HyperText Markup Language

HTML pages *reflow* in width and length to fit the screen breadth.

Major versions of HyperText Markup Language include:

HTML 3.2	W3C standard January 1997
HTML 4.01	W3C standard December 1999 revising 4.0 from 1997
XHTML 1.0	W3C standard January 2000, revised August 2002
HTML 5	W3C standard October 2014

HTML 2 supports entities, forms, headings, hyperlinks, images, lists.

It was internationalised with *charset* and *lang* attributes in 1997.

HTML 3.2 added applets, super/subscripts, tables. JEditor uses it.

HTML 4.01

- added frames, (embedded) objects, scripts, style sheets
- has 3 versions - *frameset*, *transitional* and *strict*

XHTML 1.0 reformulates HTML 4 as XML 1.0 application that was

- intended as stepping stone in migration from HTML to XML
- both HTML 4.01 and XML at same time

XHTML 1.1 modularised XHTML 1.0 and is used in ePub 2 e-books.

HTML 5.2 is current version of HTML standardised in December 2017.

All HTML versions require following pattern (after DOCTYPE)

```
<html>
  <head> <title> ... </title> ... </head>
  <body>
    ...
  </body>
</html>
```

HTML Basics

HTML documents consist of text mixed with

declaration	HTML version and its DTD (versions after 3.2)
elements	markup directives inside "<" and ">"
entity references	codes between ampersand and semicolon

HTML 5+ declaration is

```
<!DOCTYPE html>
```

HTML 4.01 Strict declaration is

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
     "https://www.w3.org/TR/html4/strict.dtd">
```

Most tags appear in complementary pairs like `<ol>` ... `</ol>`

Solo tags appear in HTML as `<br>` or in XHTML or XML as `<br/>`.

Tags can have *attributes* associating name with string value:

```
<img src="me.gif" alt="picture of wonderful me">
```

Entities represent special characters and glyphs

- can either be a *character* or *numerical* entity reference
- start with "&" and end with ";"

HTML character entities denote common and maths symbols:

&lt;	<	less than
&amp;	&	ampersand
&omega;	ω	Greek omega

Numerical names for entities reference Unicode characters

&#D;	decimal digits D like &#937; for ω or omega
&#xH;	hexadecimal digits H like &#x6CF0 for Chinese *tai*

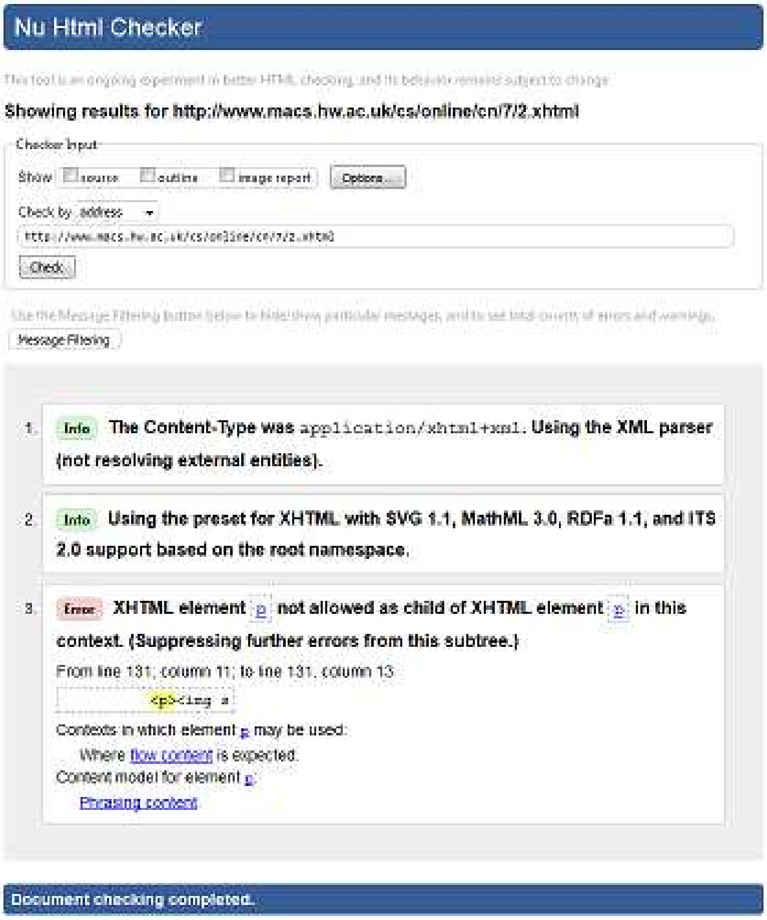
HTML Validation

HTML validation can be done via *http://validator.w3.org/*

HTML to validate can be specified by

- URL
- uploading HTML file from local file system
- pasting HTML to validate into a window

Validation report on XHTML 5 web page:



This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change.

Showing results for http://www.macs.hw.ac.uk/cs/online/cn/7/2.xhtml

http://www.macs.hw.ac.uk/cs/online/cn/7/2.xhtml

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

1. Info The Content-Type was `application/xhtml+xml`. Using the XML parser (not resolving external entities).
2. Info Using the preset for XHTML with SVG 1.1, MathML 3.0, RDFa 1.1, and ITS 2.0 support based on the root namespace.
3. Error XHTML element `p` not allowed as child of XHTML element `p` in this context. (Suppressing further errors from this subtree.)

   From line 131, column 11; to line 131, column 13

   ```
   <p><img s
   ```

   Contexts in which element p may be used:
       Where flow content is expected.
   Content model for element p:
       Phrasing content.

Document checking completed.

Validation link can be added to any HTML page with

```
<a href="http://validator.w3.org/check?uri=referer">Valid HTML?</a>
```

Clicking on link asks W3C validator to validate invoking page.

Text and Hypertext

When rendering text in HTML, the HTML processor normally

- collapses inter-word whitespaces to single output space
- wraps text to fit within the available margins
- justifies text on left margin but not on right

Line breaks can be forced on wrapping process by the <*br*> element.

HTML supports two types of container elements

block-level starts on newline and holds block-level and inline elements
examples include <*body*>, <*div*>, <*form*>, <*p*>, <*table*>

inline can only contain data and other inline elements
examples include <*a*>, <*b*>, <*code*>, <*img*>, <*q*>, <*span*>

<*div*> and <*span*> are general grouping elements that self-nest.

Paragraph tags <*p*> are block-level elements that don't self-nest.

Hypertext

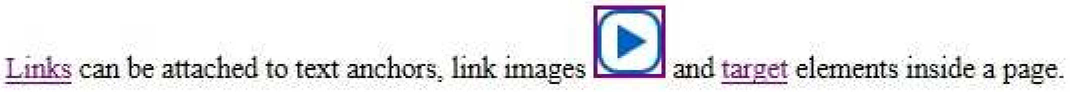
Hyperlinks and anchors for links are specified as follows:

<a href="URL">...</a>	clickable hyperlink to another page
<... id="anchor">	identifies tag of any type inside page
<a href="URL#anchor">...</a>	link to *anchor* inside another page

HTML supports hyperlinking round text, images and inline markup:

```
<a href="4.xhtml">Links</a> can be attached to text anchor,
link image <a href="6.xhtml"><img src="next.gif" alt="icon"></a>
and <a href="9.xhtml#icon">target</a> element inside a page.
```

which renders as follows:



Links can be attached to text anchors, link images and target elements inside a page.

Lists

HTML supports 3 kinds of lists

definition	<dl> ... </dl>
ordered	<ol> ... </ol>
unordered	<ul> ... </ul>

Definition lists are composed of definition titles and data:

```
<dl>
  <dt>Volume 1</dt>
    <dd>The Fellowship of the Ring</dd>
  <dt>Volume 2</dt>
    <dd>The Two Towers</dd>
  <dt>Volume 3</dt>
    <dd>The Return of the King</dd>
</dl>
```

Ordered list itemises its contents with numerical prefixes:

```
<ol>
  <li>Mercury</li>
  <li>Venus</li>
  <li>Earth</li>
</ol>
```

Deprecated <ol> attribute *type* can be "1", "a", "A", "i" or "I".

Unordered list itemises with bullet point prefixes:

```
<ul>
  <li>Health</li>
  <li>Wealth</li>
  <li>Happiness</li>
</ul>
```

Deprecated <ul> attribute *type* can be "disc", "circle" or "square".

Approved way to customise lists is by altering list style attributes.

eXtensible Markup Language

eXtensible Markup Language is meta language to structure text data.

XML in contrast to HTML was designed to

- be *extensible* - tags, attributes and values can be defined by users
- support *rich* structure - letting complex data be represented
- require *validation* or at least *well-formedness*

XML parsers *always* reject documents with *any* bad syntax in them.

XML is simpler but as useful as its meta language precursor SGML.

XML documents consist of:

declaration	definition of object types and attributes, plus values of named entities in document DTD
element	instance of object, attribute or related value as defined by DTD that appears in XML body
comment	explanatory text and data ignored by XML parser
character reference	definition of notation used to denote text and other symbols in document given by character sets
processing instruction	directive for related apps e.g. stylesheet
CDATA section	raw character data to be ignored

XML tags

- are *case* and *order sensitive*
- come in *nested* pairs or are marked as *standalone*
- have exactly one top level or *root* element
- quote all attribute values

XML Examples

Simple XML example with declaration which must be first if present:

```
<?xml version="1.0"?>
<person born="1912-06-23" died="1954-06-07">
  <name> <first>Alan</first> <family>Turing</family> </name>
  <profession>computer scientist</profession>
</person>
```

More complicated example with (non-existent) DTD specified:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE titlepage SYSTEM "http://www.id.org/dtd/hi.dtd"
    [<!ENTITY rights "GNU General Public License GPLv3">]>
<?xml-stylesheet version="1.0" type="text/xsl" href="style.xsl"?>
<titlepage>
  <!-- This is a comment! -->
  <title font="Baskerville" size="24/30" alignment="centered">
      &quot; Munde Salutem &quot; </title>
  <white-space type="vertical" amount="12"/> &rights;
  <image location="http://www.foo.bar/fleuron.eps"
         type="URL" alignment="centered"/>
  <![CDATA[ bad XML tag example: <2-faced name=Jason> ]]>
</titlepage>
```

XML supports *modularisation* of names via *namespaces* to

- distinguish elements and attributes from different tag sets
- group related elements and attributes together

Namespace example qualifies name with *prefix* and links it to *URI*:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
 <rdf:Description rdf:about="index.htm">
   <dc:title> Web Programming Lectures </dc:title>
   <dc:creator> Hamish Taylor </dc:creator>
 </rdf:Description>
</rdf:RDF>
```

Default namespace example gives it to svg, ellipse, rect tags:

```
<svg xmlns="http://www.w3.org/2000/svg"
     width="12cm" height="10cm">
  <ellipse rx="110" ry="130" />
  <rect x="4cm" y="1cm" width="3cm" height="6cm" />
</svg>
```

XML Syntax Errors

Example of several XML errors

```
<?xml version="1.0" encoding="UTF-8"?>
<item type=mango>
  <price> 6 euros </Price>
  <Description> nice
    <weight> 127 grammes
  </Description> </weight>
</item>
<item type="pear" quality="good" type="pear">
  <price> 2.53 euros
</item>
```

One way of correcting these errors:

```
<?xml version="1.0" encoding="UTF-8"?>
<basket>
  <item type="mango">
    <price> 6 euros </price>
    <Description> nice </Description>
    <weight> 127 grammes </weight>
  </item>
  <item type="pear" quality="good">
    <price> 2.53 euros </price>
  </item>
</basket>
```

Corrections

- introduce <*basket*> element to get sole root element
- put quotes round value of *type* attribute in <*item*>
- coerce </*Price*> tag to lower case
- reorder </*Description*> and </*weight*> tags
- add </*price*> tag to close child of last <*item*> tag
- remove repeated *type* attribute in last <*item*> tag

XML parsers *always* reject bad XML unlike HTML parsers.

XHTML

XHTML is HTML expressed in XML syntax

- employing the "http://www.w3.org/1999/xhtml" namespace
- that is either XHTML 1.0, XHTML 1.1 or XHTML 5+
- using a *subset* of HTML syntax with *optional* XML extras

XHTML doesn’t allow valid HTML expressions like

<I>Keyser Soze</i>	mismatched tags, XML tags are case sensitive
<img src=pic.jpg>	unquoted attribute value
<p>Finis<p>	unclosed tags - use *<p>Finis</p>* or *<p/>Finis<p/>*
<video autoplay>	valueless attribute - use *<video autoplay="autoplay">*

Polyglot XHTML is HTML and XML at same time with same meaning

- where all element and attribute names are in *lower case*
- parsed as XML when served as *application/xhtml+xml*
- parsed as HTML when served as *text/html*

Examples of non-polyglot XHTML - valid XML but invalid HTML

* </br>*	 has no end tag in HTML
<![CDATA[&]]>	*illegal HTML tag and illegal solo &*

Strict version of XHTML 1.0 is declared by

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML is best written by declaring namespace in root element.

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Add validation links to XHTML pages to aid compliance checking.

HTML 5

Up to 2009 W3C aimed to evolve modular XHTML from 1.1 into 2.0.

Web Hypertext Application Technology Working Group

- was founded in 2004 to revive interest in HTML
- was led by developers from Apple, Mozilla, Opera Software
- was unhappy about W3C shift to XHTML and then XML

In 2009 W3C agreed to work instead with WHATWG on HTML 5.

Objectives for new version of HTML were to

- base new features on HTML, CSS, DOM and JavaScript
- provide more markup to replace scripting
- reduce need for external plugins
- achieve more device independence
- handle errors better in forms and elsewhere

HTML 5 has 3 representations - HTML, XHTML, in memory DOM.

Recommended template for HTML 5 page is its XHTML serialisation:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <meta charset="utf-8" />
    <title> ... </title>
  </head>
  <body>
    ...
  </body>
</html>
```

XHTML with ".html" or ".htm" suffix is served as *text/html*.

XHTML with ".xhtml" suffix is served as *application/xhtml+xml*.

Browsers use XML parser to read latter. Errors stop page loading.

Accessing Web Pages

Browsers fetch HTML pages from local file system or a web server.

Local file system includes files

- on file systems exported to local PC - *H:\page.html*
- on file systems on Unix host - */home/cs2/bozo/page.html*

These files are referenced by the following URLs

file:///H:/page.html on Windows host

file:///home/cs2/bozo/page.html on Unix host

HTML pages fetched from a web server via HTTP

- name a web server to fetch from - *http://www.macs.hw.ac.uk*
- specify a path and file name - */cs/online/wp/index.xhtml*
- are fetched from path relative to web server's *root directory*

Different web servers may have different root directories:

http://www2.macs.hw.ac.uk/index.html/var/www/html on *anubis*

http://www.macs.hw.ac.uk/index.htm/home/www on *athena*

If path mentions a user such as *bozo*

- using URL such as *http://www2.macs.hw.ac.uk/˜bozo/page.htm*
- path is relative to web directory under his home directory

Web servers may use different names for these web directories:

http://www2.macs.hw.ac.uk/˜bozo/page.htm ˜bozo/public_html/page.htm

http://www.macs.hw.ac.uk/˜hamish/page.htm ˜hamish/www/page.htm

Things To Do

Read	Getting started with HTML - https://www.w3.org/MarkUp/Guide Dave Raggett, World Wide Web Consortium, May 2005.
Read	What’s the point of XML? Tittel E, Sun World Online, Feb 1998
Try	*Quiz on Lecture*
Do	*Exercise 3*

Key Points

- HyperText Markup Language is a readable document formatting language that supports hyperlinking, presentational markup for text, images, lists, tables, forms, embedded objects and AV media, as well as scripting and stylesheets.

- HTML is designed to support scrollable, reflowable presentation of text and other formatted content that adjusts its use of the layout to the available width without requiring lateral scrolling.

- XML is a markup language like HTML but with a simpler and more rigorous syntax that defines the structure of a textual document so it can be presented, semantically checked, transformed or filtered by programs as well as read by people.

- XHTML is HTML expressed in XML syntax enabling XHTML documents to offer the greater simplicity of XML markup, the rigor and processability of XML, and the presentational properties of HTML.

Adding Style With CSS

Cascading Style Sheets or CSS is being revised as CSS 2 version 2.

CSS directives specify how to present HTML and XML with pattern

```
selector { property: value; ...; property: value }
```

They govern presentation on screen or in print and can be given in

- *link* statement in head section of HTML page
- *<style>* tags in HTML page
- style *attribute* of any HTML tag

Style directives in file URI can be declared in HTML head section

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

Style directives for HTML can also be embedded in a page

```
<style>
  p {
    line-height: 200%;
    hyphens: none
  }
  h1 { font-weight: bolder; }
</style>
```

A style attribute can also be attached to any HTML tag

```
<h3 style="width: 50%; font-style: normal">Conclusion</h3>
```

Style can also be declared by import directives in CSS.

```
color: red;
@import url("http://style.com/newlook.css");
@import "http://style.com/default.css";
```

Stylesheet directives are given in XML via processing instructions

```
<?xml-stylesheet type="text/css" href="style.css"?>
```

Internal override external directives. Later override earlier.

This is why style sheets in CSS are said to *cascade*.

Selectively Applying CSS

All HTML elements can be given one or both of the attributes

class	value is shared identifier of one or more elements
id	value is unique identifier of that element

CSS selector syntax supports their exploitation:

tag.identifier	selects all elements of name *tag* of class *identifier*
tag#identifier	selects element *tag* whose id is *identifier*

Style declaration in CSS

```
p.blue { color: blue; font-size: smaller; text-align: center }
p.emph { color: red; font-style: italic }
h4#garish { color: fuchsia; font-family: fantasy; margin-left: 5% }
```

can be applied to HTML markup

```
<p class="emph"> Business is growing. </p>
<p class="blue"> Household debt is increasing. </p>
<p class="emph"> House prices are rising. </p>
<h4 id="garish"> Inflation is looming! </h4>
```

to apply shared styles to some elements and unique style to another.

CSS2 selectors include the following patterns:

*	matches any element
E	matches any element E
E > F	matches any element F that is child of E
E + F	matches any element F immediately preceded by E
E[foo]	matches any element E with attribute foo set to any value
E[foo="pah"]	matches any element E with attribute foo set to "pah"
E:link	matches any unvisited hyperlink

Styling Text Blocks

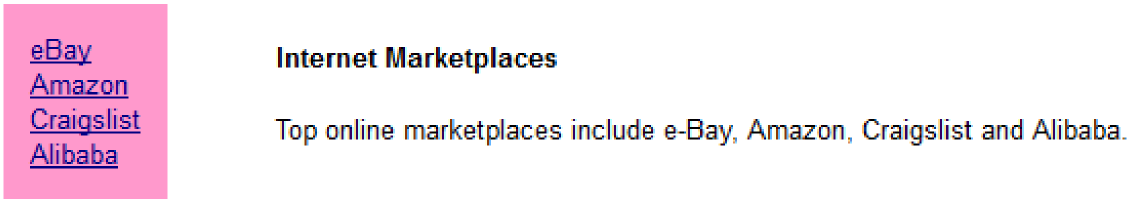
Style directives in CSS

```
main#content {
  position: absolute;
  left: 12em
}
nav.links {
  background-color: #ff99cc;
  margin-left: 2em;
  padding: 1em;
  width: 4em
}
```

can be applied to HTML markup

```
<div style="position: relative">
  <main id="content">
    <h4>Internet Marketplaces</h4>
    <p>Top online marketplaces include e-Bay, Amazon,
       Craigslist and Alibaba.</p>
  </main>
  <nav class="links">
    <a href="http://www.ebay.com">eBay</a> <br>
    <a href="http://www.amazon.com">Amazon</a> <br>
    <a href="http://www.craigslist.com">Craigslist</a> <br>
    <a href="http://www.alibaba.com">Alibaba</a>
  </nav>
</div>
```

to split markup into a part for navigation and part for content.



eBay
Amazon
Craigslist
Alibaba

Internet Marketplaces

Top online marketplaces include e-Bay, Amazon, Craigslist and Alibaba.

Add CSS validation link to any HTTP loaded (X)HTML page with

```
<a href="http://jigsaw.w3.org/css-validator/check/referer">CSS</a>
```

CSS validator can also be used via URI, file upload or cut and paste.

CSS 2.1

CSS 2.1 is latest *full-spectrum* standard and

- defines more than 90 properties
- allows vendor properties such as *-moz-border-radius*

All current CSS standards (2 and 3) are given in 2017 snapshot.

CSS 2.1 uses various size measures and %s of inherited values

em	font-size of relevant font
ex	height of x character in relevant font
in	inches - 2.54 centimetres is 1in
cm	centimetres
pt	points - 1/72 of an inch
pc	picas - 1 pc is equal to 12pt
px	pixel units - 1px is equal to 0.75pt

Colour words include 17 original and more in CSS 3 color module

aqua black blue fuchsia gray green
lime maroon navy olive orange purple
red silver teal white yellow

Use a Colour Map to specify other hues by RGB values:

rgb(x, y, z)	*x*, *y* and *z* are integer values between 0 and 255
rgb(x, y, z)	*x*, *y* and *z* are percentages between 0% and 100%
#RrGgBb	*R*, *r*, *G*, *g*, *B*, *b* are hexadecimals in range 0 .. 9, A .. F

Same colour is specified by each of

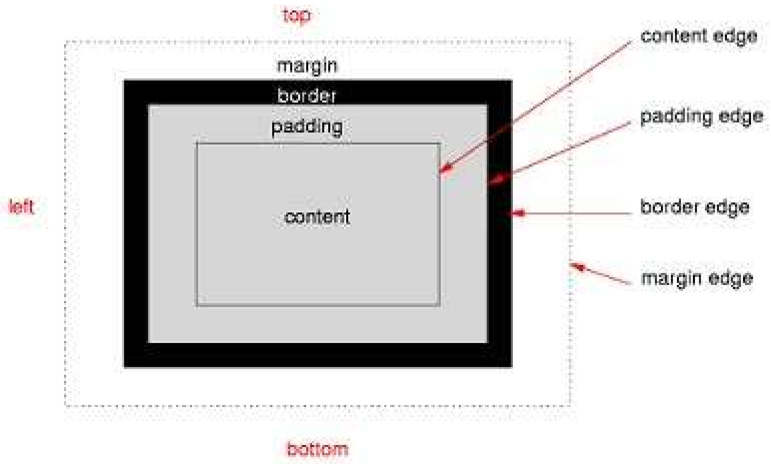
rgb(128, 128, 0) rgb(50%, 50%, 0%) #808000 yellow

CSS Box Layout

CSS 2.1 defines 4 layout modes

block	for documents
inline	for text
positioned	for explicit positioning
table	for 2D data in tabular format

CSS 2.1 box model is based on content surrounded by its context:



Properties can be set for all these surrounds:

```
<style>
  .out { border: thick dashed red;
         width: 15em;
         margin-left: 3em }
  .in { border: thin dotted blue;
        padding: 12px 6px 4px 2px;
        margin: 2px 4px 6px 12px }
</style>
<div class="out">
  <div class="in"> The truth is rarely pure and never simple. </div>
</div>
```

Borders can be set via

border-style	none, dotted, dashed, solid, ridge etc.
border-...	e.g. *border-top* for top, right, bottom, left
border	sets *border-width*, *border-style*, *border-color*

CSS and Tables

Box model replaces table attributes - *border*, *cellpadding*, *cellspacing*.

CSS property *border-collapse* has 2 models for table borders

collapse borders are merged and cell groups can have borders
border conflicts resolve towards *hidden*, *thick*, *solid*

separate default model, only cells and table have borders not rows etc
border-spacing gives distance among cell borders

Consider a table

```
<table id="t1">
 <colgroup> <col id="col1"/> <col/> <col/> </colgroup>
 <tr>
   <td class="c1">Sunday</td> <td>red top</td> <td>Sunday Post</td>
 </tr>
 <tr id="row2">
   <td class="c1">Daily</td> <td>broadsheet</td> <td>The Times</td>
 </tr>
 <tr>
   <td class="c1">Daily</td> <td>red top</td> <td>The Sun</td>
 </tr>
 <tr>
   <td class="c1">Sunday</td> <td>broadsheet</td> <td>Sunday Times</td>
 </tr>
</table>
```

It can be styled by collapsed border model with

```
table#t1 { border-collapse: collapse }
td { padding: 6px;
     border: thin solid blue }
col#col1 { border: thick solid red }
tr#row2 { border: thick solid green }
```

or styled by separated border model with

```
table#t1 { border-collapse: separate;
           border-spacing: 3px }
td { padding: 6px;
     border: thin solid blue }
td.c1 { border: thick solid red }
tr#row2 td { border: thick solid green }
```

CSS 3 and Multi-Columns

CSS 3 is being developed as modules that include:

2D/3D transforms	transformations on shape, size, position and 3D rotations
animations	keyframe animations
backgrounds/borders	resizing background, background image locus, rounded borders, boxes with shadows
flexible box model	box model generalized to apply to vertical text, properties to control marquee effect (speed, direction)
multi-columns	column count, column gap and rules between columns
text effects	text with shadows, word wrapping
transitions	changes to CSS properties over a time period
user interface	resizing elements, sizing boxes, offsetting outlines

Columns

Multi-columns are specified by CSS 3 *column* properties:

column-count	number of columns
column-gap	width between columns
column-rule	specifies color, style and width of the rule

3 columns with green rules is produced by the CSS 3 markup

```
<style>
  div#ex1 {
    column-count: 3;
    column-gap: 30px;
    column-rule: 3px outset green
  }
</style>
<div id="ex1">
  <p> ... </p>
</div>
```

CSS 3 and Text Effects

CSS 3 supports underlines, text shadows and emphasis marks.

These text decoration properties include:

text-decoration	values for line, style and color
text-decoration-line	none, underline, overline, line-through
text-decoration-style	solid, double, dotted, dashed, wavy
text-shadow	h-shadow v-shadow blur-distance color

Example style directives include

text-decoration: overline wavy green

text-decoration: underline solid red

text-shadow: 5px 5px 2px gray

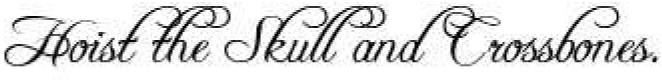
Some text effects may not be supported by your browser.

CSS 3 users can now load their own WOFF, TTF and OTF fonts:

```
<style>
  @font-face {
    font-family: myScript;
    src: url(freebooterscript.woff)
  }

  p.freebooter {
    font-family: myScript;
    margin-left: 3em;
    font-size: xx-large
  }
</style>
<p class="freebooter"> Hoist the Skull and Crossbones. </p>
```

It renders as follows:



Hoist the Skull and Crossbones.

CSS 3 and Backgrounds

CSS 3 supports flexible formatting of images in background.

These new CSS3 style properties include:

background-image	URL of source
background-position	direction percentage length
background-size	height width
background-repeat	repeat round space no-repeat
background	image position size repeat origin clip

Values of these properties can be supplied in various ways.

Repeating background is achieved by

```
<style>
  div.caption {
    background: url(water.jpg) repeat;
    width: 20em;
    padding: 1em 4em;
    color: white;
    font-size: x-large;
    font-weight: bold;
    margin-left: 3em
  }
</style>
<div class="caption"> Darkness was upon the face of the deep. </div>
```

HWU logo could be put into fixed background position with:

```
<style>
  body {
    background-image: url(hwlogo.png);
    background-attachment: fixed;
    background-position: right top;
    background-repeat: no-repeat;
  }
</style>
```

Fixed content doesn’t move when a web page scrolls.

CSS 3 and Borders

CSS 3 supports rounded corners, shadows, and border images:

border-radius	radius of curvature for corner
box-shadow	h-shadow v-shadow blur spread color
border-image	source slice width outset repeat

A box with rounded corners is achieved by

```
<style>
  .round {
    border-radius: 20%;
    border: 3px dotted blue;
    padding: 18px;
    width: 110px;
    height: 18px;
  }
</style>
<p class="round" style="width: 110px"> Let there be light. </p>
```

Border-radius of 50% with same height and width produces a circle.

Box shadows are given in much the same way as text shadows:

```
<style>
  div.rectangle {
    box-shadow: 2px 2px 2px 2px black;
    width: 15em;
    padding: 12px;
    background-color: #CCCCFF;
  }
</style>
<div class="rectangle">Am I dying, or is this my birthday?</div>
```

Spread parameter controls expansion or contraction of shadow shape.

Border-image uses inward slices from 4 image sides to define border

```
<style>
  #vines {
    border: 40px solid transparent;
    padding: 15px;
    width: 160px;
    border-image: url(vines.jpg) 155 170 repeat;
  }
</style>
<p id="vines">La belle dame sans merci.</p>
```

CSS 3 and Transforms

CSS 3 supports 2D and 3D transformation properties

transform	list of transformation functions
transform-origin	move origin to X, Y

Origin is initially set to middle of content’s bounding box i.e.

```
transform-origin: 50% 50%
```

Transformation functions include:

translate(X, Y)	2D translation in X and Y directions
scale(N, M)	2D resizing by N horizontally and M vertically
rotate(A)	2D rotation clockwise about origin through angle A
skew(X, Y)	rotation about X and Y axes
rotateX(N)	3D rotation about X axis
rotateY(N)	3D rotation about Y axis

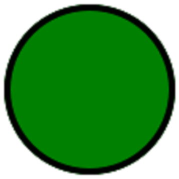
Box is spun about its centre by

```
<style>
  div#box {
    background-color: aqua;
    border: 1px solid red
    height: 2em;
    width: 7em;
    transform: translate(12em, 0em) rotate(0deg);
  }
</style>
<div id="box"> </div>
<script type="text/javascript">
  function spin(n) {
    var el = document.getElementById("box");
    el.style.transform = "translate(12em, 0em) rotate(" + n + "deg)";
    n = n + 1;
    window.setTimeout("spin("+ n + ")", 30);
  }
  spin(0);
</script>
```

JavaScript updates its rotation by 1 degree every 30 msec.

CSS 3 and Animations

CSS 3 animations can dynamically alter any CSS properties.



Animations are enumerated via keyframes and changes smoothed.

In this animation a green ball bounces forever:

```
<style>
  #ball {
    width: 100px;
    height: 100px;
    position: relative;
    background: green;
    border: 4px solid black;
    border-radius: 50%;
    animation: bounce 5s linear infinite;
  }

  @keyframes bounce {
     from { left: 250px; top:-100px; }
     25%  { left:   0px; top: 150px; }
     50%  { left: 250px; top: 400px; }
     75%  { left: 500px; top: 150px; }
     to   { left: 250px; top:-100px; }
  }
</style>
<div id="ball"></div>
```

animation property encapsulates several properties including

```
animation-name: bounce;
animation-duration: 5s;
animation-timing-function: linear;
animation-iteration-count: infinite;
```

Change timing function to its default value "ease" and duration to 3s.

CSS 3 and Transitions

CSS 3 transitions apply property changes smoothly over a period.

Transitions can create dynamic effects such as a spinning black hole.

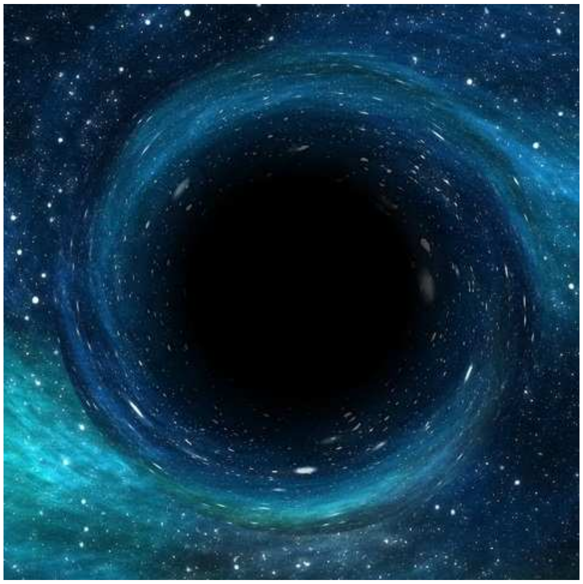


Image rotates and its border alters when hovered over:

```
<style>
  #morph {
     border: 1px dotted red;
     width: 240px;
     height: 240px;
     transition: all 1s linear 1s;
  }
  #morph:hover {
     border: 40px solid black;
     transform: rotate(360deg) scale(5, 5);
  }
</style>
<div id="morph">
  <img src="blackhole.png" alt="black hole" style="width: 240px"/>
</div>
```

CSS 3 transform is used to rotate the black hole.

transition property encapsulates several properties including

```
transition-property: all;
transition-duration: 1s;
transition-timing-function: linear;
transition-delay: 1s;
```

Change timing function to *ease*, duration to 3s and delay to 3s.

Things To Do

Read	Chapter 4, Styling with CSS *Dynamic Web Programming and HTML5* Paul S Wang, CRC Press, 2013.
Browse	CSS 3 Tutorial W3 Schools - http://www.w3schools.com/css/css3_intro.asp
Try	*Quiz on Lecture*
Do	*Exercise 11*

Key Points

- CSS directives specify how to present markup. They are given in files, inline within style tags or in style attributes. Introducing CSS rather than adding more attributes to HTML is a simpler and more generic way to specify presentation in a wider set of styles.

- CSS is collected from various sources and reconciled for consistency by favouring the inner over the outer and the later over the earlier. The CSS 2.1 standard defines more than 90 properties which are applied to HTML markup using CSS selectors.

- CSS 3 is being developed as modules addressing issues like multi-columns, text effects, backgrounds, borders, transforms, transitions and animations.