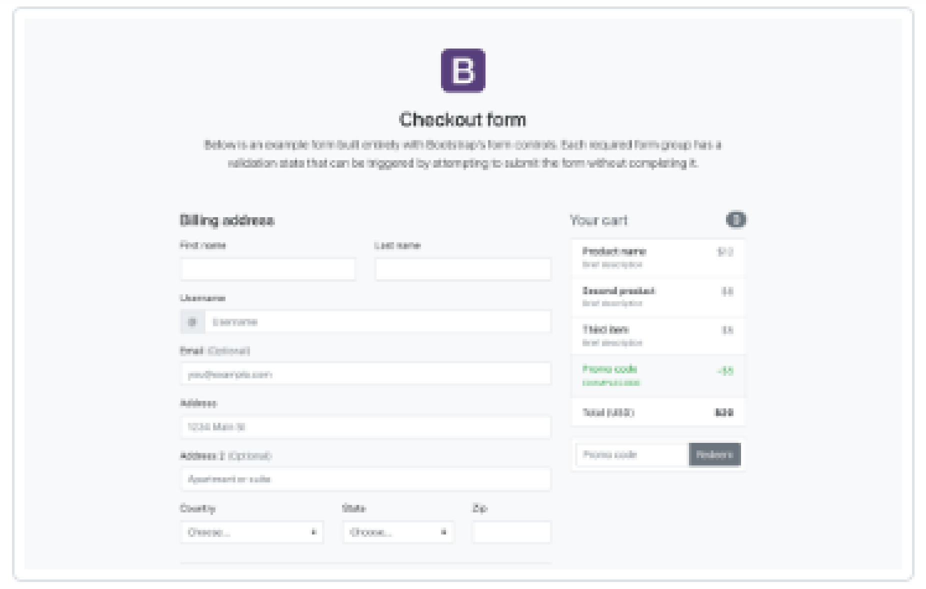
## Can lead to similar-looking webpages



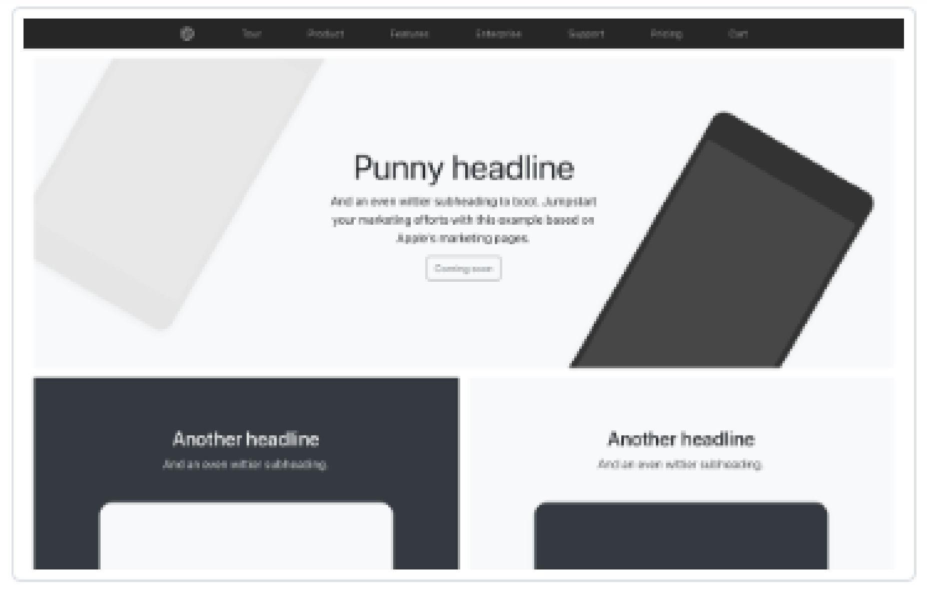
**Album**
Simple one-page template for photo galleries, portfolios, and more.



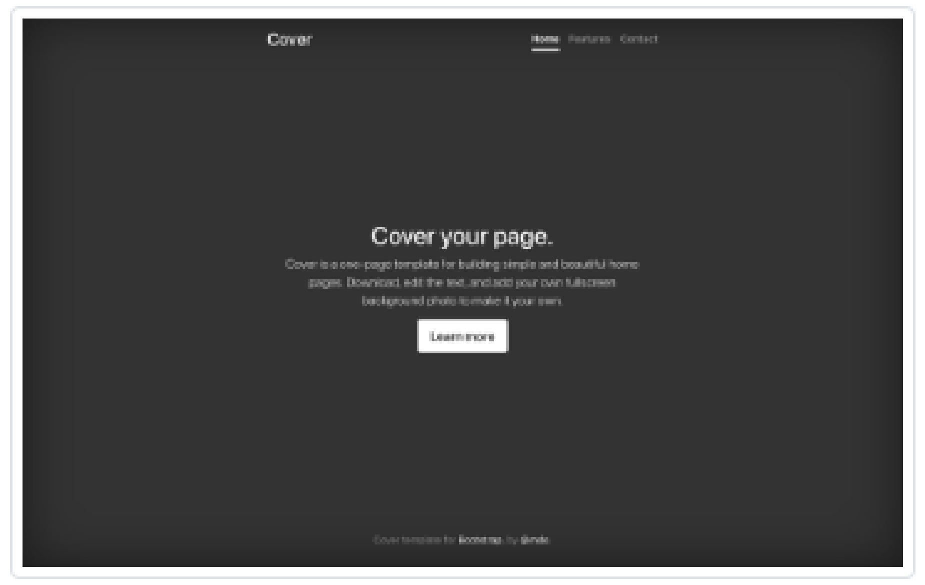
**Pricing**
Example pricing page built with Cards and featuring a custom header and footer.



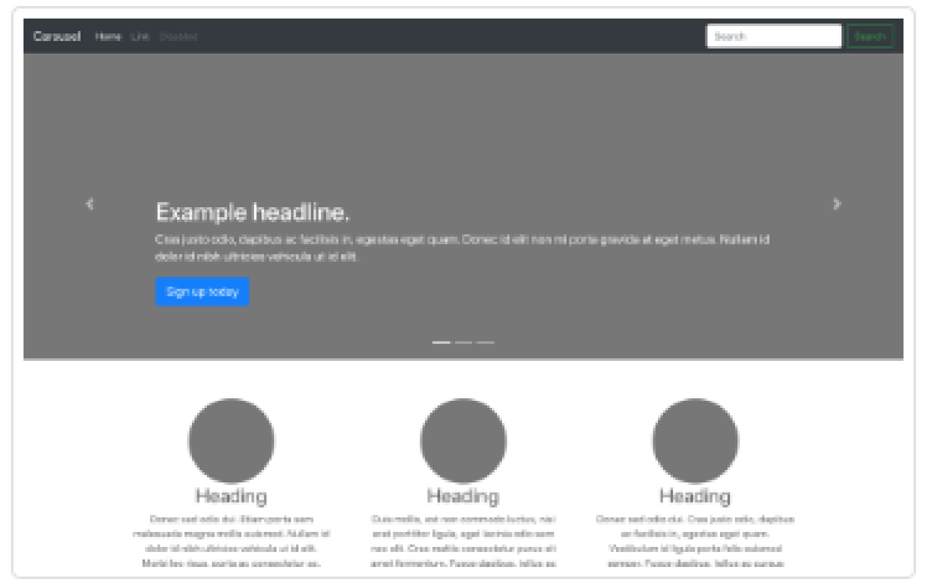
**Checkout**
Custom checkout form showing our form components and their validation features.



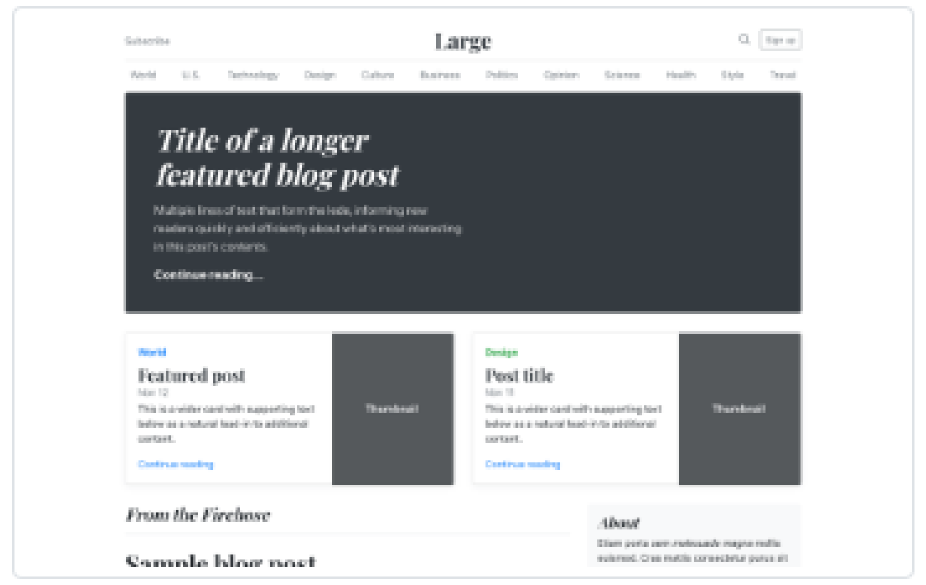
**Product**
Lean product-focused marketing page with extensive grid and image work.



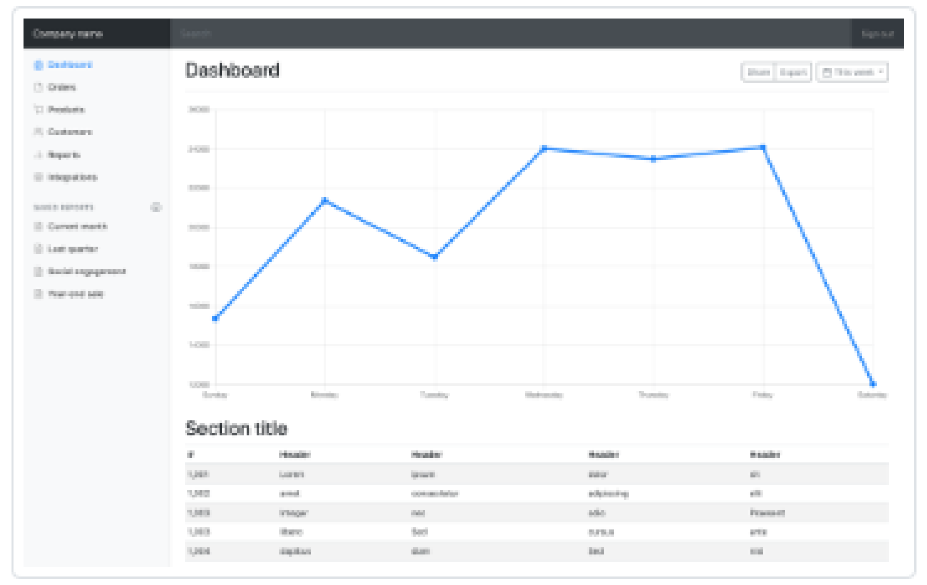
**Cover**
A one-page template for building simple and beautiful home pages.



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



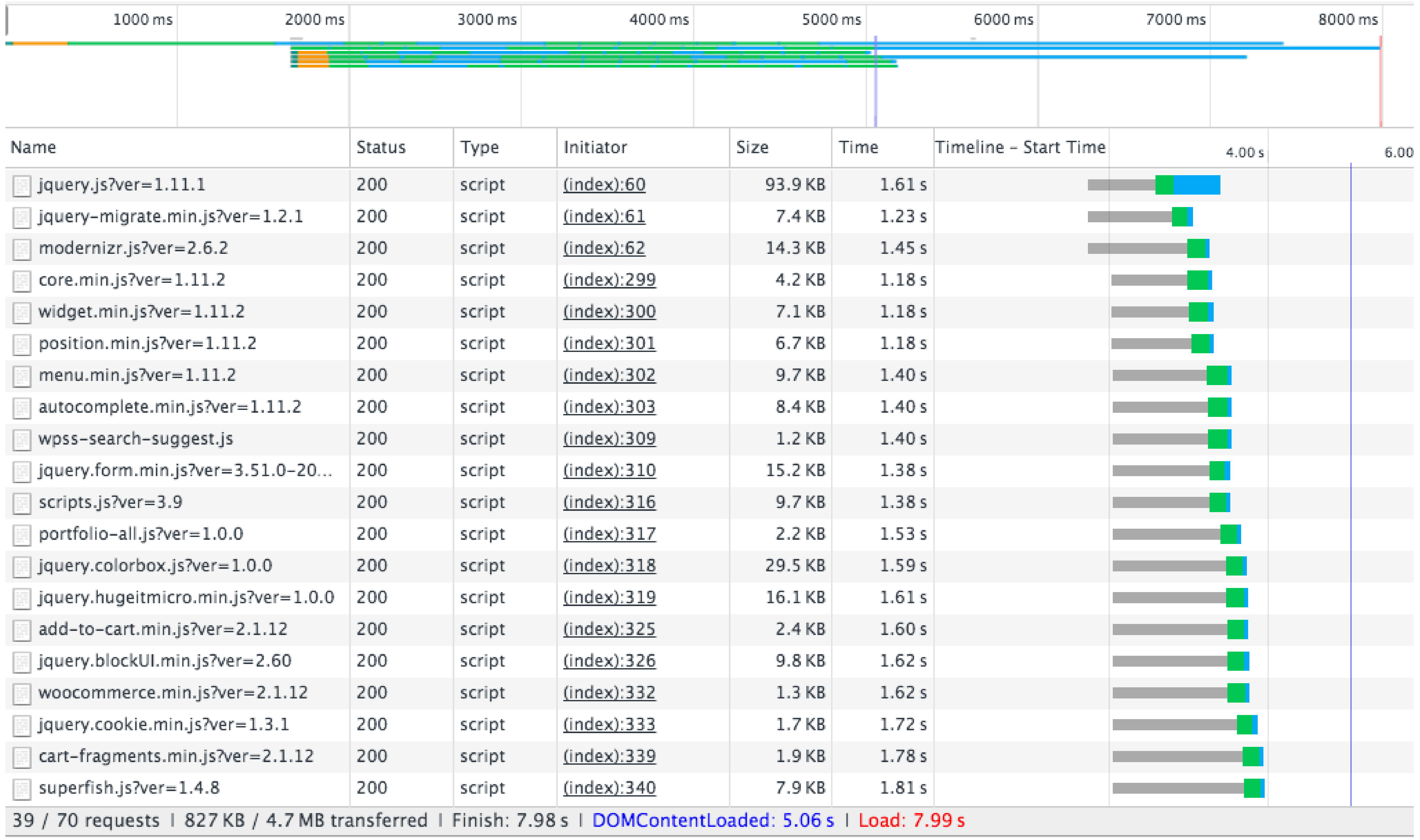
**Blog**
Magazine like blog template with header, navigation, featured content.



**Dashboard**
Basic admin dashboard shell with fixed sidebar and navbar.

# Opposition to Grid-based frameworks

## Can involve loading many files, hurting performance

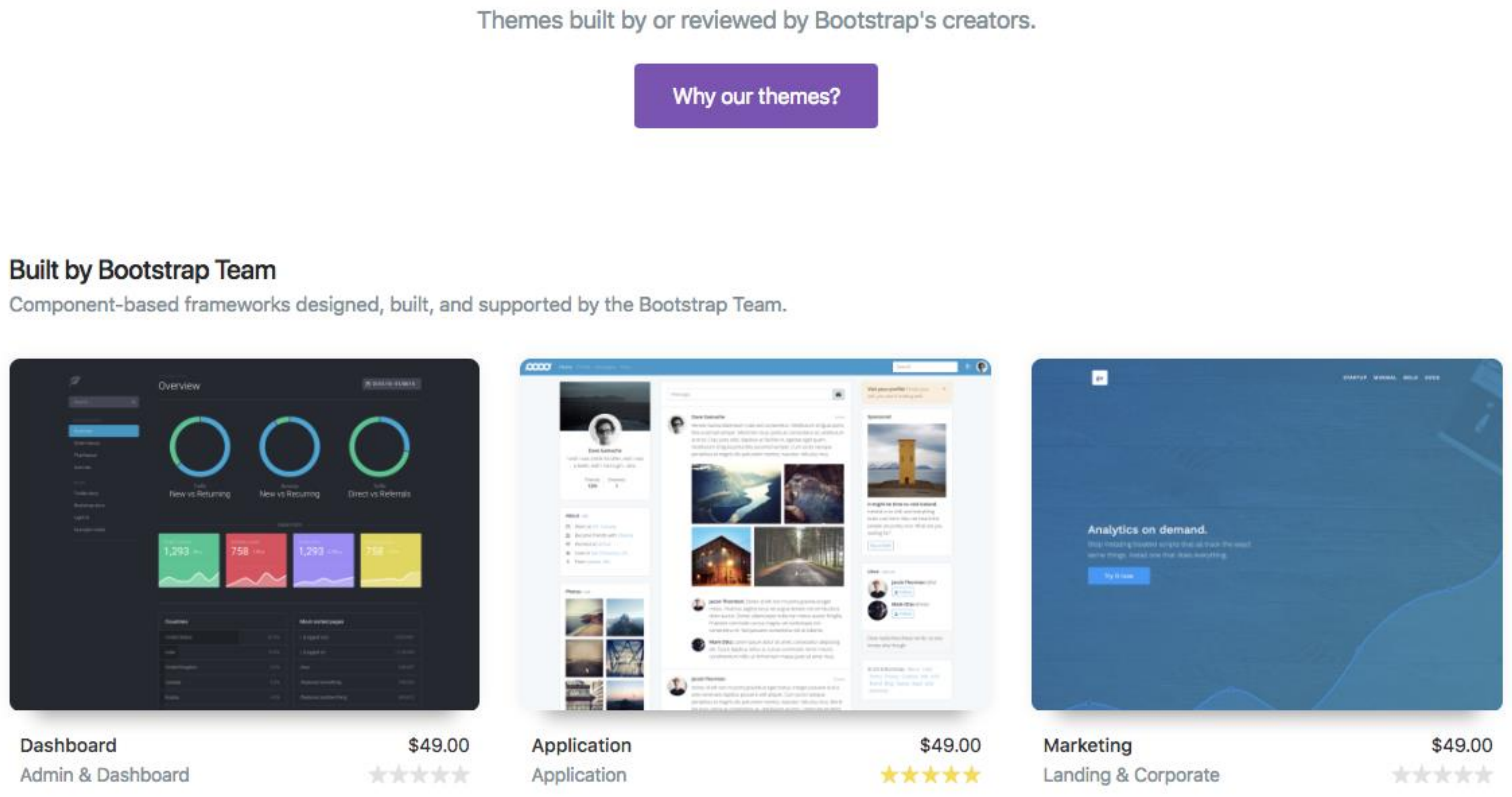


| Name | Status | Type | Initiator | Size | Time | Timeline – Start Time |
|---|---|---|---|---|---|---|
| jquery.js?ver=1.11.1 | 200 | script | (index):60 | 93.9 KB | 1.61 s | |
| jquery-migrate.min.js?ver=1.2.1 | 200 | script | (index):61 | 7.4 KB | 1.23 s | |
| modernizr.js?ver=2.6.2 | 200 | script | (index):62 | 14.3 KB | 1.45 s | |
| core.min.js?ver=1.11.2 | 200 | script | (index):299 | 4.2 KB | 1.18 s | |
| widget.min.js?ver=1.11.2 | 200 | script | (index):300 | 7.1 KB | 1.18 s | |
| position.min.js?ver=1.11.2 | 200 | script | (index):301 | 6.7 KB | 1.18 s | |
| menu.min.js?ver=1.11.2 | 200 | script | (index):302 | 9.7 KB | 1.40 s | |
| autocomplete.min.js?ver=1.11.2 | 200 | script | (index):303 | 8.4 KB | 1.40 s | |
| wpss-search-suggest.js | 200 | script | (index):309 | 1.2 KB | 1.40 s | |
| jquery.form.min.js?ver=3.51.0-20… | 200 | script | (index):310 | 15.2 KB | 1.38 s | |
| scripts.js?ver=3.9 | 200 | script | (index):316 | 9.7 KB | 1.38 s | |
| portfolio-all.js?ver=1.0.0 | 200 | script | (index):317 | 2.2 KB | 1.53 s | |
| jquery.colorbox.js?ver=1.0.0 | 200 | script | (index):318 | 29.5 KB | 1.59 s | |
| jquery.hugeitmicro.min.js?ver=1.0.0 | 200 | script | (index):319 | 16.1 KB | 1.61 s | |
| add-to-cart.min.js?ver=2.1.12 | 200 | script | (index):325 | 2.4 KB | 1.60 s | |
| jquery.blockUI.min.js?ver=2.60 | 200 | script | (index):326 | 9.8 KB | 1.62 s | |
| woocommerce.min.js?ver=2.1.12 | 200 | script | (index):332 | 1.3 KB | 1.62 s | |
| jquery.cookie.min.js?ver=1.3.1 | 200 | script | (index):333 | 1.7 KB | 1.72 s | |
| cart-fragments.min.js?ver=2.1.12 | 200 | script | (index):339 | 1.9 KB | 1.78 s | |
| superfish.js?ver=1.4.8 | 200 | script | (index):340 | 7.9 KB | 1.81 s | |

39 / 70 requests | 827 KB / 4.7 MB transferred | Finish: 7.98 s | DOMContentLoaded: 5.06 s | Load: 7.99 s

## Opposition to Grid-based frameworks

### Can stifle creativity



Themes built by or reviewed by Bootstrap's creators.

Built by Bootstrap Team

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

| | |
|---|---|
| Dashboard | $49.00 |
| Admin & Dashboard | ★★★★★ |
| Application | $49.00 |
| Application | ★★★★★ |
| Marketing | $49.00 |
| Landing & Corporate | ★★★★★ |

## Today’s goals

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

- 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
- Utilize media queries to create responsive layouts
- Develop grid-based layouts using Bootstrap