Lecture 2

Introduction to CSS

What is CSS?

- Cascading Style Sheets or CSS allow you to control the layout and look of your page easily.
- CSS tags or properties are easy to use and affect the look and feel or style of your pages.

Overview of CSS

- CSS was first developed in 1997, as a way for Web developers to define the look and feel of their Web pages. It was intended to allow developers to separate content from design so that HTML could perform more of the function that it was originally based on - the mark-up of content, without worry about the design and layout.

Why CSS?

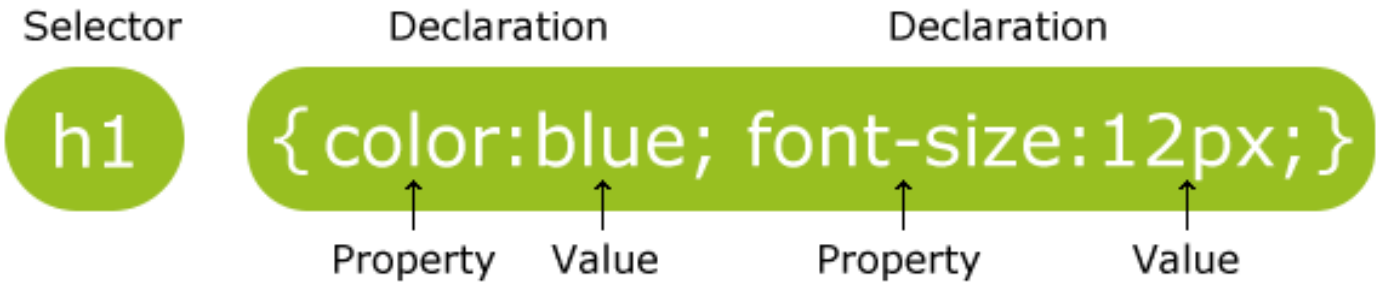
- CSS removes the presentation attributes from the structure allowing reusability, ease of maintainability, and an interchangeable presentation layer.
- CSS allows us to make global and instantaneous changes easily.

Parts of CSS

- There are essentially two parts to Cascading Style Sheets:
- the styles
- the stylesheet
  - i.e. style properties — the styles;
  - and the placement of those properties — the stylesheet.

CSS Syntax

- A CSS rule has two main parts: a selector, and one or more declarations:



CSS Syntax

- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute you want to change. Each property has a value.

CSS Example

- A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets:
- **p {color:red;text-align:center;}**
- To make the CSS more readable, you can put one declaration on each line, like this:
- **p**
  {
  color:red;
  text-align:center;
  }

CSS Comments

- Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.
- A CSS comment begins with "/*", and ends with "*/", like this:
- **/*This is a comment*/**
  p
  {
  text-align:center;
  /*This is another comment*/
  color:black;
  font-family:arial;
  }

The id and class Selectors

- In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

The id Selector

- The id selector is used to specify a style for a single, unique element.
- The id selector uses the id attribute of the HTML element, and is defined with a "#".
- The style rule below will be applied to the element with id=“header":
- #header
  {
  text-align:center;
  color:red;
  }

The Class Selector

- The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.
- This allows you to set a particular style for many HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a "."
- In the example below, all HTML elements with class="center" will be center-aligned:
- .center {text-align:center;}

Three different ways to insert CSS

- Local (Inline style)
  - confined to a **single element (tag)**
- Internal
  - affect elements in an **entire page**
- External
  - can affect **multiple pages**
- **Precedence**
  - **Local > Internal > External**

Inline Style

- To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:
- ** This is a paragraph. **

Internal Style

- An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:
- **<head>**
  <style>
  hr {color:sienna;}
  p {margin-left:20px;}
  body
  {background-image:url("images/back40.gif");}
  </style>
  </head>

External Style

- An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:
- **<head>**
  <link rel="stylesheet"
  type="text/css" href="mystyle.css">
  </head>

External Style

- An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:
- hr {color:sienna;}
  p {margin-left:20px;}

Beyond HTML elements
ID and Class Selectors

Classes and IDS

- HTML elements can be CSS selectors, but as we saw with the universal selector *, they're not the *only* selectors available.
- There are two important selectors you can use in addition to the universal selector and HTML elements: **classes**and **IDs**.
- Open up sample1.html and sample1.css

Keeping it classy

- Classes are useful when you have a bunch of elements that should all receive the same styling. Rather than applying the same rules to several selectors, you can simply apply the same class to all those HTML elements, then define the styling for that class in the CSS tab.
- Classes are assigned to HTML elements with the word class and an equals sign, like so:
- ** **

Classes in CSS

- Classes are identified in CSS with a dot (.), like so:

- **.square**
  {
  height: 100px;
  width: 100px;
  }

  This allows you to take elements of different types and give them the same styling.

Class Demo

- Create any number of HTML elements you like and give them the class "fancy". On the CSS tab, set .fancy to have a font-family of cursive and a color of #0000CD and note the differences.
- **Open up fancy.html**

ID Selector

- IDs, on the other hand, are great for when you have exactly *one* element that should receive a certain kind of styling.
- IDs are assigned to HTML elements with the word id and an equals sign:
- <div id="first”>
- IDs are identified in CSS with a pound sign (#):
- #first
- {
- height: 50px;
- }
- This allows you to apply style to a single instance of a selector, rather than *all* instances.

Class Demo

- Check out sample2.html in the editor.
- On the HTML tab:
  - Give the h2 header an ID of"intro".
  - Give the first h3 and first p a class of "standout". Don't do anything to the second h3 and p!
  - On the CSS tab:
  - Set the #intro ID's color to#B83C3A.
  - Set the .standout class's color to#F7AC5F and font-family to Verdana.

Pseudo-class selector

- A **pseudo-class selector** is a way of accessing HTML items that aren't part of the document tree.
- For instance, it's very easy to see where a link is in the tree. But where would you find information about whether a link had been clicked on or not? It isn't there!

Pseudo-class syntax

- The CSS syntax for pseudo selectors is

```
selector:pseudo-class_selector
{
 property: value;
 }
```

- It's just that little extra colon (:).
- Open up hover.html and hover.css and observe what happens.

Links

- There are a number of useful pseudo-class selectors for links, including:
  - a:link: An unvisited link.
  - a:visited: A visited link.
  - a:hover: A link you're hovering your mouse over.

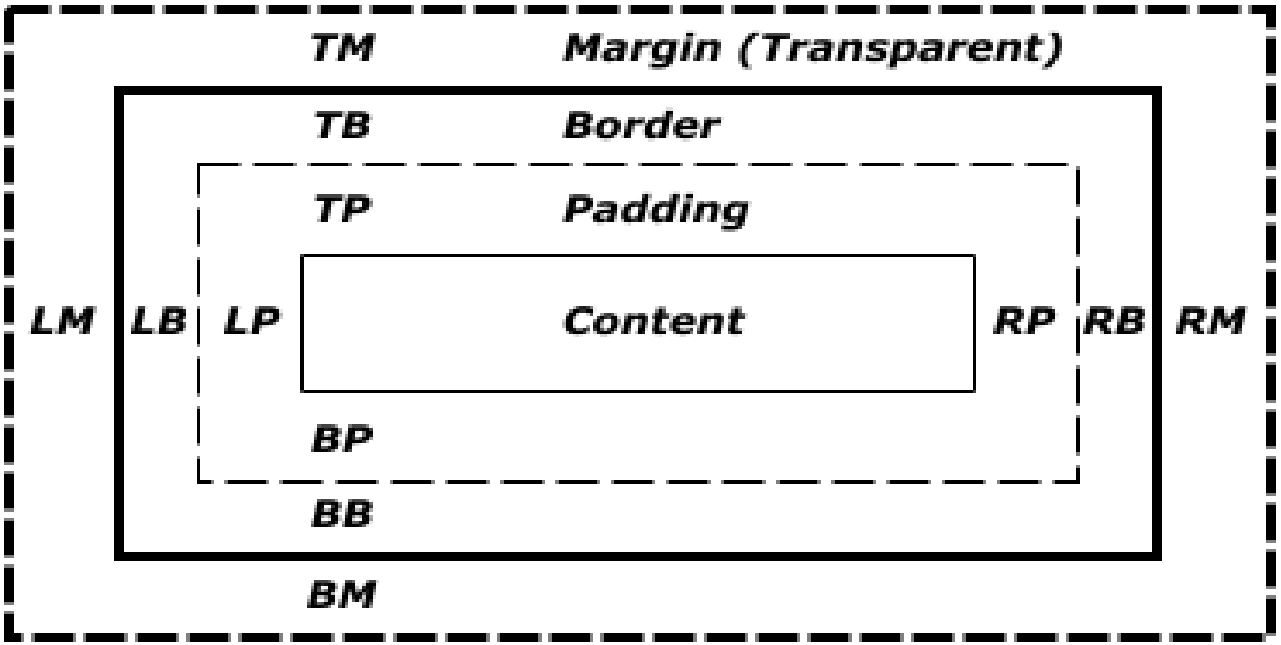
CSS Positioning

- Open up position.html and position.css on the editor, and view it on the browser.
- As you saw, the outermost box of each element went all the way across the page. This is why until now, your HTML elements have been sitting on top of one another: by default, they take up the full width of the page.

The display property

- With this property, there are four possible values:
- **block**: This makes the element a block box. It won't let anything sit next to it on the page! It takes up the full width.
- **inline-block**: This makes the element a block box, but will allow other elements to sit next to it on the same line.
- **inline**: This makes the element sit on the same line as another element, but without formatting it like a block. It only takes up as much width as it needs (not the whole line).
- **none**: This makes the element and its content disappear from the page entirely!

The CSS Box Model



Margin edge

Border edge

Padding edge

Content edge

The CSS Box Model

- The **margin** is the space around the element. The larger the margin, the more space between our element and the elements around it. We can adjust the margin to move our HTML elements closer to or farther from each other.
- The **border** is the edge of the element. It's what we've been making visible every time we set the border property.
- The **padding** is the spacing between the content and the border. We can adjust this value with CSS to move the border closer to or farther from the content.
- The **content** is the actual "stuff" in the box. If we're talking about a <p>element, the "stuff" is the text of the paragraph.

Margin

- Let's start with our margins. Adjusting our margins not only moves our element relative to other elements on the page, but also relative to the "walls" of the HTML document.
- For instance, if we take an HTML element with a specific width (such as our in the editor) and set its margin to **auto**, this tells the document to automatically put equal left and right margins on our element, centering it on the page.
- Practice with margin.html and

Margin top, right, bottom, left

- If you want to specify a particular margin, you can do it like this:
  - **margin-top: /*some value*/**
  - **margin-right: /*some value*/**
  - **margin-bottom: /*some value*/**
  - **margin-left: /*some-value*/**
- You can also set an element's margins all at once: you just start from the top margin and go around clockwise (going from top to right to bottom to left). For instance,
  - **margin: 1px 2px 3px 4px;**
- will set a top margin of 1 pixel, a right margin of 2, a bottom of 3, and a left of 4.

Padding

- Padding can be set in two ways, just like your margins. You can either select them individually, Or select them all in one declaration.
- You should also know that if you want your padding to be the same for all four sides, you can declare that value only once.
- **padding: 10px** will give your HTML element 10 pixels of padding on all sides.

Other Positioning

- Float
- Clear
- Absolute, relative, fixed and static positioning

Beyond HTML elements
ID and Class Selectors

Classes and IDS

- HTML elements can be CSS selectors, but as we saw with the universal selector *, they're not the *only* selectors available.
- There are two important selectors you can use in addition to the universal selector and HTML elements: **classes**and **IDs**.
- Open up sample1.html and sample1.css

Keeping it classy

- Classes are useful when you have a bunch of elements that should all receive the same styling. Rather than applying the same rules to several selectors, you can simply apply the same class to all those HTML elements, then define the styling for that class in the CSS tab.
- Classes are assigned to HTML elements with the word class and an equals sign, like so:
- ** **

Classes in CSS

- Classes are identified in CSS with a dot (.), like so:

- **.square**
  {
  height: 100px;
  width: 100px;
  }

  This allows you to take elements of different types and give them the same styling.

Class Demo

- Create any number of HTML elements you like and give them the class "fancy". On the CSS tab, set .fancy to have a font-family of cursive and a color of #0000CD and note the differences.
- **Open up fancy.html**

ID Selector

- IDs, on the other hand, are great for when you have exactly *one* element that should receive a certain kind of styling.
- IDs are assigned to HTML elements with the word id and an equals sign:
- <div id="first”>
- IDs are identified in CSS with a pound sign (#):
- #first
- {
- height: 50px;
- }
- This allows you to apply style to a single instance of a selector, rather than *all* instances.

Class Demo

- Check out sample2.html in the editor.
- On the HTML tab:
  - Give the h2 header an ID of"intro".
  - Give the first h3 and first p a class of "standout". Don't do anything to the second h3 and p!
  - On the CSS tab:
  - Set the #intro ID's color to#B83C3A.
  - Set the .standout class's color to#F7AC5F and font-family to Verdana.

Pseudo-class selector

- A **pseudo-class selector** is a way of accessing HTML items that aren't part of the document tree.
- For instance, it's very easy to see where a link is in the tree. But where would you find information about whether a link had been clicked on or not? It isn't there!

Pseudo-class syntax

- The CSS syntax for pseudo selectors is

```
selector:pseudo-class_selector
{
 property: value;
 }
```

- It's just that little extra colon (:).
- Open up hover.html and hover.css and observe what happens.

Links

- There are a number of useful pseudo-class selectors for links, including:
  - a:link: An unvisited link.
  - a:visited: A visited link.
  - a:hover: A link you're hovering your mouse over.

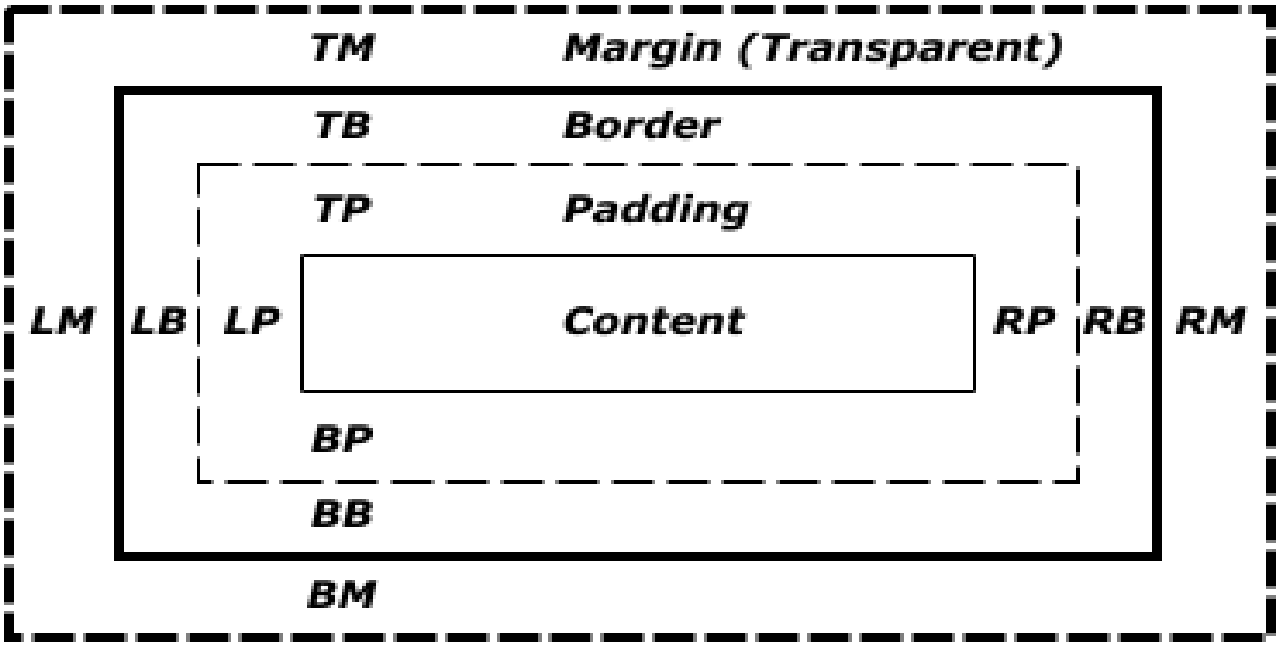
CSS Positioning

- Open up position.html and position.css on the editor, and view it on the browser.
- As you saw, the outermost box of each element went all the way across the page. This is why until now, your HTML elements have been sitting on top of one another: by default, they take up the full width of the page.

The display property

- With this property, there are four possible values:
- **block**: This makes the element a block box. It won't let anything sit next to it on the page! It takes up the full width.
- **inline-block**: This makes the element a block box, but will allow other elements to sit next to it on the same line.
- **inline**: This makes the element sit on the same line as another element, but without formatting it like a block. It only takes up as much width as it needs (not the whole line).
- **none**: This makes the element and its content disappear from the page entirely!

The CSS Box Model



Margin edge

Border edge

Padding edge

Content edge

The CSS Box Model

- The **margin** is the space around the element. The larger the margin, the more space between our element and the elements around it. We can adjust the margin to move our HTML elements closer to or farther from each other.
- The **border** is the edge of the element. It's what we've been making visible every time we set the border property.
- The **padding** is the spacing between the content and the border. We can adjust this value with CSS to move the border closer to or farther from the content.
- The **content** is the actual "stuff" in the box. If we're talking about a <p>element, the "stuff" is the text of the paragraph.

Margin

- Let's start with our margins. Adjusting our margins not only moves our element relative to other elements on the page, but also relative to the "walls" of the HTML document.
- For instance, if we take an HTML element with a specific width (such as our in the editor) and set its margin to **auto**, this tells the document to automatically put equal left and right margins on our element, centering it on the page.
- Practice with margin.html and

Margin top, right, bottom, left

- If you want to specify a particular margin, you can do it like this:
  - **margin-top: /*some value*/**
  - **margin-right: /*some value*/**
  - **margin-bottom: /*some value*/**
  - **margin-left: /*some-value*/**
- You can also set an element's margins all at once: you just start from the top margin and go around clockwise (going from top to right to bottom to left). For instance,
  - **margin: 1px 2px 3px 4px;**
- will set a top margin of 1 pixel, a right margin of 2, a bottom of 3, and a left of 4.

Padding

- Padding can be set in two ways, just like your margins. You can either select them individually, Or select them all in one declaration.
- You should also know that if you want your padding to be the same for all four sides, you can declare that value only once.
- **padding: 10px** will give your HTML element 10 pixels of padding on all sides.

Other Positioning

- Float
- Clear
- Absolute, relative, fixed and static positioning