Heath Old Boys Association Website

John R Hudson

7th May 2023

1 Introduction and structure

The HOBA website has been developed using only HTML as set out in HTML Living Standard, CSS as set out in CSS current work and the ARIA attributes for those using assistive technologies. Until 2019 it also used the earlier version of Remy Sharp's HTML enabling script which creates the HTML elements of which earlier browsers are unaware. However, in 2019 a decision was made to move to the use of CSS containers to improve the user experience of those using screenreaders. This required the removal of Remy Sharp's script and thus support for older browsers.

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 banner and crest are 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">
```

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

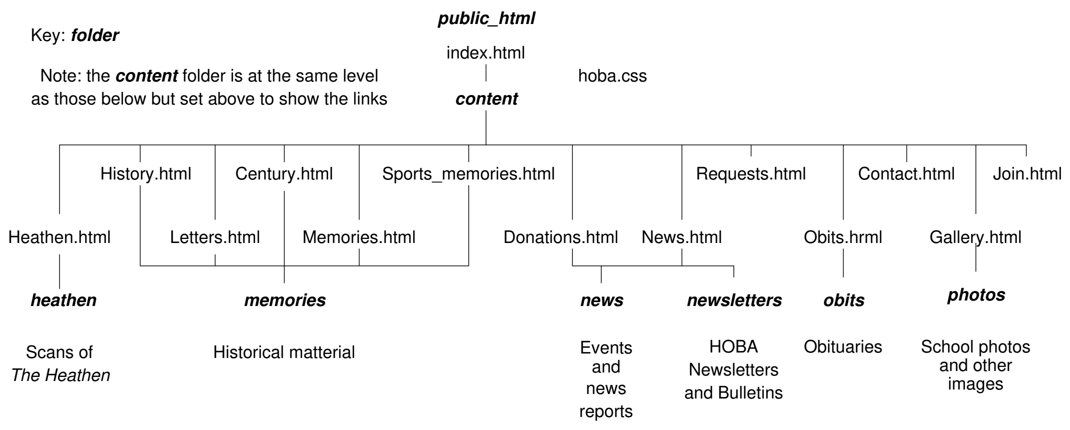

Figure 1: HOBA website structure

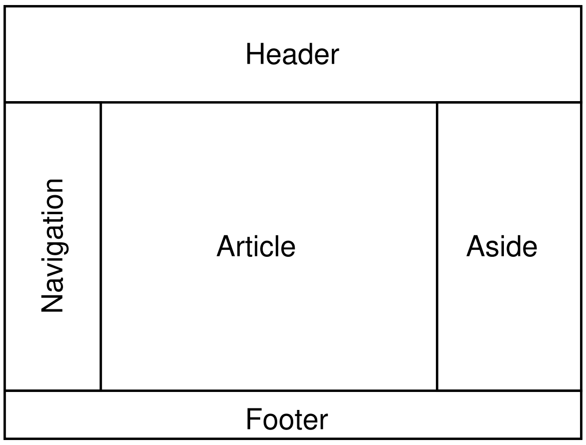

Figure 2: HTML page (thanks to Dave Fisher)

2 <head>

The `<head>` element begins with the standard character set declaration.[1] The title in the next line changes to reflect the top level page and the subject of the page; that is, all obituary pages have the title ‘`Heath Old Boys Association Obituaries`’ followed by the name of the person being remembered. This enables search engines to find references to people and subjects more easily.

This is followed by the base URL which is the same in all pages and the link to the stylesheet. 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="https://heatholdboys.org.uk/">
    <link rel="stylesheet" type="text/css" href="hoba.css"/>

</head>
```

3 <body>

Every `<body>` element contains the five semantic elements: `<header>`, `<nav>`, `<aside>`, `<article>` and `<footer>` (figure 2) with `<article>`, `<nav>` and `<aside>` inside a `<main>` element which provides a CSS flex container. 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>` element, this always reflects

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

the top level page. So 'Obituaries' appears as the second heading on every obituary page. The second `<br>` element ensures that the bottom of the background image displays.

```
<header role="banner">

    <h1>Heath Old Boys Association</h1>
    <br>
    <h2>Welcome</h2>
    <br>

</header>
```

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

5 <article>

The `<article>` element displays as the second element in the `<main>` element which provides a CSS flex container and contains the content of the page. The widths and padding 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 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 and ensures that the people in images always look into the page.

The `<section>` element provides a two column CSS grid container within the `<article>` element. This enables certain lists and two versions of the same hymn to appear in parallel columns.

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.[2] 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.

6 <nav>

The `<nav>` element displays as the first element in the `<main>` element which provides a CSS flex container and contains a menu of links to all the main pages. In second level pages, this menu

[2]For the full list of language tags, see http://www.iana.org/assignments/language-subtag-registry

contains all the links; in top level pages, the link to the showing page is omitted. The font of the `<li>` 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">
    <ul role="list">
    <li class="nav"><a href="index.html" type="text/html"
title="Welcome" accesskey="w W" role="link">Welcome</a></li>
        <li class="nav"><a href="content/News.html" type="text/html"
title="News and events" accesskey="n N" role="link">News and events</a></li>
        <li class="nav"><a href="content/Donations.html" type="text/html"
title="Donations" accesskey="d D" role="link">Donations</a></li>
        <li class="nav"><a href="content/Heathen.html" type="text/html"
title="The Heathen" accesskey="t T" role="link">The Heathen</a></li>
        <li class="nav"><a href="content/History.html" type="text/html"
title="Heath History" accesskey="h H" role="link">Heath History</a></li>
    <li class="nav"><a href="content/Century.html" type="text/html"
title="A Heathen Century" accesskey="a A" role="link">A Heathen Century</a></li>
    <li class="nav"><a href="content/Memories.html" type="text/html"
title="Memories" accesskey="m M" role="link">Memories</a></li>
    <li class="nav"><a href="content/Sports_memories.html" type="text/html"
title="Sports memories" accesskey="s S" role="link">Sports memories</a></li>
    <li class="nav"><a href="content/Gallery.html" type="text/html"
title="Gallery" accesskey="g G" role="link">Gallery</a></li>
    <li class="nav"><a href="content/Letters.html" type="text/html"
title="Letters" accesskey="l L" role="link">Letters from old boys</a></li>
    <li class="nav"><a href="content/Requests.html" type="text/html"
title="Requests" accesskey="r R" role="link">Requests</a></li>
    <li class="nav"><a href="content/Obits.html" type="text/html"
title="Obituaries" accesskey="o O" role="link">Obituaries</a></li>
    <li class="nav"><a href="content/Join.html" type="text/html"
title="Join us" accesskey="j J" role="link">Join us</a></li>
    <li class="nav"><a href="content/Contact.html" type="text/html"
title="Contact us" accesskey="c C" role="link">Contact us</a></li>
    </ul>
</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.

7 <aside>

The `<aside>` element displays as the third element in the `<main>` element which provides a CSS flex container and serves different functions on different pages. For example, on the 'News and events' page it contains links to the *HOBA Bulletins* and earlier editions of the *HOBA Newsletter*. On all other pages than on the 'Welcome' page where is 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" aria-label="Related schools">
    <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 `<li>` 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.

8 <footer>

The `<footer>` element contains a Facebook link and a Creative Commons licence, including a link to a Creative Commons icon. `<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"><img src="graphics/Facebook_25px.png" width = "25"
alt="Facebook" role="img"></a>  page and share it.</p>
<p class="footer">Licensed under the <a rel="license"
href="https://creativecommons.org/licenses/by-nc-sa/4.0/" role="link">
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International</a>
<a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_GB"
role="link"><img alt="Creative Commons BY-NC-SA"
src="http://i.creativecommons.org/l/by-nc-sa/3.0/88x31.png" role="img"></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") no-repeat;
padding-right: 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;}
main {display: flex;}
main>article {order: 2; width: 58%; padding-left: 3em; padding-right: 3em;}
main>nav {order: 1; width: 19%; padding: 1em;}
main>aside {order: 3; width: 19%; padding: 1em;}
section.grid {display: grid;
grid-template-columns: auto auto; justify-content: space-around;}
div.contactgrid {display: grid; grid-template-columns: 160px 1fr;
justify-content: left; }
pre {font: normal 100% monospace;}
*.footer {font: normal 80% sans-serif; text-align: center;}
p.first {font: normal 120% serif;}
p.right {font: italic 100% serif; text-align: right;}
li.nav {font: bold 120% cursive; list-style: none; line-height: 150%}
li.aside {font: bold 100% sans-serif; list-style: none; line-height: 150%}
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;}
sub, sup {font-size: 60%;}
```

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 1 em `padding-right` for the header were added in the light of experience;
- `main {display:  flex;}` introduces the CSS container and declares it as a flex container;
- `main>article`, `main>nav` and `main>aside` specify that these elements are in the container;
- `order` specifies the order in which the elements will appear within the container; this allows the `<article>` element to appear first in the HTML code and therefore first in the content read by a screenreader but second on the page; conversely, the `<nav>` element appears second in the HTML code and therefore second in the content read by a screenreader but first on the page, preventing visually disabled people from having to listen to the menu before they can hear the content of a page;
- only having `padding` at the sides of the `<article>` element while there is `padding` all round the `<nav>` and `<aside>` elements ensures that the text in them does not rise above that in the `<article>` element;
- `section.grid` sets up a CSS grid container with two elements automatically sized and justified and any remaining space evenly shared out to provide a two column layout in a browser on most pages which will appear as a vertical list on an older smartphone;
- `div.contactgrid` sets up a CSS grid container with two elements, the left hand one `160px` wide, to provide separate left-justified entries for each officer and the list of committee members on the 'Contact us' page to provide a two column layout in a browser and a vertical list on an older smartphone;
- `pre` sets up monospacing for the table in 'Best Heath Rugby Team;'
- `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 were originally only intended for the list of committee members on the ‘Contact us’ page which is now handled by `div.contactgrid` but happen to have no adverse impact on other tables elsewhere on the website;
- the `<span>` 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;
- the `<sub>` and `<sup>` element font sizes are set to 60% of the normal text size as is usual for sub- and superscripts.

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;
- editing the `<title>` element to reflect the subject of the page;
- 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="https://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="49vw"` 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>
    <img sizes="49vw" srcset="obits/King_Cross_CC_score_board_400px.png 400w,
        obits/King_Cross_CC_score_board_700px.png 700w,
        obits/King_Cross_CC_score_board_1000px.png 1000w,
        obits/King_Cross_CC_score_board_1300px.png 1300w"
        src="obits/King_Cross_CC_score_board_400px.png"
        alt="The derelict score board ..." role="img" aria-labelledby="kingcross">
    <figcaption id="kingcross">King Cross Cricket Club Score board in 2018
    </figcaption>
</figure>
```

Browsers that support `sizes="49vw"` select one of the sizes in the `srcset=" "` list while the `src=" "` attribute is provided for browsers that do not support the `srcset=" "` attribute.

Originally images uploaded to the website were resized to 72 pixels to the inch; from early 2021 they were resized to 96 pixels to the inch to reflect W3C recommendations.

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 `<li>` 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 the past ten *HOBA Newsletters* and a link to a separate page listing the Bulletins from 1949 to 1970 and the earlier Newsletters. After uploading a new Newsletter to the newsletters folder, copy and edit another list item in the list on the ‘News and events’ page to make the new Newsletter available at the top of the list and move the last one in the list to the top of the list on the ‘Newsletters’ page.

A 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. The advantage of the continuous flow approach is that it enables content to be laid out in ways which will fit any size or screen ratio.

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.

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>` 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>`, `<br>`, `<ul>`, `<ol>`, `<figure>` and `<blockquote>` elements took on important roles in managing the flow of the content within the continuous flows.

From 2017 the `<main>` element took on a new role as the container for CSS defined flex and grid containers which provide a more flexible way of managing the `<article>`, `<nav>` and `<aside>` elements within a web page and a more flexible way of managing the layout of elements within the `<article>` element than the older table based approach. The `<main>` element also has the advantage that using the ARIA attribute `role="main"` in the `<main>` tag offers a non-obtrusive alternative to ‘skip to main content’ links for those using screenreaders.