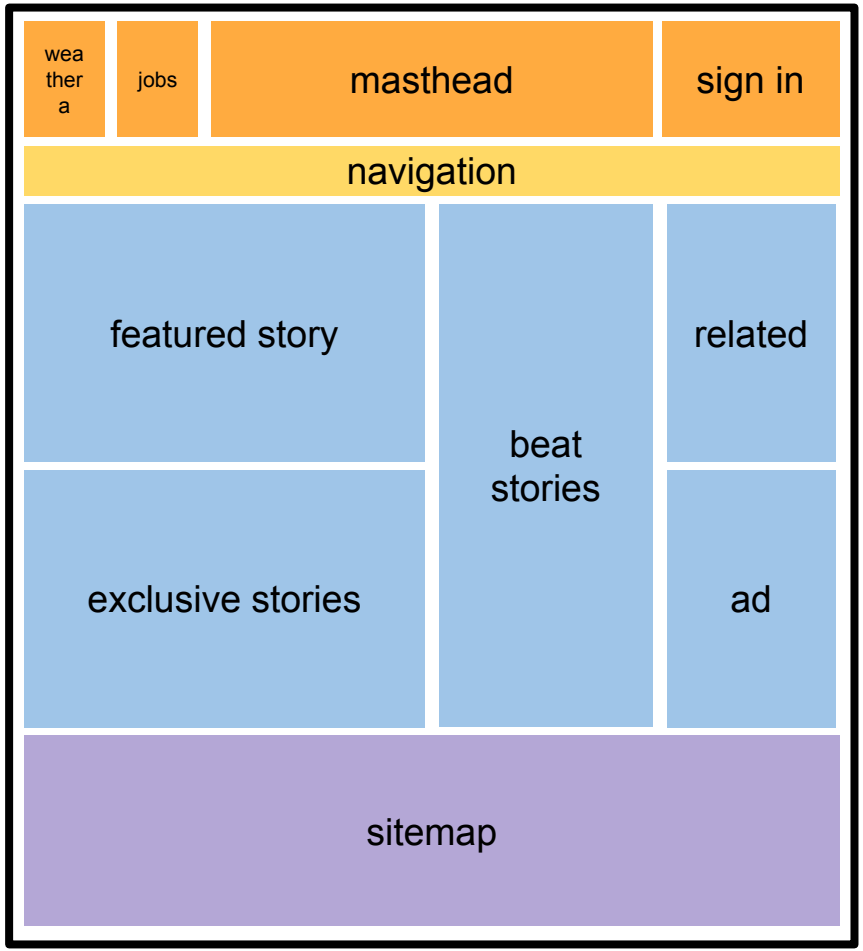
Responsive Web Design

Day 13: I399 Web Development

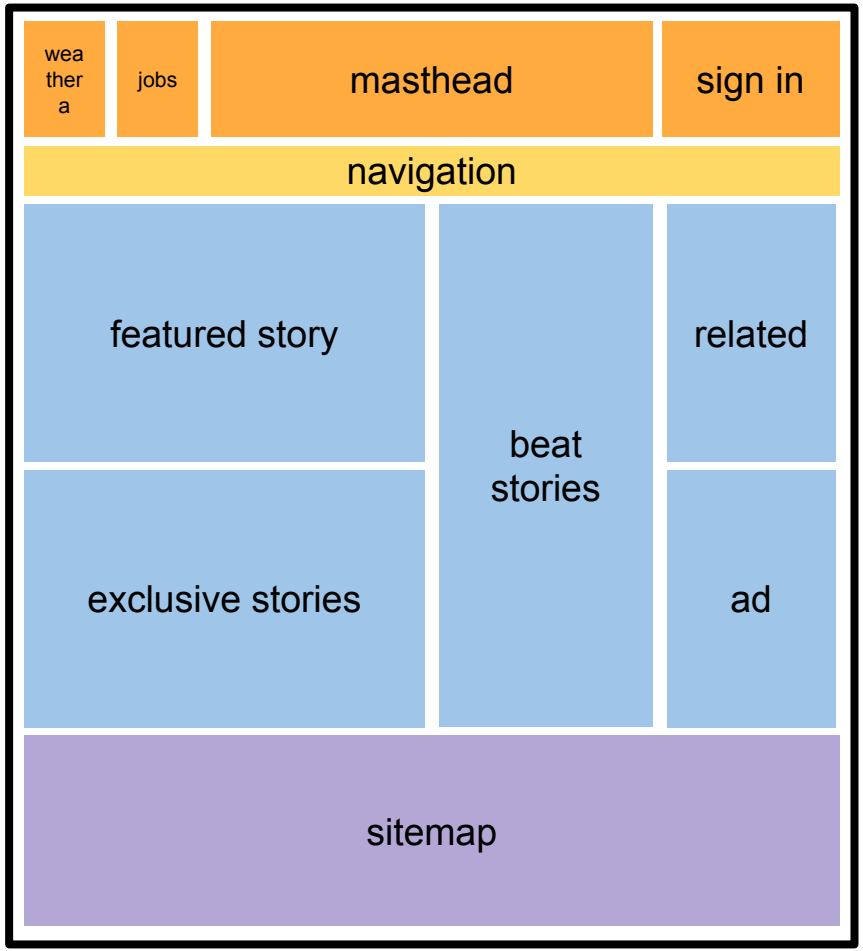


Level up:

You've been given the opportunity to try to save local news by redesigning your local newspaper's website.

You come up with a basic daily layout on the left for sites viewed on a desktop (960px or wider), but it has to work as a narrower mobile site as well (often around 320px).

Without worrying about how that is going to happen technically, **how would you structure the content so it fits into a mobile width?**



Work in pairs.

Sketch how you would structure the site content to fit a mobile width.

- Think about what content is most important -- *what items should appear in the stack first?*
- Are some item ***hidden***?
- Are some item ***condensed***?
- Are some items ***divided***?

Come up with a solution to share with the class.

What is responsive web design?

Responsive web design (RWD) is an approach to **web design** aimed at crafting sites to provide an optimal viewing and interaction experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from desktop computer monitors to mobile phones).

—https://en.wikipedia.org/wiki/Responsive_web_design

Ethan Marcotte
coined the term & proposed concept for “responsive web design”

http://alistapart.com/article/responsive-web-design

Methods for dealing with multiple viewports:

Adaptive web design
(designing for set widths)

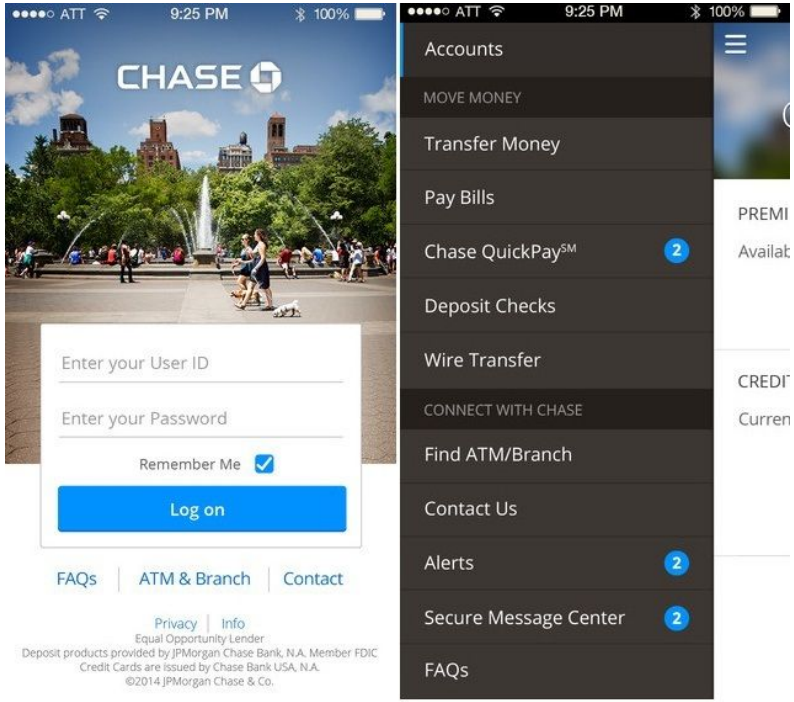
- http://www.amazon.com/

Responsive web design
(flexible with breakpoints)

- https://www.bostonglobe.com

Mobile-only site and/or an app

- https://www.chase.com/



Chase mobile app

Best practices:

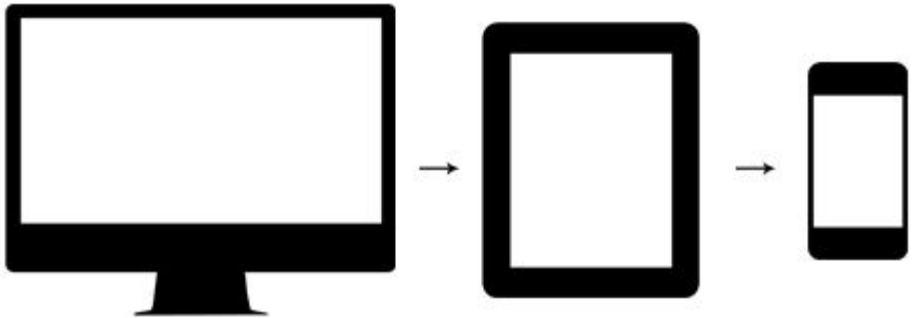
Think in systems not pages

- use modular design techniques
- design for mobile view first
- benefit of smaller code base
- optimize design and code for viewports

http://mediaqueri.es/

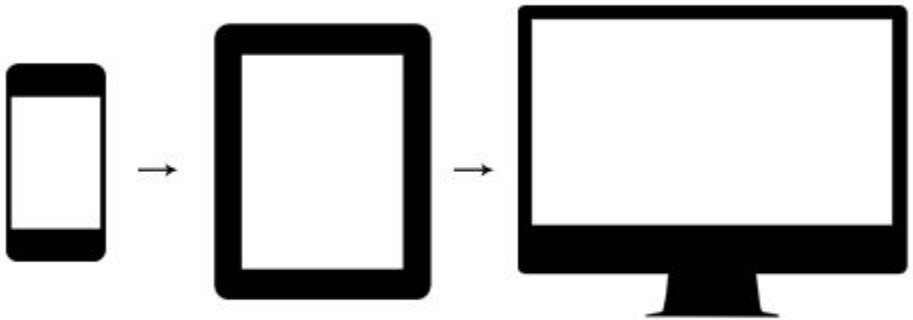
“Graceful degradation”

When mobile devices really started to catch on and “smart phones” -- devices enabled with internet and apps -- came out around 2009, designers rushed to make websites look and work better on tablets and phones. The idea was to have the complex full website ***gracefully degrade*** by removing features and elements so it would still be functional on mobile devices.



“Progressive enhancement”

This worked for a time, but eventually many designers realized that the experience needed to be great on all devices. The phone wasn’t just more convenient, it was perhaps the only experience a user might have with your site. The idea of ***progressive enhancement*** is that experience can be enhanced on larger screens or devices with more processing power, rather than degraded for mobile.

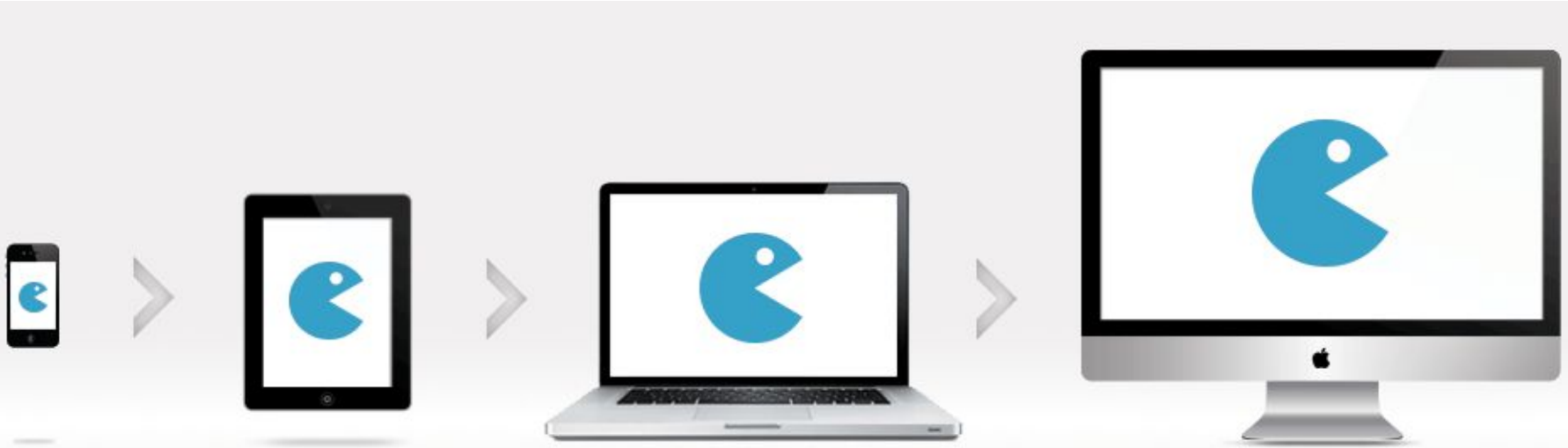


So what is “mobile first”?

A developer named Luke Wroblewski coined the term to mean the **PROCESS** of designing a website for ***mobile first,*** then use this idea of “progressive enhancement” to design a site for larger screens. People expect websites on phones to load quickly. *A site for desktop might be fantastic, but on mobile it’s like needing a packed lunch and bringing your refrigerator.*

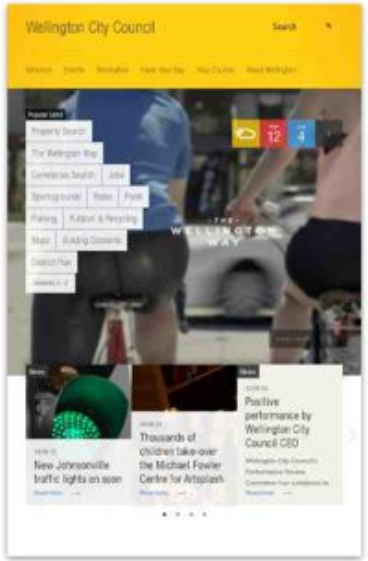
Issues with “mobile first”

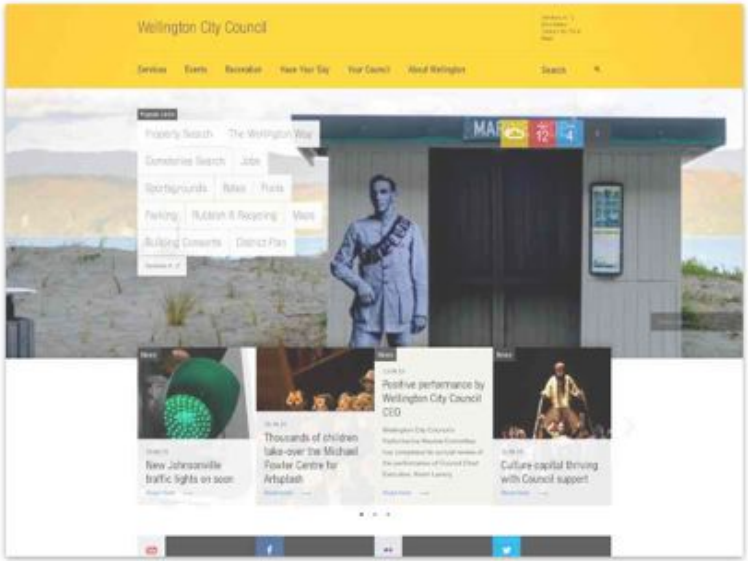
The largest issue with mobile first is that you’re already limited -- by processing power and browser choices (though much less these days for both), and mentally by the physical space allowed. Most designers like to work with a lot of screen space, and were/are used to doing so. *Now you’re starting with your peas and there is no cake -- but it does make for better usability.*



RWD Inspiration

http://mediaqueri.es/





City of Wellington

Building a RWD grid

Frameworks

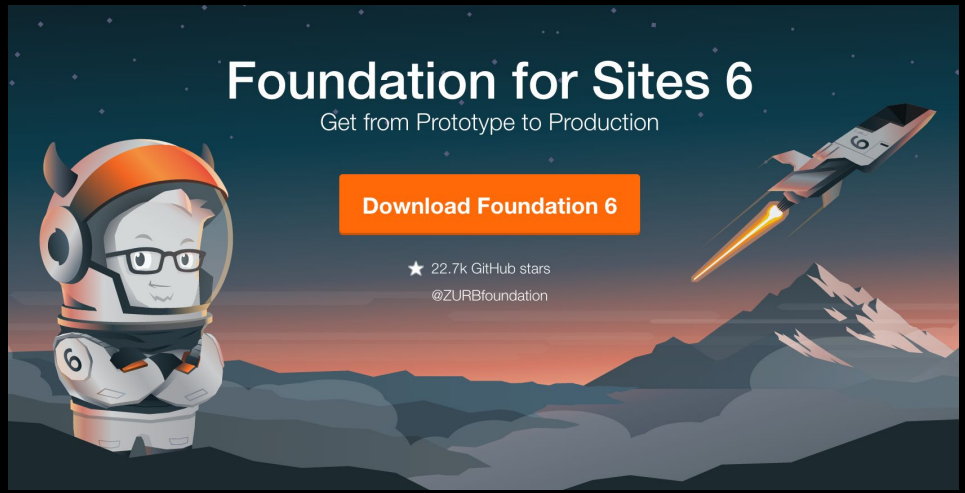
- HTML, CSS & JS template
- Includes a flexible grid
- Ready-to-use UI elements
- Quick start

‘Roll my own’

- Complete control
- Lighter weight

FYI: Project 5 = Bootstrap

http://getbootstrap.com/



Media Queries

Add ***break points*** to change your design:

- based on device viewports?
- or based on page design?

Media queries

… can use logic such as “and” and “or”
… vary greatly depending on the need
… usually rely on **min-width** or **max-width** to create flexibility

This set of queries works with the idea of progressive enhancement. Only make adjustments to styles in SM/MD/LG that need to change from how they were styled in XS. Assumes much of your XS code will work for all scenarios.

BOOTSTRAP MEDIA QUERIES

```
/* Extra small devices (phones, less than 768px) */

/* Small devices (tablets, 768px and up) */
@media (min-width: 768px) { ... }

/* Medium devices (desktops, 992px and up) */
@media (min-width: 992px) { ... }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: 1200px) { ... }
```

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries
http://www.w3schools.com/css/css_rwd_mediaqueries.asp

Headline
And some text

```
h1 { font-size: 1.0em }
p { font-size: 14px }
```

CSS primarily designed for smallest screen.

Headline
And some text

```
@media (min-width: 768px) {
    h1 { font-size: 1.5em }
}
```

First break point in this example is for screens that are at least 768px

Headline
And some text

```
@media (min-width: 992px) {
    h1 { font-size: 2.0em }
    p { font-size: 16px }
}
```

Second break point is for screens that are 992px or more wide

Headline
And some text

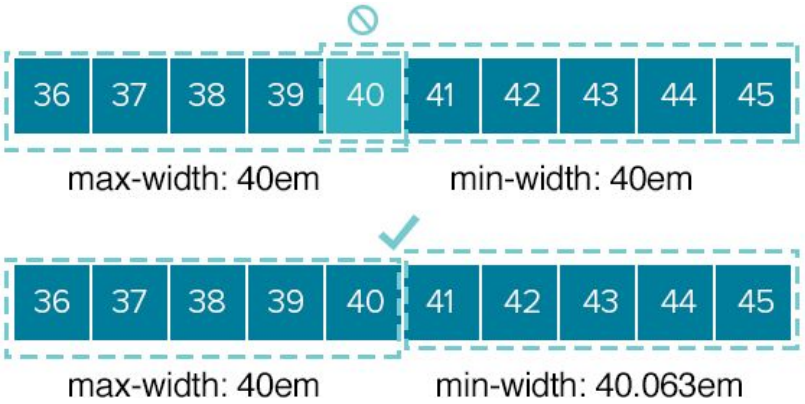
```
@media (min-width: 1200px) {
    h1 { font-size: 3.0em }
}
```

Third break point is for screens that are 1200px or more wide

Media Queries

Another way to handle media queries. Ems work well with typography choices, and that 0.063 extra (about 1px wide) allows for overlap so that browsers will only ever make one choice at those break points.

This method is less forgiving and because it doesn't rely on cascading, you do end up writing more code.



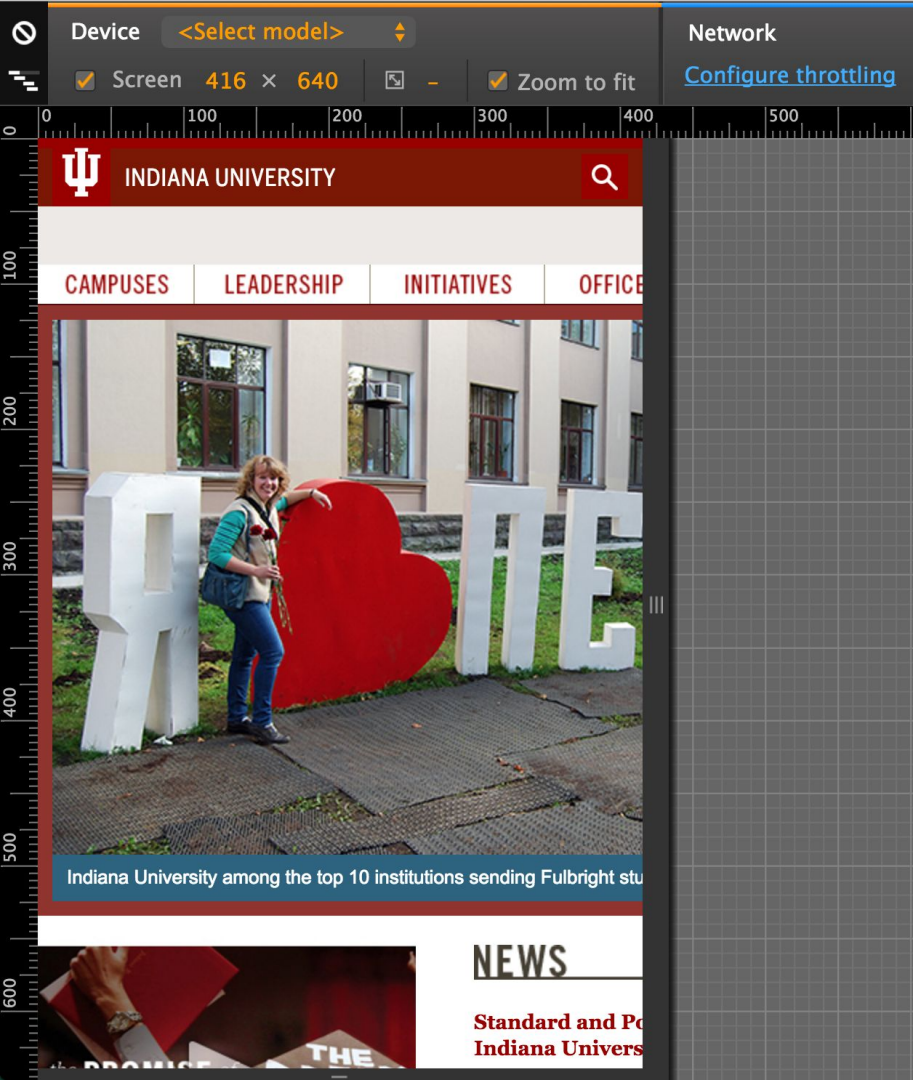
FOUNDATION MEDIA QUERIES

```
@media screen and (max-width:40em) { … }

@media screen and (min-width: 40.063em)
  and (max-width: 64em) { … }

@media screen and (min-width: 64.063em)
  and (max-width: 120em) { … }

@media screen and (min-width: 120.063em) { …
}
```



Screen size emulation tools

Chrome Device Mode

Chrome > Developer tools > Enable emulation

https://developer.chrome.com/devtools/docs/device-mode

Chrome Extensions

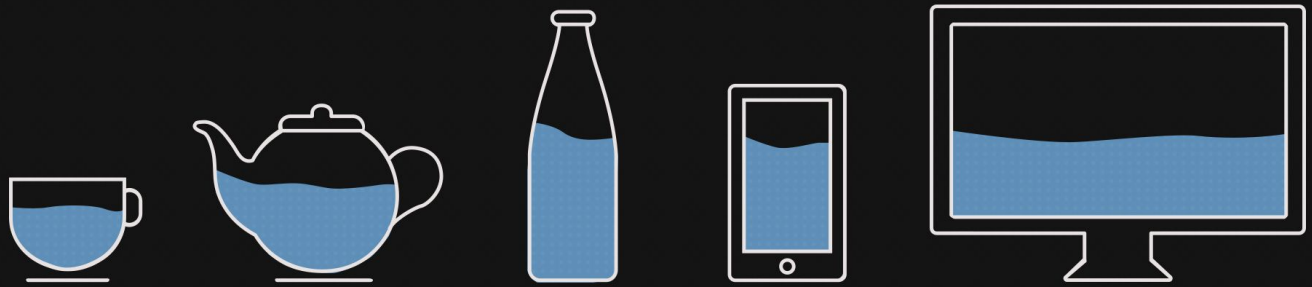
Viewport Resizer Plugin for Chrome

- Search for “chrome viewport resizer”

Online tools

http://design.google.com/resizer/#

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

HOMEWORK

https://alistapart.com/article/frameworks

This article and the terminology and concepts from today -- along with the Krug text -- will be part of your last quiz in I399, coming up the week after Spring Break.