```
/*Lecture 67: Introduction to the CSS Box Model
According to the box model, each and every element on
   a webpage can be seenas a rectangular box and each
   box can have a width, a height, padding, margins and
   a border. And we can specify all this using CSS
   properties.

Note that they are all optional —so there can be boxes
   wwhere there are no paddings / margins / borders.

PARTS OF A BOX
Content of box — contains contents
—We can specify the height and width of the box using
   'height' and 'width' properties.

Padding of box — transparent area around the content,
   but still inside of the box.
— We use padding to create whitespace inside of a box
   and we can specify padding using the 'padding'
   property.

Border of box — Border goes around the padding and the
   content.

Margin of box — Margin is whitespace that is created
   outside of the box. We can think of margin as the
   space between boxes.
—So for instance, if we wanted to create space between
   two elements, we will simply use their margin
   properties.

— Fill Area of box: Remember how text content and
   images go inside the content box? Well , the same
   does not apply for background images or the
   background colour of a box. These properties will
   actually be applied not only to the content box, but
   to the entire fill area — which INCLUDES THE PADDING
   AND THE BORDER (but NOT THE MARGIN).
*/


/*
Notice also that before implementing all of this
```

```
          (below), the elements already have some space
          between them. This is because CSS3 has some default
          values for paddings and margins for some elements,
          which would be assumed if the paddings and margins
          weren't specified.
23
24     Typically, we want to reset the paddings and margins
          of all the elements on the page, so we do something
          called a GLOBAL RESET. To do that, we need to select
          ALL the elements.
25     */
26     /* Global Reset */
27     * {
28       margin: 0;
29       padding: 0;
30       box-sizing: border-box;
31       /* Note that you should set your box sizings to
          border box. If you don't when you create an
          element, let's say: width of 100 px, and padding
          of 20px. You actually get an element with 140 px
          (100 + 20 + 20).
32
33       Border box ensures that the box sizing is fixed at
          what you declare so it will always keep the width
          at 100 px, irregardless of the padding.
34       */
35     }
36
37     body {
38       background-color: rgb(255, 247, 201);
39       font-family: Arial;
40       font-size: 20px;
41       padding: 50px; /* this will apply padding to the 4
          sides of the element, but we can also do that
          selectively. */
42     }
43
44     h1 {
45       font-size: 35px;
46       margin-bottom: 25px; /* like so */
47     }
48
```

```css
h2 {
  margin-bottom: 20px;
  text-align: center; /* {left, right, center} - this
     centers text within the box. NOTE: Only block
     elements have a padding and margin!! It doesn't
     create a box on a whole line.*/
}

p {
  margin-bottom: 20px;
}

.first {
  color: red;
}

#your-name {
  /* Element id */
  background-color: rgb(255, 220, 105);
  border: 5px solid #444; /* shorthand property -
     thickness, style, colour. Also notice that this
     border property is NOT inherited, if not all the
     child elemebts would also have a 5 pixel grey
     solid border.*/
  width: 400px;
  padding: 25px;
  margin-top: 30px;
}

input,
button {
  padding: 10px;
  font-size: 16px;
} /* apply padding and fontsize to two different
   elements */

a {
  background-color: yellowgreen; /* see that the box
     of this element is NOT the width of the screen */
}

#course-image {
```

```css
82      width: 300px;
83    }
84
85    #your-name h2 {
86      /* Child Selector - Selecting h2 selectors inside a
           form or something */
87      color: olivedrab;
88    }
89
```