# HTML

We've all used websites, but how much do we know about how they are put together? Some are very basic, some are very complicated, some are painstakingly produced by hand and some are auto-generated by other pieces of software. What they all have in common is the language used to define what goes on the page: HTML.

**HTML**, or **H**yper-**T**ext **M**arkup **L**anguage, is used to describe the building blocks of every website. We use CSS to modify their appearance and JavaScript to add extra functionality, but without HTML we can't build anything. The content of an HTML file is arranged in different **elements** denoted by keywords enclosed in angle brackets. When an HTML file is opened in a web browser the different elements are formatted correctly in the browser.

Each element has a specific meaning. Some of them will format content in a certain way (eg. headings, lists, links) while some are used to partition the page (eg. headers, footers). Every website has the same fundamental structure made up of some of these elements:

```
<!DOCTYPE html>
<html lang="en">
<!-- The html elements denote the start and end of the document-->

    <head>
    <!-- The head element includes meta-data about the site which is impo
    </head>

    <body>
    <!-- The body element contains the content which will be displayed o
    </body>
</html>
```

In this session we're going to make a (very) simple site to list some blog posts. The first thing we need to do is add a `title` element to the `head` section. This isn't displayed as part of the web page, but it will tell the browser what to display in the tab at the top of the window.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>My Blog</title>
    </head>

<!--... -->
```

# Laying Out Content - Semantic HTML

Onwards to adding some content! We could add some headings, a list of links to navigate around the site, some images and some blocks of text and they would show up just fine in the browser, but we need to think about more than just people viewing our page on a laptop. **Accessibility** is a major concern when designing a website. By ensuring our website is compatible with assistive technologies (such as screen readers), using fonts and colours which consider conditions such as colour-blindness and labelling input fields appropriately we make our site accessible to as many users as possible.

A fundamental part of this is using **semantic** HTML elements to clearly indicate where parts of our site begin and end. We'll add three elements to show the basic structure:

```
<body>
    <header>
    </header>
    <main>
    </main>
    <footer>
    </footer>
</body>
```

# The Header

In our `header` we will add the content which will be displayed at the top of the screen, in this case a title and some navigation. We'll use an `h1` element for the title. Heading elements all start with `h` and are followed by a number, with lower numbers meaning bigger text. We should only ever have a single `h1` element on a page which will represent the title.

```
<header>
    <h1>My Blog Site</h1>
</header>
```

We also want some navigation which we will display as a list of links inside a `nav` element.

By using `nav` we show that this is for navigation rather than a simple list. The `ul` inside adds an unordered list to the page with each `li` element adding a bullet point to the list ( `ol` gives an ordered list where each `li` is a numbered entry)

```
<header>
    <h1>My Blog Site</h1>
    <nav>
        <ul>
            <li>Home</li>
            <li>About</li>
            <li>Contact</li>
        </ul>
    </nav>
</header>
```

Finally we want to make these navigation tools clickable. We'll use an `a` element to do so, including an `href` property which will tell the browser where to take us if the link is clicked. We've only got one page at the moment, so for now we'll stick a link to a different site in as a placeholder.

```
<header>
    <h1>My Blog Site</h1>
    <nav>
        <ul>
            <li><a href="www.google.com">Home</a></li>
            <li><a href="www.google.com">About</a></li>
            <li><a href="www.google.com">Contact</a></li>
        </ul>
    </nav>
</header>
```

# The Posts

The `main` element we added before will contain the majority of our content. To help organise it we'll use `section` elements, which don't display anything on the page but which do help with assistive tech. We'll have two, one for some introductory text and another for the posts. Within the first block we'll have another heading, a `p` element to display some text and an image.

```
<main>
    <section>
        <h2>Welcome to my blog!</h2>
        <p>Here's a nice picture:</p>
        <img src="https://placebear.com/300/300" alt="two bears in a rive
    </section>
    <section>
        <!-- Posts will be added here -->
    </section>
</main>
```

We're using an `h2` element for this heading. We're a level down from the main heading, so we use the next size of heading element. Also note the `alt` property for the `img` tag. This is the alternate text, which will be read by screen-readers or displayed if the image fails to load.

Our second `section` has been sub-divided into another `header` and `main`, which is in turn divided into more `sections` to represent the blog posts. Each section has progressively smaller headings and some placeholder text.

```
<main>
    <section>
        <!-- Intro -->
    </section>
    <section>
        <header>
        <h2>Posts</h2>
       </header>
       <main>
        <section>
             <h3>Post 1</h3>
             <h4>Date: Yesterday</h4>
             <p>Text goes here</p>
          </section>
          <section>
             <h3>Post 2</h3>
             <h4>Date: Last Weekend</h4>
             <p>Text goes here</p>
          </section>
          <section>
             <h3>Post 3</h3>
             <h4>Date: Las Month</h4>
             <p>Text goes here</p>
          </section>
        </main>
    </section>
</main>
```

# The Footer

Many sites include a footer, which sits at the bottom of the page and often includes things such as contact details, links to relevant information and other details about the site. We'll add a fairly simple copyright notice, but we could easily put another list here or some links.

We want to use a special symbol here, which will require us to use **unicode**. Every character in the alphabet, numbers, punctuation, special symbols, even emojis have a designated unicode representation, which means any browser can recognise what should be printed. It's up to the browser (or other application) to decide precisely how to render it, which is why emojis appear differently on an iPhone compared to an Android device.

```
<footer>
    <p>&#169; Your Name 2022</p>
</footer>
```

## Next steps

We have a website! It may not look like much, but we have the basics in place. In the next part we'll make things look a bit more appealing by adding some CSS.