# Introduction

In this lesson we're going to learn how to use tags to structure content. To prepare for the lesson open up the *"structural-begin.html"* file.

Before HTML 5 we only had the `div` tags for structuring content but currently there are many more options to choose from which improve the semantics of the code.

# Div tags

So our task is to mark up sections of the document with the most appropriate tags.

The first and most common tag is the generic block tag `div` which stands for **division**. When we want to define blocks of content or data in the page, we can use the `div` tag, however, it does not add any particular meaning to what it's marking up. HTML5 introduces many new tags that help us with that.

It is useful however to have a tag containing the whole website so that we have hooks for CSS as well. So let's just wrap the whole content with the `div` tag like this:

```
<body>
    <div>
        <img src="logo.png" alt="Company Name">
        […]
        <small>2014 &copy; Company Inc. All Rights Reserved.</small>
    </div>
</body>
```

The `div` tag is useful to define generic blocks. We can use attributes like `class` and `title` to assign it more meaning if necessary.

# Header and nav tags

We have the logo at the top of the page. The logo of the company is usually a part of the header of the web page, so it's fitting to use the `header` tag here:

```
<header>
    <img src="logo.png" alt="Company Name">
</header>
```

The `header` tag, as the name implies, represents the header of a section, or of a whole document. It may be used together with headings and include information like publication date when it is used as the header of an article.

Next, we have the menu, but this list itself does not mean that it represents navigation. That's why HTML5 introduces the `nav` tag:

```
<nav>
    <ul>
        <li>
            <a href="#">Home</a>
        </li>
        <li>
            <a href="#">Products</a>
        </li>
        <li>
            <a href="#">Services</a>
        </li>
        <li>
            <a href="#">Contact Us</a>
        </li>
```

```
    </ul>
</nav>
```

The `nav` tag can be used to mark up primary page's navigation. However not all navigational blocks should use this tag. For example, a complementary menu inside the footer of a website normally would not need a `nav` tag.

Next there is the entire website's content. It will depend on the nature of the content but for now we can wrap this entire content with a `div` tag as well:

```
<div>
    <h1>Content Title</h1>
    […]
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
    excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
</div>
```

# The main tag

As another option we may use the new `main` tag:

```
<main>
    <h1>Content Title</h1>
    […]
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
    excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
</main>
```

This tag should occur only once per page and be used only for marking up the main content area. It has been recently added to the draft HTML specification, so we should see browser support increasing over time.

# The article tag

The next piece of content can be an example for using another new tag `article`:

```
<article>
    <h1>Content Title</h1>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
    excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
</article>
<h1>Another website section</h1>
[…]
```

`article` represents an independent part of the content of a page. Imagine that you can take this content between `article` tags, remove it from the site and paste it into another document or website - it still would have made sense. Examples include blog posts, magazine articles and forum posts. `article` can also contain headers and footers:

```
<article>
    <header>
        <h1>Content Title</h1>
        <p>Published: <time datetime="2015-01-20">01/20/2015</time></p>
    </header>
    […]
</article>
```

The footer could contain information about the code and it's author.

# Figure, figcaption and aside tags

Let's now take a look at the picture in the document. This is something that is related to the content, so we could use the new `figure` tag. The `figure` tag represents a unit of a content, optionally with a caption, that is self-contained. This is typically referenced as a single unit from the main flow of the document that can be moved away from the main flow of the document without affecting the document's meaning.

The HTML5 specification also introduces a `figcaption` tag to mark up the caption of the figure.

```html
<figure>
    <img src="picture.jpg" alt="Tree with pink blossoms">
    <figcaption>
        This is a tree that is widely found around my city.
    </figcaption>
</figure>
```

Notice that the `alt` text in the image is different than the `figcaption`: the `alt` text should describe what's on the image while in the caption you can explain, for example, the relation of the picture to the document.

Now imagine we need to mark up a sidebar as content that is related to the main article, but not is not necessary for its understanding. This kind of content can be marked up with the `aside` tag:

```html
<aside>
    <h2>Side content (related to main content but not necessary for its understanding)</h2>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
</aside>
```

If the `aside` tag is put inside an `article` tag, its content must be related to that article. If an `aside` tag is put outside of an `article` tag, the contents must be related to the website in general. That does not mean however that every website's sidebar should be marked up with an `aside`.

# Section and footer tags

After the content that we've wrapped with the `article` tags there is another section. If this is our basic website section that does not constitute independent content like article, we can use the generic `section` tag:

```html
<section>
    <h1>Another website section</h1>
    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ad, minima, ipsum, harum vero molestiae libero aliquam natus
excepturi nam dolore laudantium adipisci perspiciatis in repudiandae officiis provident veritatis odio rem.
    </p>
</section>
```

The `section` tag represents a thematic group of content and it should be identified by one of the heading tags: `h1` through `h6`. `h1` is used here because it is the start of a new section, unrelated to the article above.
Finally, we have the footer. This is a perfect case for the `footer` tag:

```html
<footer>
    <p>
        <a href="mailto:email@example.com">email@example.com</a>
        <br>
        100 Example Street
        <br>
        City - Country
    </p>

    <small>2014 &copy; Company Inc. All Rights Reserved.</small>
</footer>
```

The `footer` tag contains information about a section or document. It can be used as the footer of a website, or as a footer of another section or article.

Now you may see the final results in the browser. You will notice that almost nothing has changed on the page. The tags we have seen here are mainly structural tags: they help us to group content semantically, so we have better code for browsers and search engines.