Internet Technologies

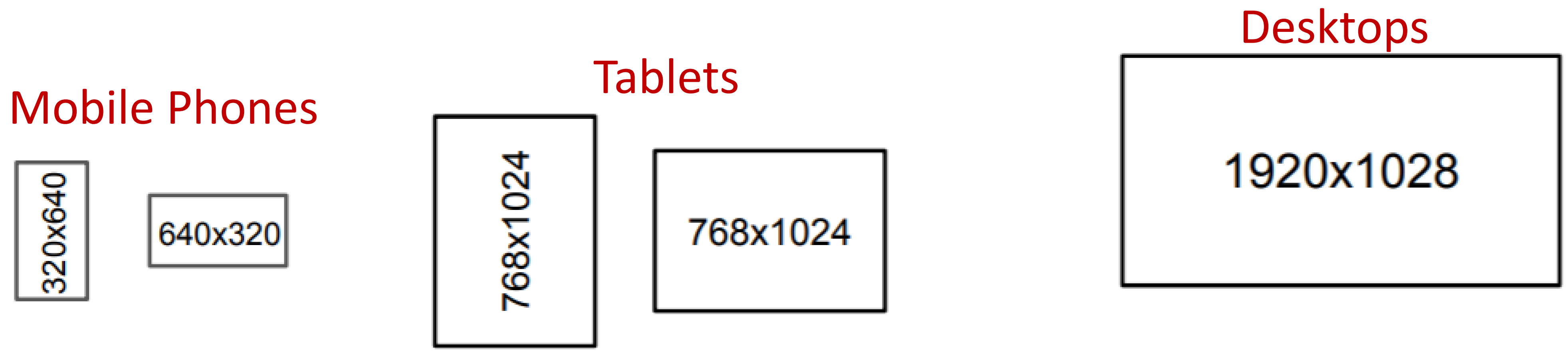
Introduction to HTML and CSS - Part 4

Responsive Webpage Development

University of Cyprus
Department of Computer Science

Responsive web design

- Use HTML and CSS to automatically resize, hide, shrink, or enlarge components of a website (images, buttons, forms, tables, font sizes, margin, padding), to make the website look good on all devices



Q: How do we do this?

Do we need to write totally different CSS for every screen size??

Meta viewport tag

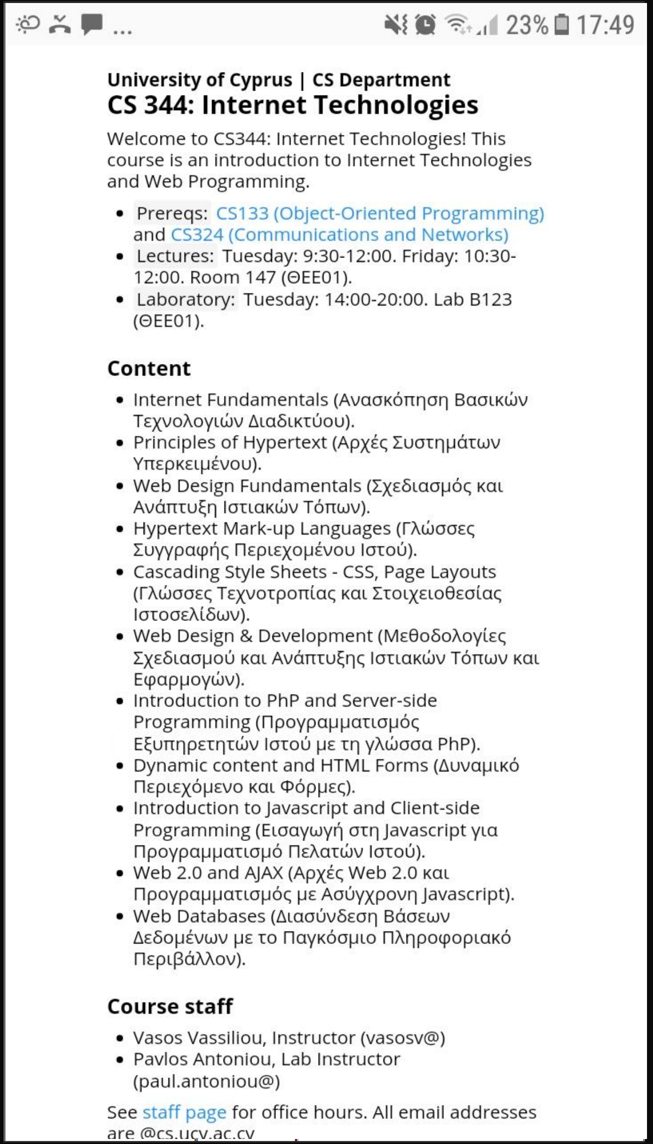
- A typical mobile-optimized site contains something like the following:

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

- Sets the viewport of a webpage: gives the browser instructions on how to control the page's dimensions and scaling.
- This belongs in the `<head>` section of your HTML.
  - Same section as the `<title>`, `<link>`, and other metadata elements.

Without the meta viewport tag

☹


23% 17:49

University of Cyprus | CS Department

CS 344: Internet Technologies

Welcome to CS344: Internet Technologies! This course is an introduction to Internet Technologies and Web Programming.

- Prereqs: CS133 (Object-Oriented Programming) and CS324 (Communications and Networks)
- Lectures: Tuesday: 9:30-12:00. Friday: 10:30-12:00. Room 147 (ΘΕΕ01).
- Laboratory: Tuesday: 14:00-20:00. Lab B123 (ΘΕΕ01).

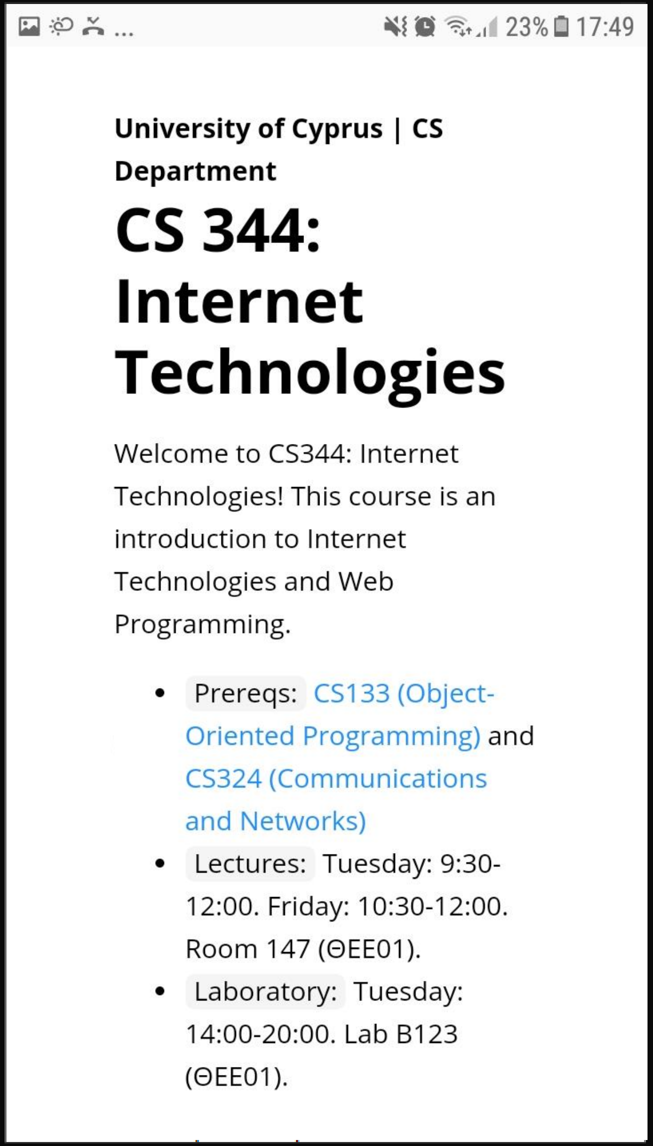
Content

- Internet Fundamentals (Ανασκόπηση Βασικών Τεχνολογιών Διαδικτύου).
- Principles of Hypertext (Αρχές Συστημάτων Υπερκειμένου).
- Web Design Fundamentals (Σχεδιασμός και Ανάπτυξη Ιστιακών Τόπων).
- Hypertext Mark-up Languages (Γλώσσες Συγγραφής Περιεχομένου Ιστού).
- Cascading Style Sheets - CSS, Page Layouts (Γλώσσες Τεχνοτροπίας και Στοιχειοθεσίας Ιστοσελίδων).
- Web Design & Development (Μεθοδολογίες Σχεδιασμού και Ανάπτυξης Ιστιακών Τόπων και Εφαρμογών).
- Introduction to PhP and Server-side Programming (Προγραμματισμός Εξυπηρετητών Ιστού με τη γλώσσα PhP).
- Dynamic content and HTML Forms (Δυναμικό Περιεχόμενο και Φόρμες).
- Introduction to Javascript and Client-side Programming (Εισαγωγή στη Javascript για Προγραμματισμό Πελατών Ιστού).
- Web 2.0 and AJAX (Αρχές Web 2.0 και Προγραμματισμός με Ασύγχρονη Javascript).
- Web Databases (Διασύνδεση Βάσεων Δεδομένων με το Παγκόσμιο Πληροφοριακό Περιβάλλον).

Course staff

- Vasos Vassiliou, Instructor (vasosv@)
- Pavlos Antoniou, Lab Instructor (paul.antoniou@)

See staff page for office hours. All email addresses are @cs.ucy.ac.cy


23% 17:49

University of Cyprus | CS Department

CS 344: Internet Technologies

Welcome to CS344: Internet Technologies! This course is an introduction to Internet Technologies and Web Programming.

- Prereqs: CS133 (Object-Oriented Programming) and CS324 (Communications and Networks)
- Lectures: Tuesday: 9:30-12:00. Friday: 10:30-12:00. Room 147 (ΘΕΕ01).
- Laboratory: Tuesday: 14:00-20:00. Lab B123 (ΘΕΕ01).

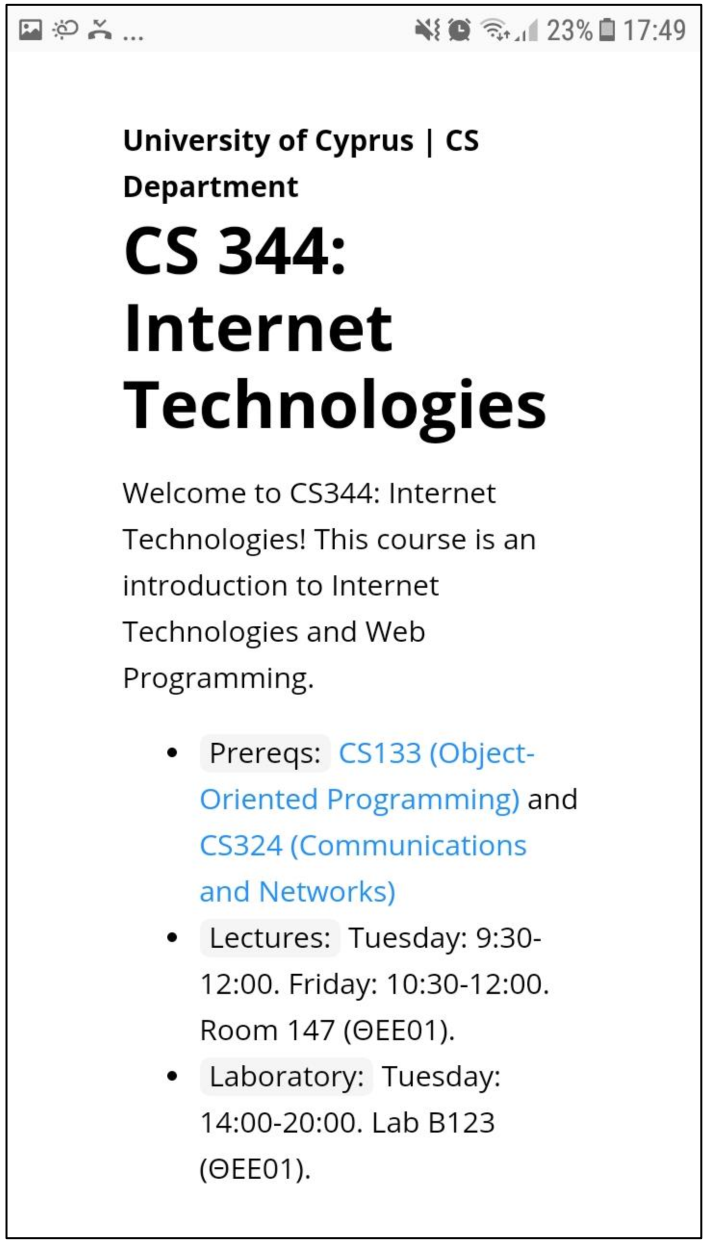
With the meta viewport tag

☺

You should always include this tag in your HTML

Making adjustments

- The meta viewport tag gets us almost all the way there, but we want to make a few adjustments.
- For example, the margin seems too big on mobile. Can we set a different margin property for mobile?



University of Cyprus | CS Department

CS 344: Internet Technologies

Welcome to CS344: Internet Technologies! This course is an introduction to Internet Technologies and Web Programming.

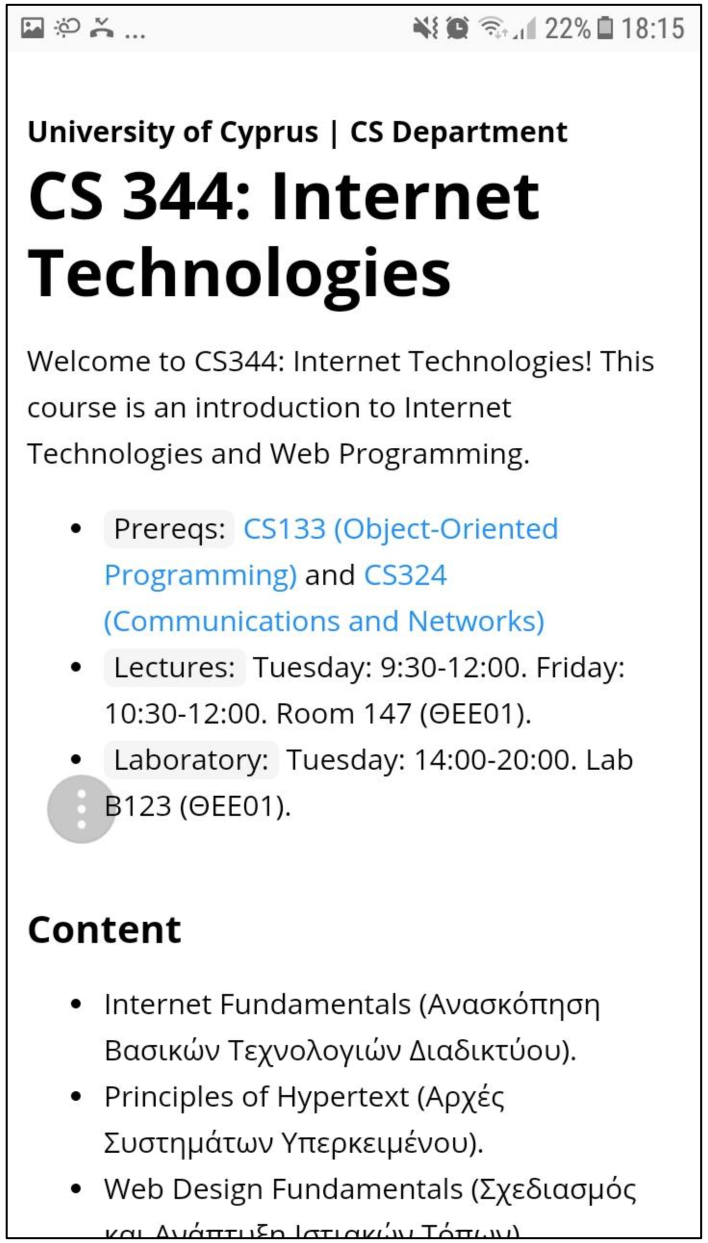
- Prereqs: CS133 (Object-Oriented Programming) and CS324 (Communications and Networks)
- Lectures: Tuesday: 9:30-12:00. Friday: 10:30-12:00. Room 147 (ΘΕΕ01).
- Laboratory: Tuesday: 14:00-20:00. Lab B123 (ΘΕΕ01).

CSS media queries

- You can define a **CSS media query** in order to change style rules based on the characteristics of the device:

CSS
```css
@media (max-width: 500px)
{
   .article {
      margin: 0 2px;
   }
}
```

- You can create much more complex media queries as well.



University of Cyprus | CS Department

CS 344: Internet Technologies

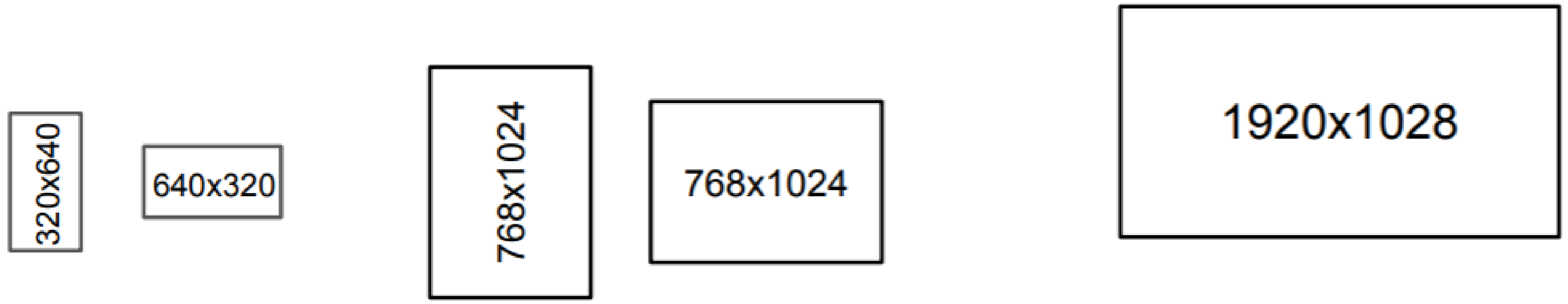
Welcome to CS344: Internet Technologies! This course is an introduction to Internet Technologies and Web Programming.

- Prereqs: CS133 (Object-Oriented Programming) and CS324 (Communications and Networks)
- Lectures: Tuesday: 9:30-12:00. Friday: 10:30-12:00. Room 147 (ΘΕΕ01).
- Laboratory: Tuesday: 14:00-20:00. Lab B123 (ΘΕΕ01).

Content

- Internet Fundamentals (Ανασκόπηση Βασικών Τεχνολογιών Διαδικτύου).
- Principles of Hypertext (Αρχές Συστημάτων Υπερκειμένου).
- Web Design Fundamentals (Σχεδιασμός

Media rules



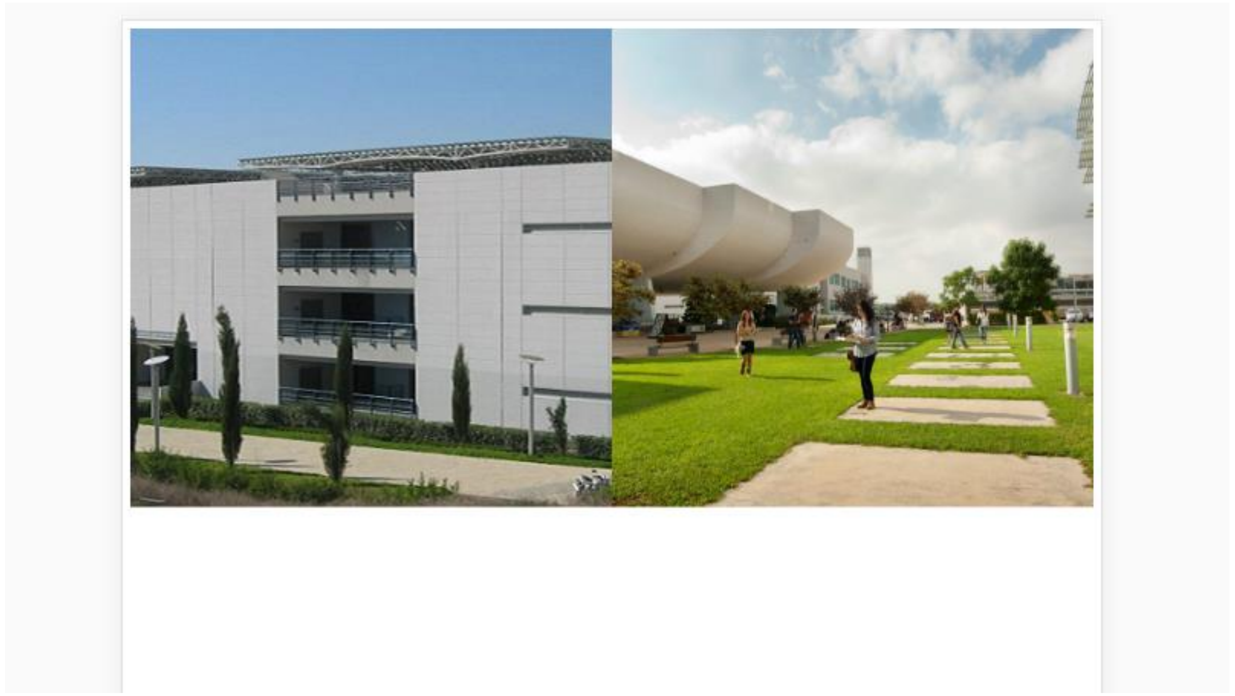
CSS

```css
@media (min-width: 1281px)  {
   /* desktops layout */
}
@media (min-width: 1025px) and (max-width: 1280px) {
   /* laptops and desktops */
}
@media (min-width: 768px) and (max-width: 1024px) {
   /* tablets (portrait) */
}
@media (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) {
   /* tablets landscape */
}
@media (min-width: 481px) and (max-width: 767px)  {
   /* low resolution tablets, mobile phones (landscape) */
}
@media (min-width: 320px) and (max-width: 480px) {
   /* mobile phones portrait */
}
```

Retina media queries for high resolution mobile displays: https://css-tricks.com/snippets/css/retina-display-media-query/

Example with images

- Webpage with 2 images, side-by-side
- We want images sizes to be responsive to browser width



Example with images

HTML

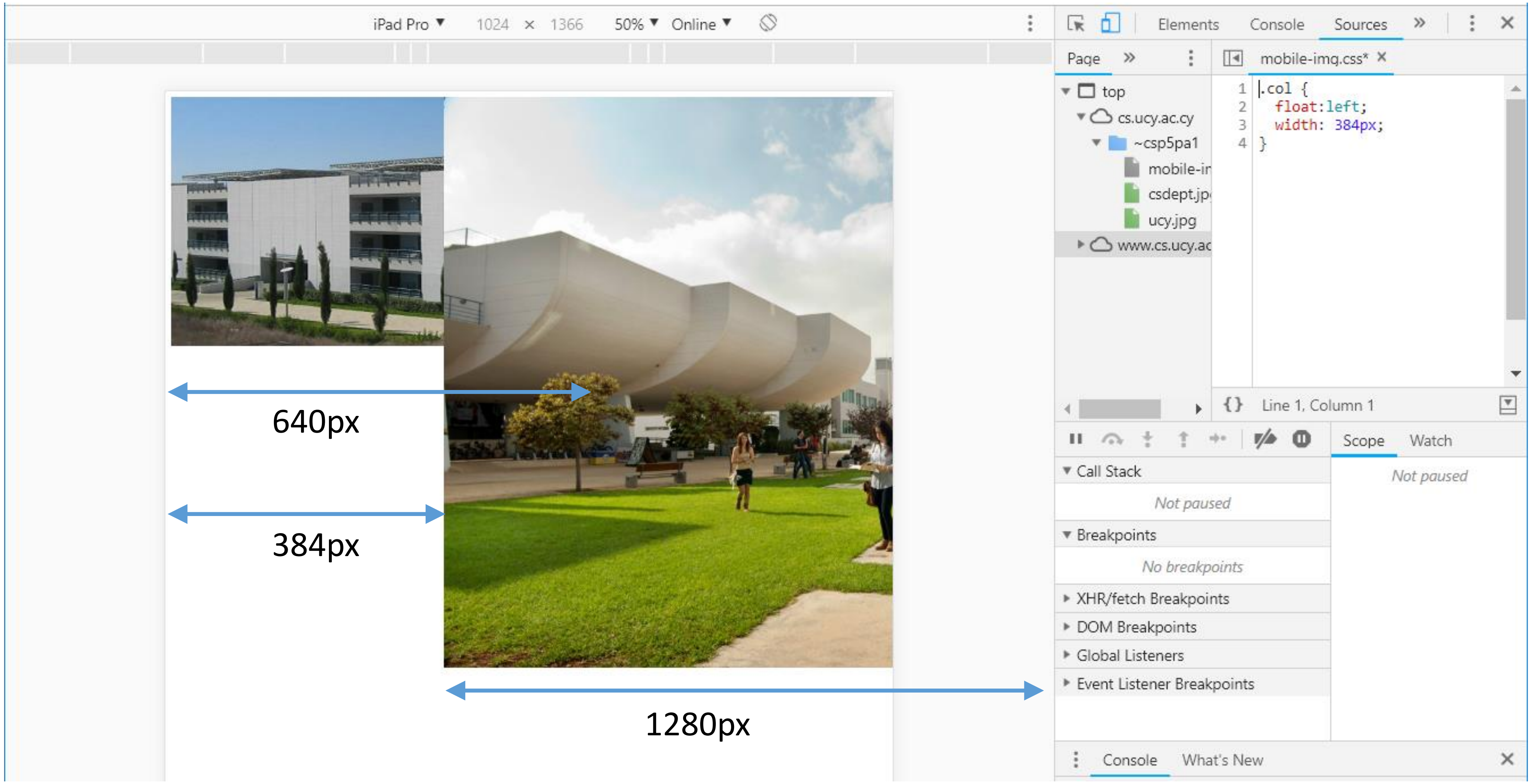
```
<body>
   <div class="row">
      <div class="col">
         <img src="csdept.jpg" alt="csdeptucy"/>
      </div>
      <div class="col">
         <img src="ucy.jpg" alt="ucy"/>
      </div>
   </div>
</body>
```

CSS

```
.col {
  float: left;
  width: 384px;
}
```

Example with images

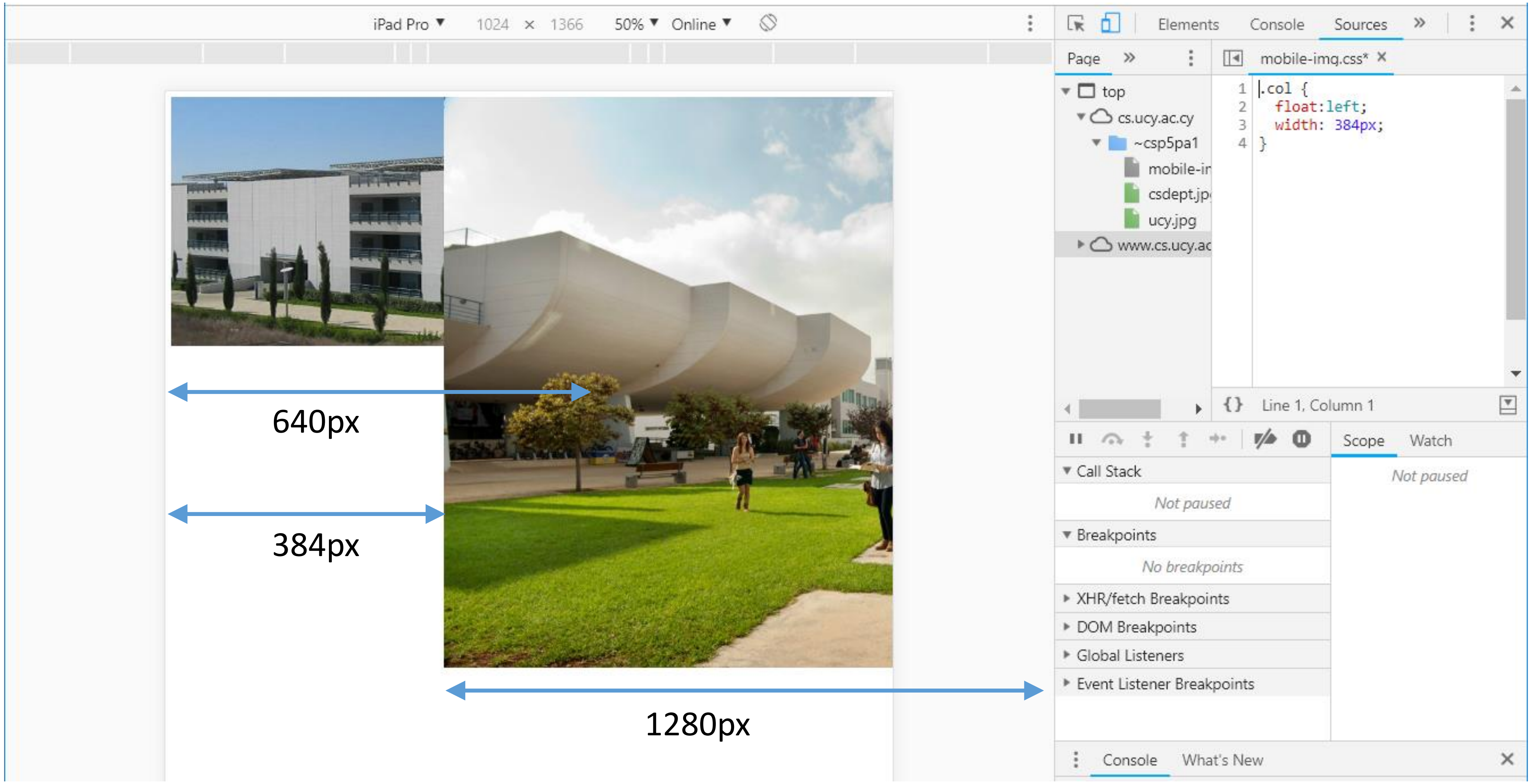
Why images are not bounded to 384px?



```
.col {
  float:left;
  width: 384px;
}
```

Example with images

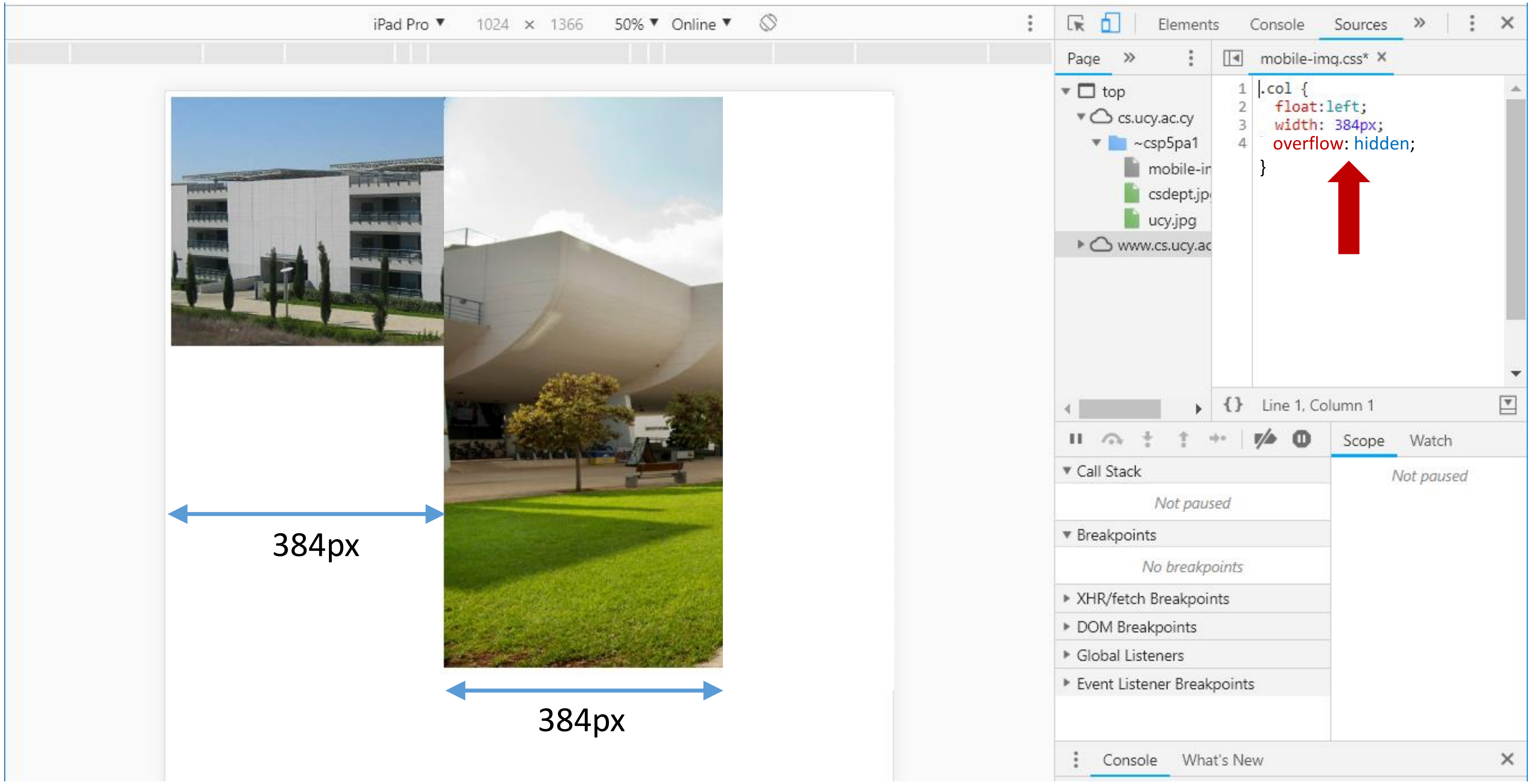
Why images are not bounded to 384px? Because the don't have a specified width. So they overflow the div.



```
.col {
  float:left;
  width: 384px;
}
```

Example with images

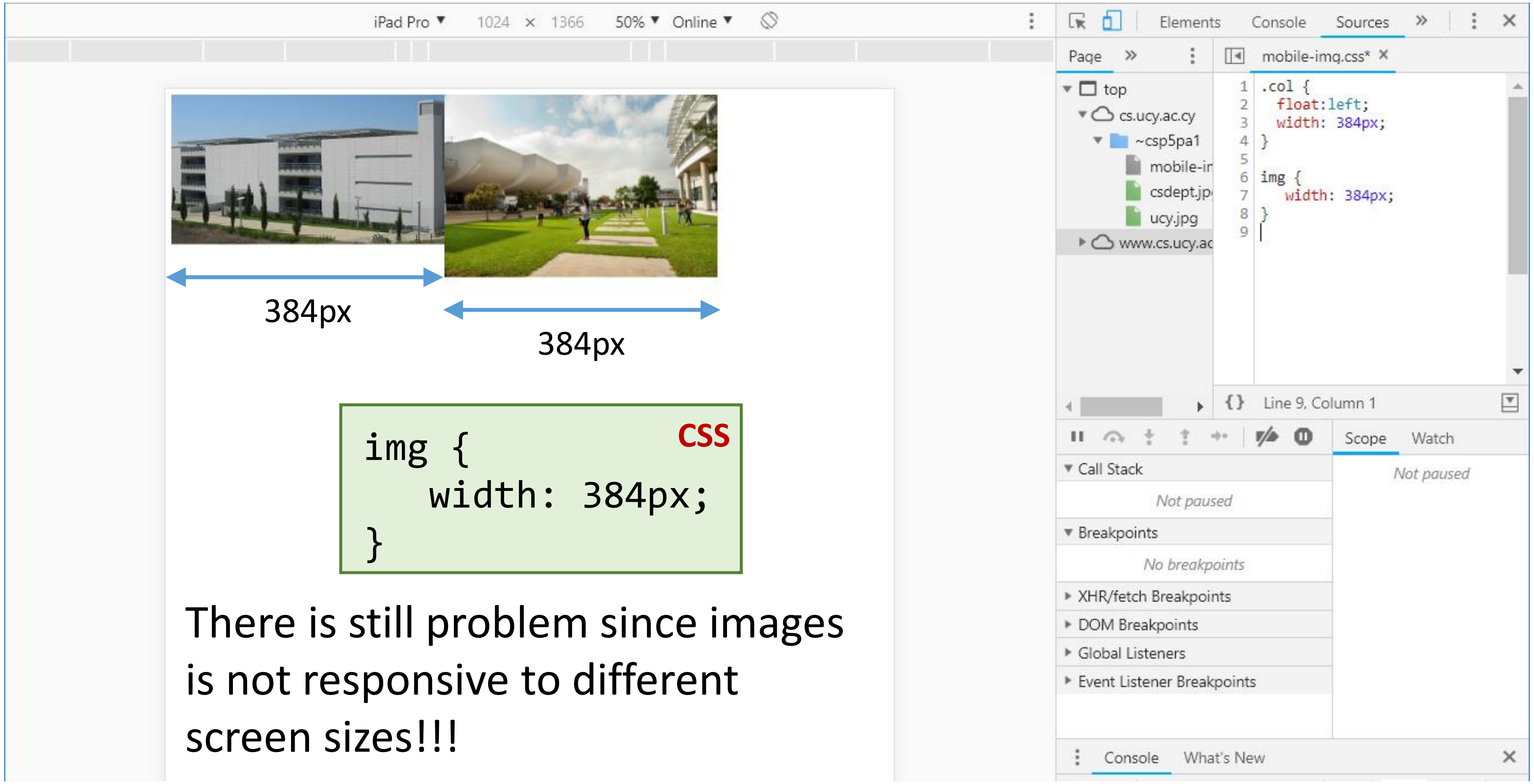
We can hide the overflowing content but this is not always desirable!



```
.col {
  float:left;
  width: 384px;
  overflow: hidden;
}
```

Example with images

Another solution to have the whole image visible… set an image width to match the containing div width…



```css
.col {
  float:left;
  width: 384px;
}

img {
   width: 384px;
}
```

CSS

```css
img {
    width: 384px;
}
```

There is still problem since images is not responsive to different screen sizes!!!

Example with images

Another solution towards responsiveness… use percentages

```
.col {                                                   CSS
  float: left;
  width: 50%;
}
```

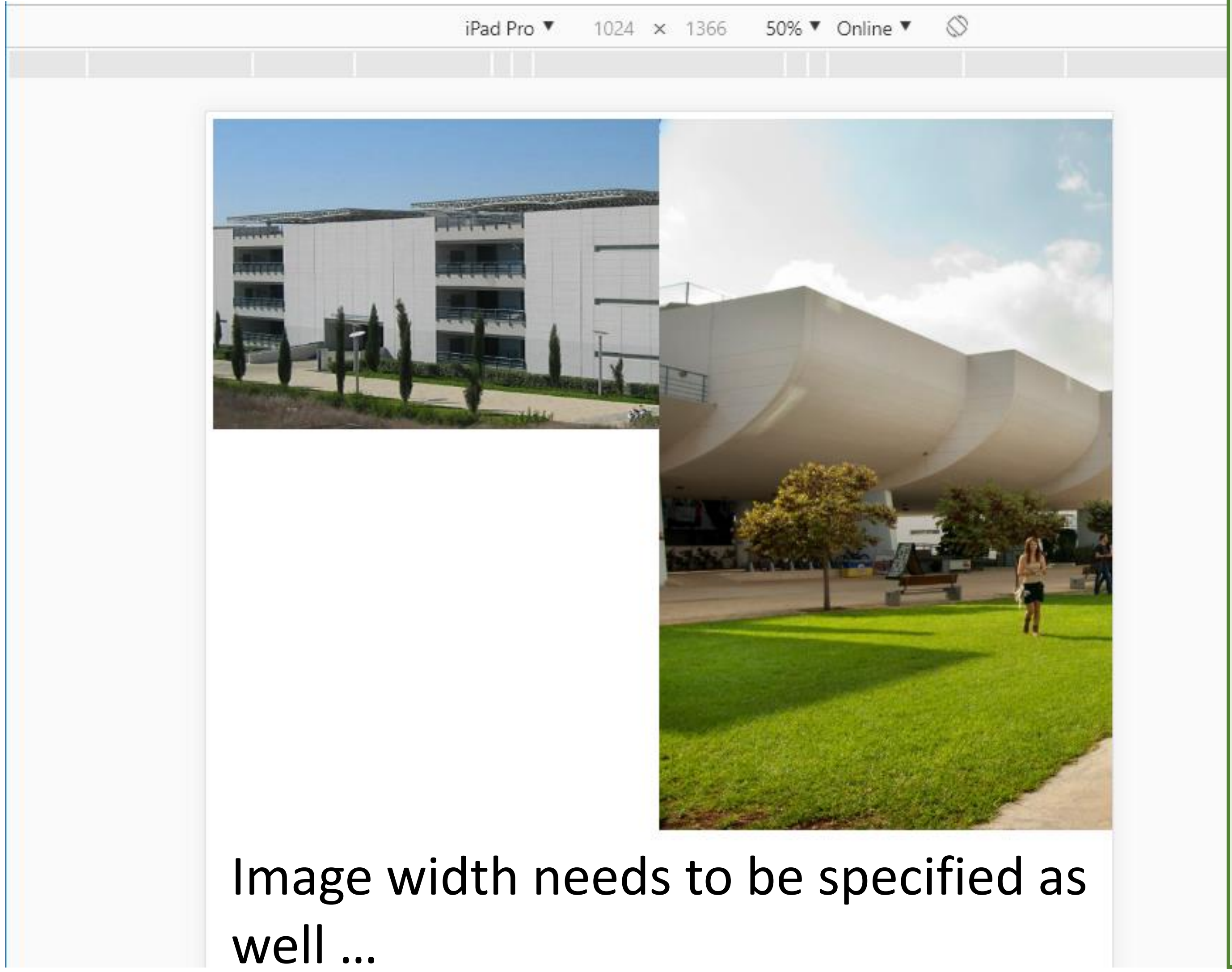
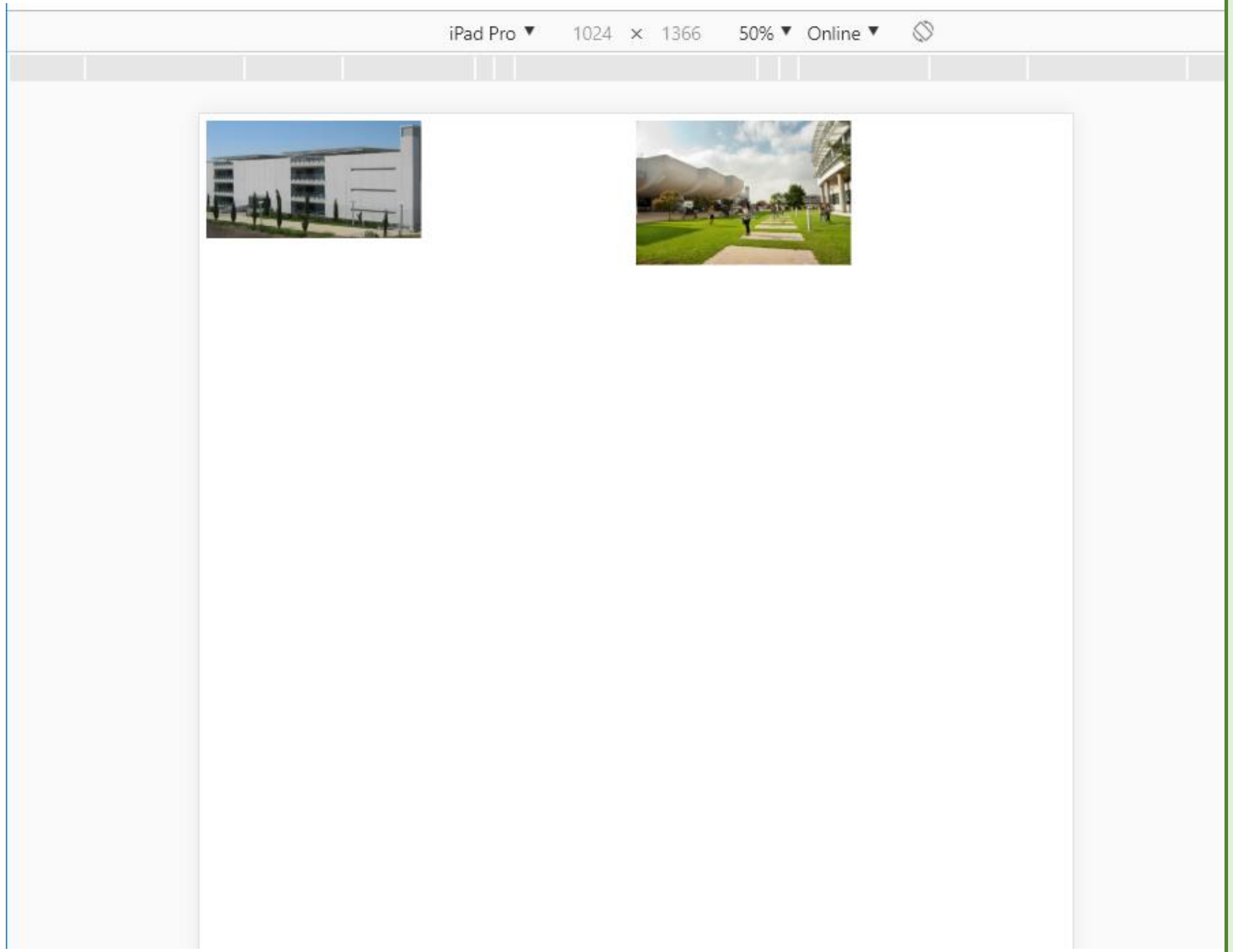


Image width needs to be specified as well …

Example with images

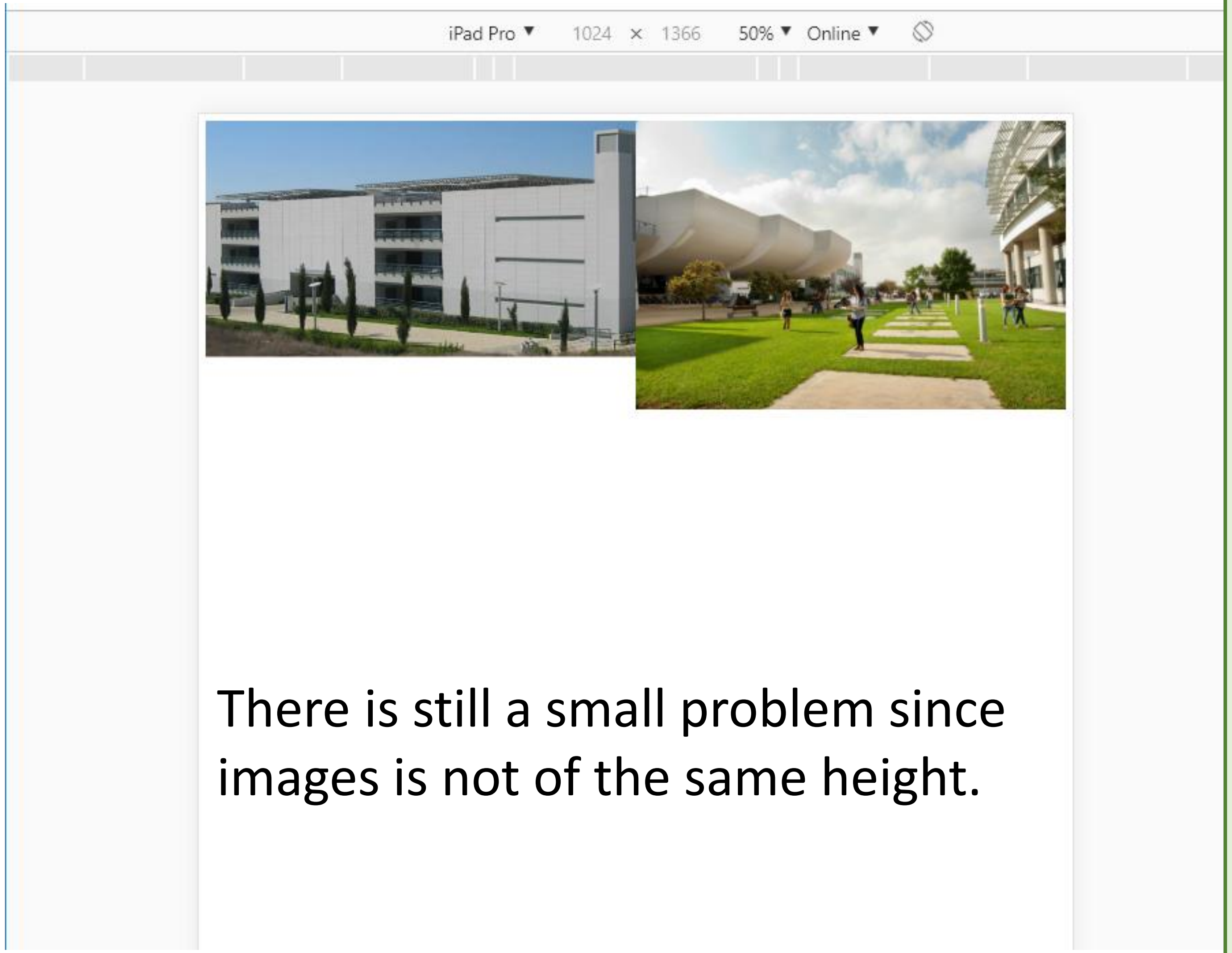


```
.col {
  float: left;
  width: 50%;
}

img {
  width: 50%;
}
```

CSS

Example with images



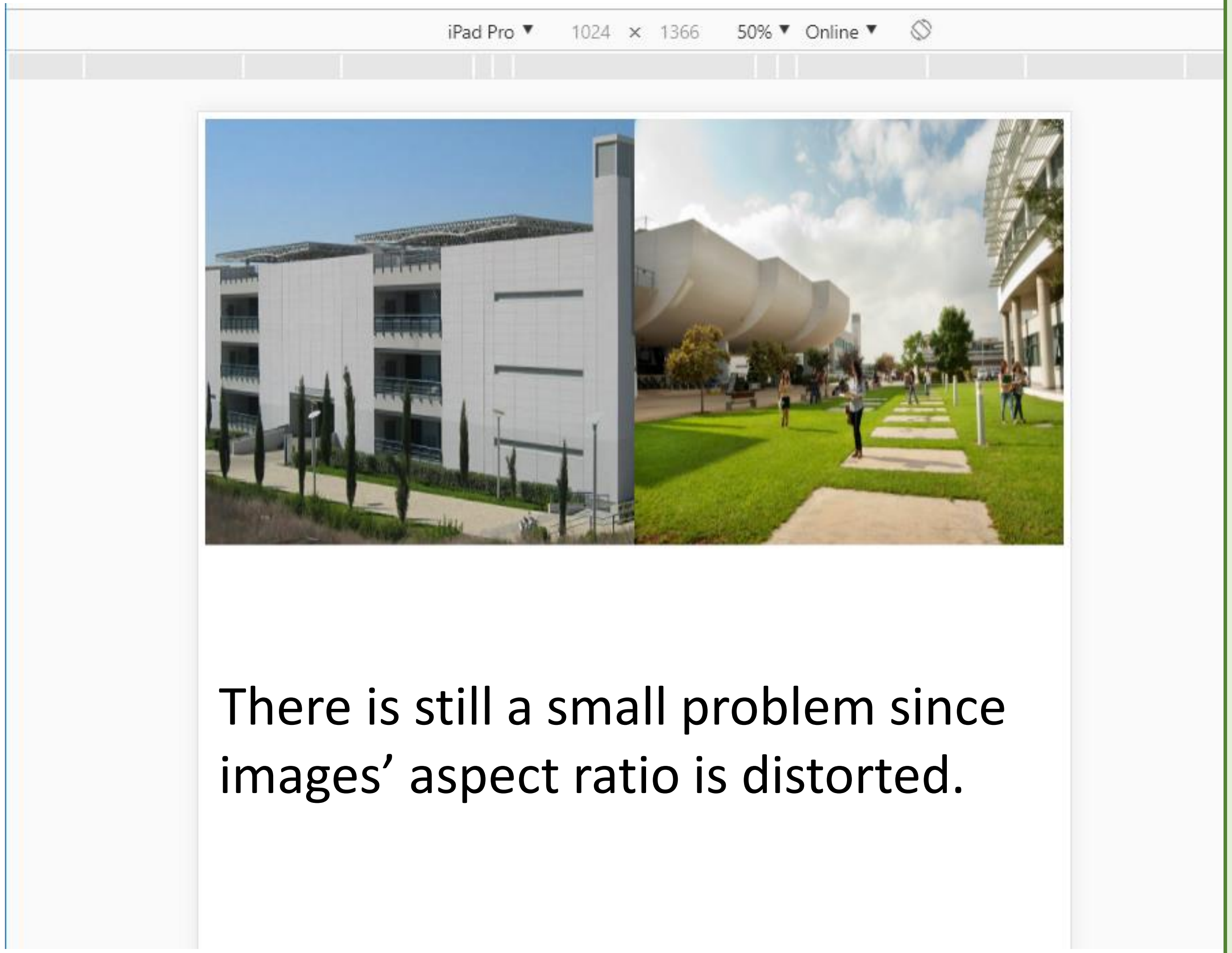
There is still a small problem since images is not of the same height.

CSS

```
.col {
  float: left;
  width: 50%;
}

img {
  width: 100%;
}
```

Example with images



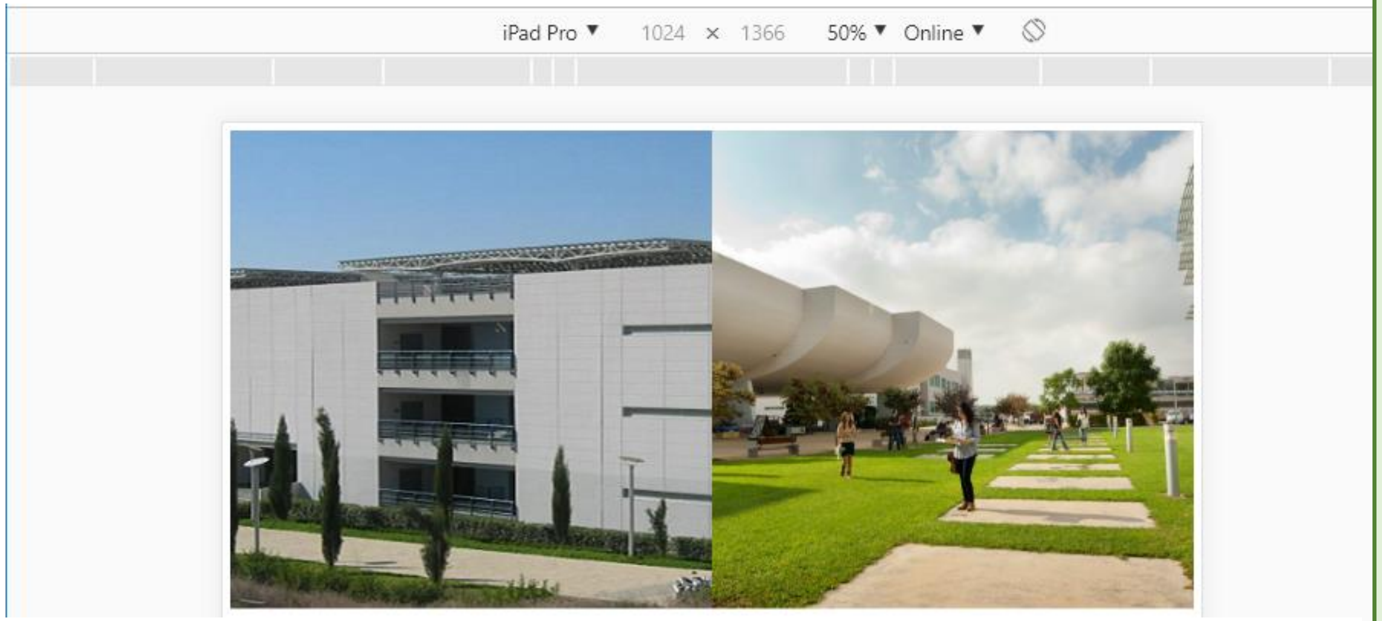
There is still a small problem since images' aspect ratio is distorted.

```css
.col {
   float: left;
   width: 50%;
   height: 500px;
}

img {
   width: 100%;
   height: 100%;
}
```

CSS

Example with images



The CSS `object-fit` property is used to specify how an `<img>` or `<video>` should be resized to fit its container.
`object-fit: cover;` cuts off the sides of the image, preserving the aspect ratio, and also filling in the space. See also [object-position]

CSS

```css
.col {
  float: left;
  width: 50%;
  height: 500px;
}

img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}
```

Important notice

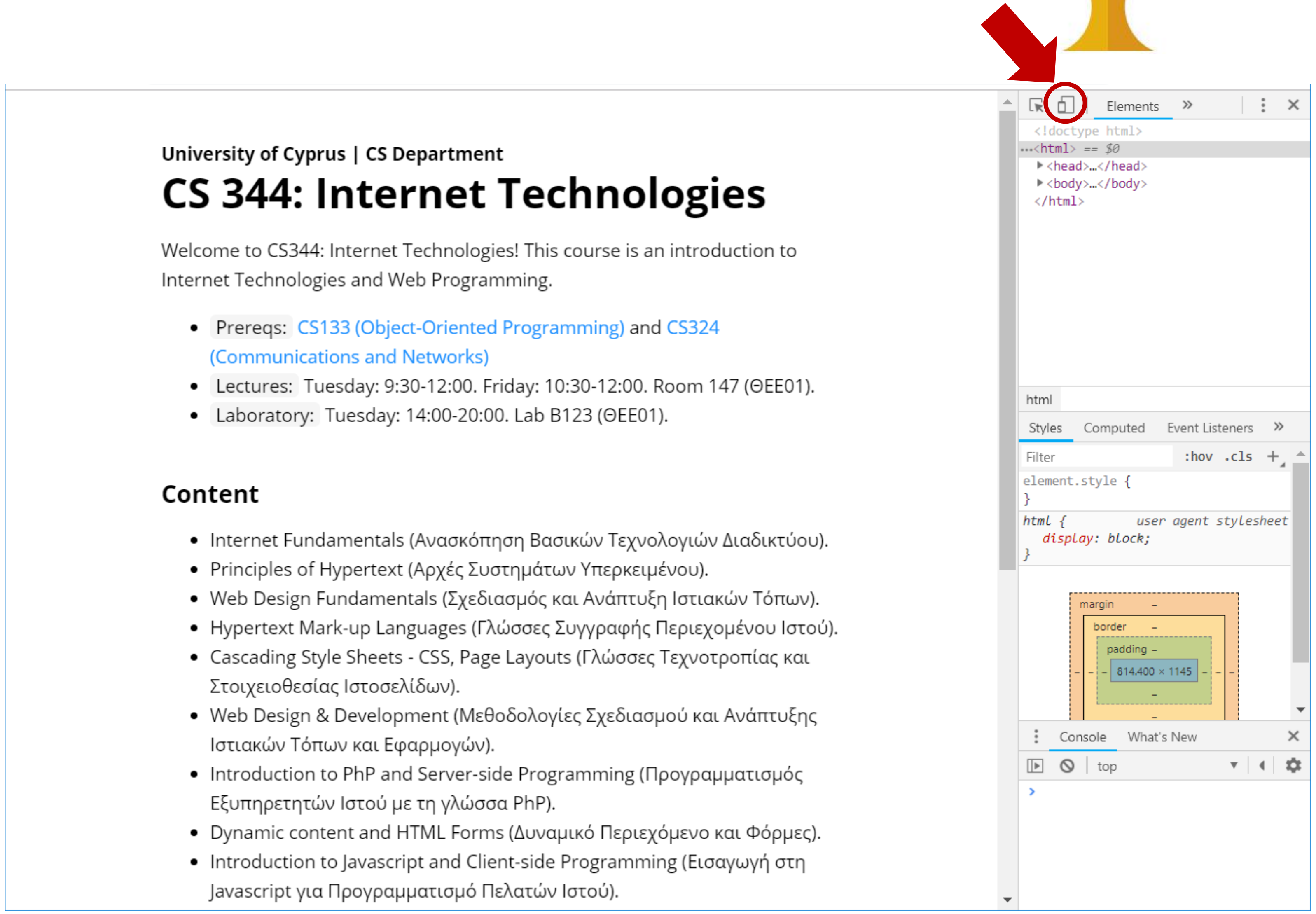
- By default in the CSS box model, the width and height you assign to an element is applied only to the element's content box.
- If the element has any border or padding, this is then **added** to the width and height to arrive at the size of the box that's rendered on the screen.
- When you set width and height, you have to adjust the value you give to allow for any border or padding that may be added.

Development strategies

- Practical question: How do you test mobile layouts?
  - Do you upload your HTML+CSS somewhere online and navigate to that URL on your phone?
  - Is there a way to connect your mobile phone to your local (laptop/deskop) device?
  - Do you run it in an Android/iOS emulator?
  - Other?

Chrome device mode

- You can simulate a web page in a mobile layout via Chrome device mode:
  - On website: Right click, and then select Inspect



University of Cyprus | CS Department

CS 344: Internet Technologies

Welcome to CS344: Internet Technologies! This course is an introduction to Internet Technologies and Web Programming.

- Prereqs: CS133 (Object-Oriented Programming) and CS324 (Communications and Networks)
- Lectures: Tuesday: 9:30-12:00. Friday: 10:30-12:00. Room 147 (ΘΕΕ01).
- Laboratory: Tuesday: 14:00-20:00. Lab B123 (ΘΕΕ01).

Content

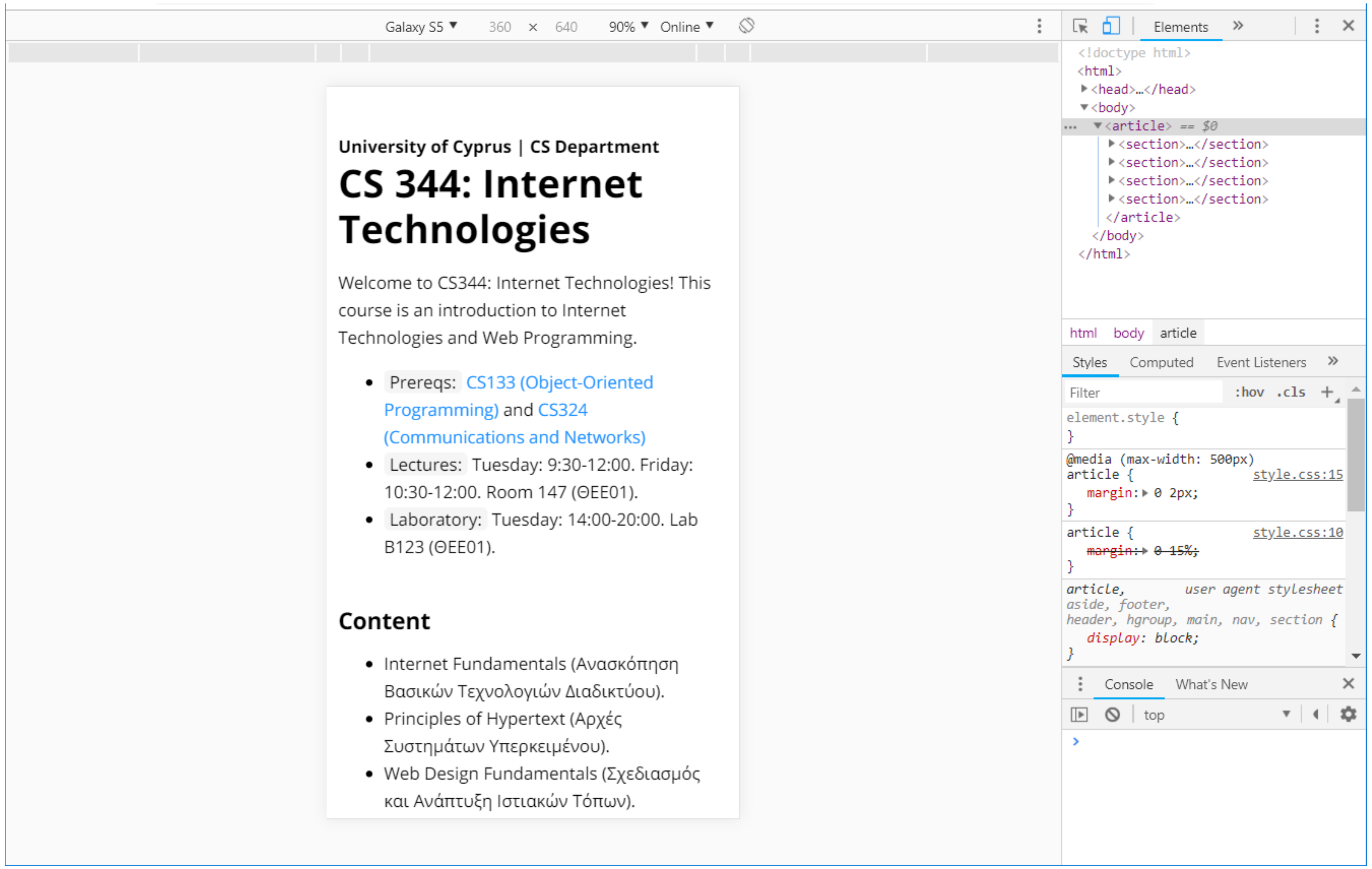
- Internet Fundamentals (Ανασκόπηση Βασικών Τεχνολογιών Διαδικτύου).
- Principles of Hypertext (Αρχές Συστημάτων Υπερκειμένου).
- Web Design Fundamentals (Σχεδιασμός και Ανάπτυξη Ιστιακών Τόπων).
- Hypertext Mark-up Languages (Γλώσσες Συγγραφής Περιεχομένου Ιστού).
- Cascading Style Sheets - CSS, Page Layouts (Γλώσσες Τεχνοτροπίας και Στοιχειοθεσίας Ιστοσελίδων).
- Web Design & Development (Μεθοδολογίες Σχεδιασμού και Ανάπτυξης Ιστιακών Τόπων και Εφαρμογών).
- Introduction to PhP and Server-side Programming (Προγραμματισμός Εξυπηρετητών Ιστού με τη γλώσσα PhP).
- Dynamic content and HTML Forms (Δυναμικό Περιεχόμενο και Φόρμες).
- Introduction to Javascript and Client-side Programming (Εισαγωγή στη Javascript για Προγραμματισμό Πελατών Ιστού).

```
<!doctype html>
…<html> == $0
  ▶<head>…</head>
  ▶<body>…</body>
</html>
```

```
element.style {
}
html {                user agent stylesheet
  display: block;
}
```

Chrome device mode

- You can simulate a web page in a mobile layout via [Chrome device mode](#):
    - On website: Right click, and then select Inspect


Galaxy S5 ▾ 360 × 640 90% ▾ Online ▾
Elements »
Styles Computed Event Listeners »
Filter :hov .cls +
Console What's New
top

University of Cyprus | CS Department

CS 344: Internet Technologies

Welcome to CS344: Internet Technologies! This course is an introduction to Internet Technologies and Web Programming.

- Prereqs: CS133 (Object-Oriented Programming) and CS324 (Communications and Networks)
- Lectures: Tuesday: 9:30-12:00. Friday: 10:30-12:00. Room 147 (ΘΕΕ01).
- Laboratory: Tuesday: 14:00-20:00. Lab B123 (ΘΕΕ01).

Content

- Internet Fundamentals (Ανασκόπηση Βασικών Τεχνολογιών Διαδικτύου).
- Principles of Hypertext (Αρχές Συστημάτων Υπερκειμένου).
- Web Design Fundamentals (Σχεδιασμός και Ανάπτυξη Ιστιακών Τόπων).

```
<!doctype html>
<html>
 ▸<head>…</head>
 ▾<body>
  ▾<article> == $0
    ▸<section>…</section>
    ▸<section>…</section>
    ▸<section>…</section>
    ▸<section>…</section>
   </article>
  </body>
</html>
```

html body article

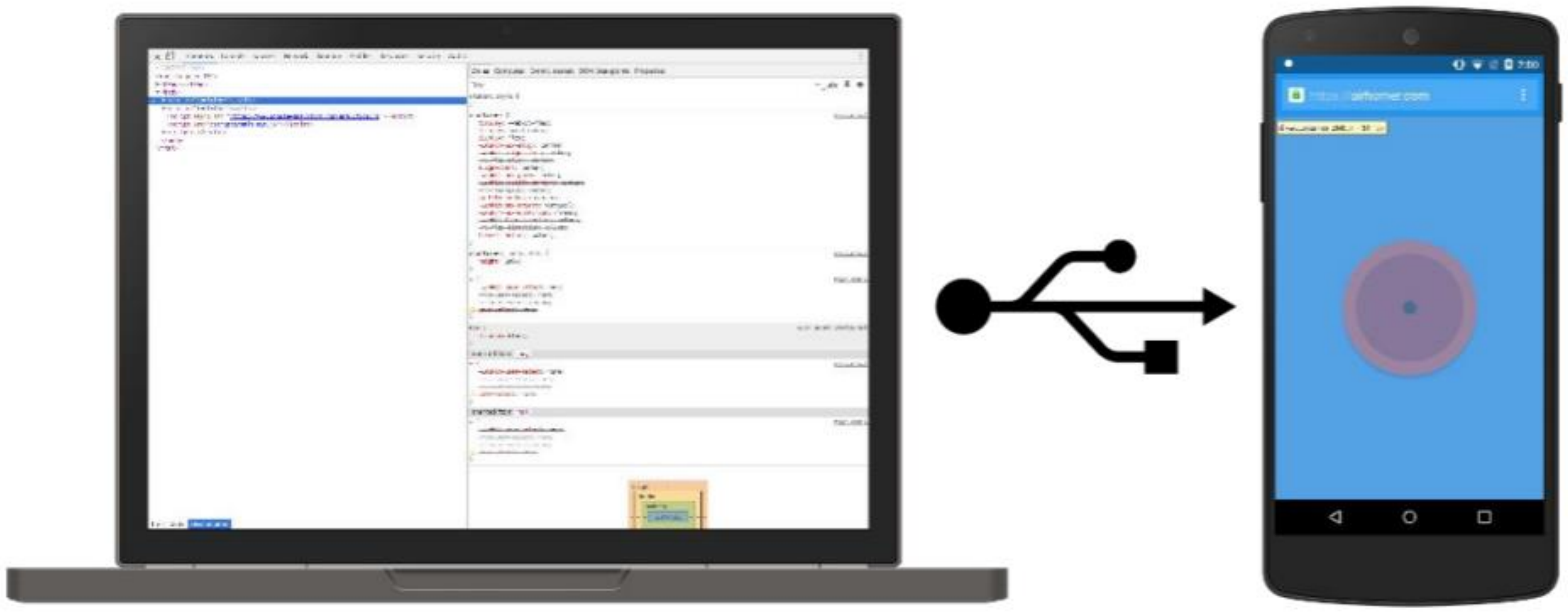
```
element.style {
}
@media (max-width: 500px)
article {                          style.css:15
  margin:▸ 0 2px;
}
article {                          style.css:10
  ~~margin:▸ 0 15%;~~
}
article,              user agent stylesheet
aside, footer,
header, hgroup, main, nav, section {
  display: block;
}
```

Chrome device mode

- **Advantages of Chrome device mode**:
  - Super convenient
  - Mostly accurate
- **Disadvantages of Chrome device mode**:
  - Not always accurate
  - iPhone particularly an issue
  - A little buggy
  - Doesn't simulate performance issues
- You should always test on real devices, too.

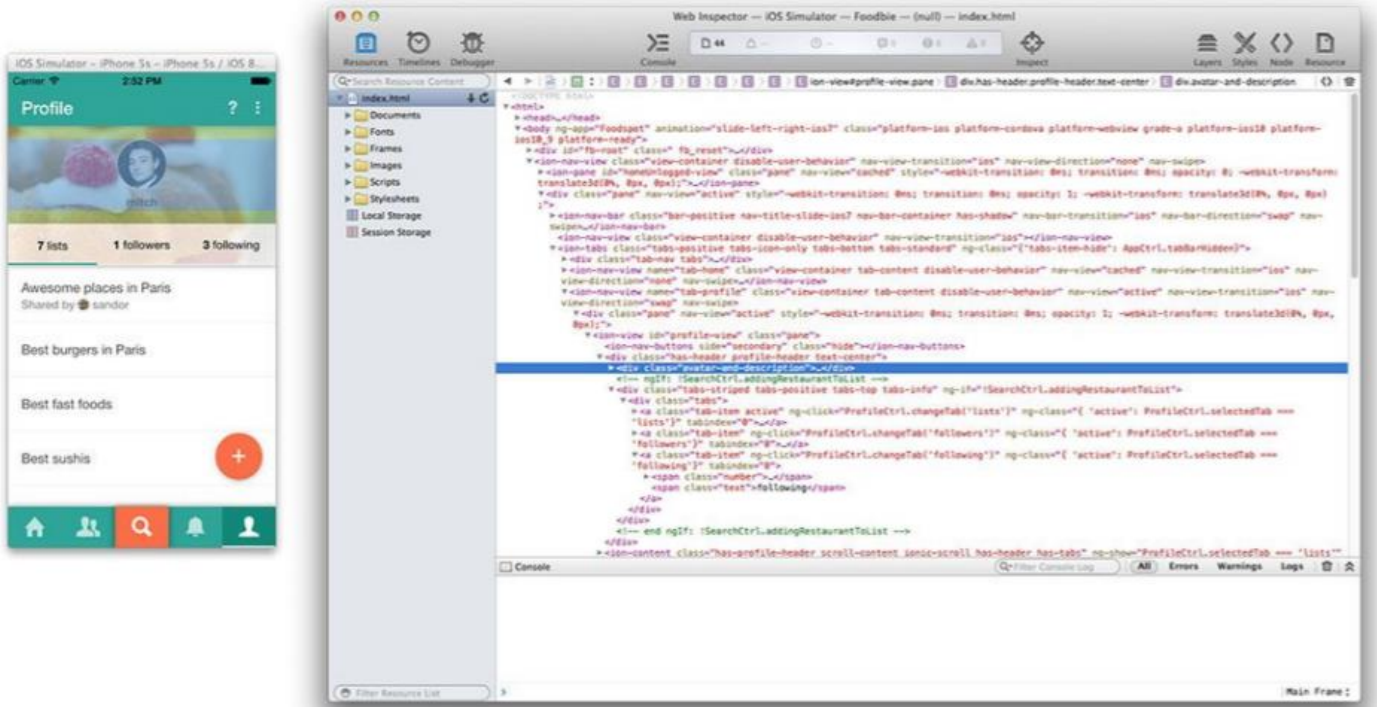
Chrome remote debugging

- If you have an Android phone, you can debug web pages on your phone via Chrome remote debugging.



Safari remote debugging

- If you have an iPhone, you can debug web pages on your phone via Safari remote debugging.



```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body ng-app="Foodspot" animation="slide-left-right-ios7" class="platform-ios platform-cordova platform-webview grade-a platform-ios10 platform-ios10_9 platform-ready">
    ▶<div id="fb-root" class=" fb_reset">…</div>
    ▼<ion-nav-view class="view-container disable-user-behavior" nav-view-transition="ios" nav-view-direction="none" nav-swipe>
      ▶<ion-pane id="homeUnlogged-view" class="pane" nav-view="cached" style="-webkit-transition: 0ms; transition: 0ms; opacity: 0; -webkit-transform: translate3d(0%, 0px, 0px);">…</ion-pane>
      ▼<div class="pane" nav-view="active" style="-webkit-transition: 0ms; transition: 0ms; opacity: 1; -webkit-transform: translate3d(0%, 0px, 0px);">
        ▶<ion-nav-bar class="bar-positive nav-title-slide-ios7 nav-bar-container has-shadow" nav-bar-transition="ios" nav-bar-direction="swap" nav-swipe>…</ion-nav-bar>
         <ion-nav-view class="view-container disable-user-behavior" nav-view-transition="ios"></ion-nav-view>
        ▼<ion-tabs class="tabs-positive tabs-icon-only tabs-bottom tabs-standard" ng-class="{'tabs-item-hide': AppCtrl.tabBarHidden}">
          ▶<div class="tab-nav tabs">…</div>
          ▶<ion-nav-view name="tab-home" class="view-container tab-content disable-user-behavior" nav-view="cached" nav-view-transition="ios" nav-view-direction="none" nav-swipe>…</ion-nav-view>
          ▼<ion-nav-view name="tab-profile" class="view-container tab-content disable-user-behavior" nav-view="active" nav-view-transition="ios" nav-view-direction="swap" nav-swipe>
            ▼<div class="pane" nav-view="active" style="-webkit-transition: 0ms; transition: 0ms; opacity: 1; -webkit-transform: translate3d(0%, 0px, 0px);">
              ▼<ion-view id="profile-view" class="pane">
                 <ion-nav-buttons side="secondary" class="hide"></ion-nav-buttons>
                ▼<div class="has-header profile-header text-center">
                  ▶<div class="avatar-and-description">…</div>
                   <!-- ngIf: !SearchCtrl.addingRestaurantToList -->
                  ▼<div class="tabs-striped tabs-positive tabs-top tabs-info" ng-if="!SearchCtrl.addingRestaurantToList">
                    ▼<div class="tabs">
                      ▶<a class="tab-item active" ng-click="ProfileCtrl.changeTab('lists')" ng-class="{ 'active': ProfileCtrl.selectedTab === 'lists'}" tabindex="0">…</a>
                      ▶<a class="tab-item" ng-click="ProfileCtrl.changeTab('followers')" ng-class="{ 'active': ProfileCtrl.selectedTab === 'followers'}" tabindex="0">…</a>
                      ▼<a class="tab-item" ng-click="ProfileCtrl.changeTab('following')" ng-class="{ 'active': ProfileCtrl.selectedTab === 'following'}" tabindex="0">
                        ▶<span class="number">…</span>
                         <span class="text">following</span>
                       </a>
                     </div>
                   </div>
                   <!-- end ngIf: !SearchCtrl.addingRestaurantToList -->
                 </div>
                ▶<ion-content class="has-profile-header scroll-content ionic-scroll has-header has-tabs" ng-show="ProfileCtrl.selectedTab === 'lists'"
```

Access local web server from mobile phone

- Run a web server locally on your laptop/desktop
  - E.g. run XAMPP locally
- Connect laptop/desktop and mobile phone in the same network (e.g., in the same WiFi)
- Find the IP of your laptop/desktop (where web server runs)
  - type ipconfig (into CMD for Windows) or ifconfig (into terminal for Unix)
    - `sudo apt-get install net-tools` to install ifconfig on Unix
- On your mobile phone browser type http://WEBSERVER-IP-ADDRESS/index.html (index.html can be omitted)

Mobile summary

- Always add the **meta viewport tag**
- Use **@media queries** to add styles for devices with certain characteristics, such as screen width
- Use the **Chrome Device Mode** to simulate mobile rendering on desktop
- For `height` and `width`, prefer percentages
- Autoscale image and videos to fit in screen region
- For fonts, prefer `em` and `rem` (see Appendix)
- Try to minimize dependent rules
  - Changing the width of one container force you to change 15 other properties to look right
- More on responsive web design

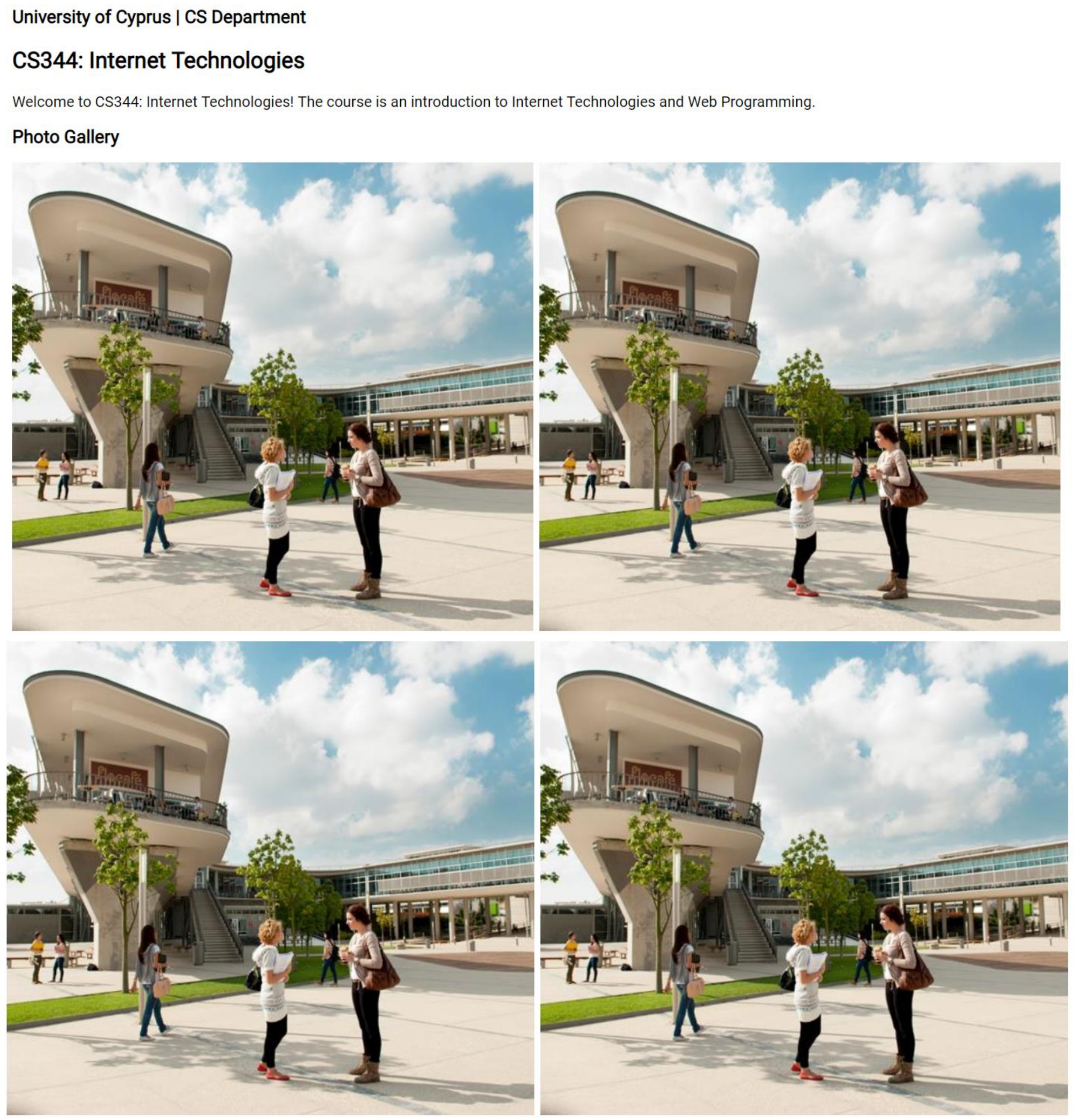
Exercise 1

- Create the responsive webpage shown in the next slides using the given guidelines.

Exercise 1

screen-size >= 1024

- SEE NEXT SLIDES FOR MORE DETAILS
- CSS code for screen sizes >= 1024px can be placed outside media queries



University of Cyprus | CS Department

CS344: Internet Technologies

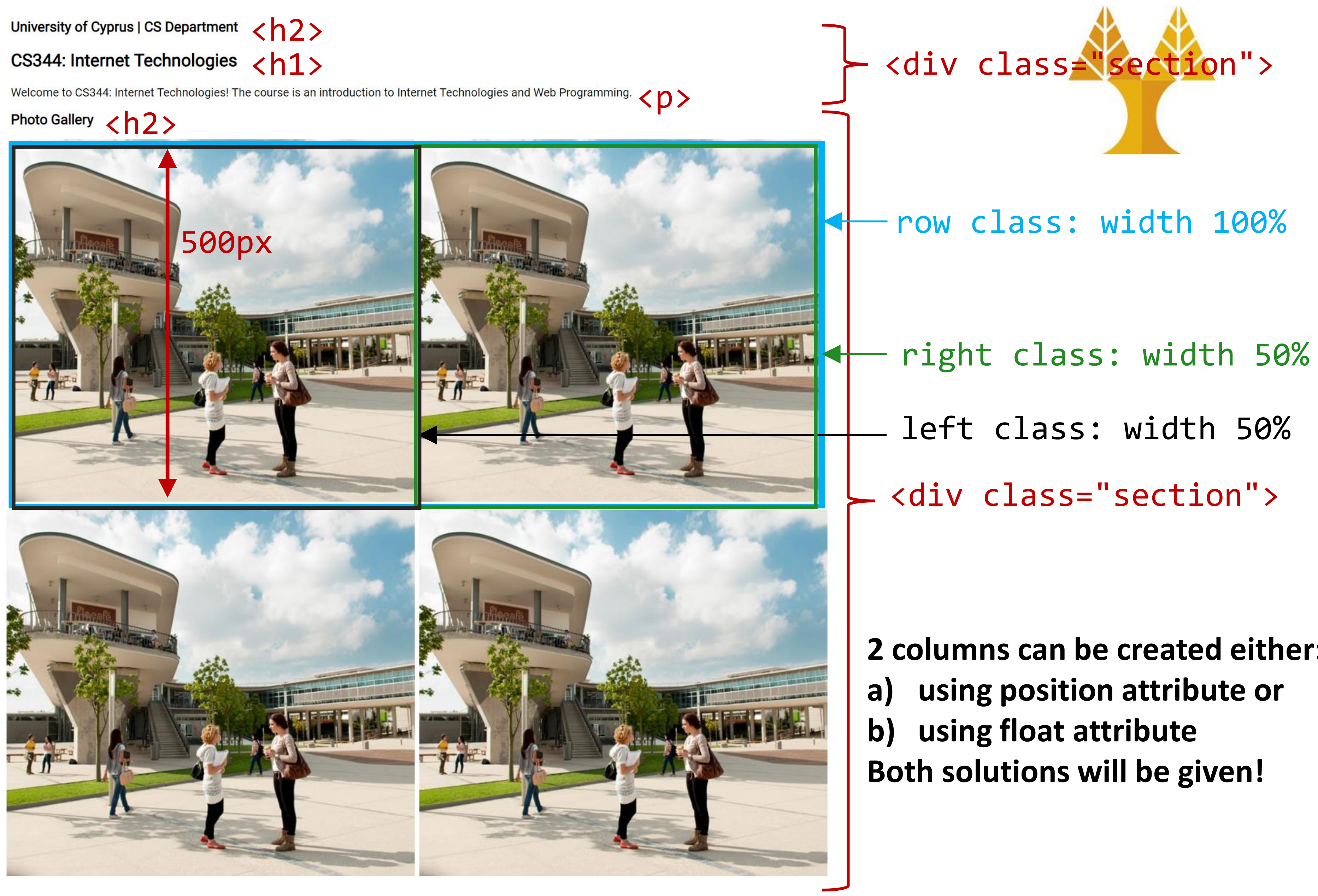
Welcome to CS344: Internet Technologies! The course is an introduction to Internet Technologies and Web Programming.

Photo Gallery

Exercise 1

screen-size >= 1024

- h1: 1.5em
- h2: 1.2em
- Images height 500px, margin right & bottom 1%, object-fit: cover (set the width properly)



University of Cyprus | CS Department

CS344: Internet Technologies

Welcome to CS344: Internet Technologies! The course is an introduction to Internet Technologies and Web Programming.

Photo Gallery

2 columns can be created either:

a) using position attribute or

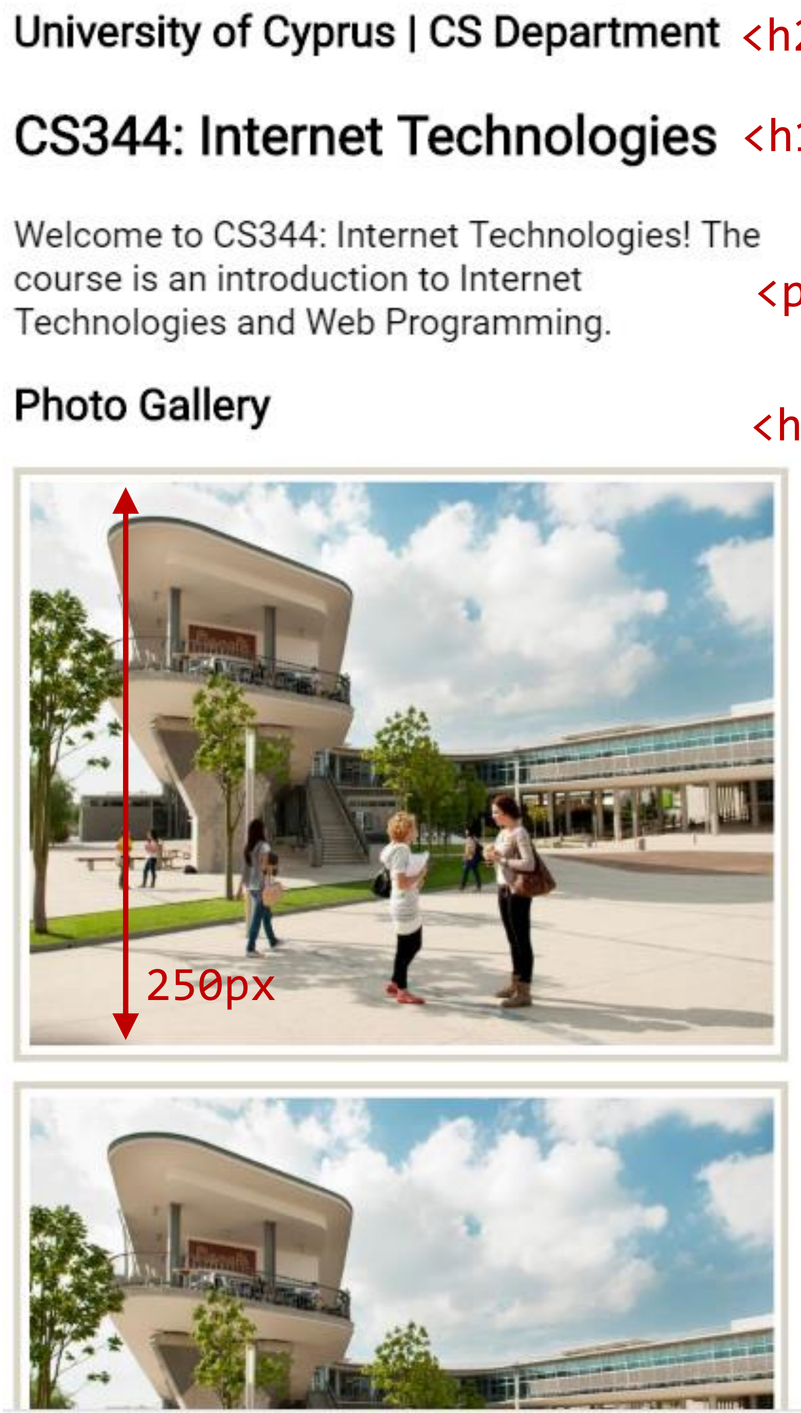
b) using float attribute

Both solutions will be given!

Exercise 1

screen-size < 1024

- h1: 1.5em
- h2: 1.2em
- Images height: 250px, only margin bottom 10px
- Border 3px with color #D9D4C6 and padding 4px
- Use media queries for styling rules that are modified in smaller screens



University of Cyprus | CS Department <h2>

CS344: Internet Technologies <h1>

Welcome to CS344: Internet Technologies! The course is an introduction to Internet Technologies and Web Programming. <p>

<div class="section">

Photo Gallery <h2>

<div class="section">

APPENDIX: Relative font sizes

percent, `em`, `rem`

Relative units

- Whenever possible, it's best to use **relative units** (like percentage) instead of absolute units (like px).
- Advantages:
  - More likely to work on different screen sizes
  - Easier to reason about; fewer magic numbers
    10% / 80% / 10% vs 122px / 926px / 122px

Relative font sizes: percent

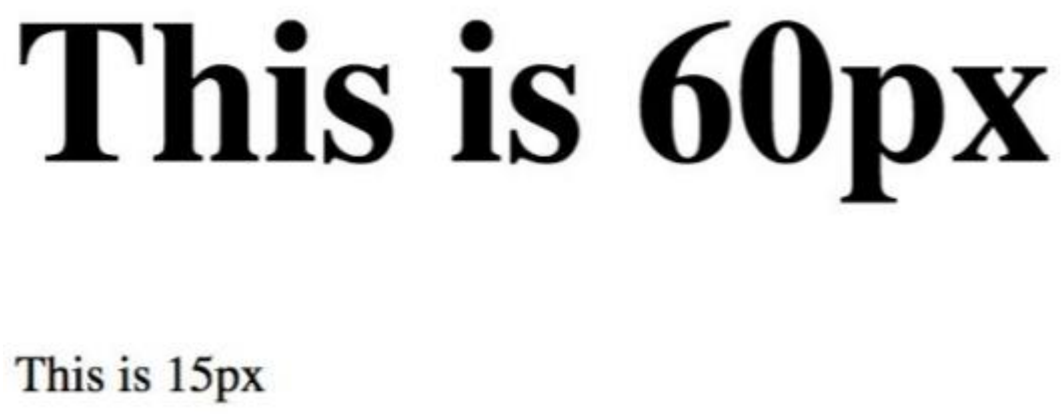
- You can define font sizes in terms of percentage:

HTML

```html
<body>
   <h1>This is 60px</h1>
   <p>This is 15px</p>
</body>
```

CSS

```css
body {
   font-size: 30px;
}
h1 {
   font-size: 200%;
}
p {
   font-size: 50%;
}
```



This is 60px

This is 15px

Relative font sizes: percent

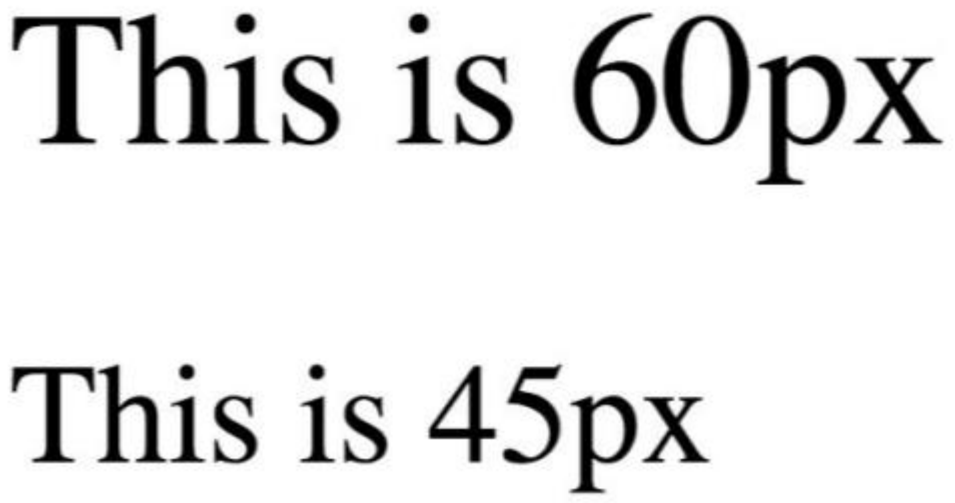
- Percent on font-size behaves exactly like percentage on width and height, in that it's relative to the parent:

HTML

```
<div>
   This is 60px
   <p>This is 45px</p>
</div>
```

CSS

```
body {
   font-size: 30px;
}
div {
   font-size: 200%;
}
p {
   font-size: 75%;
}
```



This is 60px

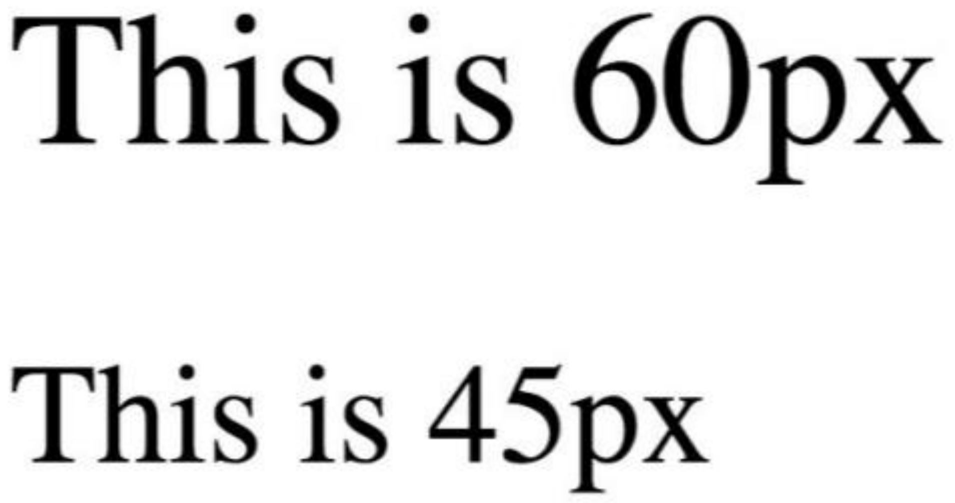
This is 45px

Relative font sizes: percent

- Percent on font-size behaves exactly like percentage on width and height, in that it's relative to the parent:

HTML
```html
<div>
   This is 60px
   <p>This is 45px</p>
</div>
```

CSS
```css
body {
   font-size: 30px;
}
div {
   font-size: 200%;
}
p {
   font-size: 75%;
}
```


This is 60px

This is 45px

p is 75% of its parent, which is 200% of 30px.
p's size: 0.75*2*30 = 45px

Relative font sizes: `em`

- But instead of percentages, relative font sizes are usually defined in terms of `em`
- `em` represents the calculated `font-size` of the element
  - `1em` = the inherited font size
  - `2em` = 2 times the inherited font size
- In other words,

`font-size: 1em;` is the same as `font-size: 100%;`

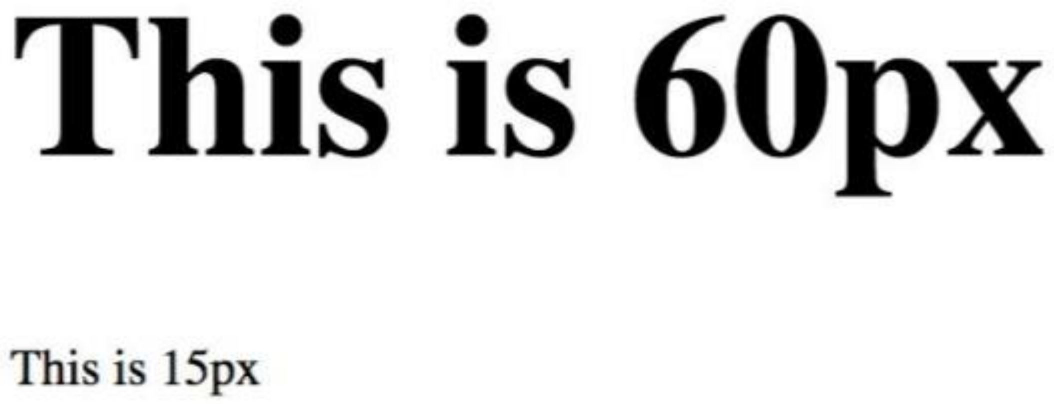
Relative font sizes: `em`

HTML

```
<body>
   <h1>This is 60px</h1>
   <p>This is 15px</p>
</body>
```

CSS

```
body {
   font-size: 30px;
}
h1 {
   font-size: 2em;
}
p {
   font-size: .5em;
}
```



This is 60px

This is 15px

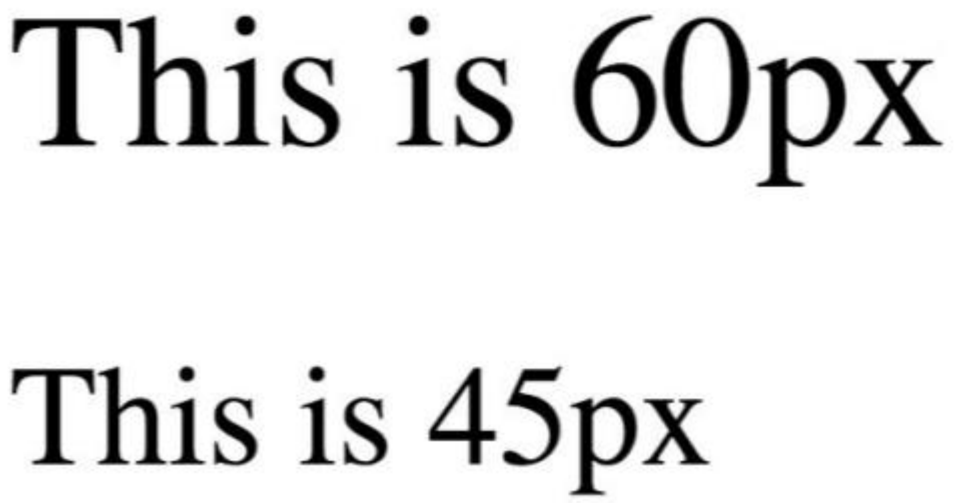
Relative font sizes: **`em`**

HTML

```
<div>
   This is 60px
   <p>This is 45px</p>
</div>
```

CSS

```
body {
   font-size: 30px;
}
div {
   font-size: 2em;
}
p {
   font-size: .75em;
}
```



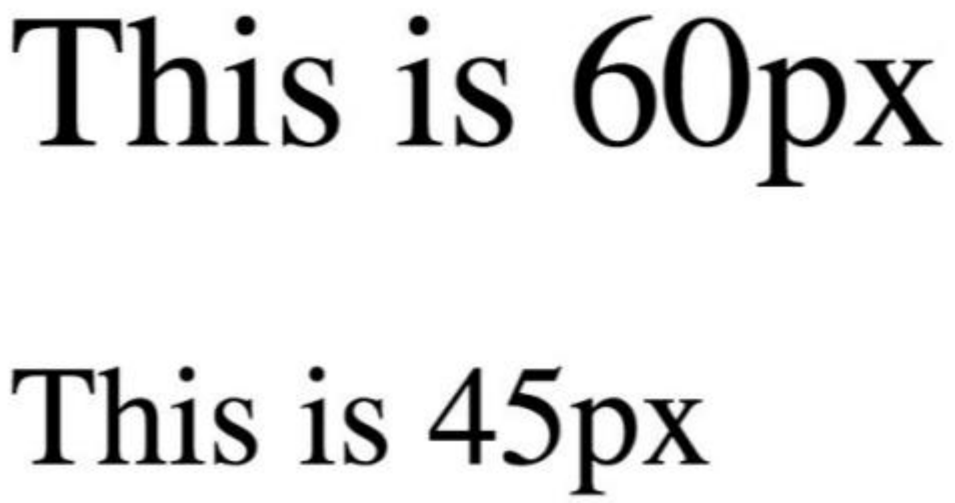
This is 60px

This is 45px

Relative font sizes: **`em`**

HTML
```html
<div>
   This is 60px
   <p>This is 45px</p>
</div>
```

CSS
```css
body {
   font-size: 30px;
}
div {
   font-size: 2em;
}
p {
   font-size: .75em;
}
```


This is 60px

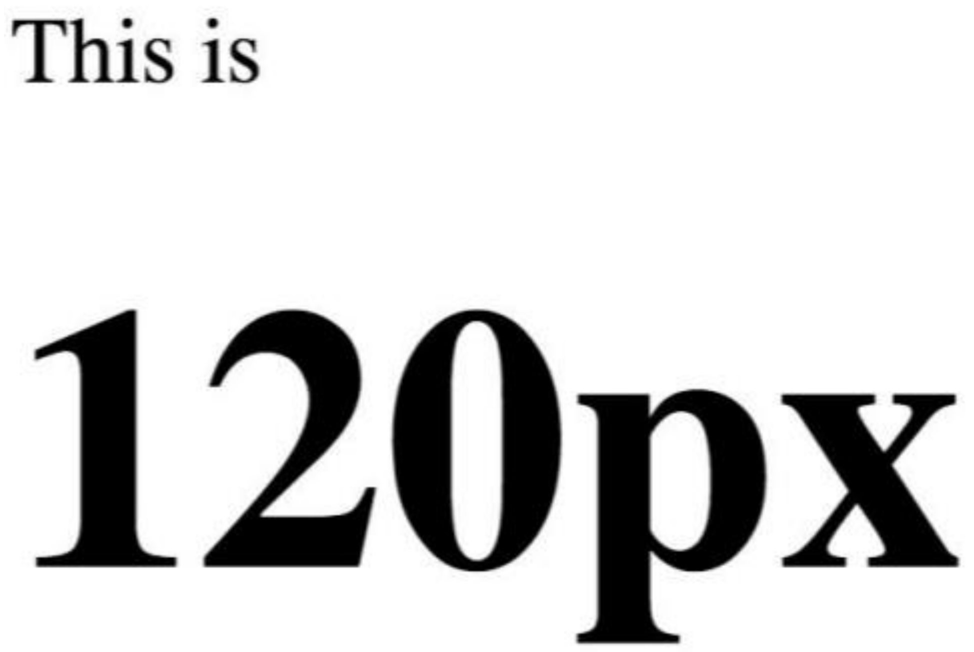
This is 45px

p's inherited font size is 2em, which is 60px.
p's size: 0.75em*60 = 45px

Relative font sizes: `em`

```
<body>
   This is
   <h1>
      <strong>120px</strong>
   </h1>
</body>
```
HTML

```
body {
   font-size: 30px;
}
strong {
   font-size: 2em;
}
```
CSS


This is

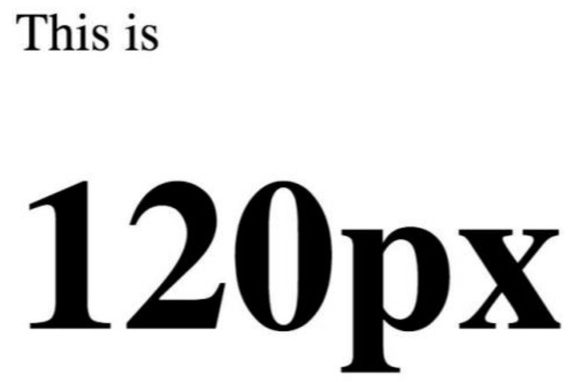
120px

Wait, why is **120px and not 60px?**

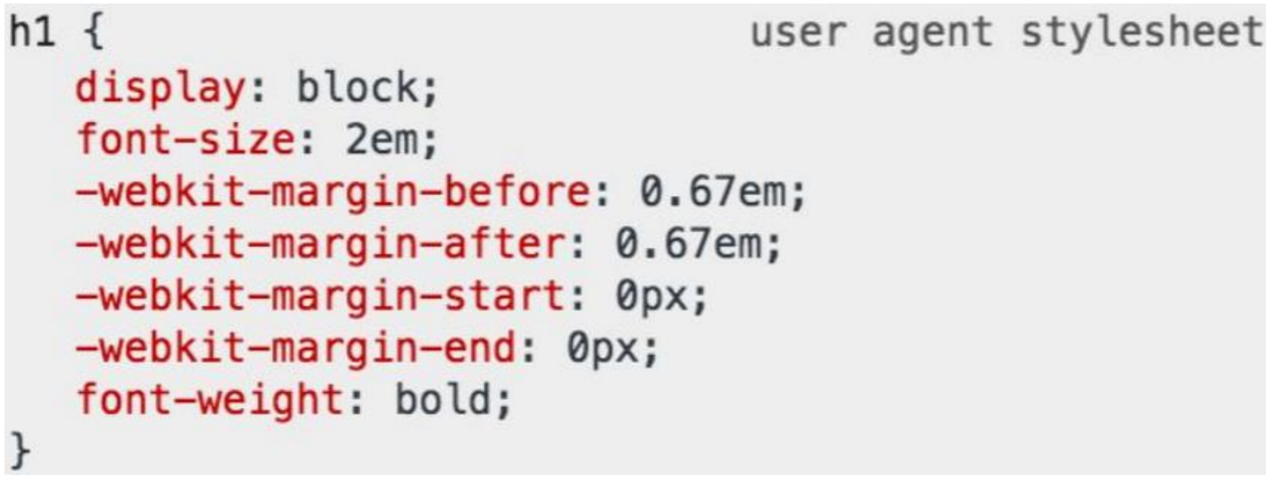
Relative font sizes: **em**

HTML
```
<body>
   This is
   <h1>
      <strong>120px</strong>
   </h1>
</body>
```

CSS
```
body {
   font-size: 30px;
}
strong {
   font-size: 2em;
}
```


This is

120px


```
h1 {                                   user agent stylesheet
  display: block;
  font-size: 2em;
  -webkit-margin-before: 0.67em;
  -webkit-margin-after: 0.67em;
  -webkit-margin-start: 0px;
  -webkit-margin-end: 0px;
  font-weight: bold;
}
```

In the Chrome Inspector, we see the default browser font-size for h1 is 2em.
So it's 30*2*2 = 120px.

Relative font sizes: `rem`

- If you do not want your relative font sizes to compound through inheritance, use `rem`
- `rem` represents the `font-size` of the **root** element (`<html>`)
  - `1rem` = the `<html>` font size (which for most browsers has a default value of 16px).
  - `2rem` = 2 times root font size

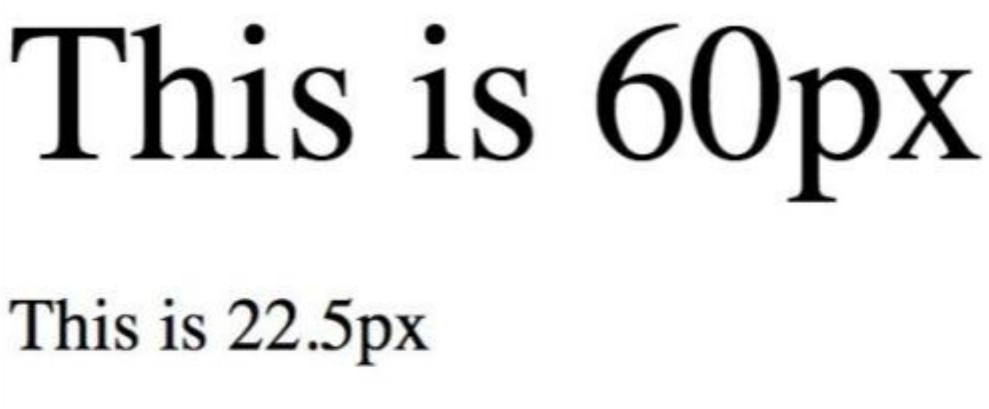
Relative font sizes: `rem`

HTML

```html
<body>
   <div>
      This is 60px
      <p>This is 22.5px</p>
   </div>
</body>
```

CSS

```css
html {
   font-size: 30px;
}
div {
   font-size: 2rem;
}
p {
   font-size: .75rem;
}
```



This is 60px

This is 22.5px

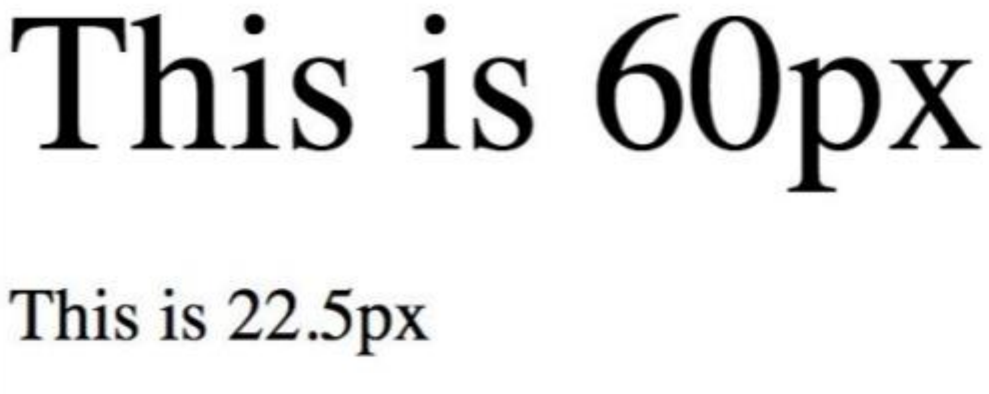
Relative font sizes: `rem`

HTML

```
<body>
   <div>
      This is 60px
      <p>This is 22.5px</p>
   </div>
</body>
```

CSS

```
html {
   font-size: 30px;
}
div {
   font-size: 2rem;
}
p {
   font-size: .75rem;
}
```



This is 60px

This is 22.5px

`font-size` is set on the `html` element, not body (or any other tag)

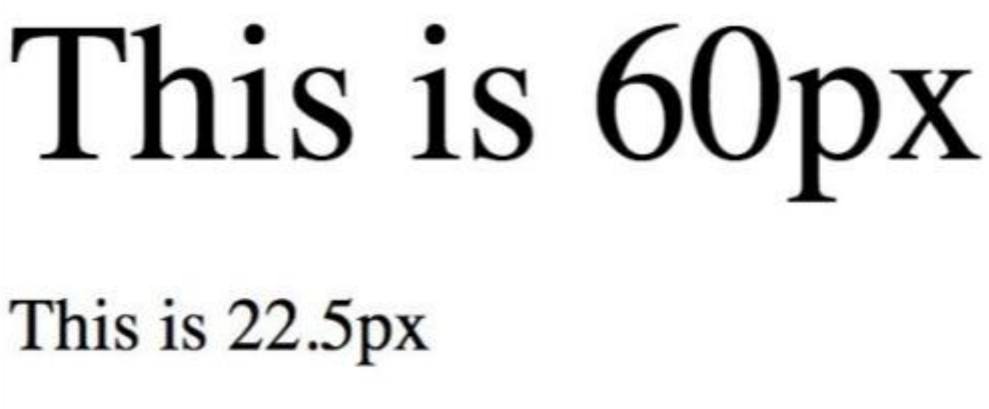
Relative font sizes: `rem`

HTML

```html
<body>
   <div>
      This is 60px
      <p>This is 22.5px</p>
   </div>
</body>
```

CSS

```css
html {
   font-size: 30px;
}
div {
   font-size: 2rem;
}
p {
   font-size: .75rem;
}
```


This is 60px

This is 22.5px

`.75em` is calculated from the root, which is `30px`, so `30*.75 = 22.5px`.

Relative font conclusions

- Use relative fonts for the same purpose as using relative `heights` and `widths`:
  - Prefer `em` and `rem` over percentages
    - Not for any deep reason, but `em` is meant for font so it's semantically cleaner
  - Use `rem` to avoid compounding sizes
  - In addition to font-size, you may consider `em`/`rem` for:
    - `line-height`
    - `margin-top`
    - `margin-bottom`