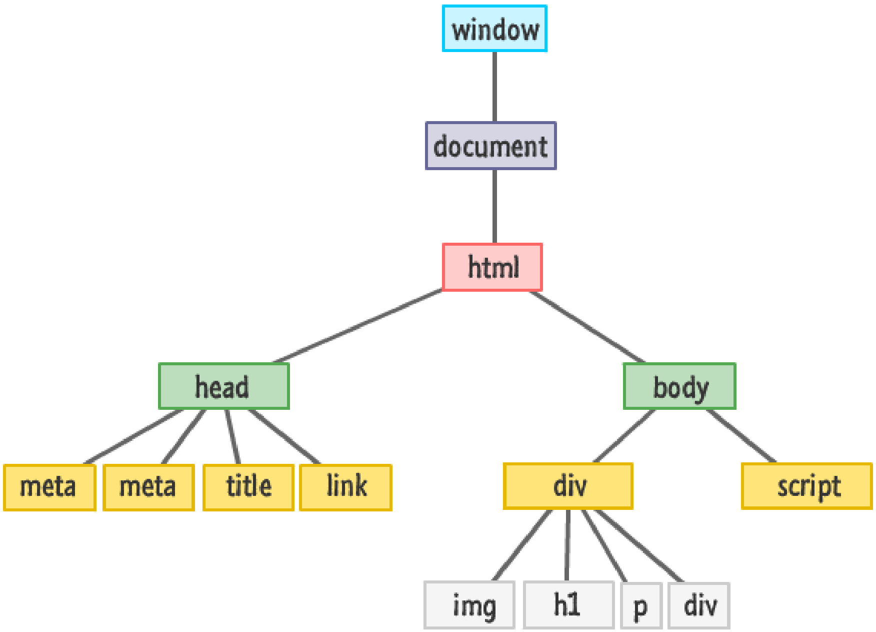
The HTML DOM (Document Object Model)

The DOM standard

Exploring the DOM of a document

When a web page is loaded, the browser creates a **Document Object Model** of the page. The **HTML DOM model** is constructed as a **tree of Objects.**

'Elements' are the pieces themselves, i.e., a paragraph, a header, and even the body are elements. Most elements can contain other elements - for example, the body element would contain header elements, paragraph elements, in fact pretty much all of the visible elements of the Document Object Model (developers call it the "DOM").



HTML code example:

```
<!DOCTYPE html>
<html lang = "en">
    <head>
        <title>Your first HTML page</title>
        <meta charset = "utf-8">
    </head>
    <body>
        <h1>My home page</h1>
        <p>Hi! My name is Ezequiel, Welcome to my Home Page!
        </p>
    </body>
</html>
```

The DOM standard

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

Exploring the DOM of a document

When a user clicks on a link or enters a URL in the address of your Web browser, it downloads the page's HTML text and builds up a model of the document's structure called the DOM (Document Object Model). This model is used to render the HTML page on the screen.

The DOM is a standard that describes how a document must be manipulated. It defines a "language- and platform neutral interface". So, **every browser offers the same JavaScript DOM API**.

The **DOM API** is a programming interface the JavaScript programmer can use to modify the HTML content or the CSS style of HTML elements on the fly.

The DOM API provides the document object as a structured object, a group of nodes represented as a tree.

The document object also exposes a large set of **methods to access and manipulate the structured document.** Through the DOM, look for nodes (html elements that compose the page), move nodes, delete nodes, modify nodes (attributes, content), and also handle their associated events.

In JavaScript, the DOM is accessible through the property **document** of the global object **window**. We rarely manipulate the window object directly as it is implicit: `window.document` is the same as `document`.

So by using this object, we can access and manipulate our page from JavaScript as a structured document.