Lesson 06 - HTML Layout

Now that we have explored a few different html tags that help us produce content, let's take a look at how we can help with the document flow using only HTML.

Container Elements

There are a set of HTML elements referred to as `containers` that, on their own, do not produce any visible effect. Their purpose is to rather act as a `container` or `parent` to a series of child elements to group them together and assist with document flow.

`<div>`

From MDN:

> The HTML Content Division element (`<div>`) is the generic container for flow content. It has no effect on the content or layout until styled using CSS.

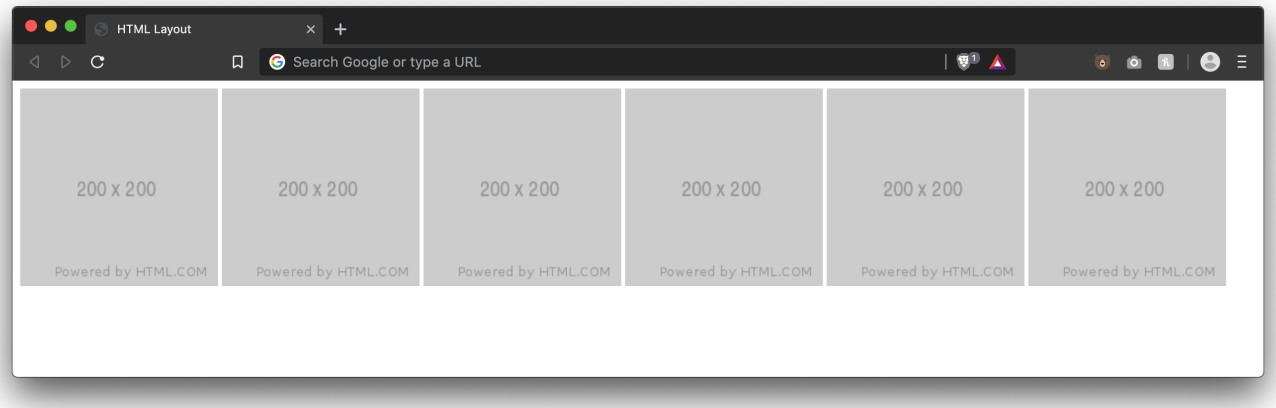
> As a "pure" container, the `<div>` element does not inherently represent anything. Instead, it's used to group content so it can be easily styled using the class or id attributes, marking a section of a document as being written in a different language (using the lang attribute), and so on.

`<div>` elements by default follow the `block` layout, thus shifting everything after it underneath it. We can use a `<div>` to group a series of elements that we want together.

Let's take a real world example. Say you have 6 images that you want to display in a 2x3 grid. By default, images display inline. So if we were to take this:

```
<img src="example-1.jpg" />
<img src="example-2.jpg" />
<img src="example-3.jpg" />
<img src="example-4.jpg" />
<img src="example-5.jpg" />
<img src="example-6.jpg" />
```

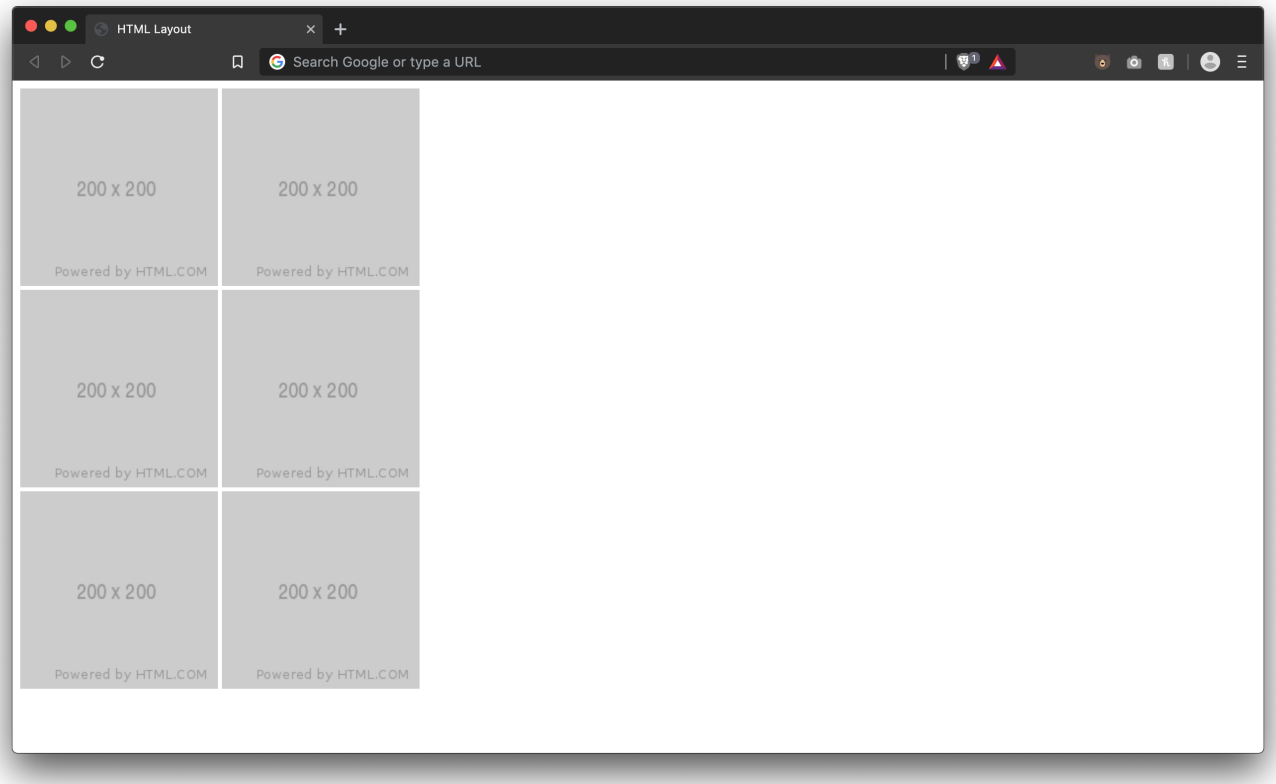
It would show all of the images one next to eachother.



Now that we have our new gained knowledge of `<div>` elements though, we can group them up to the grid that we want.

```
<div>
  <img src="example-1.jpg" />
  <img src="example-2.jpg" />
</div>
<div>
  <img src="example-3.jpg" />
  <img src="example-4.jpg" />
</div>
<div>
  <img src="example-5.jpg" />
  <img src="example-6.jpg" />
</div>
```

Which will produce the result we're after.



Semantic alternatives to `<div>`

When needing to group items together, a `<div>` is always the easiest option. However, there are other tags that behave exactly like the `<div>` but give better context as to what content is going inside, or what purpose the content serves. To your average user, they would never notice a difference. But to web-crawlers (like google) and people that require a screen reader it can make a huge difference in their overall experience.

> The `<div>` element should be used only when no other semantic element (such as `<article>` or `<nav>`) is appropriate.

(descriptions provided by MDN)

- `<main>` - represents the dominant content of the `<body>` of a document. The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.
- `<section>` - represents a standalone section — which doesn't have a more specific semantic element to represent it — contained within an HTML document. Typically, but not always, sections have a heading.
- `<header>` - represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also a logo, a search form, an author name, and other elements
- `<footer>` - represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.
- `<aside>` - represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

- `<nav>` - represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.
- `<article>` - represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). Examples include: a forum post, a magazine or newspaper article, or a blog entry.

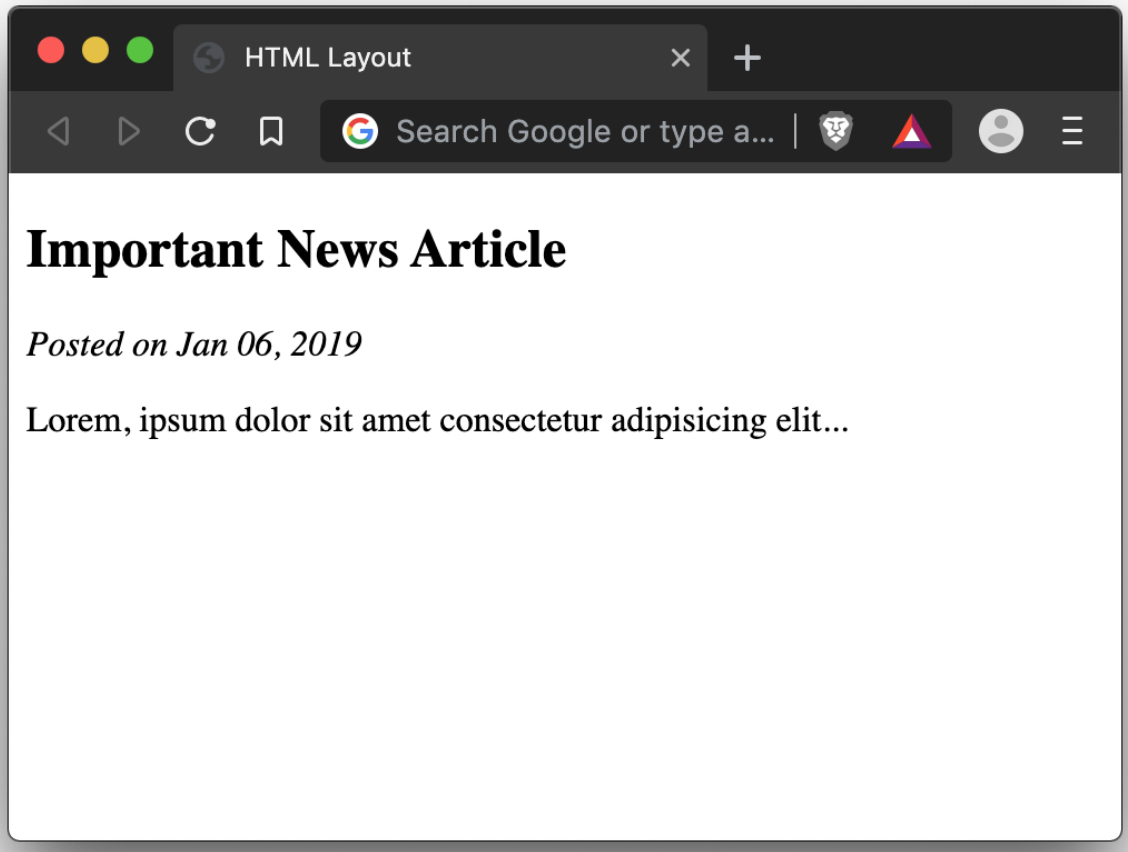
`<span>`

`<span>`, like `<div>`, is a container element that provides no default features other than to contain content. However, `<span>` is an `inline` container which makes it ideal for wrapping other `inline` elements like free-floating text.

We will cover more uses of `<span>` elements when we get to CSS, but for now let's take a look at a simple example.

Let's say you have a blog article that shows the date it was posted in a nice and short format.

```
<article>
  <h1>Important News Article</h1>
  <p><em>Posted on Jan 06, 2019</em></p>
  <p>
    Lorem, ipsum dolor sit amet consectetur adipisicing elit...
  </p>
</article>
```



But what if we want to know the exact time that it was posted? It wouldn't look very good in our layout to have that level of detail, but a common pattern in website development is to show the full timestamp of the date when you hover over top of it.

We can accomplish this by putting a `title` attribute on the element that, when hovered, will show this text.

> The `title` attribute is a global attribute that can be applied to any HTML element. It contains text representing advisory information related to the element it belongs to.

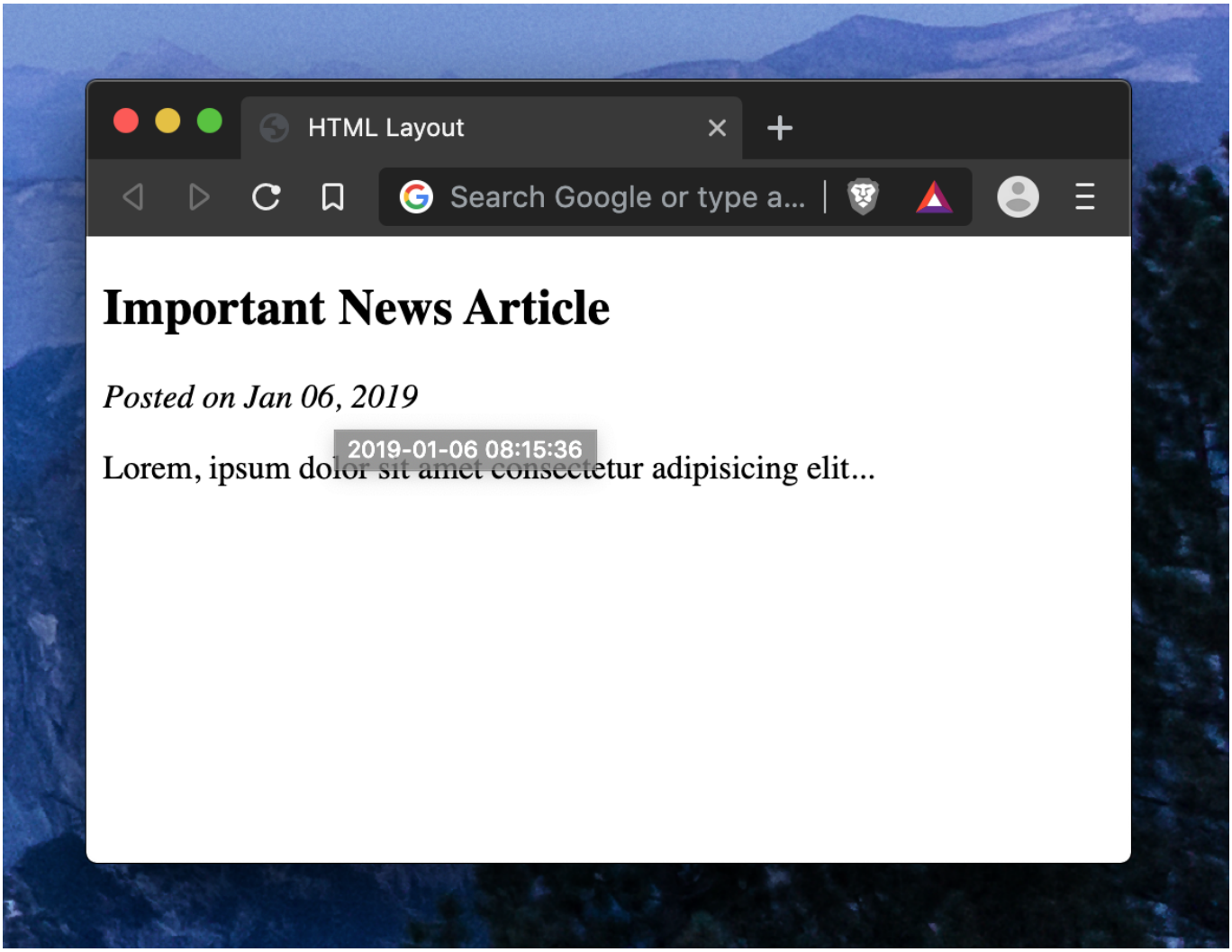
> More information about `title` on MDN

Since we want this only on the date and not the whole "Posted on..." text, we would need to somehow add a tag around only the date.

`<span>` to the rescue!

```
<article>
  <h1>Important News Article</h1>
  <p>
    <em>Posted on <span title="2019-01-06 08:15:36">Jan 06, 2019</span></em>
  </p>
  <p>
    Lorem, ipsum dolor sit amet consectetur adipisicing elit...
```

```
  </p>
</article>
```



Notice how when we added the span, the content does not shift!

If we were replace that `<span>` with a `<div>`, you can see the difference:

```
<article>
  <h1>Important News Article</h1>
  <p>
-    <em>Posted on <span title="2019-01-06 08:15:36">Jan 06, 2019</span></em>
+    <em>Posted on <div title="2019-01-06 08:15:36">Jan 06, 2019</div></em>
  </p>
  <p>
    Lorem, ipsum dolor sit amet consectetur adipisicing elit...
  </p>
</article>
```

More layout elements

There's a few more element's we can use to manipulate a page's layout. These are not containers, but they affect the flow of the document.

`<br />`

Produces a line break in text (carriage-return). It is useful for writing a poem or an address, where the division of lines is significant.

If you haven't noticed already, doing this inside of a `<p>` tag:

```
<p>
  I am on multiple lines but i appear as one
</p>
```

Will show up all in the same line. That's because in HTML line breaks do not matter and whitespace is only counted for one character (e.g. if you have `text spaced out` will be rendered as `text spaced out`).

To get around this, we can use the `<br />` element inside of paragraph tags.

```
<p>
  O’er all the hilltops<br />
  Is quiet now,<br />
  In all the treetops<br />
  Hearest thou<br />
  Hardly a breath;<br />
  The birds are asleep in the trees:<br />
  Wait, soon like these<br />
  Thou too shalt rest.
</p>
```

The above will insert a break at the end of each line.

`<hr />`

The `horiztonal rule` element. Represents a thematic break between paragraph-level elements: for example, a change of scene in a story, or a shift of topic within a section.

```
<p>1: The first rule of Fight Club is: You do not talk about Fight Club.</p>

<hr />

<p>2: The second rule of Fight Club is: Always bring cupcakes.</p>
```

Most browsers will render this as a solid grey line.

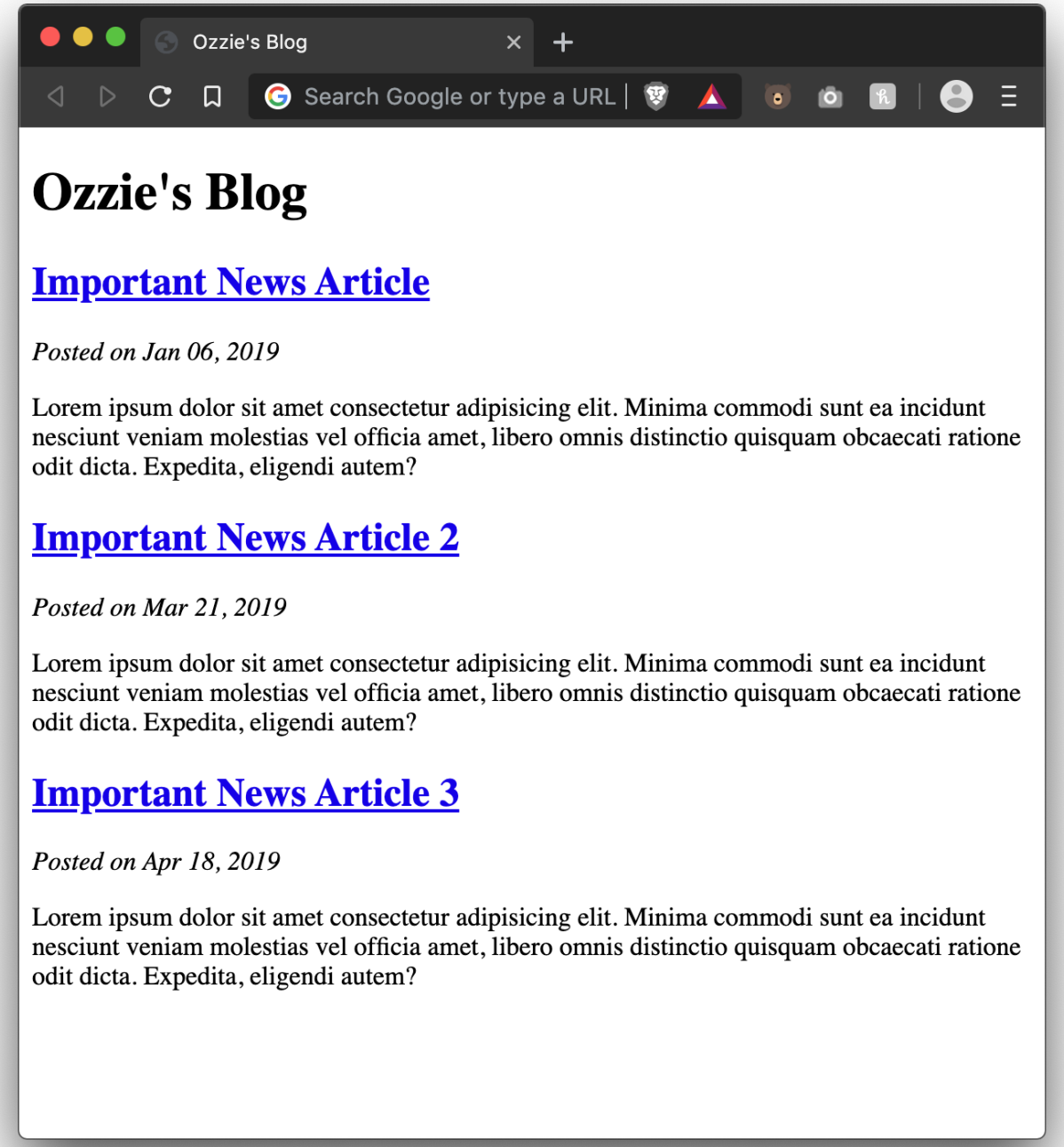
Exercise Instructions

We're going to create a small blog.

- Create a file, `index.html`
- Set the document title to `<Your Name's> Blog`
- Create a `<header>` element
- Add heading `h1` to the header, `<Your Name's> Blog`
- Create a `<main>` container underneath the `<header>`, not inside
- Create 3 `<article>` elements, each with:
    - A level 2 heading for the title, named whatever you want
    - A date underneath the title, formatted as e.g. `Jan 06, 2019`, italicized (without CSS), in a `<p>` tag. Hovering the date should reveal a more specific time
    - A single paragraph underneath the date, acting as an article summary or preview
        - Can be generated from lorem ipsum
    - Clicking on the article title should take you to the page for the post (created in next step)
- Create a `posts` folder
- Create 3 `.html` files in here to represent your blog articles titles
    - e.g. `post-1.html`, `post-2.html`, `post-3.html`
    - Make sure each article is linked to from the root `index.html` file

- Each post file should have
  - Its document title set to the title you gave the article in `index.html`
  - The same page `<header>` from `index.html`, but the heading should link back to `index.html`
  - A `<main>` container that contains
    - The title of the post (not a link)
    - The date it was posted
    - 5 paragraphs of lorem ipsum
    - Use a `<br />` to separate two of the paragraphs
    - Use `<hr />` after the last paragraph
    - Insert a `<footer>` with a `<p>` tag containing `- Your Name`

Exercise Result


Ozzie's Blog
Search Google or type a URL
Ozzie's Blog
Important News Article
Posted on Jan 06, 2019
Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?
Important News Article 2
Posted on Mar 21, 2019
Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?
Important News Article 3
Posted on Apr 18, 2019
Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?


Important News Article 2
Search Google or type a URL
Ozzie's Blog
Important News Article 2
Posted on Mar 21, 2019
Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt

nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?
Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?

Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima commodi sunt ea incidunt nesciunt veniam molestias vel officia amet, libero omnis distinctio quisquam obcaecati ratione odit dicta. Expedita, eligendi autem?

- Ozzie Neher

Further Reading

- MDN div
- MDN span
- MDN br
- MDN hr