



COM3517/6517

Web Technologies

Prof F. Ciravegna

Dr. V. Lanfranchi

Organisations, Information and Knowledge Group
Department of Computer Science
University of Sheffield

f.ciravegna@shef.ac.uk
v.lanfranchi@shef.ac.uk

- HTML Basics
- Tables, Images and Layers
- Forms
- CSS
- HTML5



The
University
Of
Sheffield.

HTML Basics

What is HTML

- HTML is a markup language for describing web documents (web pages).
 - HTML stands for Hyper Text Markup Language
 - A markup language is a set of markup tags

A small HTML document:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

HTML (Ctd)

- The DOCTYPE declaration defines the document type to be HTML
- The text between <html> and </html> describes an HTML document
- The text between <head> and </head> provides information about the document
 - metadata and declaration of files (e.g. style and javascript files)
- The text between <title> and </title> provides a title for the document
 - the title is what you see in the window's bar
- The text between <body> and </body> describes the visible page content
 - The text between <h1> and </h1> describes a heading
 - The text between <p> and </p> describes a paragraph
- Using this description, a web browser can display a document with a heading and a paragraph.
- HTML describes the formatting, not the content

Introduction

- When creating a Web page, separate the structure and the appearance
- Structure is indicated using HTML

```
<h1>Team X</h1>
```

Structure
and content

+

+

style 1

style 2

- Appearance is controlled using CSS
- Cascading Style Sheet

```
h1 {
  color: teal;
}
```

Team X

We are Team X.

- Art
- Bart
- Cara

Team X

We are Team X.

- Art
- Bart
- Cara

HTML Tags

- HTML tags are keywords (tag names) surrounded by angle brackets:
 - `<tagname>content</tagname>`
- HTML tags normally come in pairs like `<p>` and `</p>`
 - The first tag in a pair is the start tag, the second tag is the end tag
 - The end tag is written like the start tag, but with a slash before the tag name



HTML Tags

```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```


Elements, attributes and values

- A general document is made up of elements
- An element:

- **<e1>** **content** **</e1>**

start tag	mix of text and elements	matching end tag
--------------	-----------------------------	---------------------

- Empty element: **<e1 />**
- Attributes are named properties of elements
- Attributes are assigned values in elements' start tags, using an = sign

```
<li><a href="http://www.thesimpsons.com/">Bart</a></li>
```

↑
element

↑
attribute

↑
value

A sample document

(notice the indentation)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>
<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://www.thesimpsons.com/">Bart</a></li>
    <li>Cara</li>
  </ul>
</body>
</html>
```

Team X

We are TeamX.

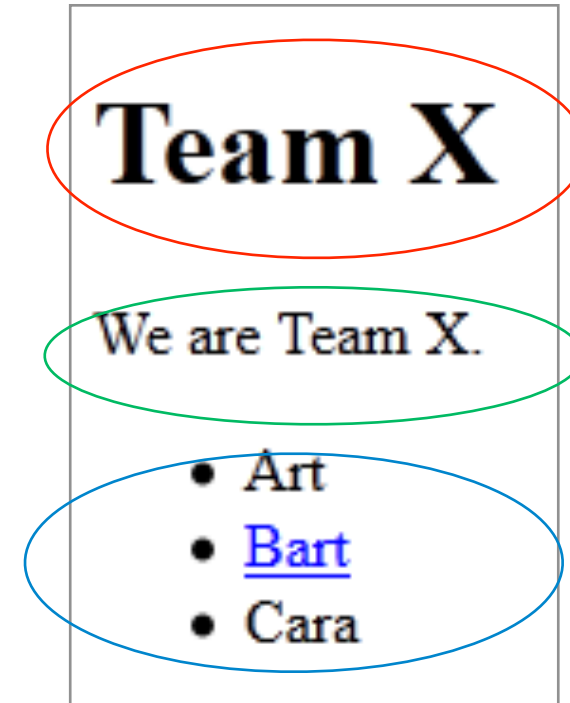
- Art
- [Bart](http://www.thesimpsons.com/)
- Cara

A sample document

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>

<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://
www.thesimpsons.com/">Bart</a></li>
    <li>Cara</li>
  </ul>
</body>
</html>
```

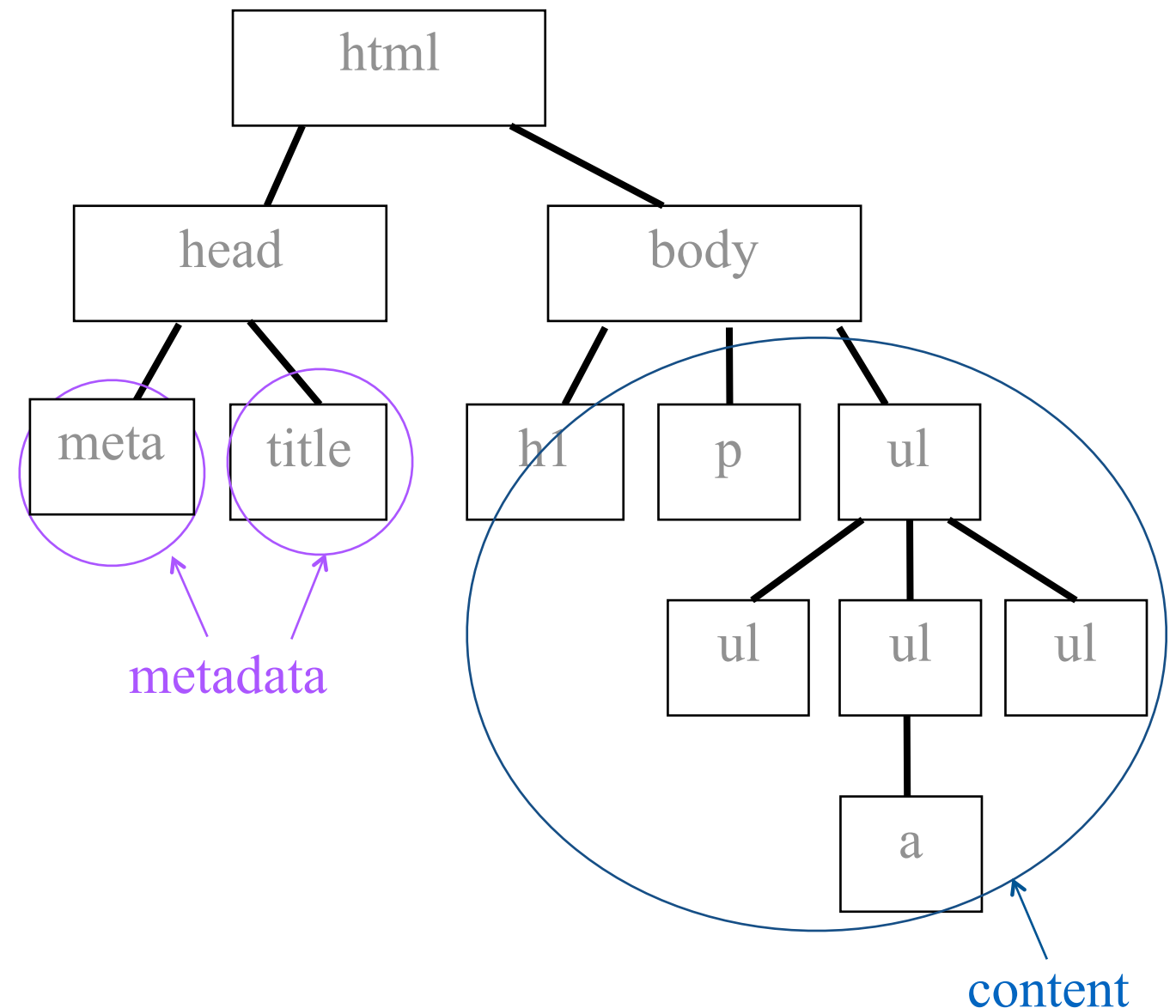


Document structure

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>

<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://
www.thesimpsons.com/">Bart
</a></li>
    <li>Cara</li>
  </ul>
</body>
</html>
```

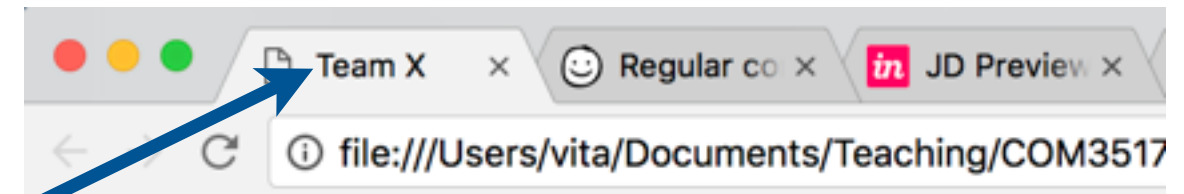


Document head

- The content of the head element is not rendered in the browser window
- The title element is compulsory and is displayed in the title bar
- The meta element provides a general-purpose mechanism for adding metadata to HTML documents

Document head

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>
<body>
<!-- content -->
</body>
</html>
```



Team X

We are Team X.

- Art
- [Bart](#)
- Cara

Doctype and language

More complex for XHTML

- Specifying the doctype triggers browsers that need it to operate in html standards mode
- The root level of the document is the **html element**
- The html element has a language attribute
 - **en** = English

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>

<body>
  <!-- content -->
</body>
</html>
```

← A comment

```
<HTML lang="fr">
<HEAD>
<TITLE>Un document multilingue</TITLE>
</HEAD>
```

www.w3.org/TR/html4/struct/dirlang.html

Charset

- Charset is an attribute of a meta tag to define the document's character encoding
 - Security risk of not setting it
 - Must be in first 512 bytes
 - Multibyte character encoding for Unicode.



UTF-8: definition



WIKIPEDIA
Encyclopedia

content
vents
article
Wikipedia

on
Wikipedia
unity portal
changes

Article

Talk

Read

Edit

View

UTF-8

From Wikipedia, the free encyclopedia

UTF-8 (**UCS Transformation Format—8-bit**^[1]) is a **variable-width encoding** that can represent every **character** in the **Unicode** character set. It was designed to avoid the complications of **endianness** and **byte order marks** in **UTF-16** and **UTF-32**.

UTF-8 has become the dominant character encoding for the **World-Wide Web**, accounting for more than half of all Web pages.^{[2][3][4]} The **Internet Engineering Task Force** (IETF) uses **UTF-8** in its **protocols** to identify the **encoding** used for character data, and the supported character encodings must include UTF-8.^[5] The **Internet Mail Consortium** (IMC) uses UTF-8 to display and create mail using UTF-8.^[6] UTF-8 is also increasingly being used as the default character encoding in **operating systems**, **programming languages**, and **applications**.^[citation needed]

UTF-8 encodes each of the 1,112,064 **code points** in the Unicode character set using one to four 8-bit **bytes** (termed "**octets**" in the Unicode Standard). Code points in the lower half of the Unicode character set (which tend to occur more frequently) are encoded using fewer bytes. The first 128 characters of Unicode, which are the same as the first 128 characters of ASCII, are encoded using a single octet with the same binary value as ASCII, making valid ASCII text valid UTF-8-encoded Unicode as well.

The official **IANA** code for the UTF-8 character encoding is **UTF-8**.^[7]

<http://en.wikipedia.org/wiki/UTF-8>

Other metadata

- Other metadata elements use name and content attributes
- Other elements
 - `link` – stylesheets (see later in this lecture)
 - `script` – JavaScript (see a later lecture)

```
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
  <link rel="stylesheet" href="teamx1.css" />
  <meta name="author" content="Steve Maddock" />
  <meta name="description" content="Team X web site for COM1004" />
  <meta name="keywords" content="Team X, sports" />
</head>
```

Meta - robots

```
<meta name="robots" content="index, nofollow" />
```

Not very popular with spiders who tend to ignore it

Keyword	Meaning
index	This document may be indexed.
noindex	This document may be not indexed.
follow	Links from this document may be followed.
nofollow	Links from this document may not be followed.
all	This document may be indexed and links from it may be followed.
none	This document may not be indexed and links from it may not be followed.

Chapman, N and J. Chapman, Web Design: A complete introduction, John Wiley & Sons, 2006.

Document body

```
<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://www.thesimpsons.com/">Bart</a></li>
    <li>Cara</li>
  </ul>
</body>
```

Headings

- HTML defines six level of headings in descending order of importance
 - **h1**, h2, h3, h4, h5, h6
- The heading element includes predefined formatting
 - Font change
 - Paragraph break
 - Space before and after

Headings

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>
<body>
  <h1>Team X</h1>
  <h2>A team from the University of Sheffield</h2>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://www.thesimpsons.com/">Bart</a></li>
    <li>Cara</li>
  </ul>
</body>
</html>
```

Team X

A team from the University of Sheffield

We are Team X.

- Art
- [Bart](http://www.thesimpsons.com/)
- Cara

Textual elements

- Paragraph - `<p>`
- Line break `
`

Lists

- HTML defines three types of lists
 - **unordered list **— group of related items in no particular order
 - **ordered list ** — group of related items in a particular order
 - **Description list <dl>** — group of name/value pairs


```
<ul>
```

```
<li>Art</li>
```

```
<li><a href="http://www.thesimpsons.com/">Bart</a></li>
```

```
<li>Cara</li>
```

```
</ul>
```

- Art
- [Bart](http://www.thesimpsons.com/)
- Cara

```
<ol>
```

```
<li>Wake up</li>
```

```
<li>Drink Coffee</li>
```

```
<li>Go to work</li>
```

```
</ol>
```

1. Wake up
2. Drink Coffee
3. Go to work

```
<dl>
```

```
<dt>Coffee</dt>
```

```
<dd>brewed drink prepared from roasted coffee beans</dd>
```

```
<dt>Sugar</dt>
```

```
<dd>generic name for sweet-tasting, soluble carbohydrates</dd>
```

```
</dl>
```

Coffee
brewed drink prepared from roasted coffee beans

Sugar
generic name for sweet-tasting, soluble carbohydrates

Hyperlinks

- Hyperlinks link a source and a destination

```
<a href="URL">link text</a>
```

- Default display: blue and underlined
- Once visited: purple and underlined

Hyperlinks

```
<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://www.thesimpsons.com/">Bart</a></li>
    <li>Cara</li>
  </ul>
</body>
```

Team X

We are Team X.

- Art
- [Bart](#)
- Cara

Team X

We are Team X.

- Art
- [Bart](#)
- Cara

Paths - as file system

- Pages are organised in folders (generally under a root called public_html)
- They can refer to each other (e.g. in a link) via either:
 - an absolute path
 - <https://mydomain.org/js/jav.js> (not suggested unless external)
 - a relative path
 - that allows navigating the filesystem:
 - format is similar to the unix/linux
 - `../` to move one directory up
 - `./` to refer to the current directory (e.g. `./js/jav.js`)
- The base of a web page (the directory pointed by the expression `./` is the folder in which the file currently displayed is
- for example:
 - public_html/index.html

Paths - as file system

https://www.w3schools.com/html/html_filepaths.asp

- Pages are organised in folders
- They can refer to each other (e.g. in a link) via either:
 - an absolute path
 - <https://mydomain.org/js/jav.js> (not suggested unless external)
 - a relative path
 - that allows navigating the filesystem:
 - Similar to Uni/Linux filesystem
 - Useful because can easily move whole Web site to a different host machine, as the links are relative

Paths - as file system

https://www.w3schools.com/html/html_filepaths.asp

Path	Description
<code></code>	picture.jpg is located in the same folder as the current page
<code></code>	picture.jpg is located in the images folder in the current folder
<code></code>	picture.jpg is located in the images folder at the root of the current web
<code></code>	picture.jpg is located in the folder one level up from the current folder

- Anchors (or bookmarks) allow readers to jump to specific parts of a Web page

1. Create an anchor:

```
<h1 id="#Top">Team X</h1>
```

2. Link to the anchor:

```
<a href="#Top">Go to the top</a>
```

3. When the user clicks the link, the page will scroll to the location

- You can create a link to an anchor in an external page

```
<a href=" ../feedback.html#Comments">Send a comment</a>
```

Block and inline elements

- How does the browser know when to start a new line?

```
<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <p>We welcome new members.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://
www.thesimpsons.com/">Bart</a> -
Club Captain</li>
    <li>Cara</li>
  </ul>
  <p>Designed by <em>a web
designer</em>, 2011</p>
</body>
```

Team X

We are Team X.

We welcome new members.

- Art
- Bart - Club Captain
- Cara

Designed by *a web designer*, 2011

Block and inline elements

- HTML distinguishes between block and inline elements
 - **Block elements**
 - Begin on new lines
 - Can contain other block and inline elements
 - Examples: p, div, ul, li, table, h1, h2, h3, h4, h5, h6
 - **Inline elements**
 - Do NOT begin on new lines
 - Can contain other inline elements or data
 - Examples: a, img, span, em, strong, code, b, i, big, small, br, cite



The
University
Of
Sheffield.

HTML Tables, Images and Layers

Tables

- How to create the following table?

	Estimate	Measured	Error
Height (cm)	40	43	+3
Width (cm)	26	25	-1

Table 1. Widget production error

Tables

- Steps:
 - The whole table
 - A row
 - A cell
 - A heading cell
 - A caption

	Estimate	Measured	Error
Height (cm)	40	43	+3
Width (cm)	26	25	-1


Table 1. Widget production error


- The whole table
- A row
- A cell
- A heading cell
- A caption

```
<table>
  <caption>Table 1. Widget production
data</caption>
  <tr>
    <td></td>
    <th>Estimate</th>
    <th>Measured</th>
    <th>Error</th>
  </tr>
  <tr>
    <th>Height (cm)</th>
    <td>40</td>
    <td>43</td>
    <td>+3</td>
  </tr>
  <tr>
    <th>Width (cm)</th>
    <td>26</td>
    <td>25</td>
    <td>-1</td>
  </tr>
</table>
```

- The insert an image, use
 - ``
- The `` tag has two required attributes:
 - `src`: the image file
 - `alt`: the text to present if the image is not available or for accessibility reasons

Attributes

 = New in HTML5.

Attribute	Value	Description
<u>align</u>	top bottom middle left right	Not supported in HTML5. Specifies the alignment of an image according to surrounding elements
alt	text	Specifies an alternate text for an image
border	pixels	Not supported in HTML5. Specifies the width of the border around an image
crossorigin	 anonymous use-credentials	Allow images from third-party sites that allow cross-origin access to be used with canvas
<u>height</u>	pixels	Specifies the height of an image
<u>hspace</u>	pixels	Not supported in HTML5. Specifies the whitespace on left and right side of an image
ismap	ismap	Specifies an image as a server-side image-map
longdesc	URL	Specifies a URL to a detailed description of an image
src	URL	Specifies the URL of an image
<u>usemap</u>	#mapname	Specifies an image as a client-side image-map
<u>vspace</u>	pixels	Not supported in HTML5. Specifies the whitespace on top and bottom of an image
width	pixels	Specifies the width of an image

- The <div> tag defines a division or a section in an HTML document.
 - Typically this is a block (of text, images, etc.)
 - The <div> tag is also used to group block-elements to format them with CSS
- <div class='purple'> this is a div where we can find lots of things blah blah blah </div>

- By default, browsers always place a line break before and after the `<div>` element
- Most browsers will display the `<div>` element with the following default values:

`div { display: block; }`

- as opposed to `gone` or `invisible`

6. The mighty div

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
  <link rel="stylesheet" href="teamx2.css" />
</head>

<body>
  <h1>Team X</h1>
  <div id="main">
    <p class="green">We are <span
class="purple">Team X</span>.</p>
    <ul>
      <li>Art</li>
      <li><a href="http://
www.thesimpsons.com/">Bart</a></li>
      <li class="italic green">Cara</
li>
    </ul>
  </div> <!-- main -->
</body>
</html>
```

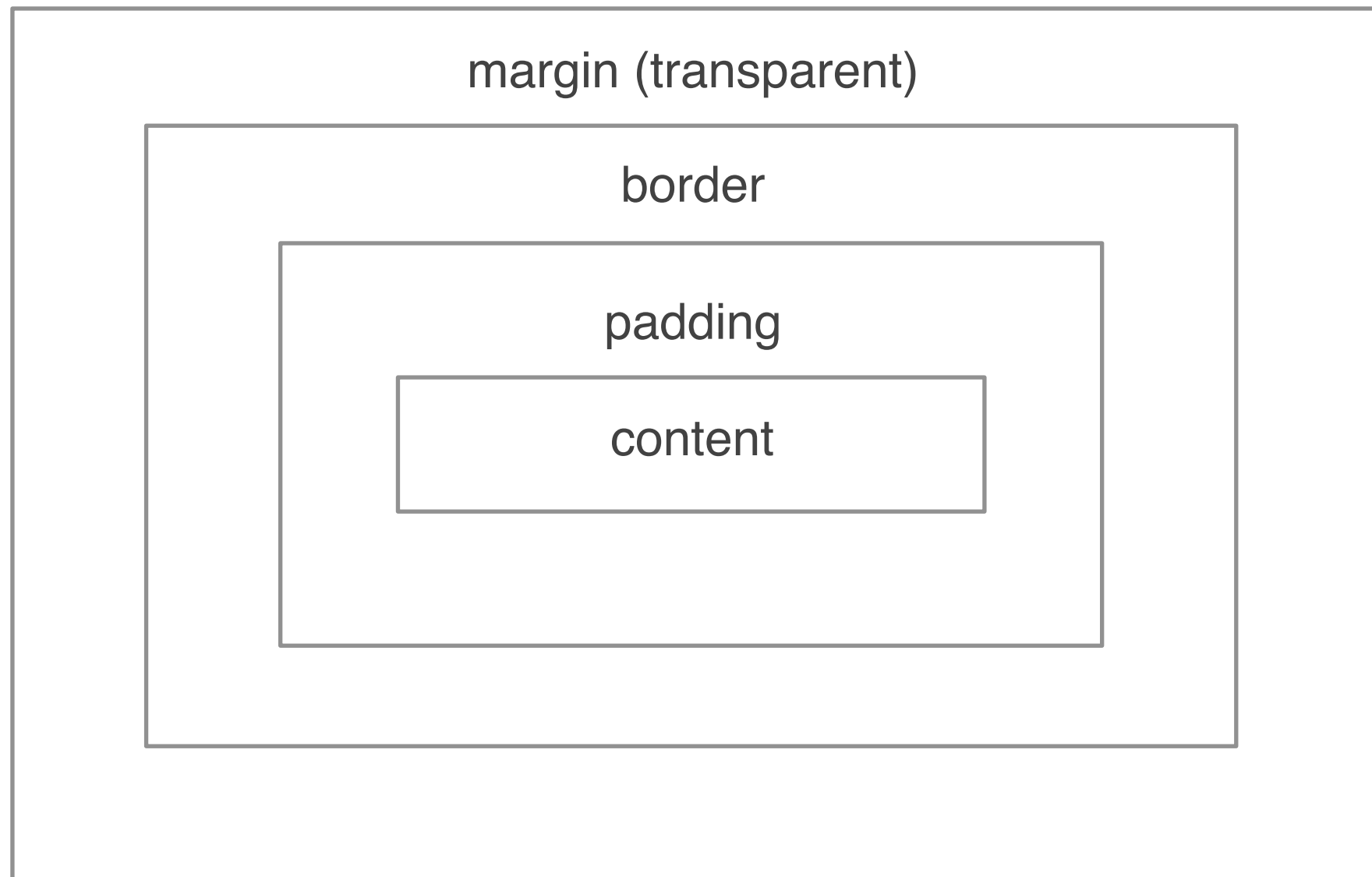
Team X

We are Team X.

- o Art
- o Bart
- o *Cara*

The Box Model

- Every element (content) is placed inside a box
- Each box region may have a thickness of zero





The
University
Of
Sheffield.

HTML Forms

- HTML forms are used to collect user input.
 - The `<form>` element defines an HTML form:
 - `<form>` *list of form elements* `</form>`
- HTML forms contain form elements.
- Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

- The `<input>` Element
 - The most important form element
 - The `<input>` element has many variations, depending on the type attribute, e.g.:
 - *text* Defines normal text input
 - *radio* Defines radio button input (for selecting one of many choices)
 - *submit* Defines a submit button (for submitting the form)



Text Input

http://www.w3schools.com/html/html_forms.asp

Text Input

<input type="text"> defines a one-line input field for **text input**:

Example

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Try it Yourself »

This is how it will look like in a browser:

First name:

Last name:

The value Attribute

The **value** attribute specifies the initial value for an input field:

Example

```
<form action="">  
First name:<br>  
<input type="text" name="firstname" value="John">  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>
```


Autocomplete

The autocomplete Attribute

The autocomplete attribute specifies whether a form or input field should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

Tip: It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.

The autocomplete attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.

Example

An HTML form with autocomplete on (and off for one input field):

```
<form action="action_page.php" autocomplete="on">  
  First name:<input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit">  
</form>
```



The min and max Attributes

The min and max attributes specify the minimum and maximum value for an <input> element.

The min and max attributes work with the following input types: number, range, date, datetime, datetime-local, month, time and week.

Example



<input> elements with min and max values:

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5">
```

Try it Yourself »

The placeholder Attribute

The placeholder attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).

The hint is displayed in the input field before the user enters a value.

The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

Example

An input field with a placeholder text:

```
<input type="text" name="fname" placeholder="First name">
```

Try it Yourself »



The required Attribute

The required attribute is a boolean attribute.

When present, it specifies that an input field must be filled out before submitting the form.

The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Example



A required input field:

```
Username: <input type="text" name="usrname" required>
```

Try it Yourself »



Radio Button

http://www.w3schools.com/html/html_forms.asp

Radio Button Input

<input type="radio"> defines a **radio button**.

Radio buttons let a user select ONE of a limited number of choices:

Example

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

- ☒ Male
- ☐ Female
- ☐ Other

The <select> Element (Drop-Down List)

The **<select>** element defines a **drop-down** list:

Example

```
<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```

The <textarea> Element

The **<textarea>** element defines a multi-line input field (**a text area**):

Example

```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

The cat was playing in the garden.

Buttons (other than submit)

The <button> Element

The **<button>** element defines a clickable **button**:

Example

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```




Submit button

http://www.w3schools.com/html/html_forms.asp

The Submit Button

<input type="submit"> defines a button for **submitting** a form to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's **action** attribute:

Example

```
<form action="action_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

Action -> the target program

The Action Attribute

The **action attribute** defines the action to be performed when the form is submitted.

The common way to submit a form to a server, is by using a submit button.

Normally, the form is submitted to a web page on a web server.

In the example above, a server-side script is specified to handle the submitted form:

```
<form action="action_page.php">
```

If the action attribute is omitted, the action is set to the current page.



Method: Post or Get?

http://www.w3schools.com/html/html_forms.asp

The Method Attribute

The **method attribute** specifies the HTTP method (**GET** or **POST**) to be used when submitting the forms:

```
<form action="action_page.php" method="get">
```

or:

```
<form action="action_page.php" method="post">
```

When to Use GET?

You can use GET (the default method):

If the form submission is passive (like a search engine query), and without sensitive information.

When you use GET, the form data will be visible in the page address:

```
action_page.php?firstname=Mickey&lastname=Mouse
```



GET is best suited to short amounts of data. Size limitations are set in your browser.

Grouping Form Data with <fieldset>

The **<fieldset>** element groups related data in a form.

The **<legend>** element defines a caption for the <fieldset> element.

Example

```
<form action="action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Try it Yourself »

This is how the HTML code above will be displayed in a browser:

Personal information:

First name:

Last name:

When to Use POST?

You should use POST:

If the form is updating data, or includes sensitive information (password).

POST offers better security because the submitted data is not visible in the page address.

The Name Attribute

To be submitted correctly, each input field must have a name attribute.

This example will only submit the "Last name" input field:

Example

```
<form action="action_page.php">
  First name:<br>
  <input type="text" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Try it Yourself »



The
University
Of
Sheffield.

CSS

Introduction

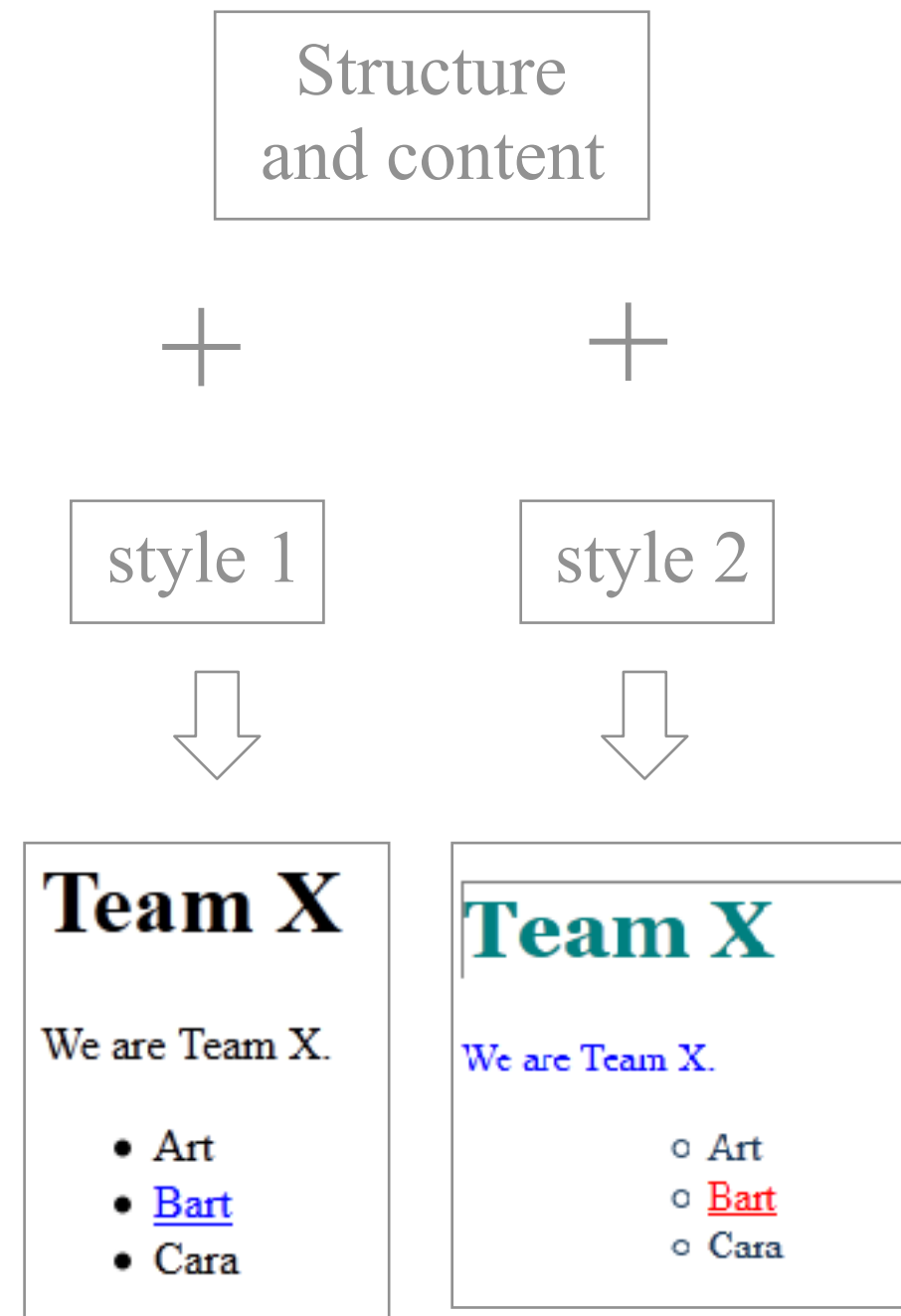
- CSS have been created to separate the structure from the appearance of a web page
- Structure is indicated using HTML

```
<h1>Team X</h1>
```

- Appearance is controlled using CSS
- Cascading Style Sheet

```
h1 {  
    color: teal;  
}
```

Introduction



Styling in older HTML

- In the past the style of each element was declared within the element itself
 - e.g. you could declare:
 - `<center>This text will be center-aligned.</center>`
 - many of them were tags in themselves
 - However this was not good
 - you may want to have different elements formatted in the same way
 - multiple declarations bring to errors

NO LONGER SUPPORTED!!!

Styling via CSS

- CSS defines the standard styling (e.g. formatting) for the **types of elements** based on:
 - their type (e.g. `<h1>`)
 - their identity (e.g. `<h1 id="myId">`)
 - their position (ul li div - a div contained in a list)
- You can still define element-level style via the style attribute writing effectively css code in the value
 - `<div style="text-align:center">`

Attaching a stylesheet

https://www.w3schools.com/html/html_css.asp

- CSS can be added to HTML in 3 ways:
 - **Inline** - by using the style attribute in HTML elements

```
<h2 style="color:red">A team from the University of Sheffield</h2>
```

- **Internal** - by using a `<style>` element in the `<head>` section

```
<head>  
  <style>  
    h2 {color: red;}  
  </style>  
</head>
```

- **External** - by using an external CSS file

```
<link rel="stylesheet" href="styles.css">
```

Inline CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>
<body>
  <h1>Team X</h1>
  <h2 style="color:red">A team from the University of
  Sheffield</h2>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://www.thesimpsons.com/">Bart</
a></li>
    <li>Cara</li>
  </ul>
</body>
</html>
```

Team X

A team from the University of Sheffield

We are Team X.

- Art
- [Bart](http://www.thesimpsons.com/)
- Cara

Inline CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
  <style>
    h2 {color: red;}
  </style>
</head>
<body>
  <h1>Team X</h1>
  <h2>A team from the University of Sheffield</h2>
  <p>We are Team X.</p>
  ....
</body>
</html>
```

Team X

A team from the University of Sheffield

We are Team X.

- Art
- [Bart](#)
- Cara

Inline CSS

```
<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Team X</title>
</head>
<meta charset="utf-8" />
  <link type="text/css" rel="stylesheet" href="css/
styles.css" />
</head>
<body>
  <h1>Team X</h1>
  <h2>A team from the University of Sheffield</h2>
  <p>We are Team X.</p>
  ....

</body>
</html>
```

Team X

A team from the University of Sheffield

We are Team X.

- Art
- [Bart](#)
- Cara

CSS rules

- A stylesheet is a set of *rules*

```
Selector { Declaration; }
```

- Example rule: change *h1* text colour to *teal*

```
h1 { color: teal; }
```

```
Selector { Property: Value; }
```

- This applies to all occurrences of the h1 element
- Multiple declarations separated by semicolons
- If property value has a space, use quotes:

```
h1 {  
    font-family: "Lucida Handwriting", Papyrus, serif;  
}
```

The HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Team X</title>
  <link rel="stylesheet" href="teamx1.css" />
</head>

<body>
  <h1>Team X</h1>
  <p>We are Team X.</p>
  <ul>
    <li>Art</li>
    <li><a href="http://www.thesimpsons.com/">Bart</a></li>
    <li>Cara</li>
  </ul>
</body>
</html>
```

Team X

We are Team X.

- o Art
- o Bart
- o Cara

The CSS

```
h1 {  
  color: teal;  
  font-family: Georgia, serif;  
  font-size: 200%;  
}  
  
p {  
  color: blue;  
}  
  
ul {  
  padding-left: 100px;  
  list-style-type: circle;  
}  
  
li {  
  color: #123456; /* hexadecimal */  
}  
  
a {  
  color: red;  
}
```

teamx1.css

A comment

Typography

15.3 Font family: the 'font-family' property

'font-family'

<i>Value:</i>	[[<u><family-name></u> <u><generic-family></u>] [, <u><family-name></u> <u><generic-family></u>]*] <u>inherit</u>
<i>Initial:</i>	depends on user agent
<i>Applies to:</i>	all elements
<i>Inherited:</i>	yes
<i>Percentages:</i>	N/A

Always include
a generic
family

```
h1 {
  font-family: Papyrus, serif;
}
```

<generic-family>

In the example above, the last value is a generic family name

- 'serif' (e.g., Times)
- 'sans-serif' (e.g., Helvetica)
- 'cursive' (e.g., Zapf-Chancery)
- 'fantasy' (e.g., Western)
- 'monospace' (e.g., Courier)

AaBbCc Sans-serif font

AaBbCc Serif font

AaBbCc Serif font (red serifs)

<https://en.wikipedia.org/wiki/Serif>

<http://www.w3.org/TR/CSS2/fonts.html>

The id selector in CSS

http://www.w3schools.com/css/css_syntax.asp

- The id selector uses the id attribute of an HTML element to select a specific element.
 - The HTML id of an element should be unique within a page, so the id selector is used to select one unique element!
 - e.g. `<div id="myId"> whatever we write here </div>`
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
 - `{ #myId color:purple }`
 - The style rule will apply to the HTML element with

The CSS class selector

http://www.w3schools.com/css/css_syntax.asp

- The class selector selects elements with a specific class attribute.
 - e.g. `<div class="purple" > XXXXXX </div>`
- To select elements with a specific class, write a period (.) character, followed by the name of the class.
 - `.purple { color=purple }`
- You can also specify that only specific HTML elements should be affected by a class.
 - Example
 - `p.center { text-align: center }`

More CSS selectors

<http://www.w3.org/TR/selectors/>

- Contextual selector: E1 E2
 - E2 is a descendant of E1
- Contextual selector: E1>E2
 - E2 is a child of E1
- Contextual selector: E1+E2
 - E2 is the immediate sibling of E1

```
ul li { color: blue; }  
/* any li nested to any level in a ul */
```

```
ul>li { color: blue; }
```

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

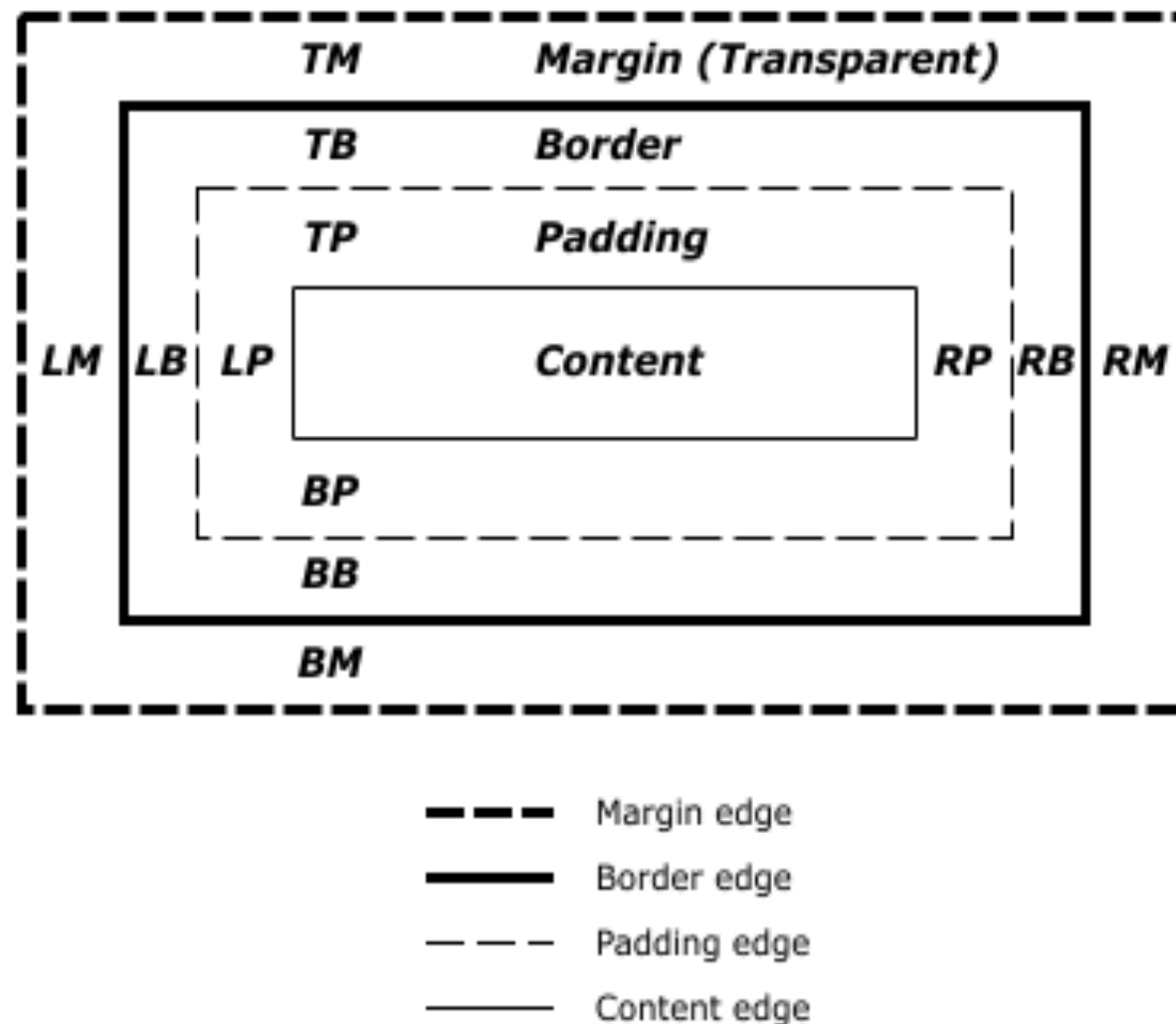
Margin, border, padding

<http://www.w3.org/TR/CSS2/box.html>

- Margins are transparent
 - Beware collapsing margins – depends on padding and border values
- Padding
 - Takes on same appearance as an element's background
- Border
 - Draws a border of finite thickness around an element
- An element has width and height attributes

Margin, border, padding

<http://www.w3.org/TR/CSS2/box.html>



- Margin delimits objects:
 - it is space around border
- Border is visible
- Padding is space between border and content

Total width = margin-left + border-left-width + padding-left + 'element width'
+ padding-right + border-right-width + margin-right

Properties

- margin-top, margin-right, margin-bottom, margin-left
- *length | percentage | auto*
- `p { margin-top: 2em; }`
- Set all at once with:
 - `body { margin: 1em 2em 3em 2em; }`

Properties

- padding-top, padding-right, padding-bottom, padding-left
 - *length | percentage*
- Set all at once with:
 - `body { padding: 2em; }`

- border-top, border-right, border-bottom, border-left
 - *length | percentage*
- Other properties:
 - border-left-color, border-right-color, border-top-color, border-bottom-colour, border-color, border-top-style, border-bottom-style, border-style, border-left-width, border-right-width, border-top-width, border-bottom-width, border-width

- Classes are used to specify one or more class names for an HTML element
- The class name can be used by CSS and JavaScript to perform certain tasks for elements with the specified class name.

```
<h2 class="city">Paris</h2>
```

```
<style>
.city {
  background-color: blue;
  color: white;
  padding: 10px;
}
</style>
```

Pre-defined pseudo-classes

<http://www.w3.org/TR/selectors/>

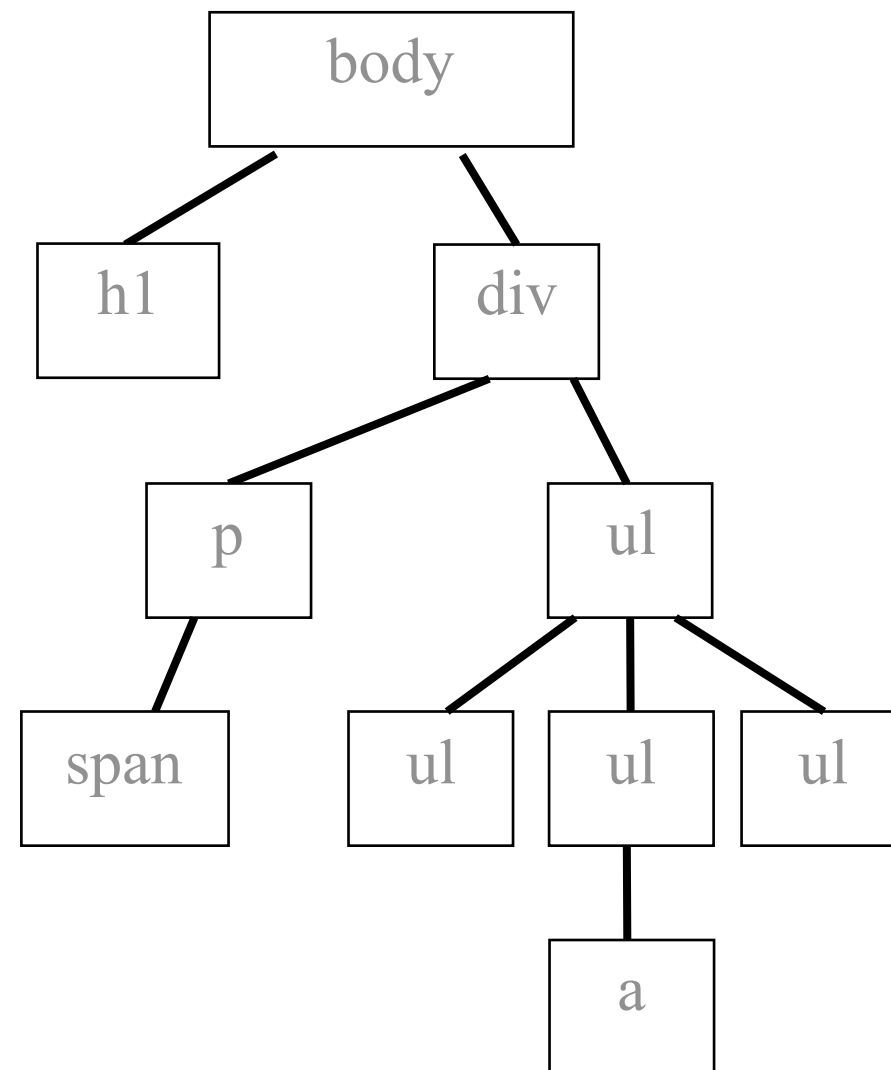
- Classes that depend on properties of the document rather than on the presence of a name in the class attribute
- The pseudo-class `:first-child` matches the first child of an element

```
h1:first-child { color: red; }
```

```
a:link { color: red; }      /* unvisited link */  
a:visited { color: gray; } /* visited link */  
a:hover { color: lime; }   /* mouse over link */  
a:active { color: lime; }  /* selected link */
```

- From the Team X example:

Inheritance



Inheritance

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Team X</title>
  <link rel="stylesheet" href="teamx2.css" />
</head>

<body>
  <h1>Team X</h1>
  <div id="main">
    <p class="green">We are <span class="purple">Team X</span>.</p>
    <ul>
      <li>Art</li>
      <li><a href="http://www.thesimpsons.com/">Bart</a></li>
      <li class="italic green">Cara</li>
    </ul>
  </div> <!-- main -->
</body>
</html>
```

Inheritance

- “Some values are inherited by the children of an element in the document tree” (<http://www.w3.org/TR/CSS2/cascade.html>)
- Unless a rule causes a different value to be explicitly assigned

```
.red { color: red; }  
.purple { color: purple; }
```

```
<p class="red">Hello <em>World</em></p>
```

Hello World

em element inherits red colour

```
<p class="red">  
Hello <em class="purple">World</em>  
</p>
```

Hello World

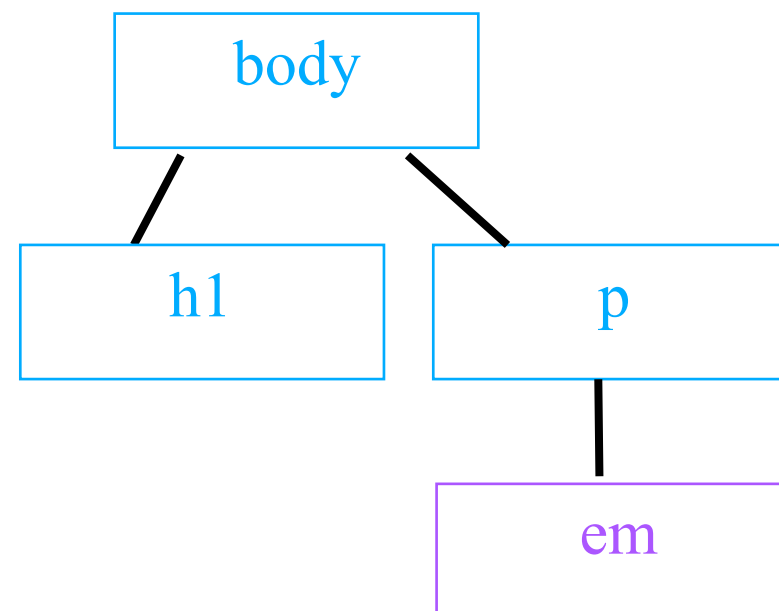
em is now of class purple which overrides the red colour

Inheritance

- Can be used to create efficient code
 - e.g. set text properties in body element, then override if necessary

```
<body>  
  <h1>text</h1>  
  <p>text <em>text</em> text</p>  
</body>
```

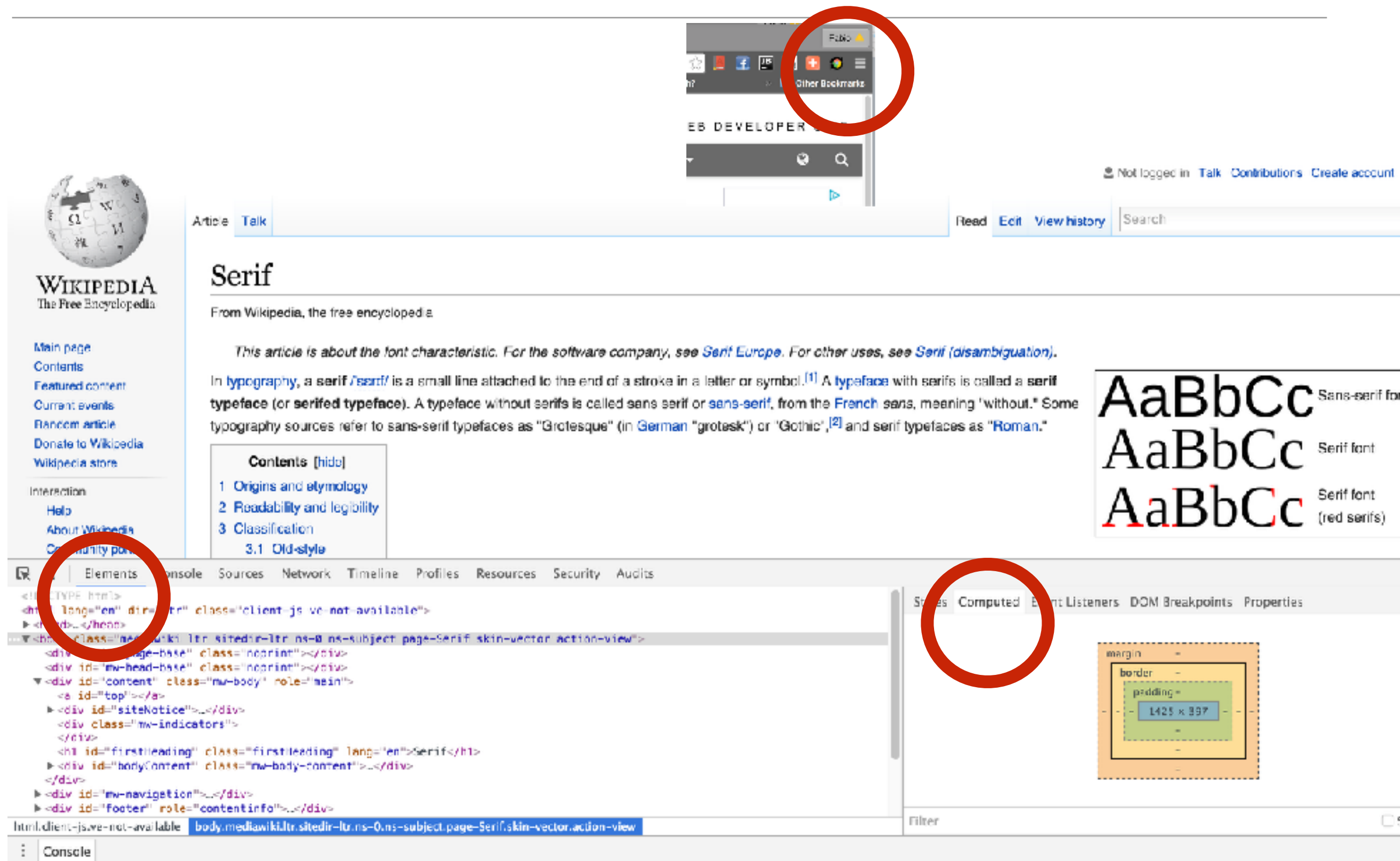
```
body { color: blue; }  
em { color: purple; }
```



Debugging with a browser

- Web browsers have in-built tools to debug HTML and Javascript
- Chrome
 - Go to >>menu on the right >> More Tools >> Developer Tools
 - on a PC it may have be slightly different names
 - Click on elements (bottom-left)
 - Click on Computed (Bottom-right)

Debugging with Chrome



The screenshot shows a web browser displaying the Wikipedia article "Serif". The browser's address bar shows the URL "https://en.wikipedia.org/wiki/Serif". The page title is "Serif". The page content includes a definition of serif typefaces and a list of contents. The Chrome DevTools interface is open, showing the "Elements" panel on the left and the "Properties" panel on the right. A red circle highlights the "Elements" panel, and another red circle highlights the "Properties" panel.

Wikipedia Article Content:

Serif

From Wikipedia, the free encyclopedia

This article is about the font characteristic. For the software company, see [Serif Europe](#). For other uses, see [Serif \(disambiguation\)](#).

In **typography**, a **serif** /*ˈsɛrɪf*/ is a small line attached to the end of a stroke in a letter or symbol.^[1] A **typeface** with serifs is called a **serif typeface** (or **serified typeface**). A typeface without serifs is called sans serif or **sans-serif**, from the **French** *sans*, meaning 'without.' Some typography sources refer to sans-serif typefaces as "Grotesque" (in **German** "grotesk") or "Gothic",^[2] and serif typefaces as "**Roman**."

Contents [hide]

- Origins and etymology
- Readability and legibility
- Classification
 - Old-style

Font Examples:

AaBbCc	Sans-serif font
AaBbCc	Serif font
AaBbCc	Serif font (red serifs)

Chrome DevTools:

Elements Panel:

```
<!DOCTYPE html>
<html lang="en" dir="ltr" class="client-js ve-not-available">
  <head>
    <script class="mw-head-base" class="noprint"></script>
    <div id="mw-head-base" class="noprint"></div>
    <div id="content" class="mw-body" role="main">
      <a id="top"></a>
      <div id="siteNotice"></div>
      <div class="mw-indicators"></div>
      <h1 id="firstHeading" class="firstHeading" lang="en">Serif</h1>
      <div id="bodyContent" class="mw-body-content"></div>
    </div>
    <div id="mw-navigation"></div>
    <div id="Footer" role="contentinfo"></div>
  </html>
```

Properties Panel:

margin: -

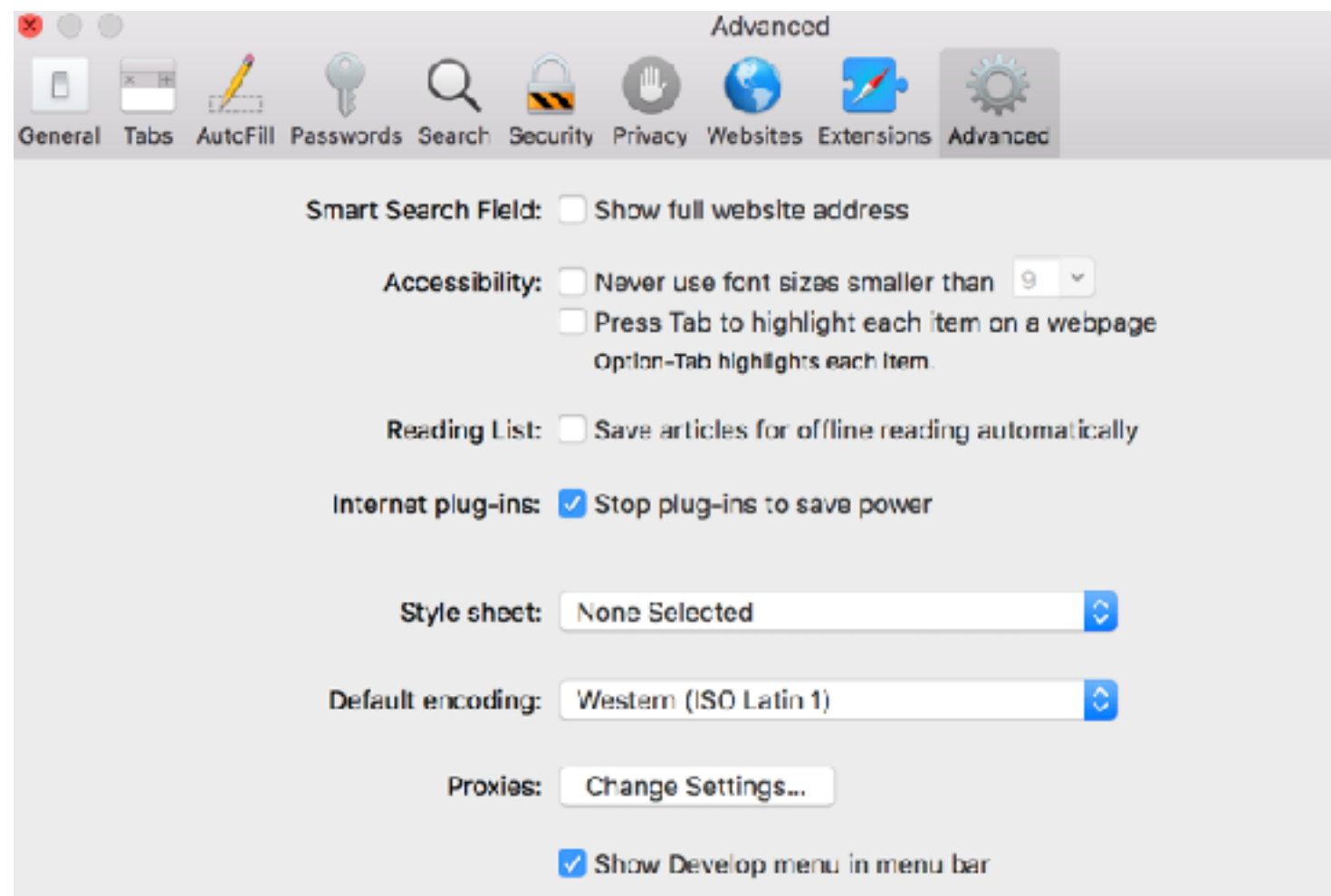
border: -

padding: -

1425 x 397

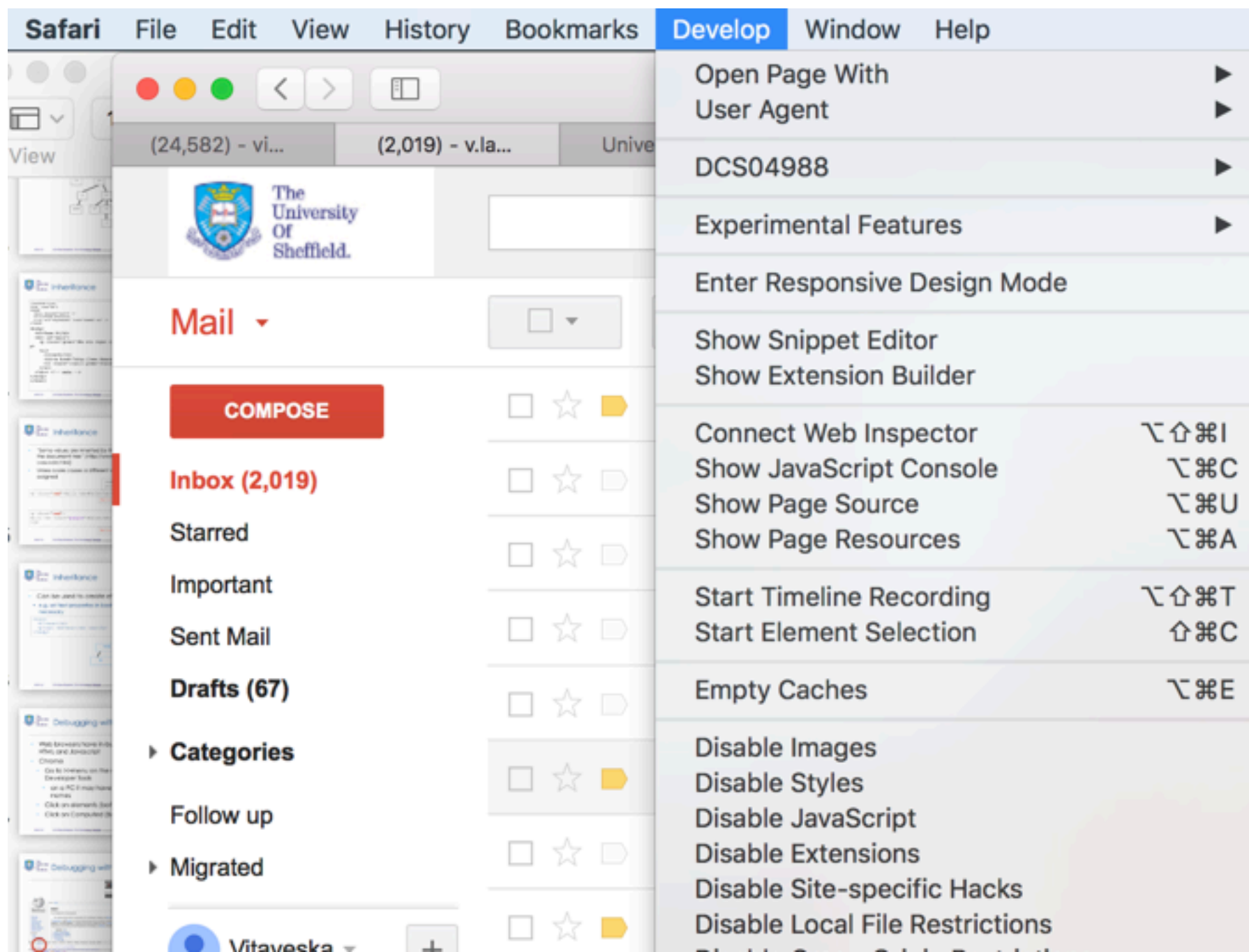
Debugging with a browser

- Safari
- Enable it from Safari->Preferences->Advanced



Debugging with a browser

- When enabled you can use the Develop menu





The
University
Of
Sheffield.

HTML5

-
- HTML5 is the new development for HTML but a quite radically different one
 - It allows semantics (as opposed to formatting) to be declared
 - this means that you can declare the type content of context expected
 - and the interpreter will check it

HTML5 declaration

- To declare you are using HTML5 you need to start your document with the following

```
<!DOCTYPE html>
```

- The browsers will interpret the new tags they support
- Ignore what they don't support yet

- Section are a new semantic tag that identifies a section of a document
- Most people use it to replace <div>
- **BE CAREFUL**
- Respect the semantic meaning
- The section element divides a block of text into sections
- All section elements should be followed by a heading tag
 - If you don't have one you don't need a section!

HTML5 article and header

- Article is used to define a block of text that makes sense on its own
 - e.g. the content of an about page
- Header is used to define a header section
 - Always followed by a h1....h6 tag
- hgroup is used to group a title and a subtitle

```
<article>
  <header>
    <hgroup>
      <h1>All about flour</h1>
      <h2>Wheat flour is the backbone of the baked goods we love</h2>
    </hgroup>
    <p>Published on January 12th</p>
  </header>
</article>
```


HTML5 - other elements

- New semantic elements like `<footer>`.

HTML5 - form validation

- Earlier in the lesson we saw how to create forms in HTML
- Traditionally you used Javascript with HTML to validate form content
- With HTML5 you can let the browser handle the validation!
- e.g. attribute “required”

```
<form>  
  <input type="text" required /><br />  
  <input type="submit" value="Submit now" />  
</form>
```

HTML5 - form validation

- HTML5 can handle 2 types of email address validation
 - Standard email address

```
<form>  
  <input type="email" required /> <br />  
  <input type="submit" value="Submit Now!">  
</form>
```

- Patter-based email address

```
<form>  
  <input pattern="/^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-  
zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/ " required />  
  <br />  
  <input type="submit" value="Submit Now!">  
</form>
```

- Validation of phone numbers

```
<form>  
  Your phonenumber: <input type="tel" required />  
  <input type="submit" value="Submit now" />  
</form>
```

- Not yet fully supported
- No pattern validation

BUT

- When using mobile browser the keyboard switches to numbers!

HTML5 - form validation

- Validation of date (YYYY-MM-DD)

```
<form>
  <input type="date">
  <input type="submit" value="Submit Now!">
</form>
```

- Validation of date (MM-DD)

```
<form>
  <input type="month">
  <input type="submit" value="Submit Now!">
</form>
```

- Validation of date (HH:mm:ss.ss)

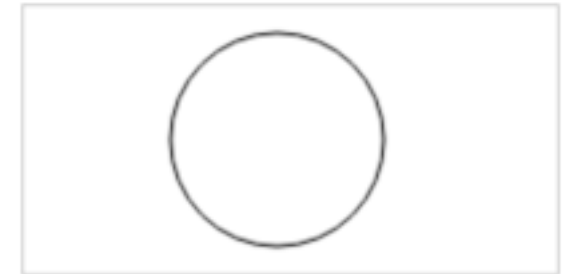
```
<form>
  <input type="time">
  <input type="submit" value="Submit Now!">
</form>
```

- HTML5 introduce the canvas element
 - A container to draw graphics on a web page
 - a rectangular area with no border and no content.
 - Graphics are drawn on the fly, via JavaScript
 - has several methods for drawing paths, boxes, circles, text, and adding images

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- Example: draw a circle

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95, 50, 40, 0, 2*Math.PI);  
ctx.stroke();
```



- Example: draw a gradient

```
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
  
// Create gradient  
var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10, 10, 150, 80);
```



- Canvas can be animated
- Canvas can be interactive
- Canvas can be used for games

HTML5 - let's try a little game

- Get your phone/tablet out
- Go to kahoot.it
- Enter the Game PIN when it appears on screen
- Have fun!

Summary

- We have learnt about
 - HTML
 - Basic tags
 - Forms
 - Tables
 - Images
 - HTML5
 - CSS
- Tomorrow during the lab we will try creating an HTML page

Designing for multiple

- As we said last time, mobile phones are the most used device for browsing the Web
- All websites *must* be designed for both computers AND mobiles
 - For all operating systems as well!
- This means
 - Strict use of standards (e.g. HTML5)
 - Test and display on all architectures
- A good way of doing it is to adopt libraries that allow this flexibility as a built in property

Questions?

