

IT1100 - Internet and Web Technologies

Lecture 03

Basics of the client-side development

Re Cap - Recap - Lecture 01

1. Data networks and the Internet
2. Network Services and Protocols
3. Web server and the Browser
4. Markup languages

Recap - Lecture 02

1. Distributed systems and their architectures
2. Main concepts of web
3. E-commerce systems and forms

HTML - Web forms

Recap

- Demo – html5_forms_Demo.html

Student Registrations

File | E:/OneDrive/OneDrive%20-%2...

Online TAXI

Customer Details

Customer Name

Customer Phone Number

Email

Pick-up Address

Vehicle Details

Vehicle Type

Air Condition
☒ with A/C ☐ without A/C

Tour Staff
☒ with Driver ☐ with Tour guide

Trip date

starting date Time

End date Time

Content

- Development of the **browser-based** web application uses 3 main technologies in the client-side
 1. HTML – To develop the content
 2. CSS – To format/decorate the content
 3. JavaScript (JS) – To develop the application (processing) components

1. What is and Why CSS?

- CSS – Cascading Style Sheets
- Used to
 - Decorate / Format web page content
- Advantages
 - Reduce HTML formatting tags
 - Easy modification
 - Save lot of work and time
 - Faster loading

1. What is and Why CSS?

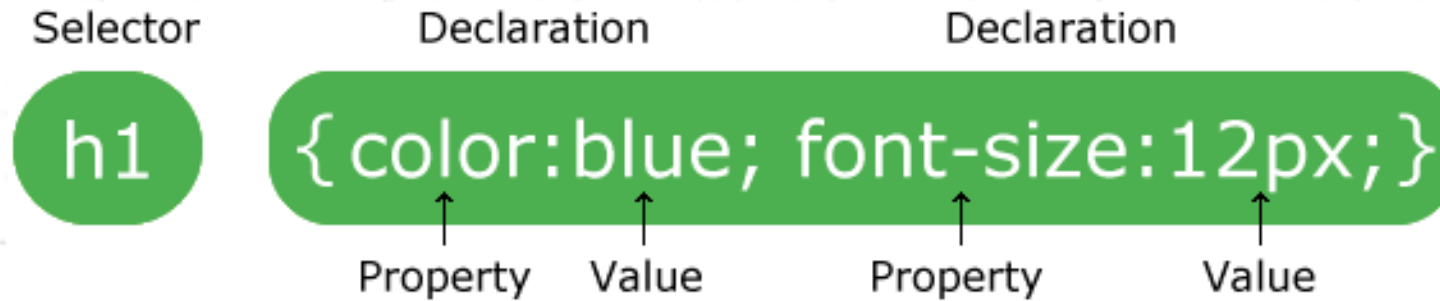


Demo

- Without CSS - [demo1 default.html](#)
- With CSS - [demo2 default.html](#)

2. CSS statements

Selector, Properties and Values



- 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.

2. CSS statements

Selector, Properties and Values

- Examples

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

```
h1 {  
    color: white;  
    text-align: center;  
}
```

```
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

My First CSS Example

This is a paragraph.

2. CSS statements with Properties and Values

Some formatting categories

- Positioning
- Size
- Alignment
- Font / Text
- Color / Background / Border

Activity

Find the available CSS rules under the following categories

- Positioning
- Size
- Alignment
- Font / Text
 - https://www.w3schools.com/css/css3_fonts.asp
- Color / Background / Border
 - https://www.w3schools.com/css/css3_backgrounds.asp

3. How to use CSS?

1. Inline

2. Internal sheets

3. External sheets

3.1. Inline

- Used as HTML attribute

`<tag style = "CSS Statements" >`

- Example

`<div style="color:#3425AA;">`

Content goes here

`</div>`

What is this
color code?

Question 1

```
<h1>my timetable</h1>
```

Ex. Apply below CSS rules to the element above

- text-align: center;
- color: #0000CC;
- font-family: Tahoma, Geneva, sans-serif;
- font-size: 32px;

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<h1 style="text-align:center; color:#0000CC;  
font-family:Tahoma, Geneva, sans-serif; font-  
size:32px;">my timetable </h1>
```

```
</body>
```

```
</html>
```

Output

my timetable

3.1. Inline

- `<h1 style="text-align:center; color:#0000CC; font-family:Tahoma, Geneva, sans-serif; font-size:32px;">my timetable</h1>`
- `<h1 style="text-align: center; color:#0000CC; font-family: Tahoma, Geneva, sans-serif; font-size: 32px;">my timetable</h1>`

This is
more
readable

3.2. Internal style sheet

- Why not inline CSS?
 - Difficult to modify
 - (why?)
- Solution
 - Internal style sheet

3.2. Internal style sheet

- Where to write?
 1. In the **head** section of the HTML document
 2. Using the **style** element
`<style></style>`

- How to write?

```
<head>  
  <style>  
    Selector  
    {  
      /* CSS Statements */  
    }  
  </style>  
</head>
```

3.2. Internal style sheet

- We use selectors to access the content
- Selectors
 1. **By Element** (h1, p, a)
 2. **By ID** (#MainHeading, #Section1)
 3. **By Class** (.Menu, .Header, .Footer)

3.2. Internal style sheet [By element]

```
<!DOCTYPE html>
<html>
<head>
  <style>
    a {
      color:#F00;
    }
  </style>
</head>
<body>

<a href="www.sliit.lk">visit SLIIT</a>

</body>
</html>
```



Output

visit SLIIT

3.2. Internal style sheet [**By ID**]

- Use the **ID** attribute for the element in the **body**
 - ID should be unique

```
<p id="ParaMain">My paragraph </p>
```

- CSS in head <style></style>

```
#ParaMain{  
    color:#F00;  
}
```

3.2. Internal style sheet [By ID]

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #ParaMain{
      color:#F00;
    }
  </style>
</head>
<body>

<p id="ParaMain">My paragraph </p>
<p >My paragraph </p>
<p>My paragraph </p>

</body>
</html>
```



Output

My paragraph

My paragraph

My paragraph

3.2. Internal style sheet [By class]

Use the **class** attribute for the element(s) in the **body**

- There can be multiple element, who belong to the same class

```
<p class="MainFont">My paragraph </p>
```

```
<a class="MainFont" href="www.google.com">Google</a>
```

- CSS in head <style></style>

```
.MainFont  
{  
    font:sans-serif;  
    font-size:12px;  
}
```

3.2. Internal style sheet [By class]

```
<!DOCTYPE html> <html> <head>
  <style>
    .MainFont_1
    {
      font-size:12px; color:blue;
    }
    .MainFont_2
    { font-size:12px; color:red; }
  </style>
</head> <body>
  <p class="MainFont">My paragraph </p>
  <a class="MainFont_1" href="www.google.com">Google</a>
  <a class="MainFont_2" href="www.google.com">Google</a>
</body> </html>
```

Output

My paragraph

Google Google

3.3. External style sheets

Why External?

- Easy to modify
- Uniform formatting in the site
- Design themes
- Faster loading

3.3. External style sheets

CSS is coded in an external file

- Extension of the file should be **.css**
- The code in the file starts with the line

`@charset "utf-8";`

3.3. External style sheets

```
@charset "utf-8";  
/* CSS Comments*/
```

```
.MainFont  
{  
    font: Tahoma, Geneva, sans-serif;  
    font-size: 12px;  
}
```


3.3. External style sheets

How to link the external CSS file with the web page?

```
<head>
```

```
  <link rel="stylesheet" href="CSS/CSSMain.css" />
```

```
</head>
```

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="html_external.css" />
</head>
<body>
    <p class="MainFont">My paragraph </p>
    <a class="MainFont_1" href="www.google.com">Google</a>
    <a class="MainFont_2" href="www.google.com">Google</a>
</body>
</html>
```



```
.MainFont_1
{
    font:Tahoma, Geneva, sans-serif;
    font-size:12px;
    color:blue;
}
.MainFont_2
{
    font:Tahoma, Geneva, sans-serif;
    font-size:12px;
    color:red;
}
```

Output

My paragraph

Google Google

4. Advanced Selectors

Pseudo Classes/Elements

- A CSS pseudo-element is used to style specified parts of an element.
- For example, it can be used to:
 - Style the first letter, or line, of an element
 - Insert content before, or after, the content of an element

Notation

```
Selector:pseudo-class {  
    /* CSS Statements */  
}
```

```
Selector::pseudo-element {  
    /* CSS Statements */  
}
```

Example

```
<!DOCTYPE html>
<html> <head>
<style>
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}
</style>
</head> <body>
<p>You can use the ::first-line pseudo-element to add a
special effect to the first line of a text. Some more text.
And even more, and more, and more, and more, and
more, and more, and more, and more, and more, and
more, and more, and more.</p>
</body> </html>
```

Output

YOU CAN USE THE ::FIRST-LINE PSEUDO-ELEMENT TO ADD A SPECIAL EFFECT TO THE FIRST LINE OF A TEXT. SOME more text. And even more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more, and more.

4. Advanced selectors

Pseudo Classes/Elements

- Some useful pseudo classes
 - :link
 - :visited
 - :hover
- Some useful pseudo elements
 - first-letter
 - first-line
 - first-child

4. Advanced selectors

Pseudo Classes/Elements

```
/* unvisited link */
```

```
a:link
```

```
{ color:#090; }
```

```
/* visited link */
```

```
a:visited
```

```
{ color:#C60; }
```


4. Advanced selectors

Pseudo Classes/Elements

```
/* mouse over link */
```

```
a:hover
```

```
{ color:#C0C; }
```

```
/* selected link */
```

```
a:active
```

```
{ color:#0FF; }
```

4. Advanced selectors

Exercise 1

- Find the available pseudo classes/elements in CSS3
- Find how to access different input element types in CSS
- Find how to apply CSS rules to multiple selections (Using combinators)

4. Advanced selectors

- Learn all the advanced selectors in the presentation below

<http://estelle.github.io/selectors/#slide1>

5. CSS – Some other essentials

5.1. Specificity

- **Specificity** is the means by which browsers decide which CSS property values are the most relevant to an element and, therefore, will be applied. Specificity

is based on the matching rules which are composed of different sorts of [CSS selectors](#) [Mozilla]

<https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>

http://estelle.github.io/CSS-Workshop/part_02_specificity.html

5. CSS – Some other essentials

5.2. Web fonts

CSS3 web fonts allow you to use custom fonts other than device fonts (web safe fonts)

http://estelle.github.io/CSSWorkshop/part_07_fonts.html#slide1

5. CSS – Some other essentials

5.3. Colors, gradients, backgrounds

http://estelle.github.io/CSS-Workshop/part_06_colors.html

<http://estelle.github.io/gradients/>

http://estelle.github.io/CSS-Workshop/part_08_backgrounds.html

5. CSS – Some other essentials

5.4. Transformations and animations

http://estelle.github.io/CSS-Workshop/part_10_transforms.html

<http://estelle.github.io/animation/>

5. CSS – Some other essentials

- http://estelle.github.io/CSS-Workshop/part_03_generated.html
- http://estelle.github.io/CSS-Workshop/part_04_media.html
- http://estelle.github.io/CSS-Workshop/part_04_media.html#slide10
- http://estelle.github.io/CSS-Workshop/part_12_features.html

CSS – Summary

1. What is and Why CSS?
2. CSS statements with Properties and Values
3. How to use CSS?
4. Advanced selectors
5. Some other essentials

Examples

```
<!DOCTYPE html>
<html>
<head>
<style>

</style>
</head>
<body>

<h1>This is a heading</h1>
<h2>This is a smaller heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
  h1, p {
    color: red;
  }
</style>
</head>
<body>

<h1>This is a heading</h1>
<h2>This is a smaller heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

Examples

```
<!DOCTYPE html>
<html>
<head>
<style>

</style>
</head>
<body>

<h1>This is a Heading</h1>
<p id="para1">This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    color: red;
}
</style>
</head>
<body>

<h1>This is a Heading</h1>
<p id="para1">This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```


Exercise

```
<!DOCTYPE html>
<html> <head> <style>
h1 {
  text-transform: uppercase;
}
p {
  text-transform: capitalize;
}
</style> </head> <body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body> </html>
```

Result:

THIS IS A HEADING

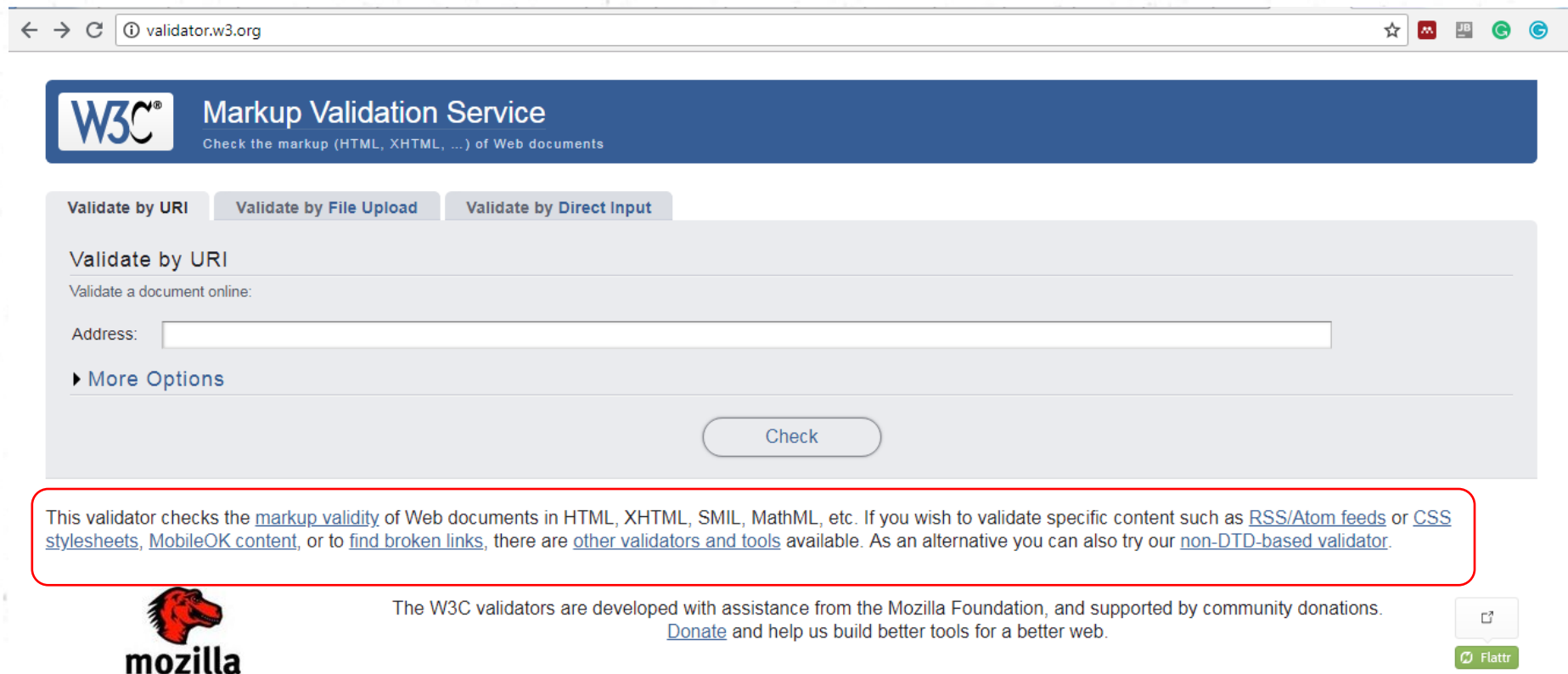
This Is A Paragraph.

This Is Another Paragraph.

Reference

<http://validator.w3.org/>

- This validator checks the [markup validity](#) of Web documents



W3C® Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI


Validate a document online:

Address:

► [More Options](#)

[Check](#)

This validator checks the [markup validity](#) of Web documents in HTML, XHTML, SMIL, MathML, etc. If you wish to validate specific content such as [RSS/Atom feeds](#) or [CSS stylesheets](#), [MobileOK content](#), or to [find broken links](#), there are [other validators and tools](#) available. As an alternative you can also try our [non-DTD-based validator](#).

 mozilla

The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web.

[Flattr](#)