CSCI 4140 – Tutorial 6

Responsive Web Design with Bootstrap 3

Matt YIU, Man Tung (mtyiu@cse)
SHB 118
Office Hour: Wednesday, 3-5 pm

2016.03.03

Outline

- Mobile web development
- Font Awesome
- Responsive web design (RWD)
- CSS media queries
- Bootstrap 3
- jQuery

Mobile web development

- In Assignment 2, we will build a "mobile web app" called YouTube Remote *(more details in the next tutorial)*
- What will you learn in the coming weeks?
  - **Front-end development:** Responsive web design (RWD) with Bootstrap, jQuery, YouTube Iframe API, React.js, Backbone.js, …
  - **Back-end development:** Node.js, Express.js, WebSocket, Socket.IO, RESTful APIs, MongoDB, ...
  - **Deployment:** Heroku
  - As a matter of fact, we are using the **MEAN stack** (though AngularJS is replaced by React.js / Backbone.js)

Mobile web development

- Differences for designing for mobile devices:
  - Working with **small screens**
    - Solved by **responsive web design**
  - Working with **touch screens**
    - Solved by using **DOM Touch events** (https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Touch_events)
  - Optimizing **images**
    - Also solved by **responsive web design**
    - Use images with retina screen support (keyword: **@2x**)
  - **Mobile APIs**
    - Introduced by **HTML5**
    - New possibilities offered by mobile devices, such as **orientation** and **geolocation**

Ref.: https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile

Font Awesome

It provides a comprehensive set of icons with the power of CSS!

What is Font Awesome?

- What is the format of this icon?
  - Old fashioned answer: GIF, PNG, JPG, SWF (!?)…
  - Modern answer: This is a **font**!

```
.fa {                                   font-awesome.css:14
  display: inline-block;
  font: ▶normal normal normal 14px/1
      FontAwesome;
  font-size: inherit;
  text-rendering: auto;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  transform: translate(0, 0);
}
```

```
.fa-pied-piper-alt:before   font-awesome.css:1376
{
  content: "\f1a8";
}
```

What is Font Awesome?

- It provides **500+ vector icons** for **FREE**
- Prefect on **retina** screens
- Pure CSS – no JavaScript involved
- Extremely easy to use!
  - The simplest way to use it is to import the CSS from BootstrapCDN (CDN stands for **content delivery network**)

```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css">
```

You may need to insert "http:" here if you are testing your site locally.

Font Awesome: Example

```
<html>
<head>
    <title>CSS Example: Font Awesome</title>
    <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/font-
awesome/4.5.0/css/font-awesome.min.css">
    <style> h3 { color: red; } </style>
</head>
<body>
    <h1><i class="fa fa-smile-o"></i> Hello
World!</h1>
    <h2><i class="fa fa-exclamation-circle"></i>
Different size...</h2>
    <h3><i class="fa fa-thumbs-up"></i> Different
color...</h3>
</body>
</html>
```

css/fa.html

Font Awesome: How to use?

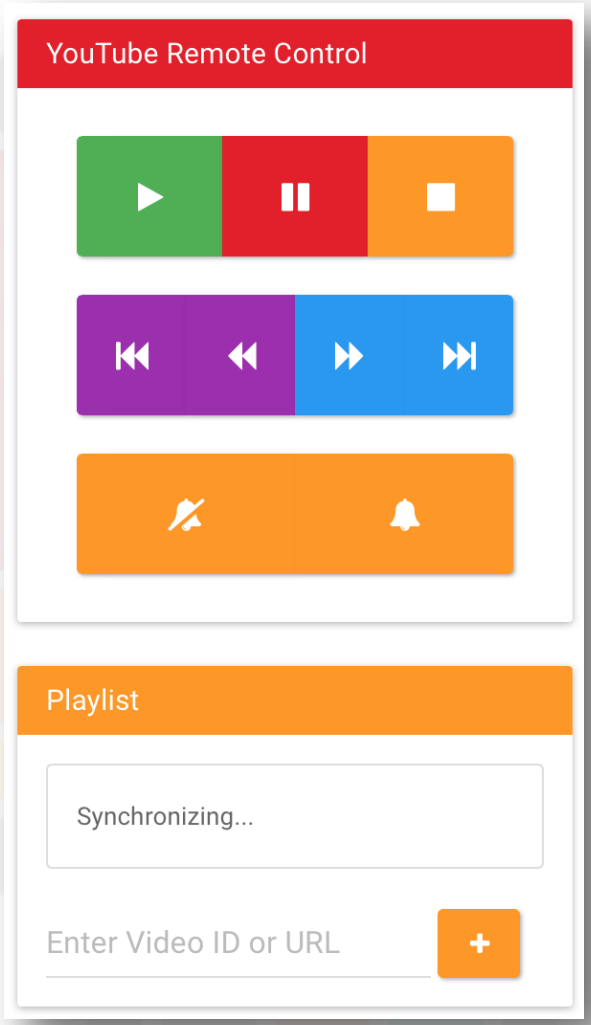
- From the example, you just need to insert the following code at the desired position:

```
<i class="fa fa-[icon name]"></i>
```

  - The CSS will insert the icon for us!
- Where can we check the icon's name?
  - http://fortawesome.github.io/Font-Awesome/icons/

Font Awesome

- You can use this library in Assignment 2
  - And I used a lot
- There are also some useful classes in the library
  - Read http://fortawesome.github.io/Font-Awesome/examples/ for more examples
  - It can even produce animations with CSS



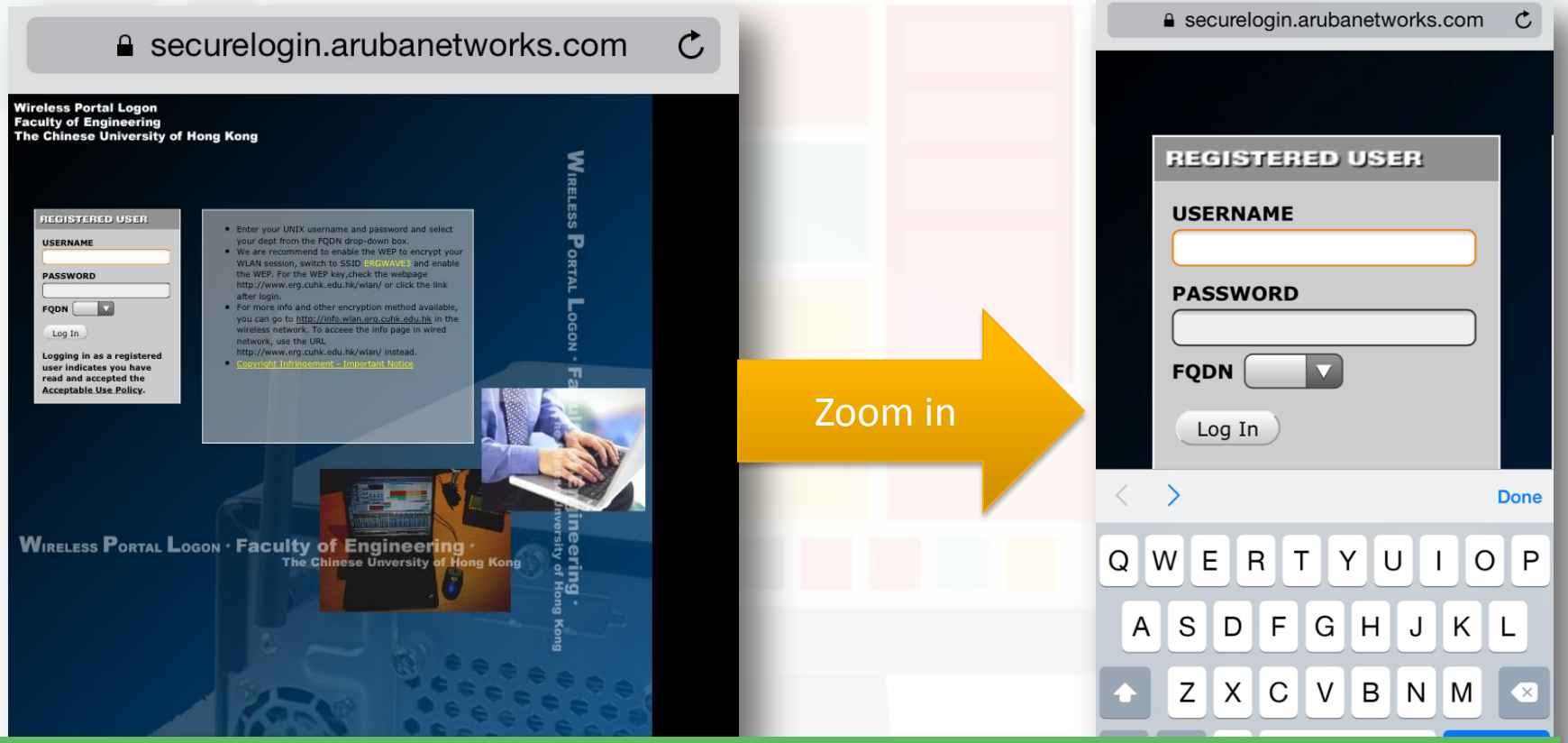
Responsive web design (RWD)

Believe me, you are already viewing it everyday..

Responsive web design: Introduction

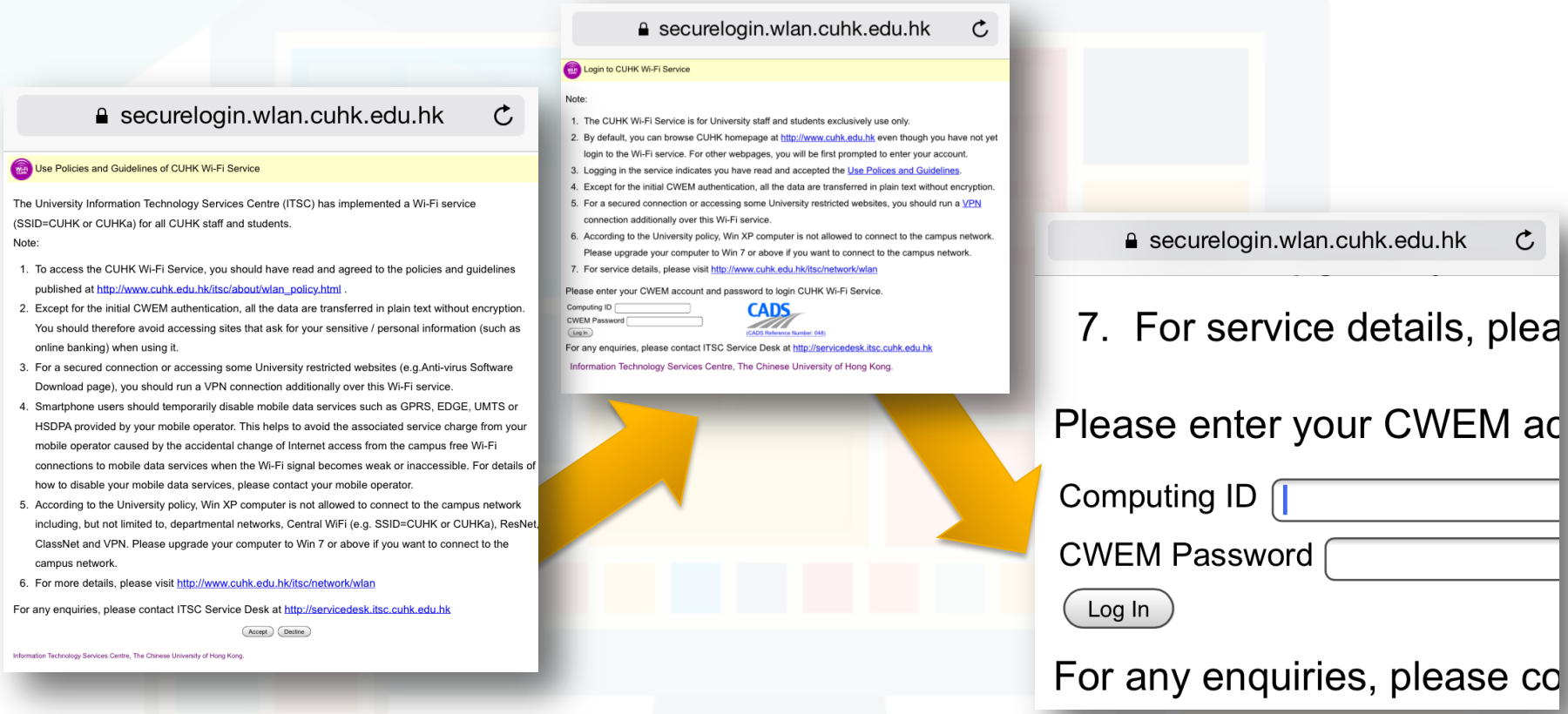
- How many “smart” devices do you have?
  - Smartphones, tablets, notebook, …
  - Designing web pages to display in these devices without degrading **user experiences** (UX) are extremely challenging!
    - Different screen sizes, resolutions, pixel per inch (ppi), browsers, …
    - It is not feasible to build and maintain multiple versions of the same page for each devices!
- Some sites just give up…
  - Let’s see some examples in CUHK ;-)

Bad Example: ERGWAVE Portal Login



You need to zoom in and scroll to login as the site is designed for desktop *only*…

Bad Example: CUHK Wi-Fi Service Login



CUHK Wi-Fi Service Login page is even more terrible in terms of user experience…

Responsive web design: Introduction

- To solve this problem, an approach called "**responsive web design**" (RWD) is proposed
  - A combination of **fluid, proportion-based grids**, **flexible images**, and **CSS3 media queries**, an extension of the **@media**…(from Wikipedia)
- Examples:
  - Course homepage: http://tywong-mole.com/~csci4140/
  - Tutorial resource page: http://mtyiu.github.io/csci4140-spring16/
  - CUHK: http://www.cuhk.edu.hk/english/index.html
  - Harvard University: http://www.harvard.edu
  - Kickstarter: https://www.kickstarter.com

CSS media queries

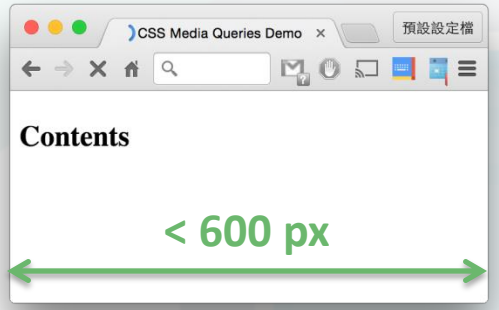
An essential ingredient of responsive web design…

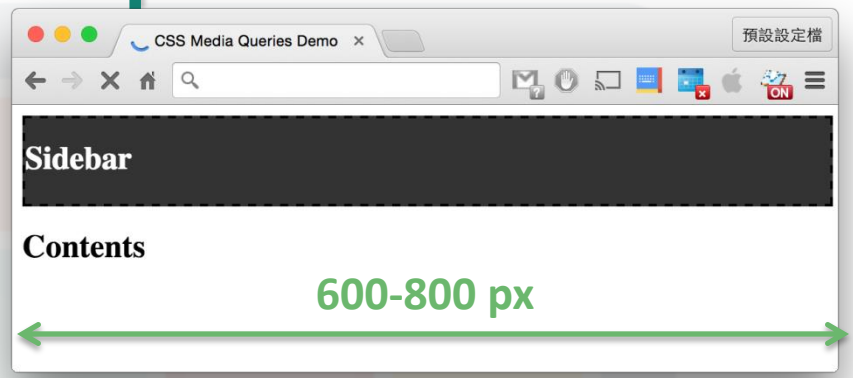
CSS media queries

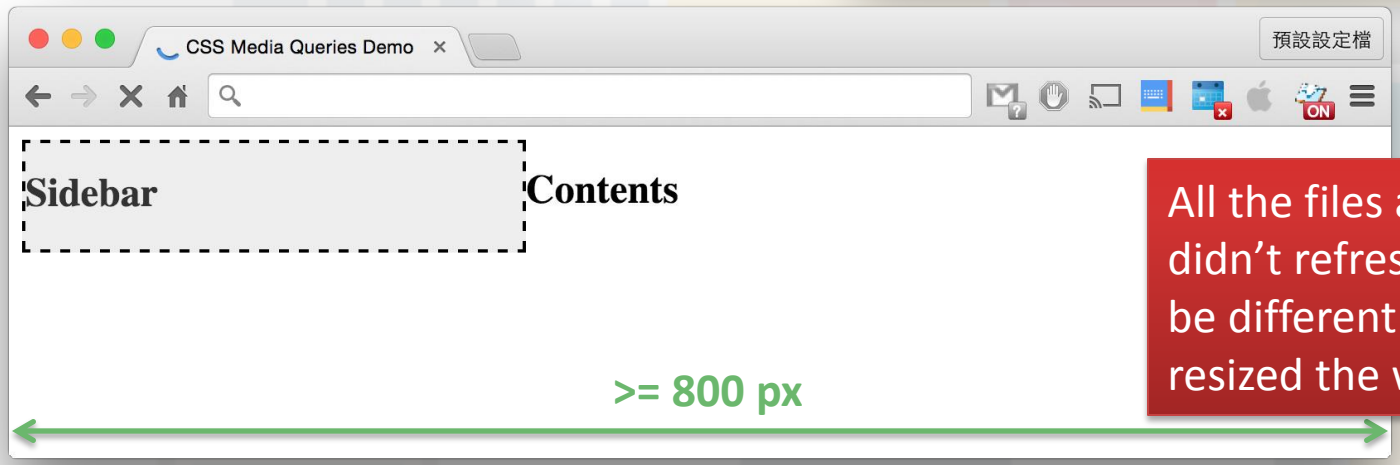
- Introduced in CSS3
- Consists of
  - A **media type**; and
  - At least one **expression** that limits the style sheets' **scope** by using **media features**, such as width, height, and color
- Media queries let the **presentation** of content be tailored to a specific range of output devices without having to change the content itself
- You need to specify the **viewport** (which controls how a webpage is displayed on a mobile device) in a **meta tag**:

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

CSS media queries: Example







All the files are the same, and I didn't refresh! Why there can be different layout when I resized the window?

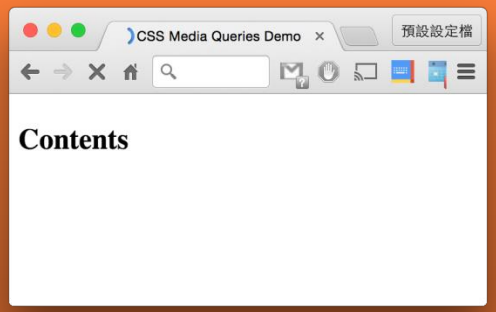
`media-query/sidebar.html & media-query/sidebar.css`

CSS media queries: Example

```
.sidebar { display: none; }
.contents { display: inline; float: l
@media screen and (min-width: 500px)
    .sidebar {
        display: inline; float: left;
        width: 100%; background-color
    }
}
@media screen and (min-width: 600px) {
    .sidebar {
        display: block; width: 300px;
        background-color: #eee; color: #333;
    }
}
```

media-query/sidebar.css

By default, the sidebar is hidden.



CSS media queries: Example

```
.sidebar { display: none; }
.contents { display: inline; float: lef
@media screen and (min-width: 500px) {
    .sidebar {
        display: inline; float: left; b
        width: 100%; background-color:
    }
}
@media screen and (min-width: 600px) {
    .sidebar {
        display: block; width: 300px;
        background-color: #eee; color:
    }
}
```

media-query/sidebar.css

When this criterion is satisfied, apply this set of styles!

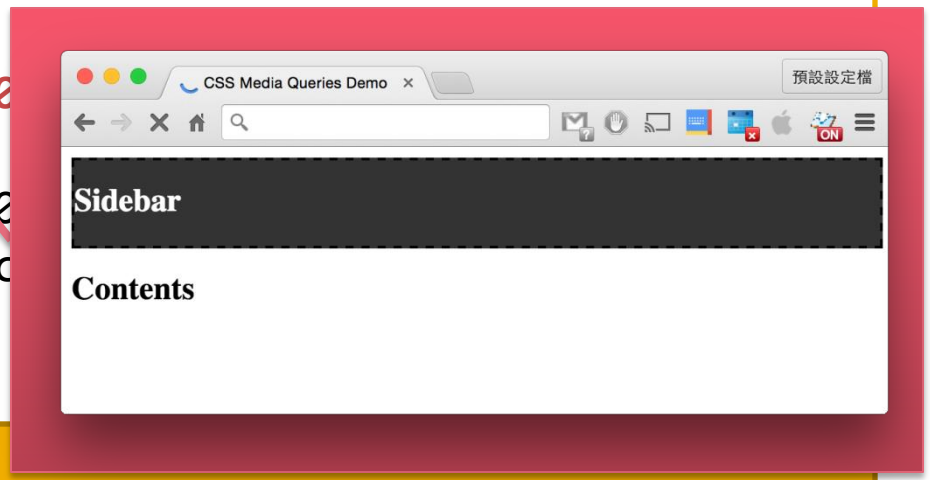
- `@media` works like an "`if`"
- `screen`: The page is viewed with a screen
- `min-width`: The minimum width of the rendering surface of the output device

The whole statement means: "if the media is at a screen and the width is at least 500px"… This is called a **media query**.

CSS media queries: Example

```
.sidebar { display: none; }
.contents { display: inline; float: left; }
@media screen and (min-width: 500px) {
    .sidebar {
        display: inline; float: left; border: 2px dashed #000;
        width: 100%; background-color: #333; color: #fff;
    }
}
@media screen and (min-width: 600
    .sidebar {
        display: block; width: 3
        background-color: #eee; c
    }
}
```

media-query/sidebar.css



CSS media queries: Example

```
.sidebar { display: none;
.contents { display: inlin
@media screen and (min-wid
    .sidebar {
        display: inline; f
        width: 100%; backg
    }
}
@media screen and (min-width: 600px) {
    .sidebar {
        display: block; width: 300px;
        background-color: #eee; color: #333;
    }
}
```

media-query/sidebar.css

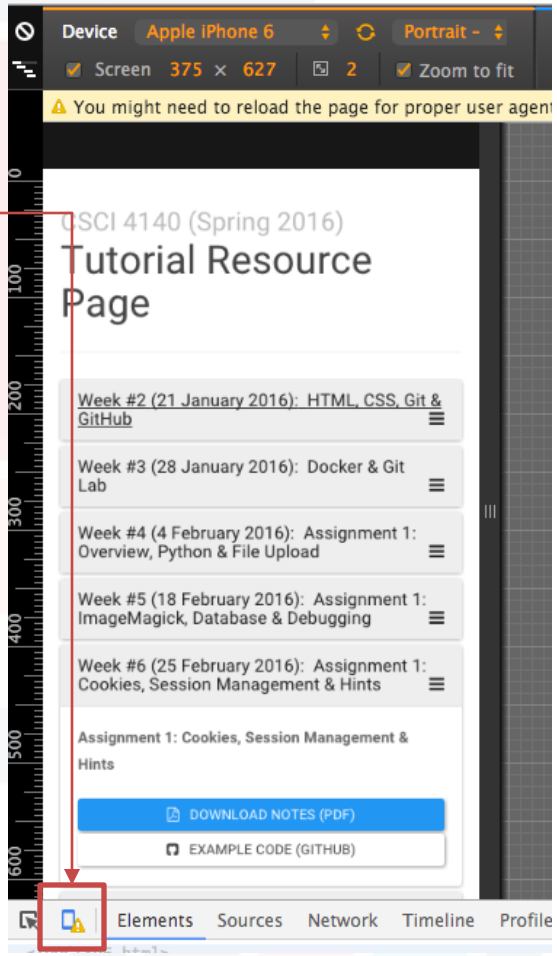
This media query checks “if the media is at a screen and the width is at least 600px”…

CSS media queries: Another example

- Read the code yourself:
  - `media-query/demo.css` & `media-query/demo.html`
- There are a lot more to explore!
  - You can learn more yourself (I don't have time to cover more)…
    - Reference: https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Media_queries

CSS media queries: Debugging

- Chrome has a useful tool for debugging responsive layout:
  - **Developer Tools** > Click this button
- Now let's move to **Bootstrap** – a convenient framework for developing **responsive web pages**



Bootstrap 3

A framework for developing responsive, mobile first web projects

Bootstrap: Introduction

- A **front-end HTML + CSS + JS framework** developed by **Twitter**
- Since version 3.0, it adopted a **mobile first** design philosophy, emphasizing **responsive design** by default
  - Of course, it uses **CSS media queries**
- Also include a collection of JavaScript plugins
- Examples:
  - Bootstrap: http://getbootstrap.com
  - Font Awesome: http://fortawesome.github.io/Font-Awesome/

Bootstrap: How to use?

- Again, there is a Bootstrap CDN:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">

<!-- Optional theme -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-
theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"
></script>
```

- You can also download the archive for compiled and minified CSS, JavaScript, and fonts here *(check my example code)*

Bootstrap: Overview

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <title></title>
        <link href="css/bootstrap.min.css" rel="stylesheet">
    </head>
    <body>
        <div class="container-fluid">
        </div>
    </body>
</html>
```

bootstrap/template.html

Bootstrap: Overview

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <title></title>
        <link href="css/boots
    </head>
    <body>
        <div class="container-fluid">
        </div>
    </body>
</html>
```

bootstrap/template.html

Bootstrap makes use of certain HTML elements and CSS properties that require the use of the HTML5 doctype.

To ensure proper rendering and touch zooming, add the viewport meta tag to your <head>.

Bootstrap: Overview

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Comp
        <meta name="viewport" conte
initial-scale=1">
        <title></title>
        <link href="css/bootstrap.min.css" rel="stylesheet">
    </head>
    <body>
        <div class="container-fluid">
        </div>
    </body>
</html>
```

bootstrap/template.ht

I only include the CSS of Bootstrap. For the JavaScript part, please study yourself.
By the way, you are not allowed to use any external JavaScript library except Socket.IO in your assignments.

Bootstrap requires a containing element to wrap site contents and house our grid system.

- `.container`: A responsive fixed width container
- `.container-fluid`: Full width container, spanning the entire width of your viewport

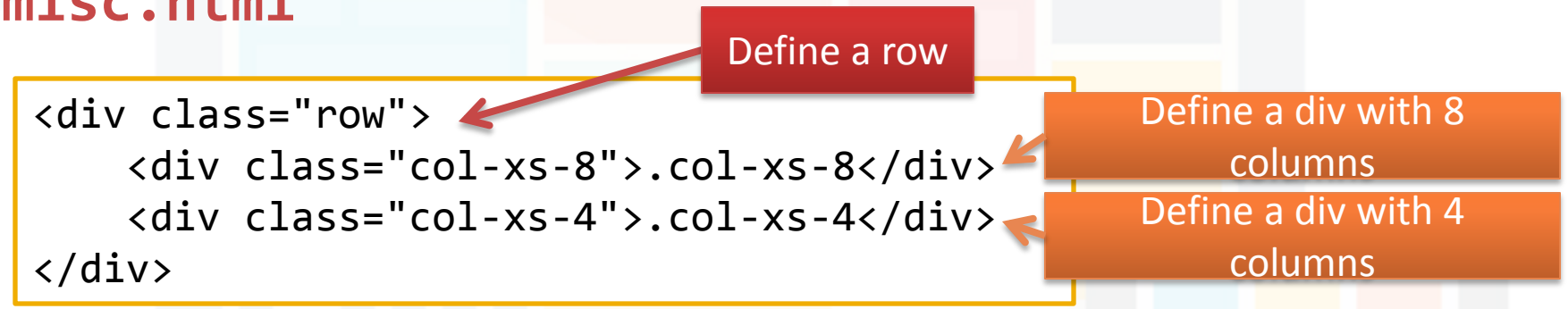
(See `bootstrap/container.html` in example code)

Bootstrap: Grid system

- A responsive, mobile first **fluid grid system** that appropriately scales up to 12 columns as the **device** or **viewport size** changes
  - The page is divided into **12 columns** and an arbitrary number of **rows**
  - We place our contents into a certain number of columns
  - Bootstrap CSS will **scale** the size of the contents or **stack** them according to the screen width
- Bootstrap defines 4 device sizes:
  - `xs`: Extra small devices, e.g., phones (< 768 px)
  - `sm`: Small devices, e.g., tablets(>= 768 px)
  - `md`: Medium devices, e.g., desktops (>= 992 px)
  - `lg`: Large devices, e.g., desktops (>= 1200 px)

Bootstrap: Grid system

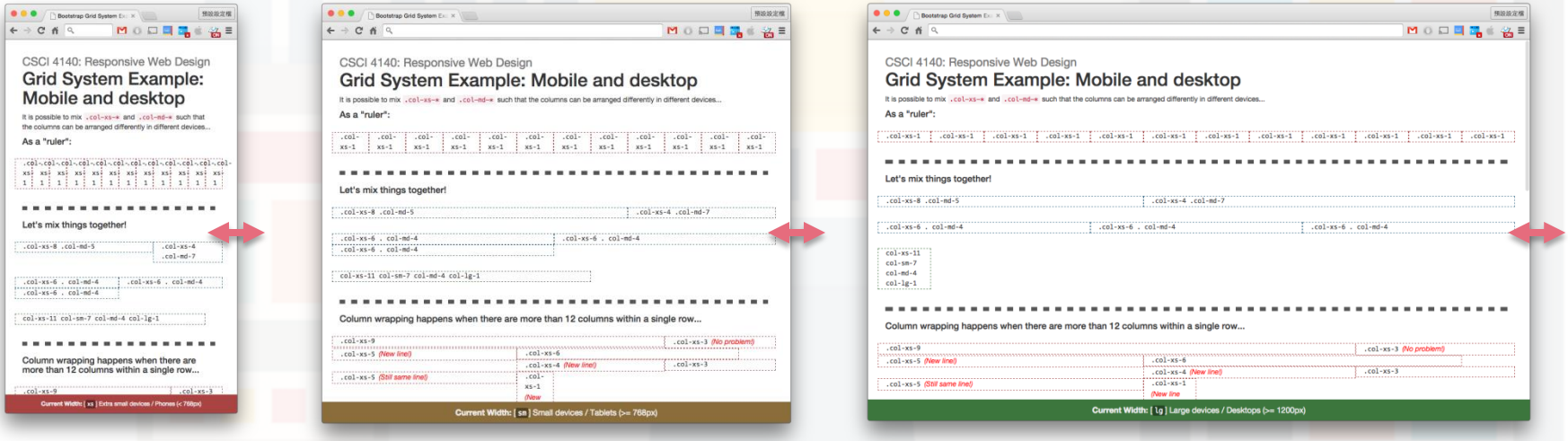
- Examples: **bootstrap/grid.html** and **bootstrap/grid-misc.html**



  - Study the code and layout with Chrome Developer Tools
- Differences between **.col-xs-***, **.col-sm-***, **.col-md-*** and **.col-lg-***
  - When you define a row using, for example, **md** but the current screen width is within the range of **xs** or **sm**, the columns will be **stacked**; otherwise it is horizontal

Bootstrap: Grid system

- Try to **resize** the browser window to see how the columns are arranged
  - Most examples come from the official documentation, which is great: http://getbootstrap.com/css/
  - I developed a simple JavaScript plugin to tell you the current width



Bootstrap: Grid system

- Another example: Responsive ERGWAVE Portal Login
  - This example uses the grid system to make the page responsive
  - Original: **`ergwave/login.html`** & **`ergwave/styles.css`**
  - Responsive version: **`ergwave/login_responsive.html`** & **`ergwave/styles_responsive.css`**
- There are a lot of great things in Bootstrap
  - Explore yourself! The documentation is fantastic!
  - http://getbootstrap.com/css/ & http://getbootstrap.com/components/
  - Feel free to use them in Assignment 2

Bootstrap: Responsive utilities

- In Assignment 2, you need to hide some elements on the page when the screen width changes
- Bootstrap provides utility classes for displaying / hiding content in response to the width
- You will find the following classes useful:
  - `.visible-xs-*`, `.visible-sm-*`, `.visible-md-*`, `.visible-lg-*`, `.hidden-xs`, `.hidden-sm`, `.hidden-md`, `.hidden-lg`
- Read http://getbootstrap.com/css/#responsive-utilities
  - Apply the classes to a `<div>` and resize the window to see the effect

Bootstrap: Bonus materials

- For those who are not satisfied with the default layout of Bootstrap: http://bootswatch.com
  - You only need to change the `bootstrap.min.css`
- Another alternative to Bootstrap – ZURB Foundation
  - http://foundation.zurb.com
  - The grid system is similar to that of Bootstrap
  - Example:
    - My homepage (http://www.cse.cuhk.edu.hk/~mtyiu/)
    - UHS *(out of my expectation!)*: http://www1.uhs.cuhk.edu.hk/
- If you are a fan of CSS animations:
  - http://daneden.github.io/animate.css/

I can't write better than the documentation (http://api.jquery.com/)…

Come to the tutorial to see live demo of using jQuery!

Reminder

- We will use **Node.js** and **Express** to implement the backend of Assignment 2
  - They allow us to use JavaScript to implement **server-side programs**
- More details will be given in later tutorials
- For now, please install them on your computer first
  - You can either install them on your computer; or
  - Use **Docker** again!
- Also, please register an account on Heroku (https://www.heroku.com/)

– End –