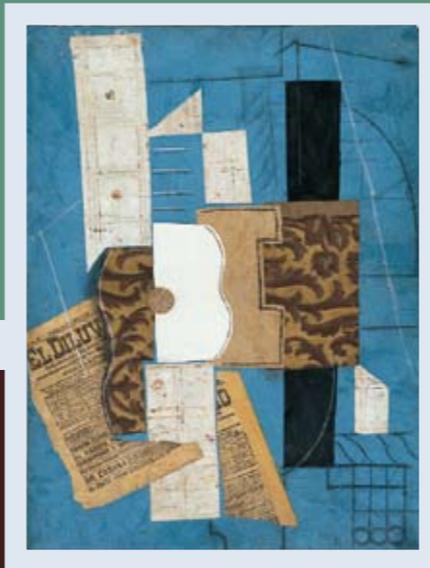


digital *multimedia*

nigel chapman and jenny chapman



Graphics and Colour
Video and Animation
Sound
Text and Typography
Hypermedia
Flash and DOM Scripting
Multimedia and Networks

**Third
Edition**

10

Hypermedia

Based on material from
***Digital Multimedia*, 3rd edition**
published by John Wiley & Sons, 2009
© 2009 Nigel Chapman and Jenny Chapman

These lecture slides © 2009
Nigel Chapman and Jenny Chapman

All figures © MacAvon Media Productions

Text Layout Using XHTML and CSS

Text in an XHTML document is marked up with tags that delineate the document elements corresponding to the logical divisions of the text.

Each element consists of a start tag, followed by its content and an end tag.

A start tag consists of the element name written in angle brackets.

An end tag is the same except that a slash precedes the name, e.g. `<p>...</p>`.

**Empty elements have no content. Their start and end tags may be combined, e.g. `
`.**

Elements may have attributes, whose values are assigned in the start tag.

Character entity references and numeric character references may be used to represent characters that are used as part of the markup or which are not available on a conventional keyboard.

Stylesheet rules written in CSS may be used to control the layout of elements.

The selector of each rule determines which elements it applies to, and the corresponding declarations set properties that control their appearance.

The class attribute is used to distinguish between different subsets of an element type, so that finer control can be exerted over their layout.

An XHTML document begins with a document type declaration. Its root is the html element, which contains the head and body elements.

Stylesheet rules may appear in a style element in the head.

Block-level elements are normally formatted as discrete blocks; inline elements are run in to the surrounding text.

The block-level elements in XHTML include

**p (paragraph),
h1–h6 (headings),
br (line break),
hr (horizontal rule),
blockquote and
pre (pre-formatted).**

Unordered, ordered and definition lists are marked up as ul, ol or dl elements, which contain li elements (ul or ol) or pairs of dt (term) and dd (definition) elements.

Divisions of a document that should be treated as a unit are identified by div elements.

Inline elements include em (emphasis) and strong; span is used to identify arbitrary inline divisions.

Any element may have a class attribute, and/or an id attribute with a unique identifying value.

Font characteristics are controlled by the font-family, font-style, font-variant, font-weight and font-size properties in CSS.

The font-family property's value is a list of font names in decreasing order of preference.

Generic font families (serif, sans-serif, monospace, cursive and fantasy) may be used as fallback values.

The font-style property can have the values normal, italic, or oblique.

The font-weight property can have the values normal or bold, or 100, 200, ..., 900. The values bolder or lighter may be used to set the weight relative to the inherited value.

The font-size property can have the values xx-small, x-small, small, medium, large, x-large, xx-large, larger or smaller.

Font sizes can also be set numerically in absolute units (usually px), relative units (em), or as a percentage of the inherited value.

Leading is specified using the line-height property.

All of the font properties may be set at once using the font property.

The colour of text is set using the color property; a background colour may be set using background-color.

Colours in the sRGB colour space may be specified as percentages, numbers or a hexadecimal value.

A few named colours may also be used.

The text-align property controls alignment (left, right, center or justify).

Each element is placed in a box.

Inline elements are wrapped inside the enclosing block's box.

Block element boxes are arranged vertically.

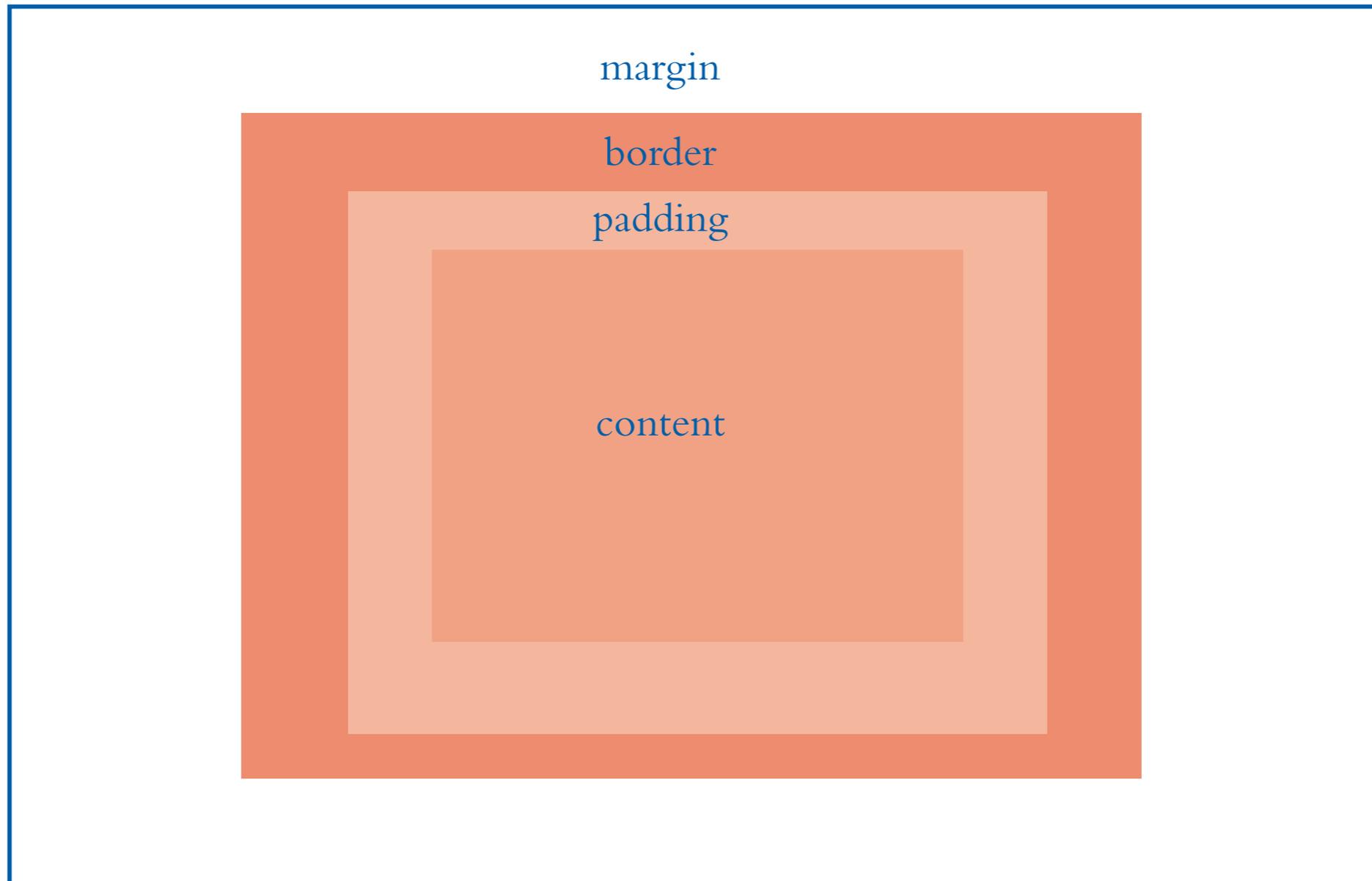
text text text inline text text text text text text text
text text text text text text text text text text text text text text text
text text text text text text text text text text text text text text text
text text text text text inline text text text text text text
text inline text
text text text text text text text text text text text text text text text
text text text text text text text text text text text text text text text
text text text text text text inline text
text text text text text text text text text text text text text text text
text text text text text text text text text text text text text text text

block

block

CSS layout algorithm for normal flow

Each box may have a border, separated from the box's contents by some padding. Margins separate the box from its neighbours.



CSS box model

Padding and margin widths, and border width, colour and style can all be set with collections of properties that set the top, right, bottom and left values independently, or a single property that sets them all at once.

Boxes can be floated to the left or right margin using the float property.

```
<body>
<p class="leftfloater">
Left floated text will move to the left ...
</p>
<p class="rightfloater">
Right floated text will move to the right...
</p>
<p>
The main text flows past ...
</p>
<p class="clear">
When a paragraph belonging to the class "clear" is
encountered...
</p>
</body>
```

```
p.leftfloater {
margin-left: 0;
float: left;
width: 30%;
}
p.rightfloater {
margin-right: 0;
float: right;
width: 30%;
}
p.clear {
clear: both;
width: 90%;
}
```

Left floated text will move to the left of the page, while the main body flows sublimely past it.

The main text flows past the floaters, accommodating itself to the space in between them.

Right floated text will move to the right of the page.

When a paragraph belonging to the class "clear" is encountered, the layout resumes below the floated material, like this.

Floated elements

Setting the position property to absolute allows the coordinates of a box's corners to be set explicitly so it can be positioned anywhere on the page.

The * selector matches any element. It is omitted if followed by a class name.

Context-sensitive selectors are used to apply rules to elements that occur within an element that matches some selector.

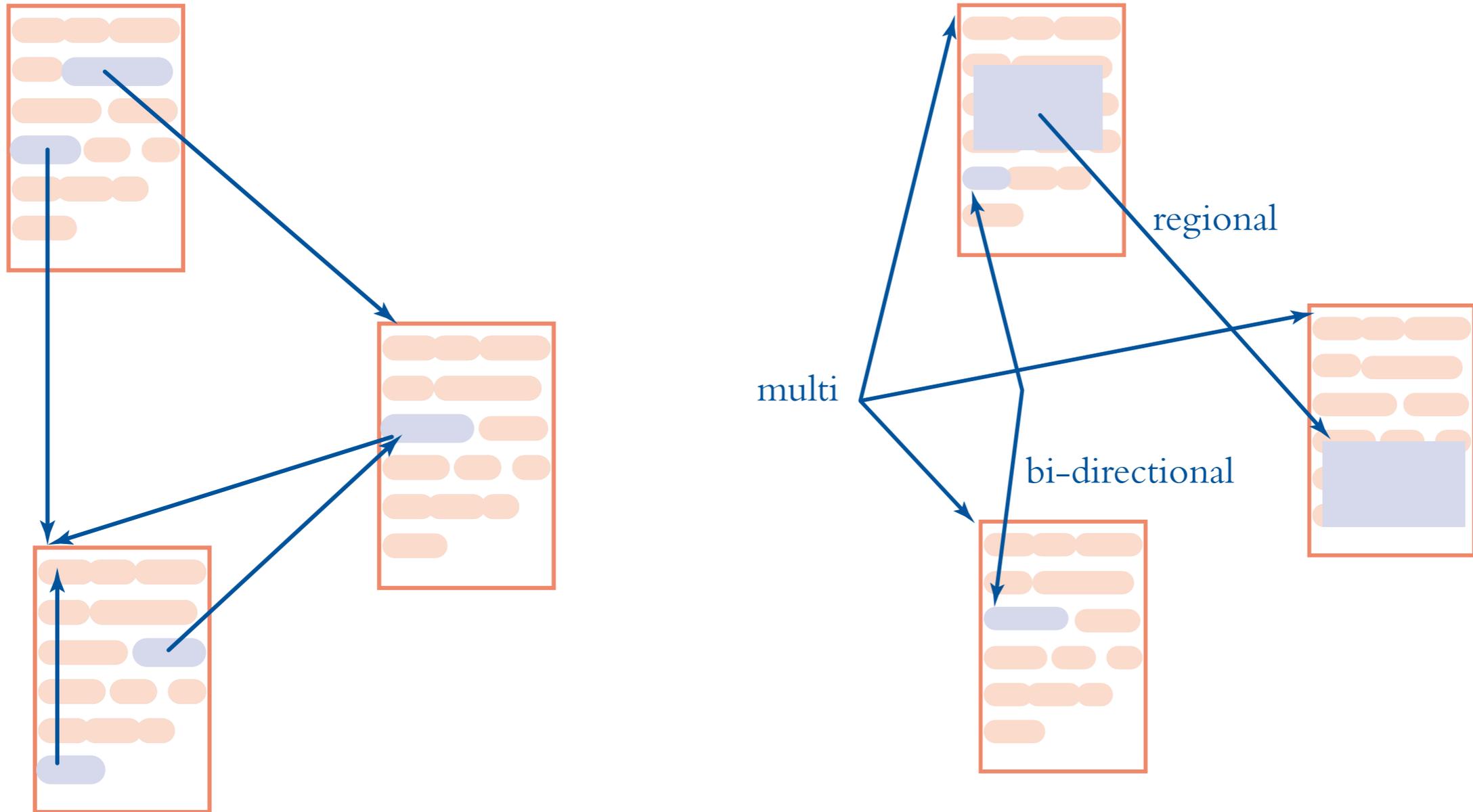
Pseudo-classes can be used to select first and last child elements.

Links

Links are used to make connections between nodes in a hypertext system.

Simple uni-directional links point from a single source to a destination.

More complex extended links (regional, bi-directional and multi-links) are possible, but are rarely used in practice.



Simple uni-directional links *Extended links*

URLs are used to identify destinations on the World Wide Web.

A Web page URL consists of the prefix `http://`, followed by a domain name identifying a host machine and a path which gives the location of the resource on the host.

Certain characters, including spaces, must be represented by a `%` followed by their ASCII code in hexadecimal when they appear in URLs.

An absolute URL has all three components.

In a relative URL some may be omitted, and resolved with reference to the URL used to retrieve the document.

A URL may have a fragment identifier appended, consisting of a # and a name that identifies a location within a page.

The a element is used in XHTML as the source of a link.

Its href attribute's value is the URL of the link's destination.

Special CSS pseudo-classes (:link, :visited, :active, :hover) are used to apply styling to links in different states.

The id attribute of any element can be used to give it a unique identifier that can be used in a fragment identifier.

The hreflang, charset and title attributes can be used to add extra information to a link. The rel attribute may be used to indicate its purpose.

The link element is used to create links between entire documents.

The Web and Hypermedia

Internet media types (also known as MIME types) provide a means of identifying the type of a resource.

An Internet media type takes the form type/subtype where type provides a broad indication of the sort of data and subtype specifies a more precise format.

Common examples include image/jpeg, text/css, audio/mpeg and application/octet-stream.

Plug-ins and ActiveX controls are used to extend the capabilities of browsers, allowing them to display additional types of media.

The `img` element is used to embed bitmapped images in Web pages.

Its `src` attribute is a URL pointing to the image data; its `alt` attribute provides a textual alternative.

The `width` and `height` attributes allow the browser to speed up page display by reserving space for the image before it has been downloaded.

Let us show you a [picture of some cows](#).

These cows have more presence:



Linking to (top) and embedding (below) an image

The object element is the officially preferred way of embedding multimedia, including video and Flash movies, in Web pages.

If the media object cannot be displayed, the content of the object is used as a substitute.

For plug-ins, the data attribute points to the media data and the type attribute gives its media type.

The width and height attributes are used to reserve space on the page.

Within the object's content, param elements, with name and value attributes, are used to pass parameters to the plug-in.

For ActiveX controls, the value of the classid attribute provides a Globally Unique Identifier for the control.

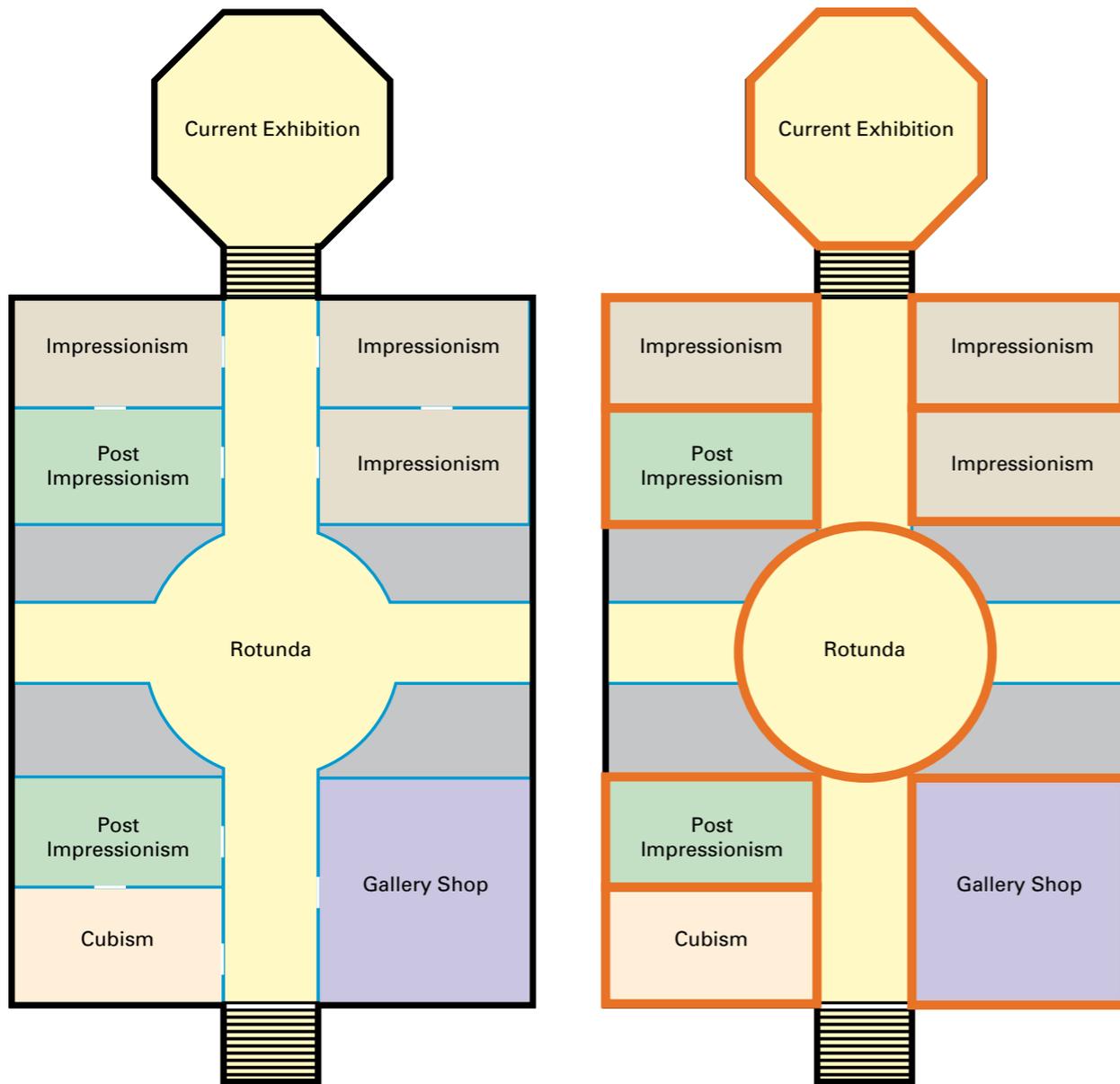
For video and Flash movies, a param element whose name is movie and whose value is a URL points to the movie.

Conditional comments or JavaScript must be used to combine both types of object so that media will display in any browser.

An image placed within an a element can act as a graphic link.

Where an image is used as a link it is essential to provide a textual alternative, as the value of the alt attribute, for the benefit of people using non-visual user agents.

Hot spots (active regions) can be added to an image with the map element and usemap attribute to create image maps.



An image map

The map element contains area elements defining the geometry of the hot spots.

Shape	Coordinates	Interpretation
rect	<i>left-x, top-y, right-x, bottom-y</i>	coordinates of top left and bottom right corners of the rectangle
circle	<i>centre-x, centre-y, radius</i>	coordinates of the centre and radius of the circle
poly	$x_1, y_1, x_2, y_2, \dots, x_n, y_n$	coordinates of the polygon's vertices

Image map areas