CS3226

WEB PROGRAMMING AND APPLICATIONS

LECTURE 06 - FRONT-END (BOOTSTRAP)

DR STEVEN HALIM

STEVENHALIM@GMAIL.COM

For 'not newbies': Try this Bootstrap quiz @ W3Schools first

OUTLINE

MOTIVATING EXAMPLES

Review: CSS Zen Garden (from CSS lecture)

Now, presenting AWWWARDS

Knowing HTML5+CSS3+JS are not enough...
Good web design is an art and not purely science

Not-so-good Fact: Most (but not all) of us in Computer *Science* are technical people with "not-so-good design skill"

Can we do better?

FRONT-END (CLIENT-SIDE) DESIGN

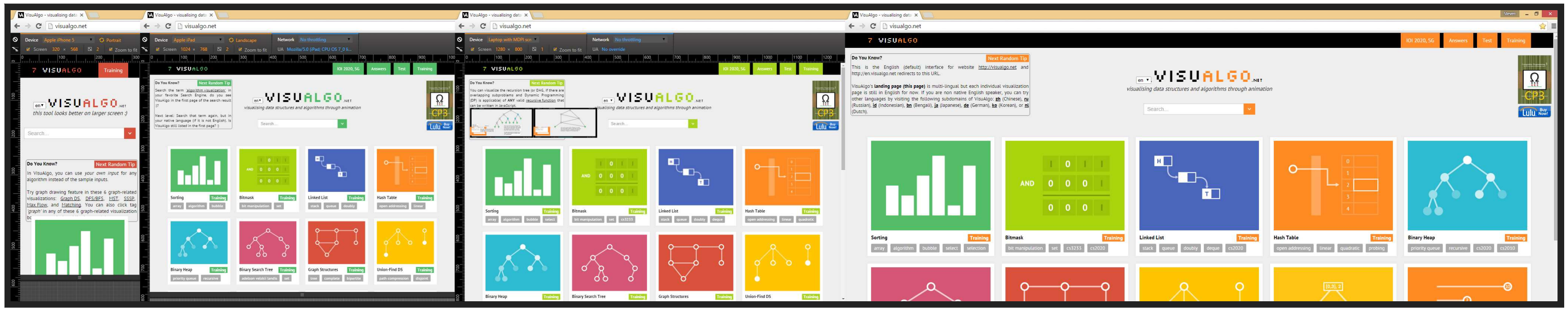
To reach good website design, you need to ~~know~~master the language of the web and beyond...

What is shown to the *first-time* visitor of your web application is your front-end/client-side interface: it can be a make-or-break moment

SCREEN RESOLUTION

But... we have a BIG (or actually SMALL) problem...

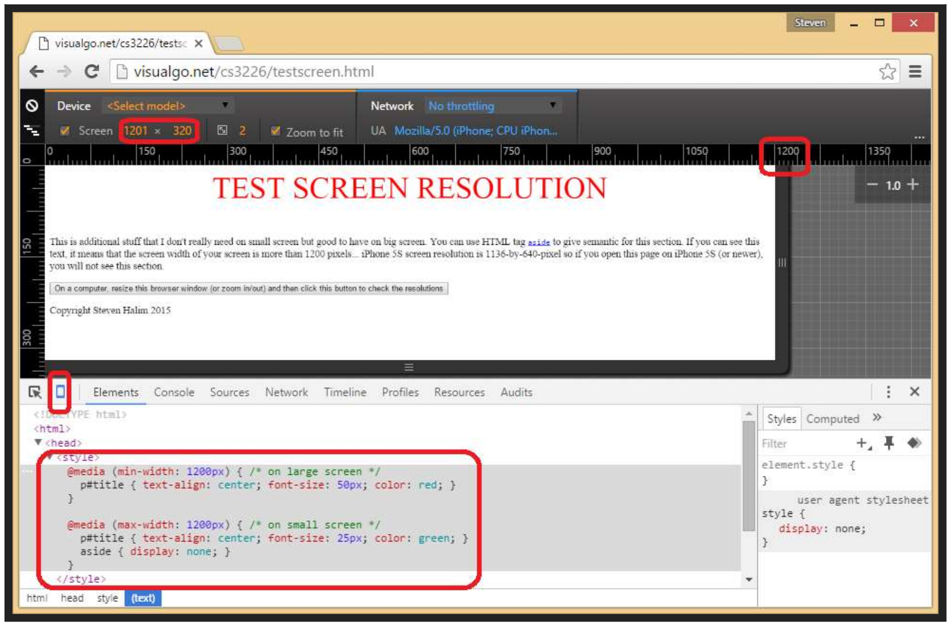
Not all visitors of your website (web application) have (viewing) device(s) with the same screen resolution as yours, thus we need a good "screen real estate" management



GOOGLE CHROME "DEVICE MODE"

We can use Google Chrome "device mode" to (virtually*) check how our website looks like on <u>different</u> devices

Check our test HTML


visualgo.net/cs3226/testsc
visualgo.net/cs3226/testscreen.html
Steven
Device <Select model>
Network No throttling
Screen 1201 × 320
2
Zoom to fit
UA Mozilla/5.0 (iPhone; CPU iPhon...
1.0
Elements Console Sources Network Timeline Profiles Resources Audits
Styles Computed
Filter
html head style (text)

TEST SCREEN RESOLUTION

This is additional stuff that I don't really need on small screen but good to have on big screen. You can use HTML tag aside to give semantic for this section. If you can see this text, it means that the screen width of your screen is more than 1200 pixels... iPhone 5S screen resolution is 1136-by-640-pixel so if you open this page on iPhone 5S (or newer), you will not see this section.

On a computer, resize this browser window (or zoom in/out) and then click this button to check the resolutions

Copyright Steven Halim 2015

```
<!DOCTYPE html>
<html>
▼<head>
 ▼<style>
    @media (min-width: 1200px) { /* on large screen */
      p#title { text-align: center; font-size: 50px; color: red; }
    }

    @media (max-width: 1200px) { /* on small screen */
      p#title { text-align: center; font-size: 25px; color: green; }
      aside { display: none; }
    }
  </style>
```

```
element.style {
}
      user agent stylesheet
style {
  display: none;
}
```

SIMPLISTIC WAY: FIXED LAYOUT

What is a 'fixed' layout and what are the pros/cons of it?

Answer

CHECKING SCREEN RESOLUTION

How can our webpage (web application) check the screen resolution and react accordingly so that our visitor can view our webpage better in his/her device?

#1. CSS3 WAY (CSS MEDIA QUERY)

Apply different CSS rules on different media
(that has different screen resolution/width)

```
/* In testscreen.html, we have these two rules */
/* if large screen (min-width: 1200px), apply these rules */
@media (min-width: 1200px) {
  p#title { text-align: center; font-size: 50px; color: red; } }
/* if small screen (max-width: 1200px), apply these rules instead */
@media (max-width: 1200px) {
  p#title { text-align: center; font-size: 25px; color: green; }
  aside { display: none; } }
```

Easy to use for simple applications,
but difficult for more complex ones

#2. JS (JQUERY) WAY

Check `screen.width`, `$(window).width()`, or `$(document).width()` and manipulate the DOM programmatically

```
// example in testscreen.html
if ($(document).width() < 1200) // controlled with JS
  $('footer').hide();
else
  $('footer').show();
```

This JS way can be more flexible than CSS media query way as JS is much more expressive than the simple (implicit) if-else statement of CSS media query

#3. ANY OTHER SIMPLER WAY(S)?

Read on...

RESPONSIVE WEB DESIGN

The term *responsive** is given to a webpage that can adapt it's content when loaded on different devices

1. Large desktops (e.g. 1920x1080),
2. Portable laptops (e.g. 1280x800),
3. Mobile tablets (e.g. 1024x768/iPad2),
4. Smartphones (e.g. 1136x640/iPhone5)

Let's explore several examples of responsive webpages

DESIGN PHILOSOPHY

It is easier to go bigger if we start small (Mobile-First Design Philosophy)

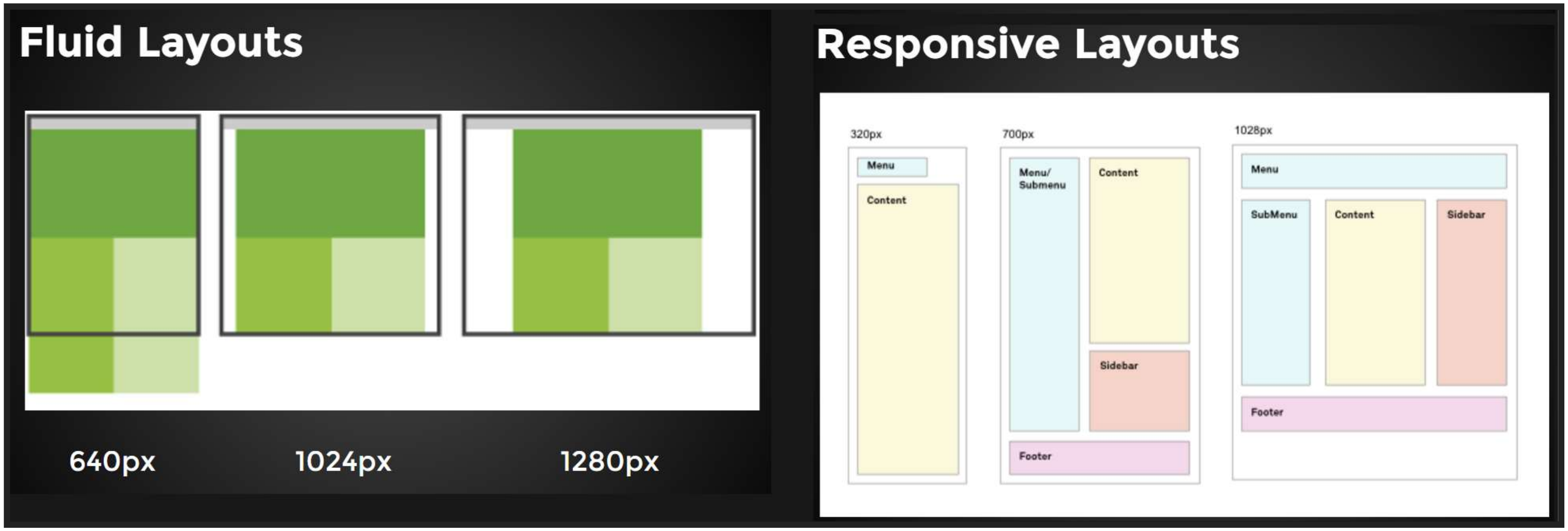
It is usually harder to do the other direction

But again it is all depends on the purpose of the web application, e.g. for VisuAlgo visualization pages, we decide not to support visualization on extremely small screen

KEY IDEAS

Some key ideas for Responsive Web Design are:

1. All sizes are in **percentages**, not in absolute numbers
2. Elements are categorized based on importance and can be removed/hidden as needed



BOOTSTRAP

It is <u>not</u> easy to create websites (web applications) that are responsive from scratch, e.g. (the front page of) VisuAlgo, but we can use some help :)

Introducing Bootstrap...

BOOTSTRAP BACKGROUND

Started by two Twitter engineers back in 2011 and have since been massively developed by various contributors

Their Tagline:

"Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web"

BOOTSTRAP IS A FRAMEWORK

Back in the Introduction lecture, we discussed the pros and the cons of using templates (frameworks)

However, using Bootstrap is more akin to using jQuery (on top of vanilla JavaScript) rather than using pure website templates, so I recommend it's usage to speed up the development of the front-end of your web applications

BOOTSTRAP VS COMPETITORS

There are other competing frameworks/libraries out there

We are using one of the most popular framework in CS3226

BOOTSTRAP VS SELF-MADE DESIGN

Once you are (much) more familiar with website design, you may start seeing the (current) limits of Bootstrap (e.g. your website design look 'too similar' too many other website that uses Bootstrap too) and will want to design your own (or contribute to this big project)

BOOTSTRAP - SETUP

Let's start by downloading the latest version
(and put them in our local copy, open source)

Or alternatively, we can just use this CDN* links:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstra
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstra
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/boots
```

BOOTSTRAP - FIRST EXAMPLE

The first simple example on using Bootstrap
(to transform your frontend from "plain" to "cool")

Open multtable.html, the solution for this JavaScript challenge — "a standard HTML5 table with basic styling"

What's Next?

Answer

BOOTSTRAP - GRID SYSTEM (1)

This is one of the most important feature of Bootstrap

Basically, everything is inside a `<div>` container which can be fixed (default) or fluid (fill the width of the browser)

Inside container, we have one (or many) `<div>` rows
Each row is divided into 12 equal sized columns

BOOTSTRAP - GRID SYSTEM (2)

For each row, we decide how many columns a certain `<div>` —that contains the actual HTML element(s) content— has on different screen resolution by adding class(es):

1. `.col-xs-*` (very small)
2. `.col-sm-*` (small devices)
3. `.col-md-*` (medium devices)
4. `.col-lg-*` (large devices)

BOOTSTRAP - GRID SYSTEM (3)

We can give empty column offsets on different screen resolution using `.col-xs/sm/md/lg-offset-*`

We can also choose to hide a `<div>` on different screen resolution using `.hidden-xs/sm/md/lg`

See gridsystem.html example to get a clearer idea
All `<div>`s are highlighted with red border for clarity

BOOTSTRAP TOUR (1)

Using CS3226 & CS3233 Private IVLE

First, Bootstrap CSS classes:

1. Grid system, revisited (more varieties)
2. Tables, revisited (more classes)
3. Buttons

BOOTSTRAP TOUR (2)

Next, Bootstrap components:

4. Badges
5. Navs
6. Navbar
7. Alerts

BOOTSTRAP TOUR (3)

Next, Bootstrap JavaScripts:

8. Modal Dialog Box
9. ScrollSpy (<nav> + JavaScript)
10. ToolTips (attribute + JavaScript)
11. Alert (class + JavaScript)

NOTES

For much more examples and detailed documentations, just visit Bootstrap website directly

Or read this freemium Bootstrap tutorial from udemy

OTHER DESIGN ASPECTS

Many other aspects have not been discussed, e.g.

1. Choosing Color (scheme)
2. Typography (font)
3. Mouse Hover issue on Mobile
4. Interaction Design, User eXperience/UX (CS3240)
5. Supporting older web browsers?
6. Etc...

ART VS SCIENCE

Again, good web design is (very) important for a successful web application, but this skill is an art and is hard to master

THE END

If you haven't do it,
try this Bootstrap quiz @ W3Schools for self check

Chinese
New Year
2016
猴
Year of the Monkey
猴
designed by freepik.com