

Heath Old Boys Association Website

John R Hudson

25th February 2018

1 Introduction and structure

The HOBA website has been developed using only HTML as set out in [HTML Living Standard](#),¹ with one exception, CSS as set out in [CSS current work](#), the [ARIA attributes](#) for those using assistive technologies and the earlier version of Remy Sharp's [HTML enabling script](#) which creates the HTML elements of which earlier browsers are unaware.²

The slightly cumbersome structure (figure 1) of the old frame-based website folders was retained to enable an easy transition from the old website to the current website. In addition to the content folders shown in the diagram, the HTML enabling script is held in the `js` folder and the banner, crest and hosting company logo in the `graphics` folder. All other images are held in the same folder as the page to which they relate.

Apart from the change in HTML version, the main changes have been to add a separate page covering news of the Association's charitable donations and to allocate most of the elements which had formerly appeared under Heath History to a number of separate pages. *The Heathen* got its own page while a number of items under Heath History which represented requests were merged with the items which had formerly appeared under the Noticeboard page to create the new Requests page. The remaining new pages represent sections of the old Heath History page.

Every page begins with the lines:

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

¹See [A note on layout](#) for the reasons for this.

²Later versions of this script no longer support the earlier versions of Internet Explorer.

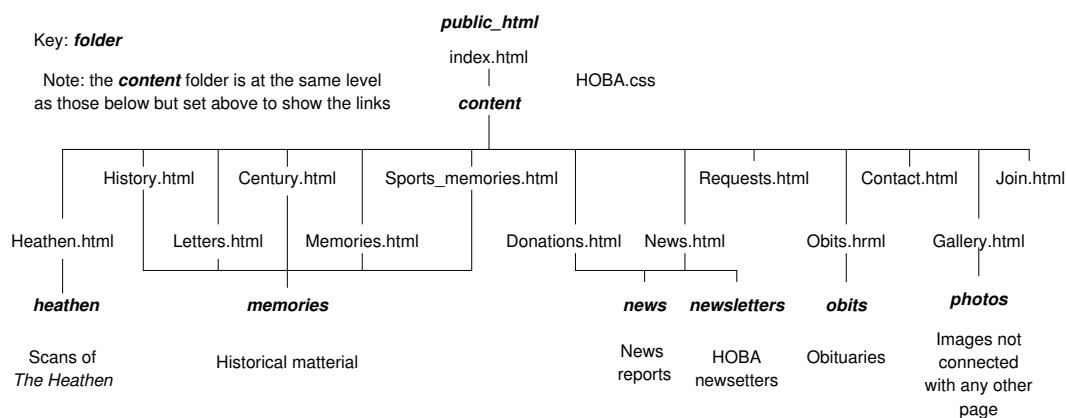


Figure 1: HOBA website structure

indicating that the page follows the post-2011 HTML standards and that its language is English.

2 <head>

The <head> element begins with the standard character set declaration.³ The title in the next line changes to reflect the top level page; that is, all obituary pages have the title ‘Heath Old Boys Association Obituaries.’ This is followed by the base URL which is the same in all pages and the link to the stylesheet. The <head> element concludes by running the earlier version of Remy Sharp’s script which is contained in a Javascript file in the js folder.⁴ Apart from the different titles, this element is the same in all pages.

```
<head>
  <meta charset="UTF-8"/>
  <title>Heath Old Boys Association Welcome</title>
  <base href="http://heatholdboys.org.uk/">
  <link rel="stylesheet" type="text/css" href="HOBA.css"/>
  <!--[if lt IE 9]>
    <script src="js/html5shiv.js"></script>
  <![endif]-->
</head>
```

3 <body>

Every <body> element contains the five semantic elements: <header>, <nav>, <aside>, <article> and <footer> (figure 2). The ARIA role attribute of the <body> element and of any of its child elements which have an ARIA role attribute is included in the opening tag.

```
<body role="document">
```

4 <header>

The <header> element has a background image and two headings which display on the right, the second being the page title. As with the <title> element in the head, this always reflects the top level page. So ‘Obituaries’ appears as the second heading on every obituary page. The second
 element ensures that the bottom of the background image displays.

```
<header role="banner">
  <h1 role="heading">Heath Old Boys Association</h1>
  <br>
  <h2 role="heading">Welcome</h2>
  <br>
</header>
```

Apart from the second heading, this element is the same on all pages.

³Google recommends including <meta name=viewport content="width=device-width, initial-scale=1"> in the <head> element in order to deal with a bug/feature in iOS but this ruins the display on other mobile phone browsers; so it has been omitted.

⁴All the folders used were part of the earlier website.

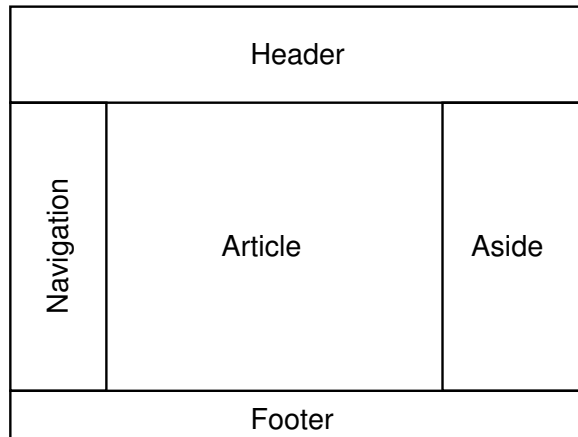


Figure 2: HTML page (thanks to [Dave Fisher](#))

5 <nav>

The <nav> element floats to the left and contains a menu of links to all the main pages. In second level pages, this menu contains all the links; in top level pages, the link to the showing page is omitted. The font of the elements in the <nav> element is defined as *cursive* in the CSS file so that they are clearly different from any other type of list and their line height is increased to provide more space between hyperlinks as recommended by Google.

```
<nav role="navigation">
  <br>
  <menu label="Home page" type="toolbar" role="toolbar">

  <li class="nav"><a href="index.html" type="text/html" accesskey="w W"
  role="link">Welcome</a></li>
  <li class="nav"><a href="content/News.html" type="text/html"
  accesskey="n N" role="link">News and events</a></li>
  <li class="nav"><a href="content/Donations.html" type="text/html"
  accesskey="d D" role="link">Donations</a></li>
  <li class="nav"><a href="content/Heathen.html" type="text/html"
  accesskey="t T" role="link">The Heathen</a></li>
  <li class="nav"><a href="content/History.html" type="text/html"
  accesskey="h H" role="link">Heath History</a></li>
  <li class="nav"><a href="content/Century.html" type="text/html"
  accesskey="a A" role="link">A Heathen Century</a></li>
  <li class="nav"><a href="content/Memories.html" type="text/html"
  accesskey="m M" role="link">Memories</a></li>
  <li class="nav"><a href="content/Sports_memories.html" type="text/html"
  accesskey="s S" role="link">Sports memories</a></li>
  <li class="nav"><a href="content/Gallery.html" type="text/html"
  accesskey="g G" role="link">Gallery</a></li>
  <li class="nav"><a href="content/Letters.html" type="text/html"
  accesskey="l L" role="link">Letters from old boys</a></li>
  <li class="nav"><a href="content/Requests.html" type="text/html"
  accesskey="r R" role="link">Requests</a></li>
  <li class="nav"><a href="content/Obits.html" type="text/html"
  accesskey="o O" role="link">Obituaries</a></li>
```

```

    <li class="nav"><a href="content/Join.html" type="text/html"
    accesskey="j J" role="link">Join us</a></li>
    <li class="nav"><a href="content/Contact.html" type="text/html"
    accesskey="c C" role="link">Contact us</a></li>
  </menu>
</nav>

```

The `accesskey` attribute has been added to every menu item — for which reason, there was careful selection of page titles to ensure that no two pages had the same initial letter.

The `
` element at the start of the `<nav>` element lowers the start of the `<menu>` element so that it does not appear above the main heading in the `<article>` element

6 `<aside>`

The `<aside>` element floats to the right and serves different functions on different pages. For example, on the ‘News and events’ page it contains links to earlier editions of the *HOPA Newsletter*. On all other pages than on the ‘Welcome’ page where it contains links to the most recent additions to the website, it contains the contact details of the Crossley Heath and Savile Park at Heath schools, the latter occupying part of the former Heath Grammar School site.

```

<aside role="complementary">
  <br>
  <p><a href="http://www.crossleyheath.calderdale.sch.uk/" type="text/html"
  role="link"><strong>Crossley Heath School</strong></a></p>
  <p>Savile Park<br>HALIFAX<br>HX3 OHG</p>
  <p>Tel: (01422) 360272<br>Fax: (01422) 349099</p>
  <p>Email: <a href="mailto:admin@crossleyheath.org.uk" type="text/html"
  role="link">admin@crossleyheath.org.uk</a></p>
  <p><a href="http://savileparkprimary.org.uk/" type="text/html"
  role="link"><strong>Savile Park Primary School</strong></a></p>
  <p>Free School Lane<br>HALIFAX<br>HX1 2PT</p>
  <p>Tel: (01422) 352844<br>Fax: (01422) 893506</p>
</aside>

```

The `
` element at the start of the standard `<aside>` element lowers the link to Crossley Heath School so that it does not appear above the main heading in the `<article>` element; in other `<aside>` elements, a heading element performs this function.

The `` elements within other `<aside>` elements are defined in the CSS file as having no list style to distinguish them from other types of list and their line height is increased to provide more space between hyperlinks as recommended by Google.

7 `<article>`

The `<article>` element contains all the main content; the margins set in the CSS file avoid any overlap with the `<nav>` and `<aside>` elements and ensure that those using very oblong screens do not have very long lines of text to read.

Three headings are defined in the CSS file for use within the article element: `<h3>` provides centred headings while `<h4>` and `<h5>` provide successively smaller left justified headings.

`<p class="first">` defines a larger font for use where the first paragraph of an article constitutes an introductory summary; this is not used on many pages, such as ‘A Heathen Century’ or ‘Obituaries,’ where the first sentence is not obviously an introductory summary.

`<p class="right">` defines a right justified paragraph to contain the publication date of an article.

The `<time datetime="YYYY-MM-DD">` element is used wherever a date consists of more than a year to provide assistive technologies with a machine readable date.

The `<figure>` element may take the attributes `class="left"` or `class="right"` which make them float to the left or right. This enables images or long quotations within a `<figure>` element to appear to the left or right of the body text.

The `<blockquote>` element is used to contain long quotations whether within an article or when the entire article consists of material from another source, such as a newspaper obituary. In the latter case, the `<blockquote>` element is placed within a `<figure>` element and a `<figcaption>` added to indicate the source of the quotation.

Short quotations are enclosed by the `<q>` element which generates double quotes; single quotes are used wherever quotation marks have been used but not to enclose direct speech, such as when reporting nicknames.

`<cite>` is used for the titles of books and artistic works and `<i lang=" " >` to indicate any foreign words and the source language.⁵ The latter is also useful for assistive technologies.

The en rule – is used between dates as in ‘Heath 1932–39’ and the em rule — to separate portions of text. If your HTML editor does not generate these easily, you can copy and paste them from other pages or from here.

8 <footer>

The `<footer>` element contains a Creative Commons licence, including a link to a Creative Commons icon, and, in older pages, has the name of the hosting company but now contains a Facebook link. `<p class="footer">` defines a smaller sans-serif font and centres the paragraph to distinguish it from text in other parts of the page.

```
<footer role="contentinfo">

  <p class="footer">‘Like’ our <a href="https://m.facebook.com/Heath-Old-Boys-Association-409705452726963/?fref=ts" type="text/html" role="link"></a> page and share it.</p>
  <p class="footer">Licensed under the <a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_GB" role="link">Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License</a> <a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_GB" role="link"></a></p>

</footer>
```

9 The CSS file

```
body {font: normal 100% serif;}
header {border-top: medium solid #C71585; border-right:
medium solid #C71585; border-left: medium solid #C71585;
background: transparent url("graphics/Topback_new.png")
```

⁵For the full list of language tags, see <http://www.iana.org/assignments/language-subtag-registry>

```

no-repeat; padding-right: 1em;}
nav {float: left; width: 19%; padding: 1em;}
aside {float: right; width: 19%; padding: 1em;}
article {margin-left: 22%; margin-right: 22%; padding: 1em;}
h1 {font: bold 180% sans-serif; text-align: right; color: gold;}
h2 {font: bold 160% sans-serif; text-align: right; color: gold;}
h3 {font: bold 150% sans-serif; text-align: center; color: #C71585;}
h4 {font: bold 120% sans-serif; color: #C71585;}
h5 {font: bold 100% sans-serif; color: #C71585;}
*.footer {font: normal 80% sans-serif; text-align: center;}
p.first {font: normal 120% serif;}
p.right {font: italic 100% serif; text-align: right;}
ul.link {line-height: 140%}
li.nav {font: bold 120% cursive; list-style: none; line-height: 150%}
li.aside {font: bold 100% sans; list-style: none; line-height: 140%}
li.alpha {list-style: lower-alpha;}
li.roman {list-style: lower-roman;}
td {padding-right: 1em; vertical-align: top;}
figure.left {float: left;}
figure.right {float: right;}
figcaption {text-align: center; font-style: italic;}
span {color: #C71585;}

```

Most of the contents of the CSS file have been explained already; the outstanding points are:

- all sizes are ratios so that the pages can appear on any size of device;
- the body text is serif; this is a deliberate choice as most of the content represents text which would naturally appear in print media;
- the school colours were claret and gold; however, there is no SVG colour directly corresponding to claret; so ‘mediumvioletred’ (hex: #C71585) was chosen as the best match;
- the border and 1em padding-right for the header were added in the light of experience;
- the line height declarations in ul.link, li.nav and li.aside provide more space between hyperlinks as recommended by Google;
- li.alpha provides alphabetic numbering and li.roman lower case Roman numbering of a list, normally one indented within an existing list;
- the td attributes are only intended for the list of committee members on the ‘Contact us’ page but happen to have no adverse impact on other tables elsewhere on the website;
- the element is used with ‘Recent additions’ on the Welcome page to hold comments like ‘updated’ where an item has been updated since it was originally posted.

10 Adding new pages

In most cases, new pages can be added by:

- copying and renaming an existing page from the same folder as the page is intended to be in;

- replacing the content between `<article role="article">` and `</article>` with new content;
- adding a suitable link to the relevant top level page and to the `<aside>` in `index.html`;
- altering the `<base href=" " >` definition to:

```
<base href="http://heatholdboys.org.uk/">
```

(this will normally have been different on the development site).

When adding a a new scan of *The Heathen* it is only necessary to upload it and copy and edit a link to it in the ‘Heathen’ page.

10.1 Images

Images with a width of less than 400 px are contained in a `<figure>` element with `class="left"` or `class="right"` to float them to the left or right of the content. They will then normally be centred on an older mobile ‘phone.

For larger images, a series of increasingly smaller images are created, the smallest being no wider than 400px, the links to them all being included in a `srcset=" "` attribute with the addition of their width in pixels. The `sizes` attribute is set to `sizes="45vw"`⁶ so that the image takes up the width of the `<article>` element, or roughly half the width of the screen, allowing for a margin on either side of the image.

```
<figure>
  
  <figcaption>Edwards, Potter, ... </figcaption>
</figure>
```

Browsers that support `sizes="45vw"` select one of the sizes in the `srcset=" "` list while the `src=" "` attribute is provided for browsers that do not support the `srcset=" "` attribute.

10.2 Links to `<aside>`

The ‘Welcome’ page `<aside>` element contains links to all recently added pages; normally, all that is needed is to copy over the link in the top level page and add the `class="aside"` attribute to the `` element. This list will be pruned if it gets too long and will not contain links to items more than a year old (which should have appeared in the most recent Newsletter).

The ‘News and events’ page `<aside>` element contains links to past *HOBA Newsletters*. After uploading a new Newsletter to the newsletters folder, it is only necessary to copy and edit another list item in the list to make it available.

⁶The relative value `v`, or 1% of the viewport, is at the later stages of approval and not yet formally part of CSS but is supported by Chrome, chromium, Firefox, Safari and Edge and is the only instance of using something which is not yet part of a standard.

A note on layout

Broadly speaking, there are two approaches to laying out digitally generated pages: using frames or using continuous flow. Programs like Adobe InDesign, Quark Express and Scribus use frames which act as containers for the content; certain types of frames can be linked so that content flows freely between them. Most word processors and the \LaTeX typesetting program use continuous flow with content which is not part of the continuous flow, such as figures and tables or ‘boxouts,’ being placed in boxes which may alter the flow but do not stop it.

In the early part of the 21st century, it became customary to use HTML tables to lay out HTML pages in a manner similar to the ways in which frames are used to manage the layout in frame based programs. However, because HTML code was increasingly being displayed on screens of widely differing sizes and ratios, from 2011 this approach was deprecated in favour of a continuous flow approach. Like frame based approaches to print media, table based approaches to web pages work best when the dimensions and ratio of the screen are known. Where they are unknown, a table based layout typically fails to use the whole of some screens.

So, from 2011, the `<header>`, `<footer>`, `<nav>` and `<aside>` elements took care of material that was not part of the continuous flow and the `<figure>` element took on the role of the box within the continuous flow. The `<article>` element became the primary container for the continuous flow with the `<section>` and `<main>` elements containing secondary continuous flows.

To avoid clashes between headings inside and outside the continuous flow, or between the headings in primary and secondary continuous flows, the `<hgroup>` element was added while the `<p>`, `
`, ``, ``, `<figure>` and `<blockquote>` elements took on important roles in managing the flow of the content within the continuous flows.

While the continuous flow approach enables content to be laid out in ways which will fit any size or screen ratio, the downside for those used to a frame or table based approach is that they have to learn or relearn skills in handling continuous flow which they never needed when using frames or tables.

