

Cascading Style Sheets

Unit 2

Refer Web technology a developers perspective

by N.P.Gopalan/J.Akhilandeswari

Introduction

- CSS describes **how HTML elements are to be displayed on screen**
- It can control the layout of multiple web pages all at once

Introduction

- A CSS rule-set consists of a selector and a declaration block
- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces

```
h1{color:blue;}
```

1.The element Selector

```
p {  
  text-align: center;  
  color: red;  
}
```

2.The id Selector

- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- The style rule below will be applied to the HTML element with id="para1":

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

3.Class selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.
- In the example below, all HTML elements with class="center" will be red and center-aligned

```
.center {  
text-align: center;  
color: red;  
}
```

4.Grouping Selectors

```
h1, h2, p {  
    text-align: center;  
    /* This is a single-line comment */  
    color: red;  
}
```

CSS Examples

1.<h1 style="background-color:DodgerBlue;">Hello World</h1>

```
2.body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-x;  
}
```

```
3.body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed; /* image should be fixed (will not scroll  
with the rest of the page)*/  
}
```


Examples

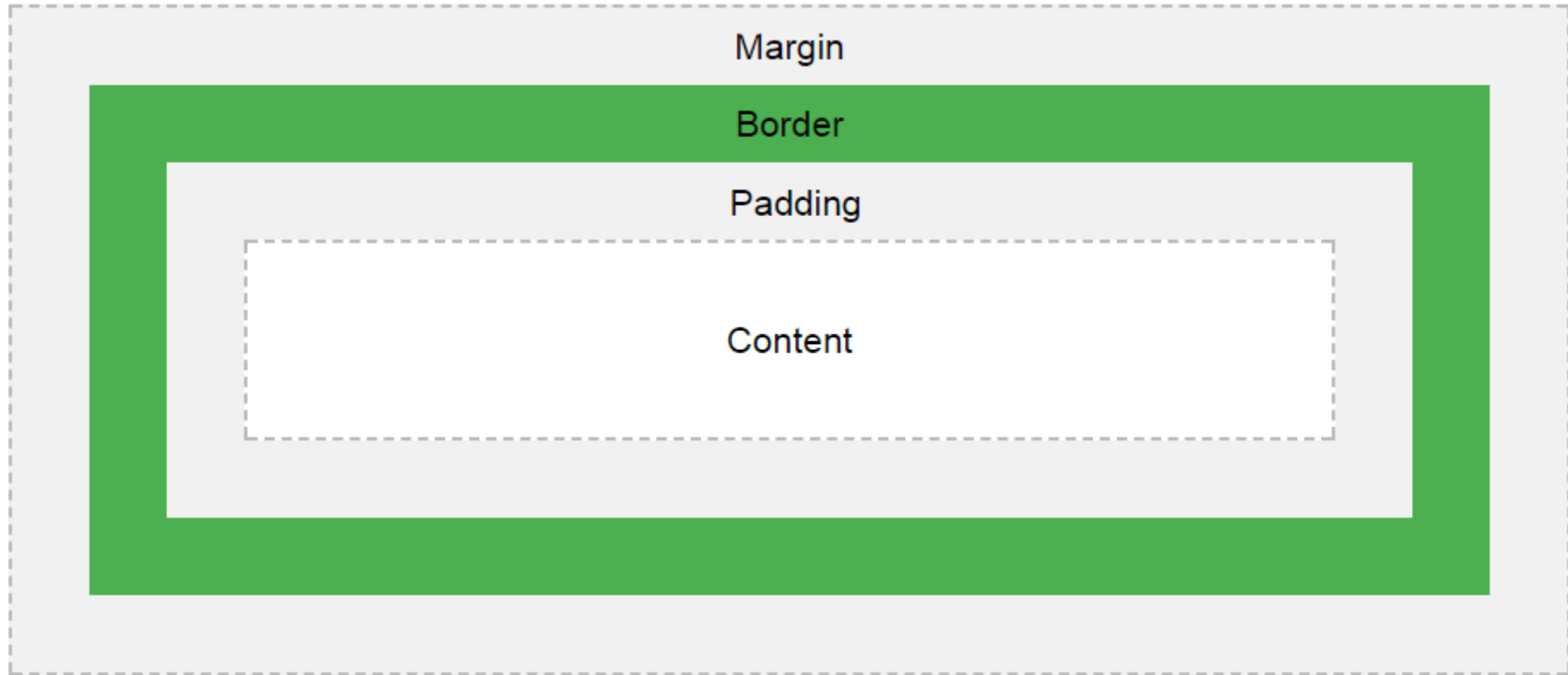
```
4. p {  
    margin: 25px 50px 75px 100px;/*top right bottom left*/  
}
```

```
5. div {  
    max-width: 500px;  
    height: 100px;  
    background-color: powderblue;  
}
```

```
6.p {  
    font-family: "Times New Roman", Times, serif;  
}
```

```
7.h1 {  
    font-size: 40px;  
}
```

CSS box model



CSS box model

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example CSS box model

- `div {
width: 300px;
border: 25px solid green;
padding: 25px;
margin: 25px;
}`

Three Ways to Insert CSS

1. External style sheet
2. Internal style sheet
3. Inline style

External Style Sheet

- With an external style sheet, you can change the look of an entire website by changing just one file
- Each page must include a reference to the external style sheet file inside the `<link>` element.
- The `<link>` element goes inside the `<head>` section:

```
<head>
```

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

```
</head>
```

External Style Sheet

- The file should not contain any html tags. The style sheet file must be saved with a .css extension.
- Here is how the "mystyle.css" looks:

```
body {  
    background-color: lightblue;  
}
```

```
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

Internal Style Sheet

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the `<style>` element, inside the `<head>` section of an HTML page:

Internal Style Sheet

```
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```

Inline Styles

- Used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
- `<h1 style="color:blue;margin-left:30px;">This is a heading</h1>`

Inline Styles

- Suppose multiple styles defined for h1 tag one in external and other in internal then which style is applied?

```
<head>
```

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

```
<style>
```

```
h1 {
```

```
    color: orange;
```

```
}
```

```
</style>
```

```
</head>
```

Inline Styles

- h1 will be orange

Styling priority

1. Inline style (inside an HTML element)
 2. External and internal style sheets (in the head section)
 3. Browser default
- So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the `<head>` tag, or in an external style sheet, or a browser default value.

Additional information

CSS-Input

```
input {  
    width: 100%;  
}
```

- The example above applies to all <input> elements. If you only want to style a specific input type, you can use attribute selectors:
- `input[type=text]` - will only select text fields
- `input[type=password]` - will only select password fields

CSS-Input

```
input[type=text] {  
    width: 100%;  
    padding: 12px 20px;  
    margin: 8px 0;  
    box-sizing: border-box;  
}  
input[type=text] {  
    border: 2px solid red;  
    border-radius: 4px;  
}  
input[type=text] {  
    border: none;  
    border-bottom: 2px solid red;  
}
```


CSS-Input

- `input[type=text] {
 background-color: #3CBC8D;
 color: white;
}`
- `input[type=text]:focus {
 background-color: lightblue;
}`
- `input[type=text]:focus {
 border: 3px solid;
}`

CSS-Input

```
input[type=text] {  
    background-color: white;  
    background-image: url('searchicon.png');  
    background-position: 10px 10px;  
    background-repeat: no-repeat;  
    padding-left: 40px;  
}
```

```
textarea {  
    width: 100%;  
    height: 150px;  
    padding: 12px 20px;  
    border: 2px solid;  
    border-radius: 4px;  
    background-color: #f8f8f8;  
    resize: none;  
}
```

CSS-select/button

- ```
select {
 width: 100%;
 padding: 16px 20px;
 border: none;
 background-color: #f1f1f1;
}
```
- ```
input[type=button], input[type=submit], input[type=reset] {  
  background-color: #4CAF50;  
  border: none;  
  color: white;  
  padding: 16px 32px;  
  text-decoration: none;  
  margin: 4px 2px;  
  cursor: pointer; /*zoom-in,zoom-out/*  
}
```

Few Questions with Solutions

Semester Questions

1. Explain the different ways to style html page?-4 marks
2. Compare html/xml and css/xsl?-4 marks
3. What do you understand by css ?explain different types css with examples.--6 marks
4. Write the html code to create scrolling display containing image with a border on page?-6 marks
5. Built dropdown menu using css and html only?-4 marks

Semester Questions

d) Design a cascaded style sheet for the following HTML document:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title>Home page</title></head>
```

```
<body>
```

```
<p>Home</p>
```

```
<a href="contact.html">Contact us </a>
```

```
</body>
```

```
</html>
```

Solution question4

```
<marquee behavior="scroll" direction="left">  
  
</marquee>
```

Solution question4:Create A Hoverable Dropdown

Create a dropdown menu that appears when the user moves the mouse over an element.

Step 1) Add HTML:

```
<div class="dropdown">  
  <button class="dropbtn">Dropdown</button>  
  <div class="dropdown-content">  
    <a href="#">Link 1</a>  
    <a href="#">Link 2</a>  
    <a href="#">Link 3</a>  
  </div>  
</div>
```


Solution question4:Create A Hoverable Dropdown

- Use any element to open the dropdown menu, e.g. a `<button>`, `<a>` or `<p>` element.
- Use a container element (like `<div>`) to create the dropdown menu and add the dropdown links inside it.
- Wrap a `<div>` element around the button and the `<div>` to position the dropdown menu correctly with CSS.

Solution question4:Create A Hoverable Dropdown

Step 2) Add CSS:

- ```
/* Dropdown Button */
.dropbtn {
 background-color:blue;
 color: white;
 padding: 16px;
 font-size: 16px;
 border: none;
}
```

```
/* The container <div> - needed to position the dropdown content */
.dropdown {
 position: relative;
 display: block;
}
```

# Solution question4:Create A Hoverable Dropdown

- ```
/* Dropdown Content (Hidden by Default) */  
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color:grey;  
  min-width: 160px;  
}  
  
/* Links inside the dropdown */  
.dropdown-content a {  
  color: black;  
  padding: 12px 16px;  
  text-decoration: none;  
  display: block;  
}
```

Solution question4:Create A Hoverable Dropdown

- ```
/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color:pink}

/* Show the dropdown menu on hover */
.dropdown:hover .dropdown-content {
 display: block;
}

/* Change the background color of the dropdown button when the
dropdown content is shown */
.dropdown:hover .dropbtn {
 background-color: red;
}
```

# Solution question4:Create A Hoverable Dropdown

Dropdown

Dropdown

Link 1

Link 2

Link 3

Dropdown

Link 1

Link 2

Link 3



# Hello World!

This example contains some advanced CSS methods you may not have learned yet. But, we will explain these methods in a later chapter in the tutorial.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
 margin-left: 200px;
 background: #5d9ab2 url("img_tree.png") no-repeat top left;
}
.center_div {
 width: 90%;
 background-color: #d0f0f6;
 text-align:left;
 padding: 8px;
}
</style>
```

```
</head>
```

```
<body>
```

```
<div class="center_div">
```

```
<h1>Hello World!</h1>
```

```
<p>This example contains some advanced CSS methods you may not
have learned yet. But, we will explain these methods in a later chapter
in the tutorial.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```