CSCI 4140 – Tutorial 5

Responsive Web Design with Bootstrap 3

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

2015.02.12

Outline

- Assignment 2 overview – Layout design
- CSS (Cascading Style Sheets)
- Font Awesome
- Responsive web design (RWD)
- CSS media queries
- Bootstrap 3
- Setting up the development environment for Node.js and Express

Assignment 2 Overview – Layout Design

If you know how to implement the layout, you are free to leave!

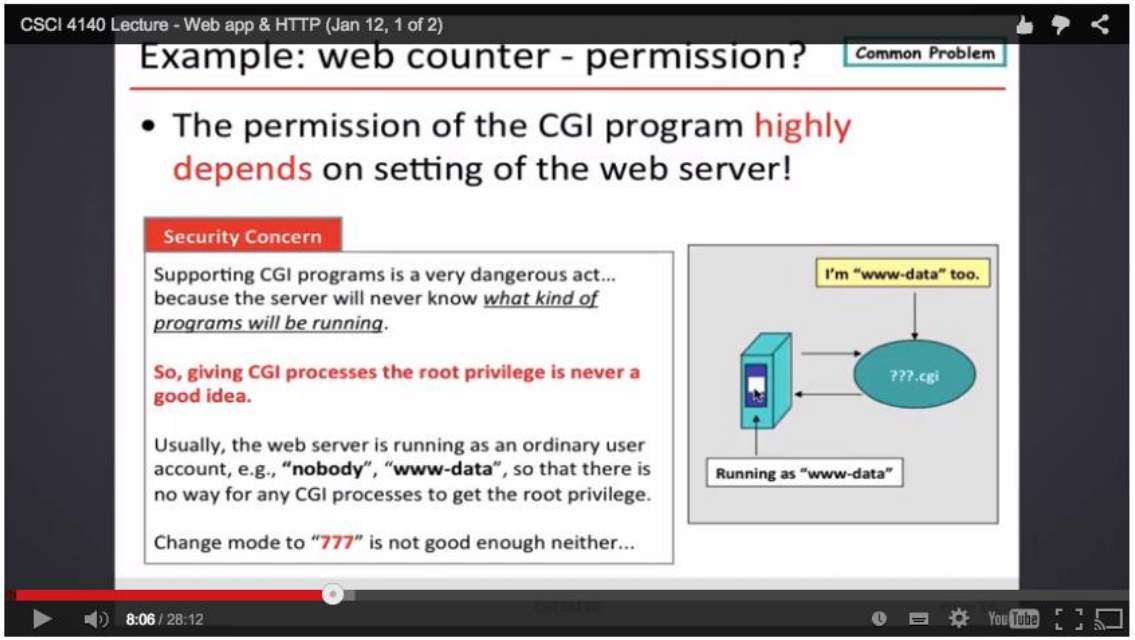
When screen size ≥ 992 px…

CSCI 4140: Assignment 2

youtube-mtyiu.rhcloud.com/session/2670

YouTube Remote (Desktop)

YouTube Player — *Last Command : -*

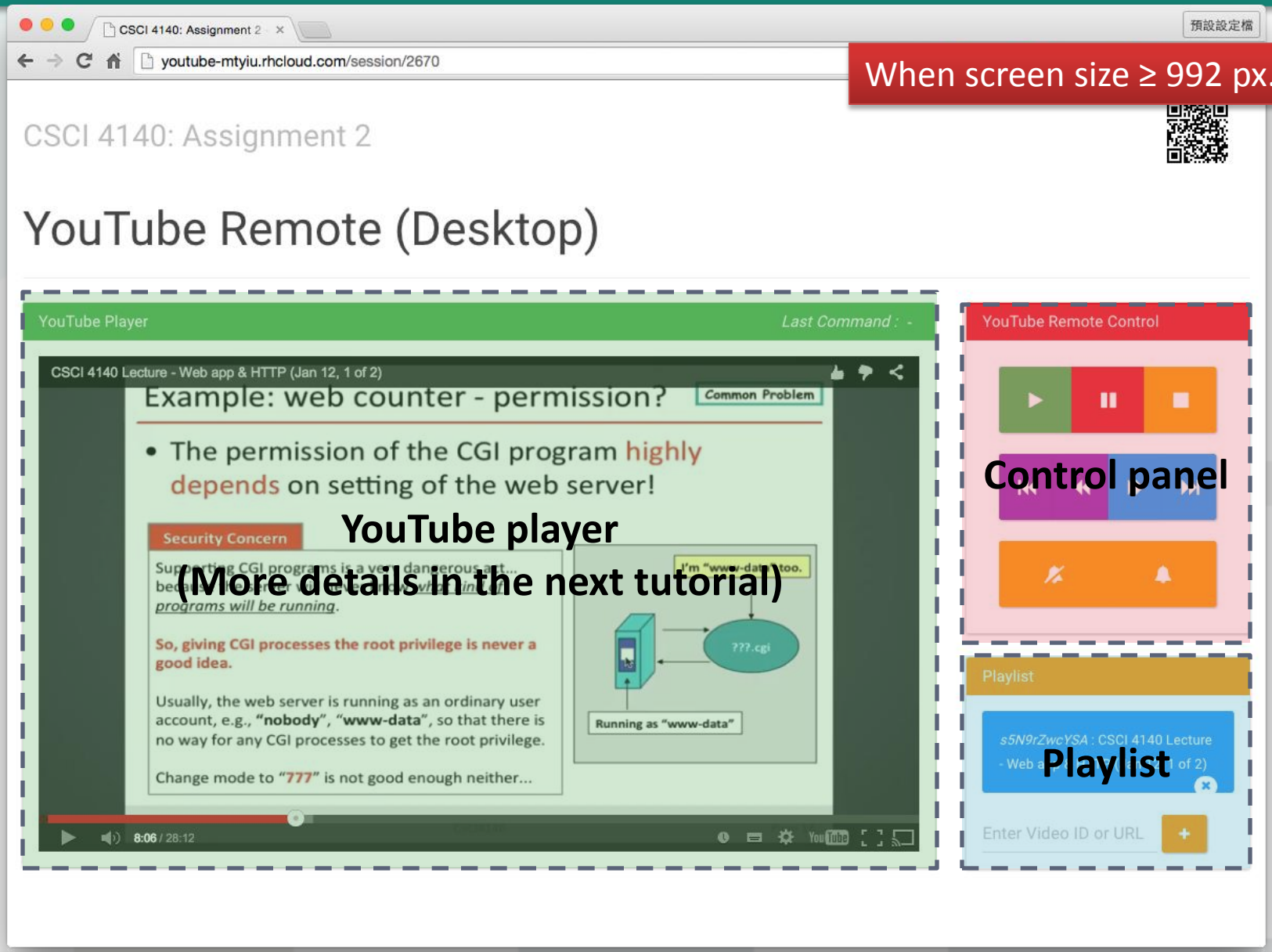


YouTube Remote Control

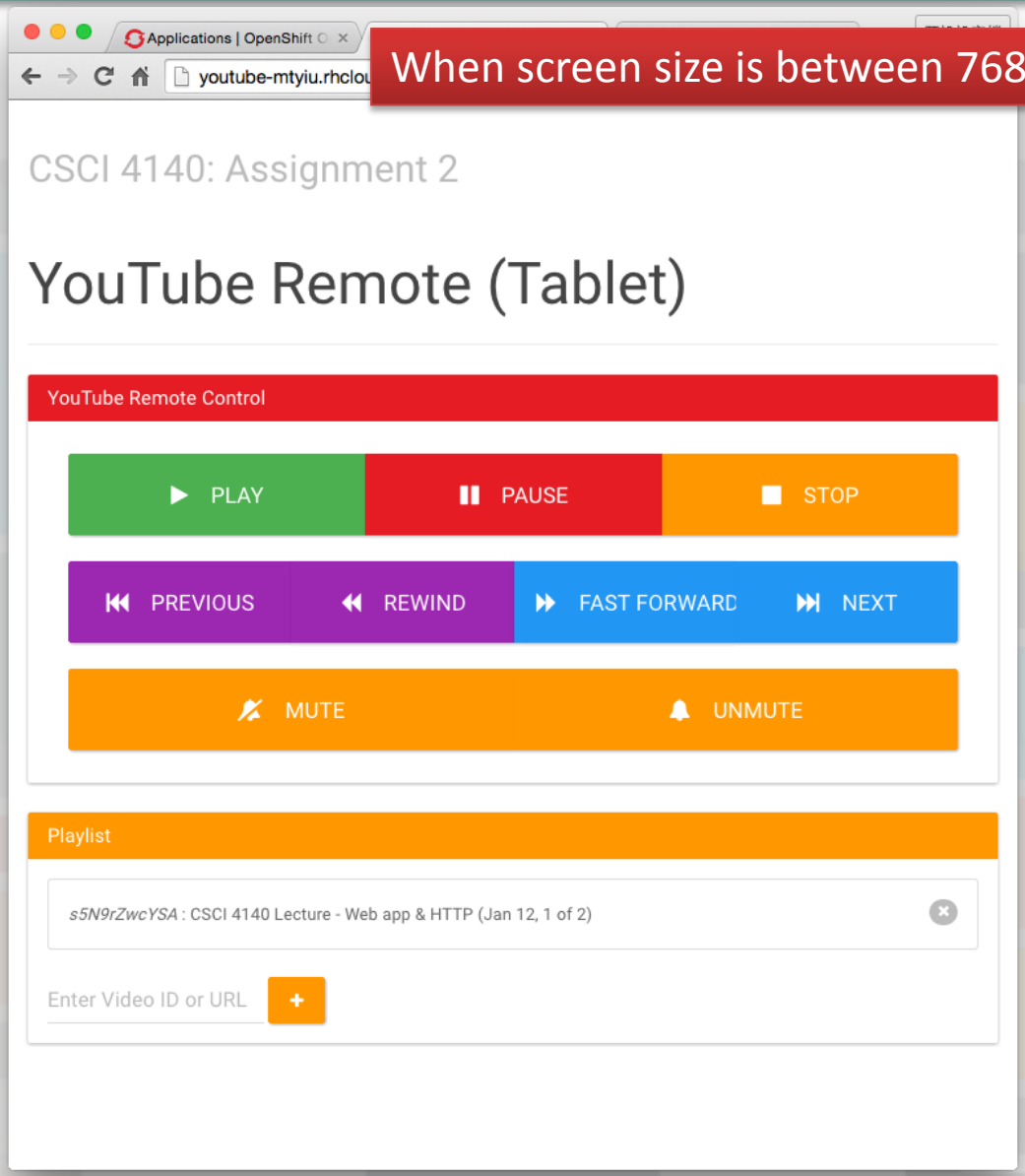
Playlist

s5N9rZwcYSA : CSCI 4140 Lecture - Web app & HTTP (Jan 12, 1 of 2)

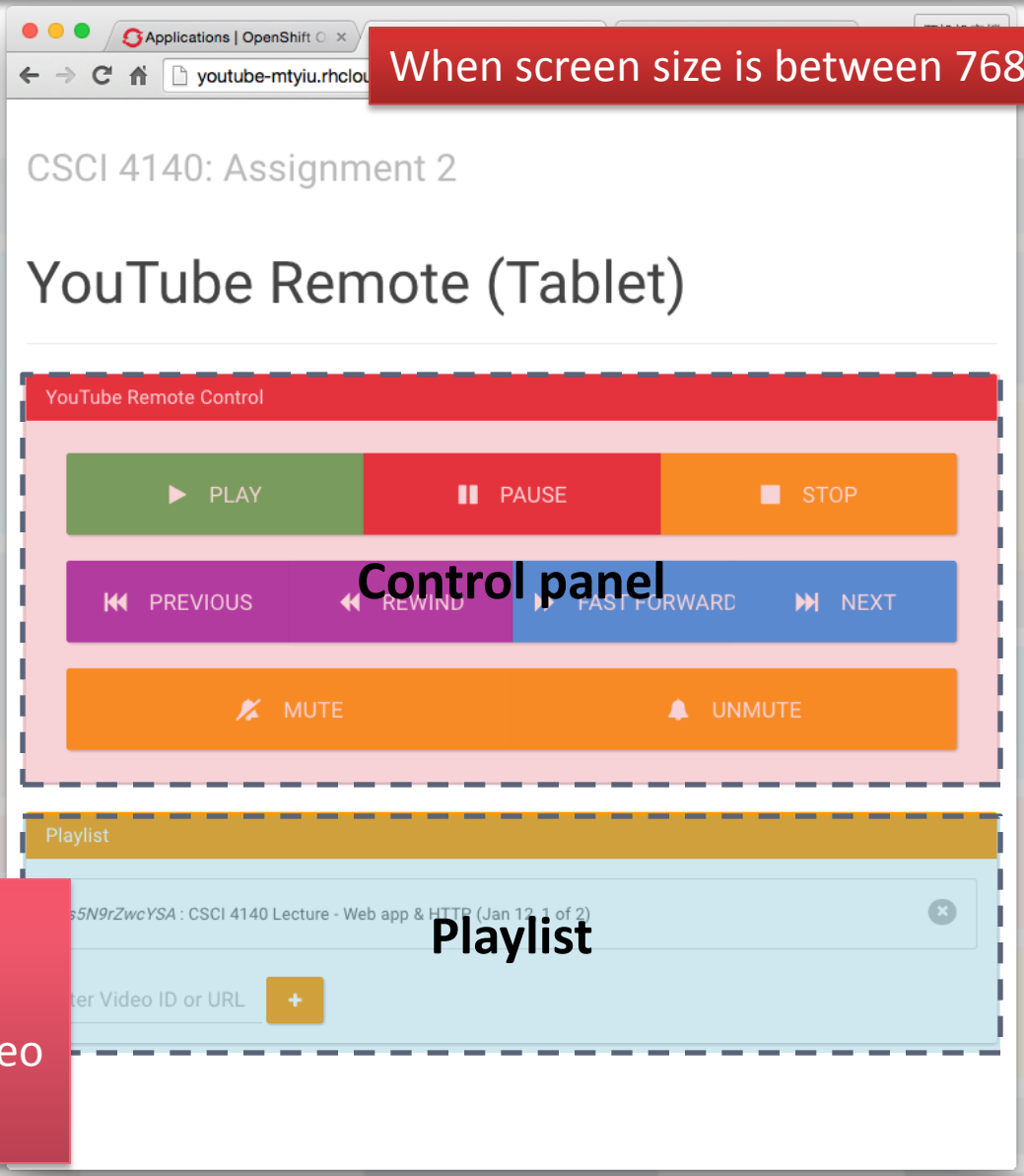
Enter Video ID or URL


CSCI 4140: Assignment 2
youtube-mtyiu.rhcloud.com/session/2670
When screen size ≥ 992 px...
CSCI 4140: Assignment 2
YouTube Remote (Desktop)
YouTube Player
Last Command : -
CSCI 4140 Lecture - Web app & HTTP (Jan 12, 1 of 2)
Example: web counter - permission?
Common Problem
The permission of the CGI program highly depends on setting of the web server!
Security Concern
So, giving CGI processes the root privilege is never a good idea.
Usually, the web server is running as an ordinary user account, e.g., "nobody", "www-data", so that there is no way for any CGI processes to get the root privilege.
Change mode to "777" is not good enough neither...
???.cgi
Running as "www-data"
8:06 / 28:12
YouTube player
(More details in the next tutorial)
YouTube Remote Control
Control panel
Playlist
s5N9rZwcYSA : CSCI 4140 Lecture
Playlist
Enter Video ID or URL

When screen size is between 768px and 992 px…



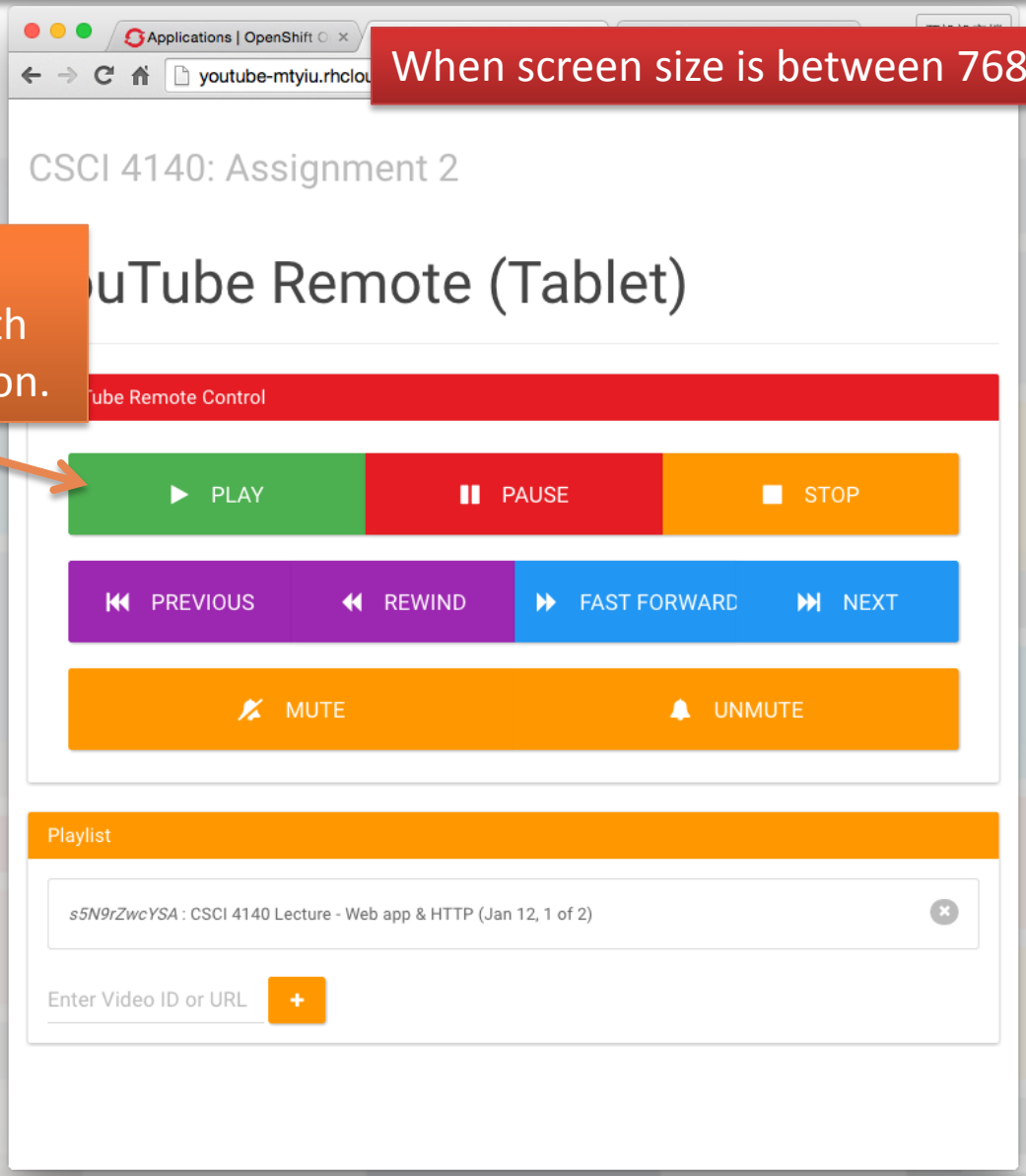
When screen size is between 768px and 992 px…



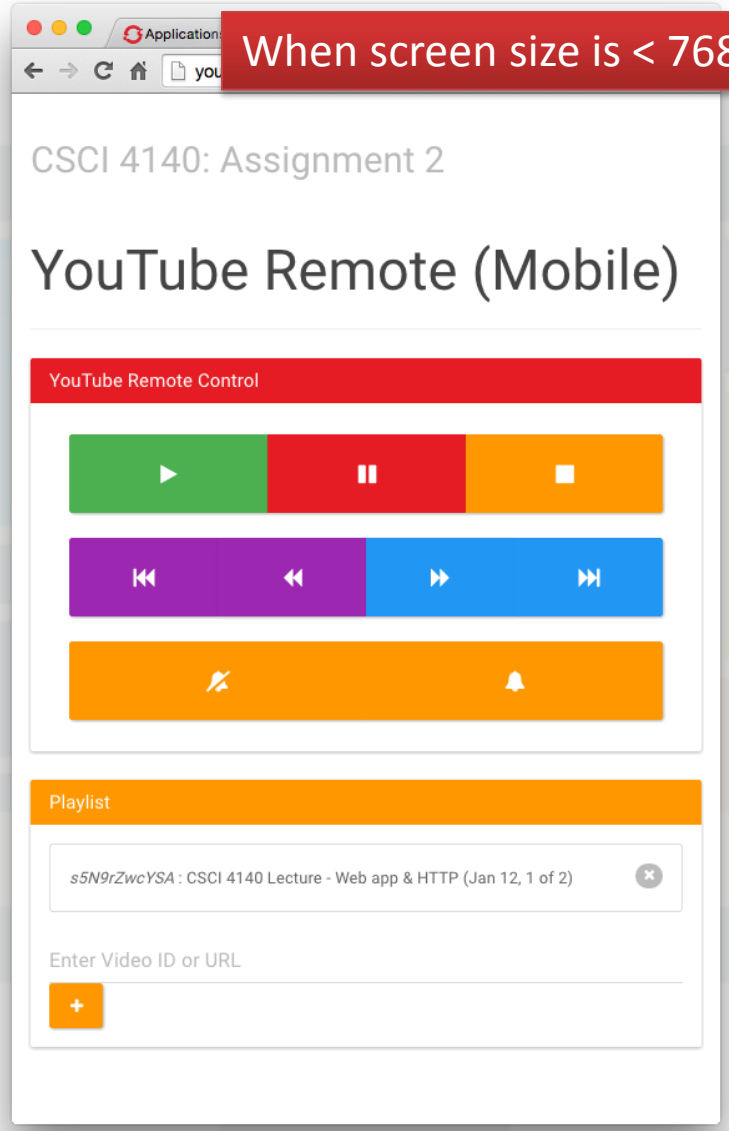
Note: When we reduce the width of the window, the video player is hidden.

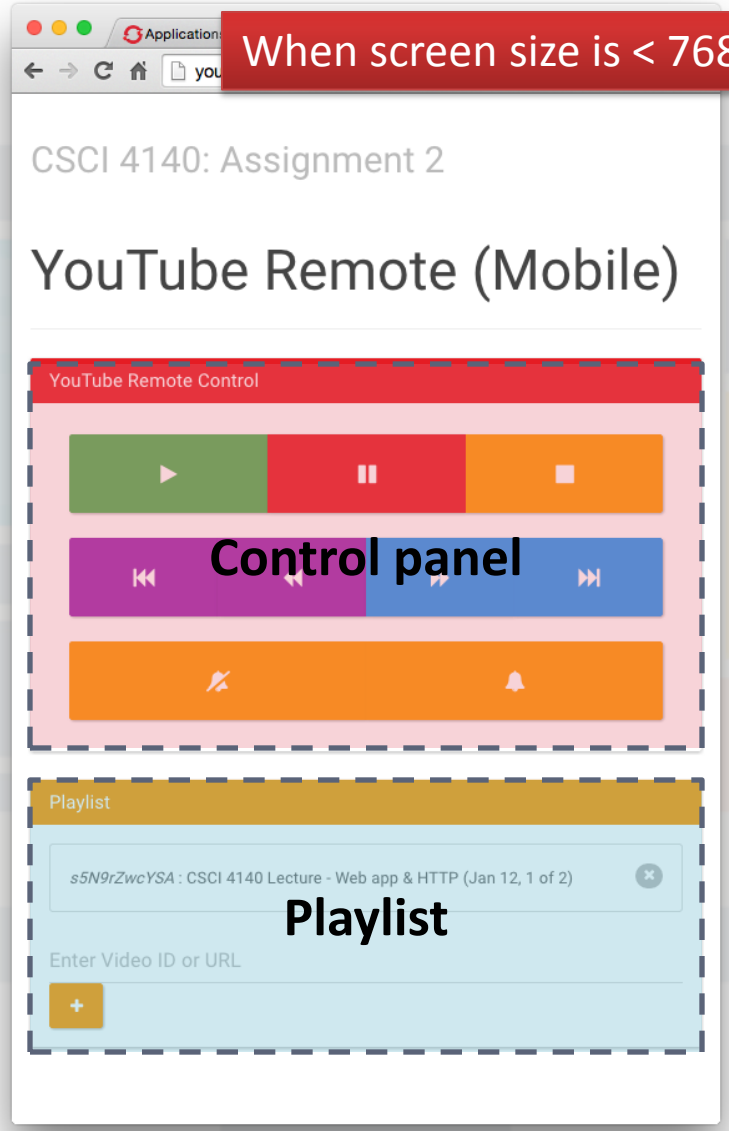
When screen size is between 768px and 992 px…

The buttons are displayed with both icon and description.



When screen size is < 768px...

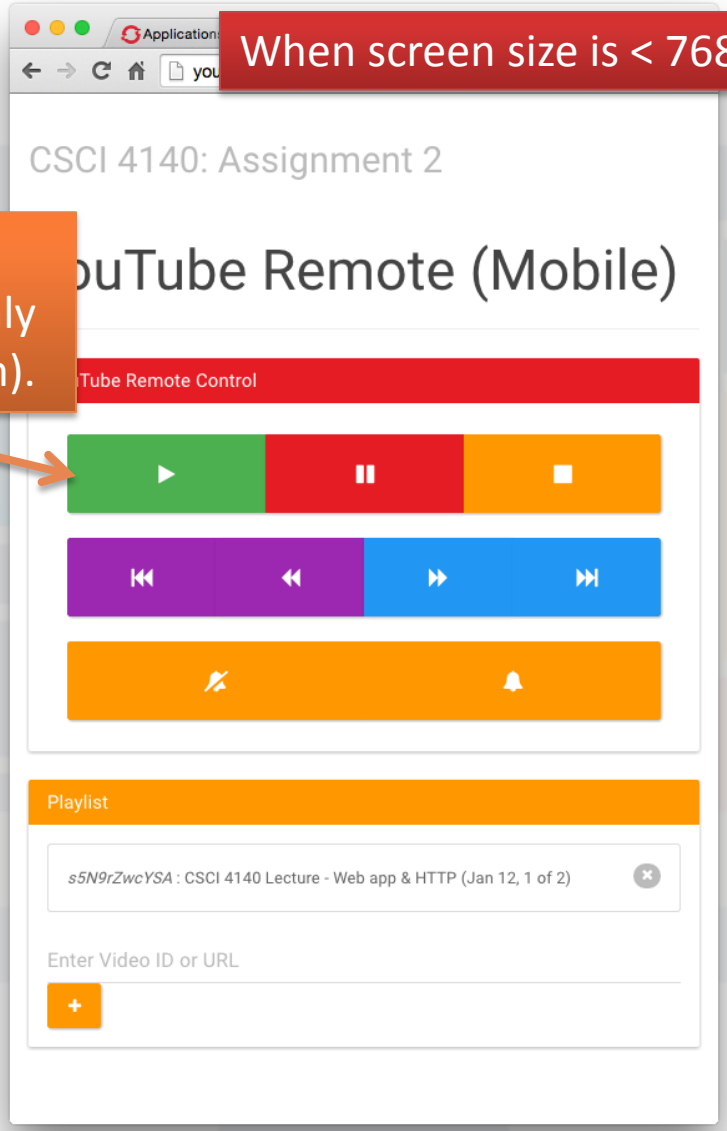




Note: When we reduce the width of the window, the video player is hidden.

When screen size is < 768px...

The buttons are displayed with icon only (**WITHOUT** description).



Assignment 2 Overview – Layout Design

- You are asked to implement a **responsive UI**
- We will resize the window **WITHOUT refreshing the page**, so you are forced to do the screen width detection in **client side**
  - Suggested solution: Bootstrap / CSS media queries
  - Not recommended: JavaScript
- You are free to rearrange the components, but they must meet the requirements in the specification
- After this tutorial, you are able to implement the UI!

```
h1 { color: white;
  background: orange;
  border: 1px solid blac
  padding: 0 0 0 0;
  font-weight: bold;
}
/* begin: seaside-theme */

body {
  background-color:white;
  color:black;
  font-family:Arial,sans-serif;
  margin: 0 4px 0 0;
  border: 12px solid;
}
```

CSS

CSS is almost a must in modern web development..

CSS

- CSS = **C**ascading **S**tyle **S**heets
- A **stylesheet language** used to describe the **presentation** of a document written in HTML or XML
- CSS describes how the structured element must be **rendered** on screen, on paper, in speech, or on other media
- CSS has various **levels** and profiles:
  - CSS1: Published on December 17, 1996
  - CSS2: Published in May 1998
  - CSS2.1: Published on June 7, 2011
  - CSS3: Earliest drafts published in June 1999; published as modules
- Ref: MDN, Wikipedia

CSS: Example

```
<html>
<head>
    <title>CSS Example: H1</title>
</head>
<body>
    <h1>Hello world!</h1>
</body>
</html>
```

css/h1.html

Hello world!

CSS: Example

```
<html>
<head>
    <title>CSS Example: H1</title>
    <style>
        h1 {
            color: #FF0000;
            border: 1px dashed #333333;
        }
    </style>
</head>
<body>
    <h1>Hello world!</h1>
</body>
</html>
```

css/h1-css.html

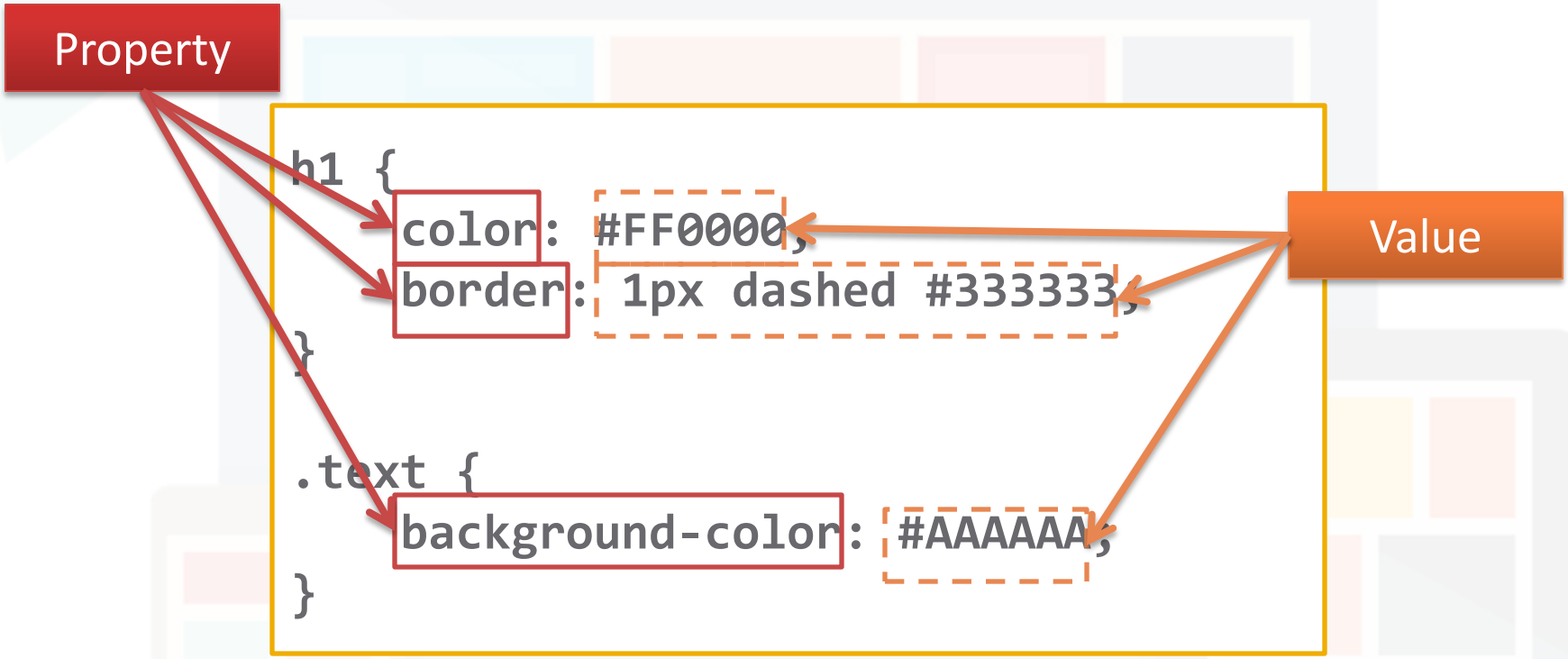
CSS: Syntax


Selector(s)
h1 {
color: #FF0000;
border: 1px dashed #333333;
}
.text {
background-color: #AAAAAA;
}
Declaration

CSS: Syntax


Declarations block
h1 {
color: #FF0000;
border: 1px dashed #333333;
}
.text {
background-color: #AAAAAA;
}

CSS: Syntax


Property
Value
h1 {
color: #FF0000;
border: 1px dashed #333333;
}
.text {
background-color: #AAAAAA;
}

CSS: Syntax

```
h1 {
    color: #FF0000;
    border: 1px dashed #333333;
    /* You can write comments. */
}

.text {
    background-color: #AAAAAA;
}

/* You can also
   write comments
   in multiple lines. */
```

CSS: How to insert?

- Inline styles *(not recommended)*

```
<div style="background-color: red; border: 1px solid;">
    This is ugly...
</div>
```

- Internal stylesheet *(still not recommended)*

```
<html>
<head>
    <style> h1 { color: #FF0000; }</style>
</head>
<body>
    <h1>Hello world!</h1>
</body>
</html>
```

Put the whole CSS declarations block in the `style` attribute.

Embed the CSS declaration blocks within a `<style>` tag.

CSS: How to insert?

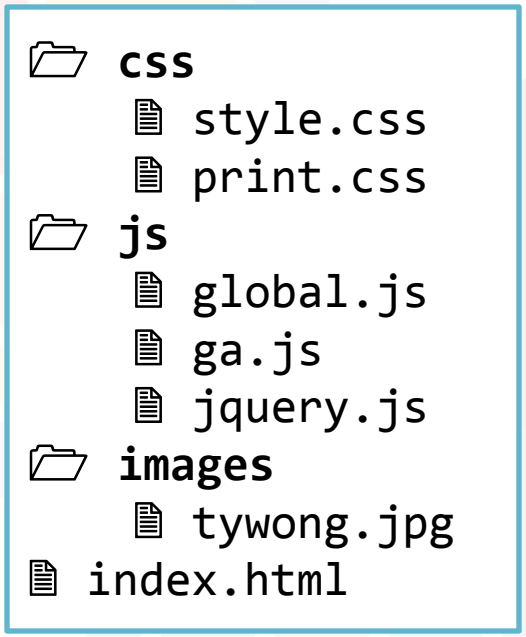
- External stylesheet(s)

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

  - Write the CSS declaration blocks in a **separate file**
  - **Import** the CSS inside the <head> section
  - You can import **multiple CSS**
  - When the browser renders the page, the CSS styles are applied in the **same order of their appearance**
  - This is recommended!
    - Why?

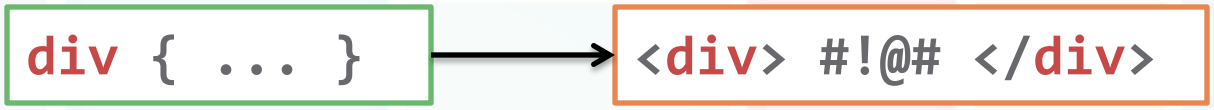
Sidetrack: How to store the code?

- Although there are no standards, most website uses similar structure to store the code (HTML, CSS, JavaScript, …)
- One of the goals of CSS: **Separation of document content from document presentation**
- If you use inline styles or internal stylesheets and you want to change the color of h1, how many file you need to edit if there are 1000 `.html` files!?



CSS: Selectors

- To apply CSS on an element, we need to select it with selectors
  - Element selector

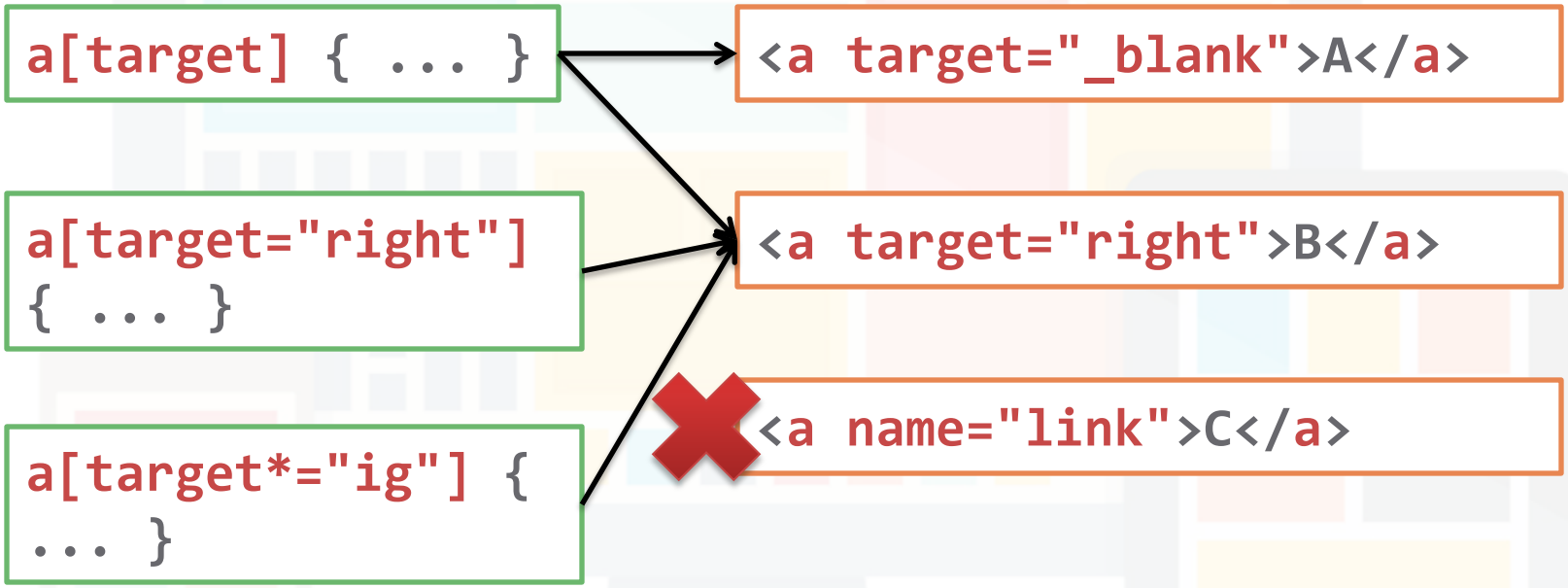


  - ID selector ('#' + ID of the element)

  - Class selector ('.' + class of the elements)

CSS: Selectors

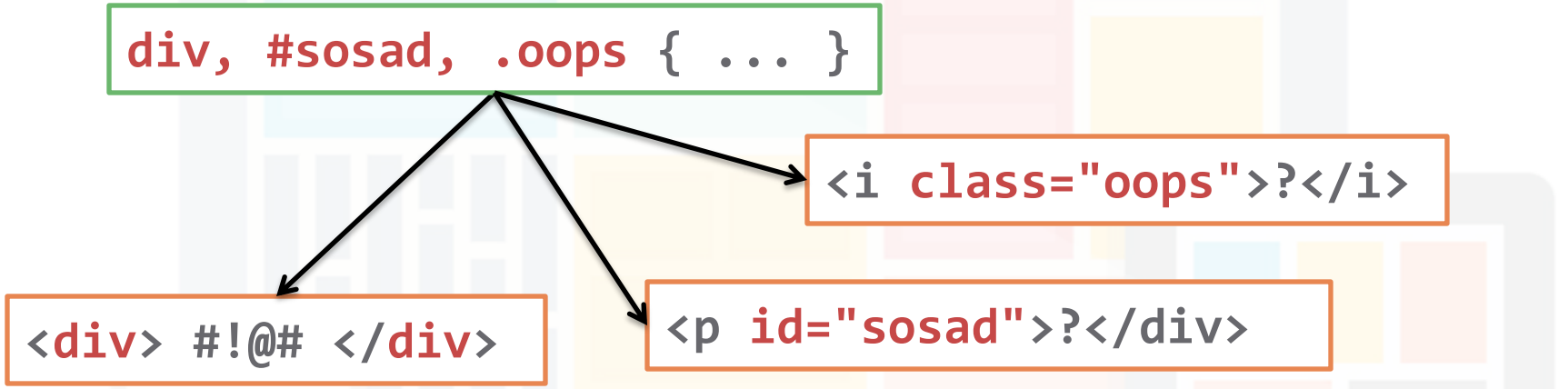
- To apply CSS on an element, we need to select it with selectors
  - Attribute selector



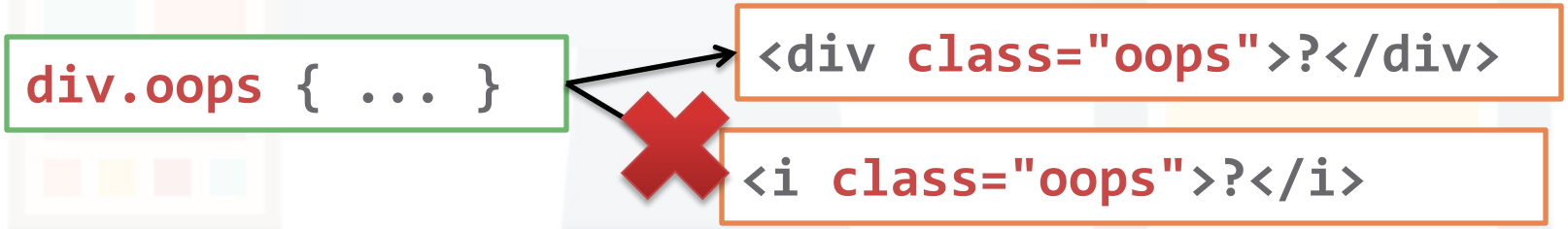
  - Read http://www.w3schools.com/css/css_attribute_selectors.asp for more examples!

CSS: Selectors

- To apply CSS on an element, we need to select it with selectors
  - Multiple selectors (use "," – works like "OR")

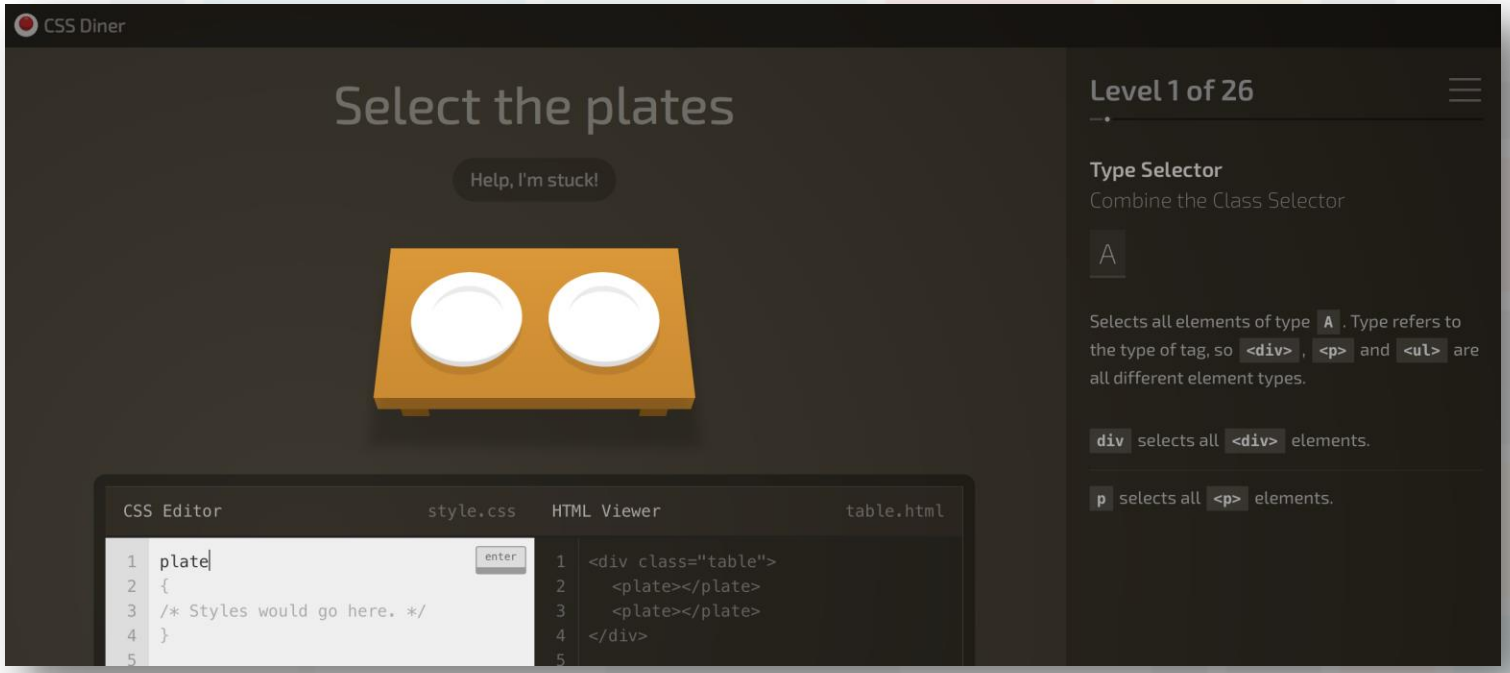


  - Multiple selectors (works like "AND")



CSS: Selectors

- There are more variations in using CSS selectors
- Recommendation: http://flukeout.github.io
  - An addictive game for learning CSS selectors!



CSS: Properties

- There are too many of them…
  - Start from http://www.w3schools.com/css/css_background.asp to learn the available properties yourself
  - Yet another good reference (a "man page" of CSS): https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
  - You can almost customize everything (and forget about IE)
- Demo: Using **Chrome Developer Tools** to modify and try the CSS properties

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";
}
```

  - I used this library in the tutorial page of CSCI 4180 in the last semester…Did you notice that? ☺

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**)

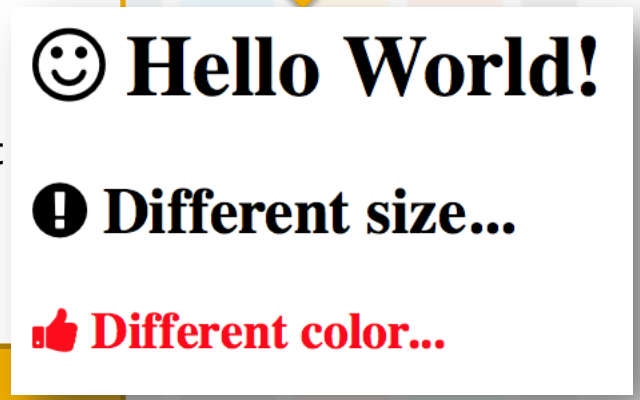
```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css">
```

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/4.3.0/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

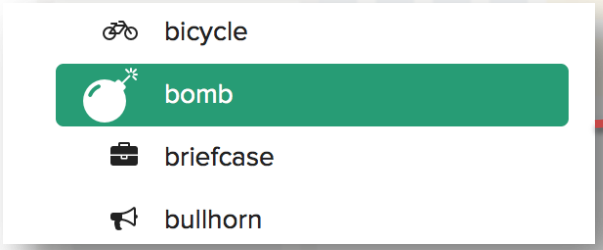


Font Awesome: How to use?

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

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

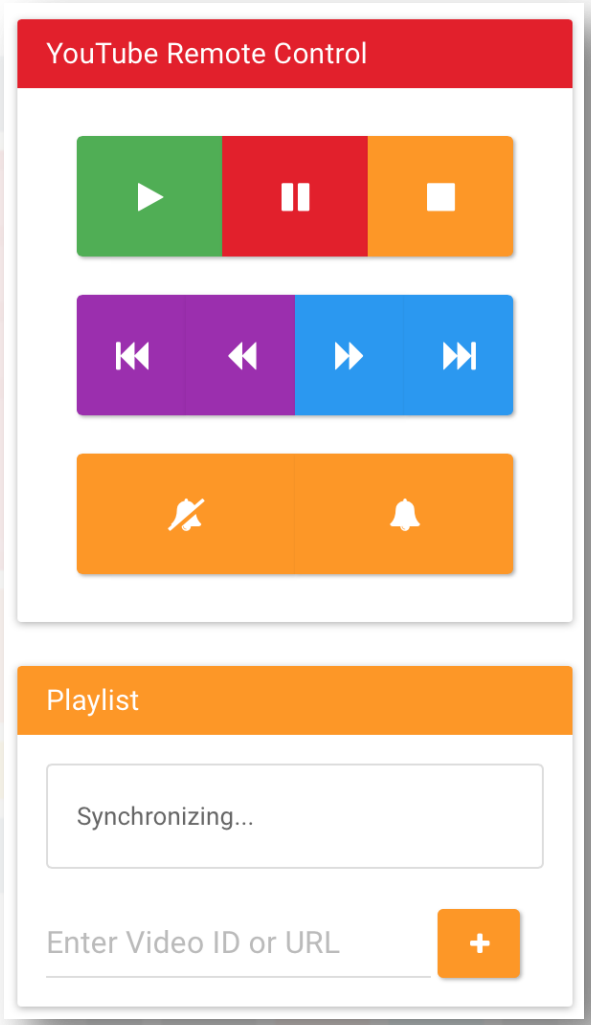
  - The CSS will insert the icon for us!
- Where can we check the icon's name?
  - http://fortawesome.github.io/Font-Awesome/icons/




```
<i class="fa fa-bomb"></i>
```

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



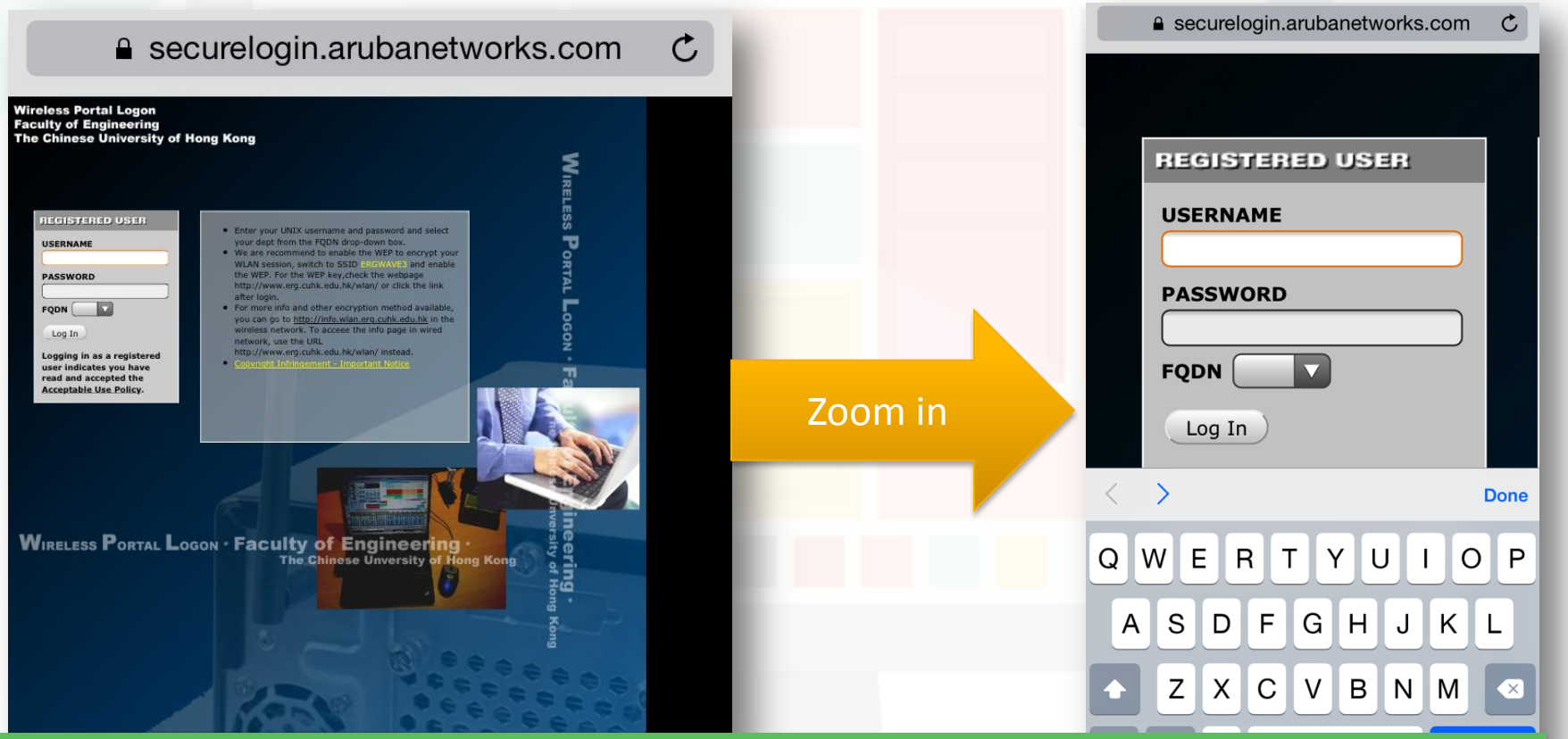
Responsive web design (RWD)

Believe me, you are already viewing it everyday..

Responsive web design: Introduction

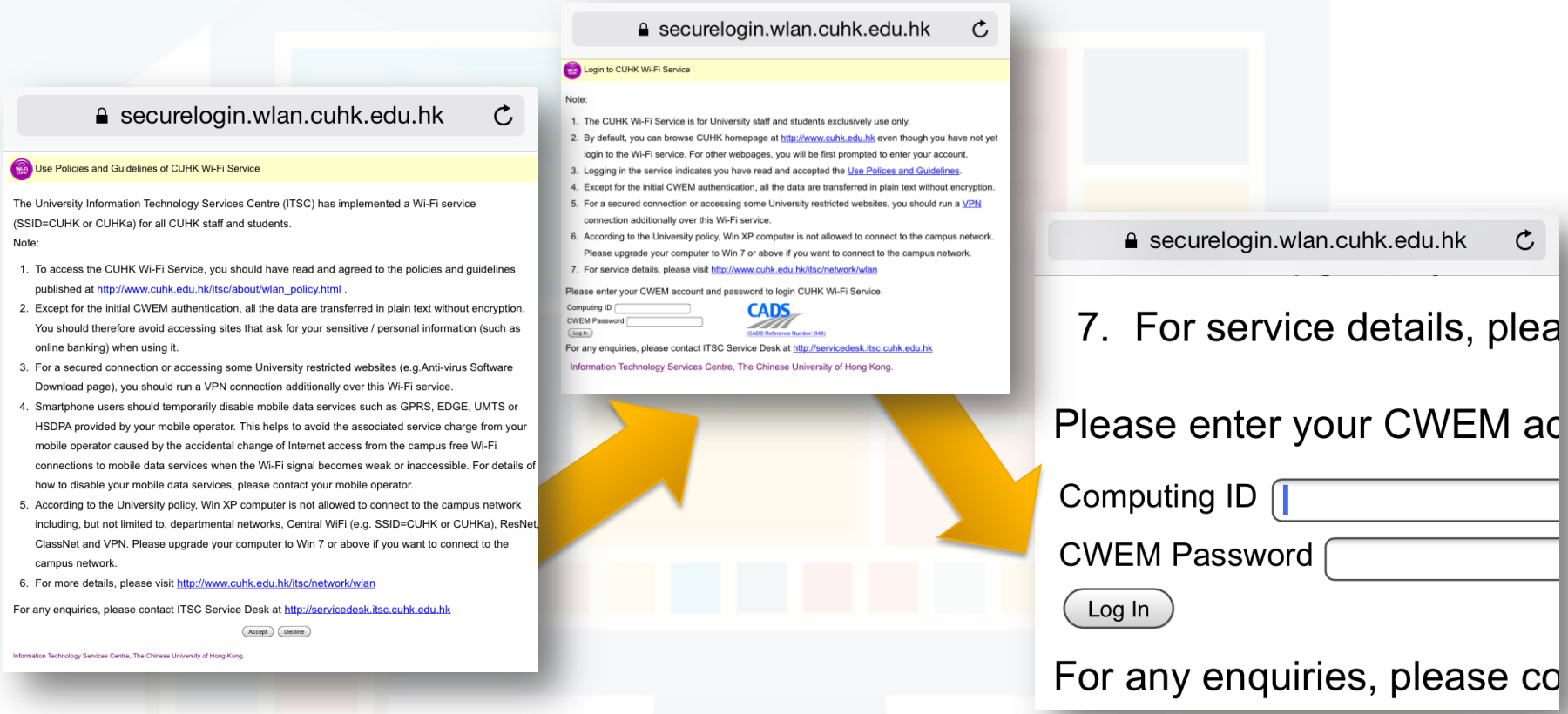
- How many “smart” devices do you have?
  - Smartphones, tablets, notebook, …
  - Designing web pages to display in these devices without degrading **user experiences** (UX) are extremely challenging!
    - Different screen sizes, resolutions, pixel per inch (ppi), browsers, …
    - It is not feasible 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://www.cse.cuhk.edu.hk/csci4140/
  - Tutorial resource page: http://mtyiu.github.io/csci4140-spring15/
  - Harvard University: http://www.harvard.edu
  - Kickstarter: https://www.kickstarter.com

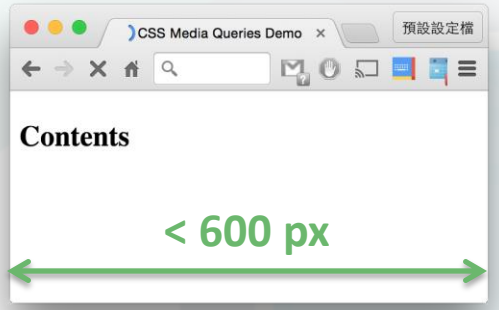
CSS media queries

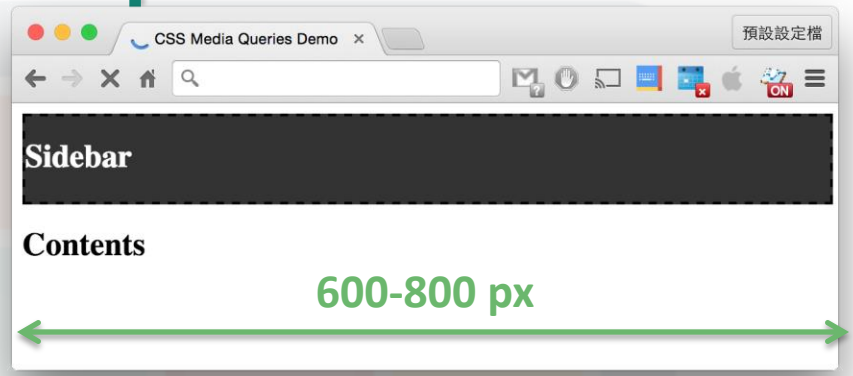
An essential ingredient of responsive web design…

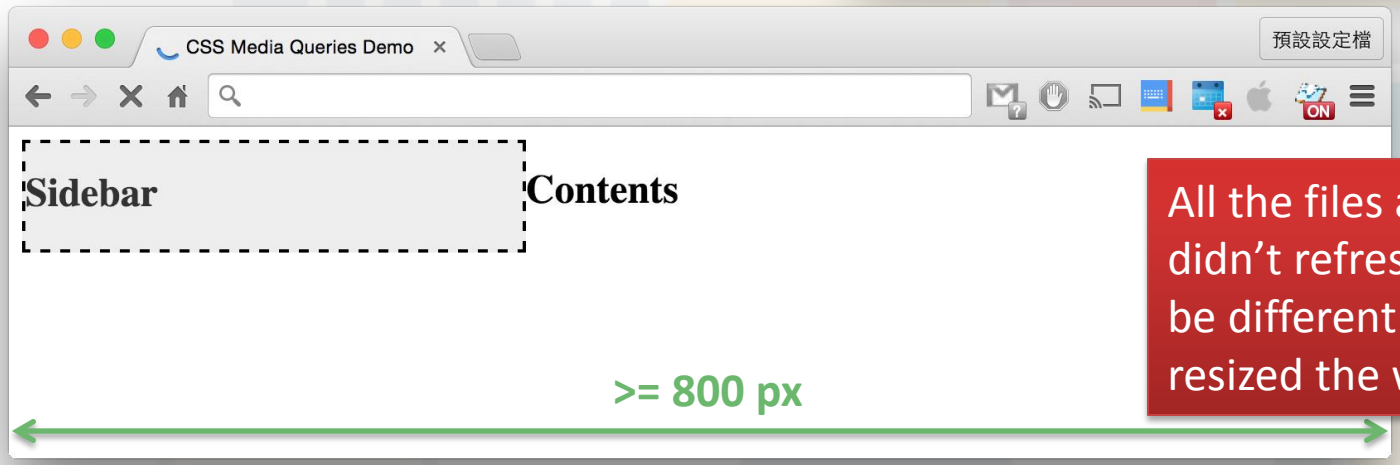
CSS media queries

- 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

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?

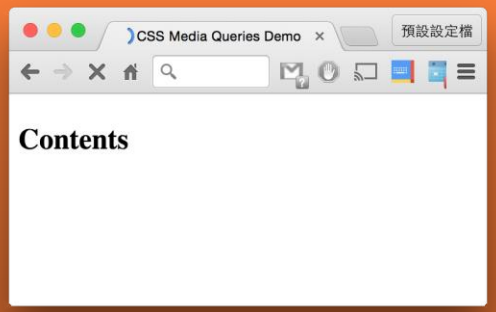
`media-query/sidebar.html & media-query/sidebar.css`

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!

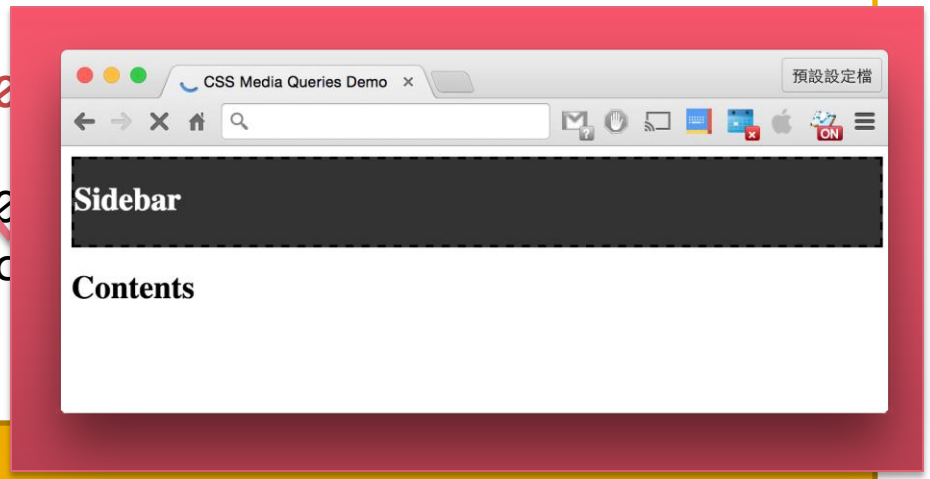
- `@media` works like an "`if`"
- `screen`: The page is viewed with a screen
- `min-width`: The minimum width of the rendering surface of the output device

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
- 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/
  - Course homepage: http://www.cse.cuhk.edu.hk/csci4140/
  - Tutorial resource page: http://mtyiu.github.io/csci4140-spring15/

Bootstrap: How to use?

- Again, there is a Bootstrap CDN:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/boots
trap.min.css">

<!-- Optional theme (can be omitted) -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/boots
trap-theme.min.css">

<!-- Latest compiled and minified JavaScript -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/js/bootstr
ap.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 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/template.html

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.html

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.

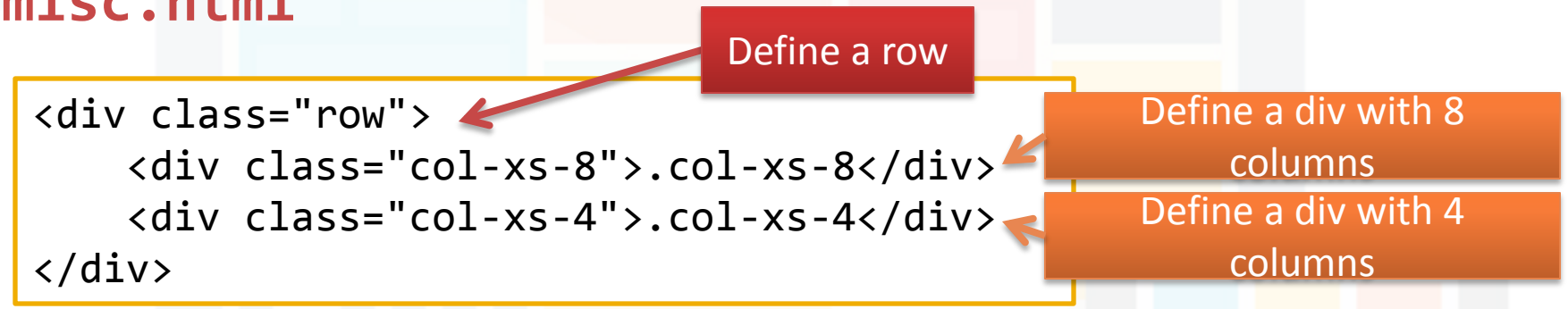
- **`.container`**: A responsive fixed width container
- **`.container-fluid`**: Full width container, spanning the entire width of your viewport

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

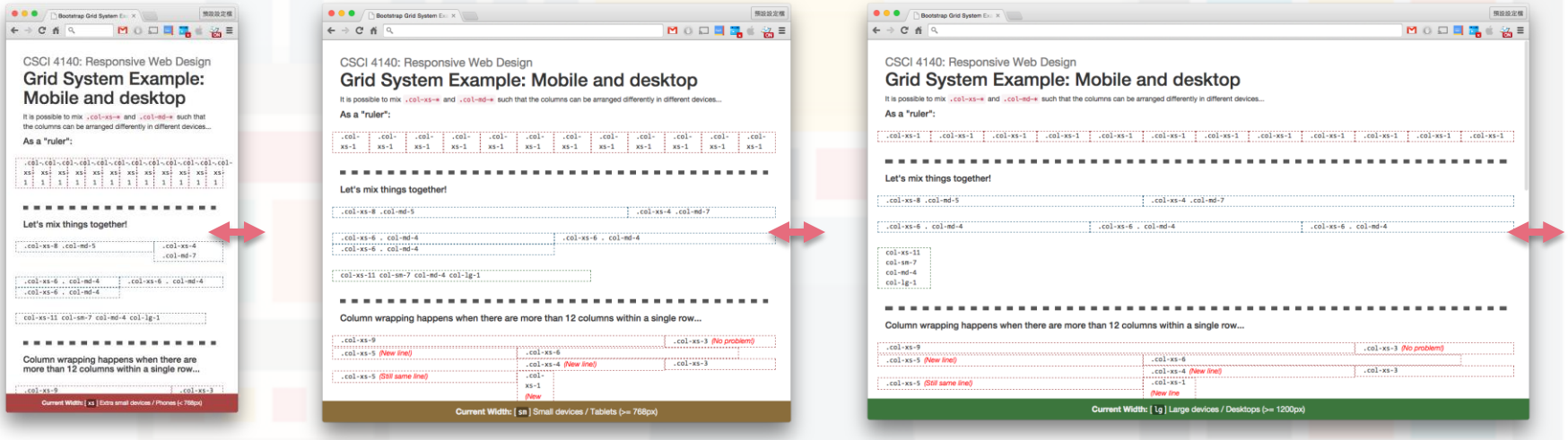
- Examples: **bootstrap/grid.html** and **bootstrap/grid-misc.html**



  - 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 *(I don't have time to go through them one by one…)*
  - 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
  - The JavaScript part is forbidden in assignments, but you can use in your projects

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 CSS part is also allowed in your assignments
  - The grid system is similar to that of Bootstrap
  - Example: My homepage (http://www.cse.cuhk.edu.hk/~mtyiu/)
- If you are a fan of CSS animations: http://daneden.github.io/animate.css/

Reminder

- We will use **Node.js** and **Express** to implement the backend of Assignment 2
  - No more Python ☺
- 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
  - Go to the **tutorial resource page** to download the instructions for installing Node.js and Express on Windows, Linux or Mac

– End –