Introduction to the DOM

This chapter will help you discover how a web page is shown by a browser.

WE'LL COVER THE FOLLOWING

You already know that a web page is a document that contains text and tags such as headings, paragraphs, links, etc. This happens in a language called HTML. Let's take this simple web page as an example. Feel free to add your own information!

Output

HTML

```
<!doctype html>
<html>

<head>
    <meta charset="utf-8">
    <title>My web page</title>
</head>

<body>
    <h1>My web page</h1>
    <p>Hello! My name's Baptiste.</p>
    <p>I live in the great city of <a href="https://en.wikipedia.org/wiki/Bordeaux">Bordeaux<
</body>

</html>
```

To create this result, the browser first takes the HTML code and builds a representation of its structure. It then displays this structure in the browser.

The browser also offers *programmatic access* to its structured representation

of a displayed web page. Using this interface, you can dynamically update the

page by adding or removing elements, changing styles, etc. This is how you create *interactive* web pages.

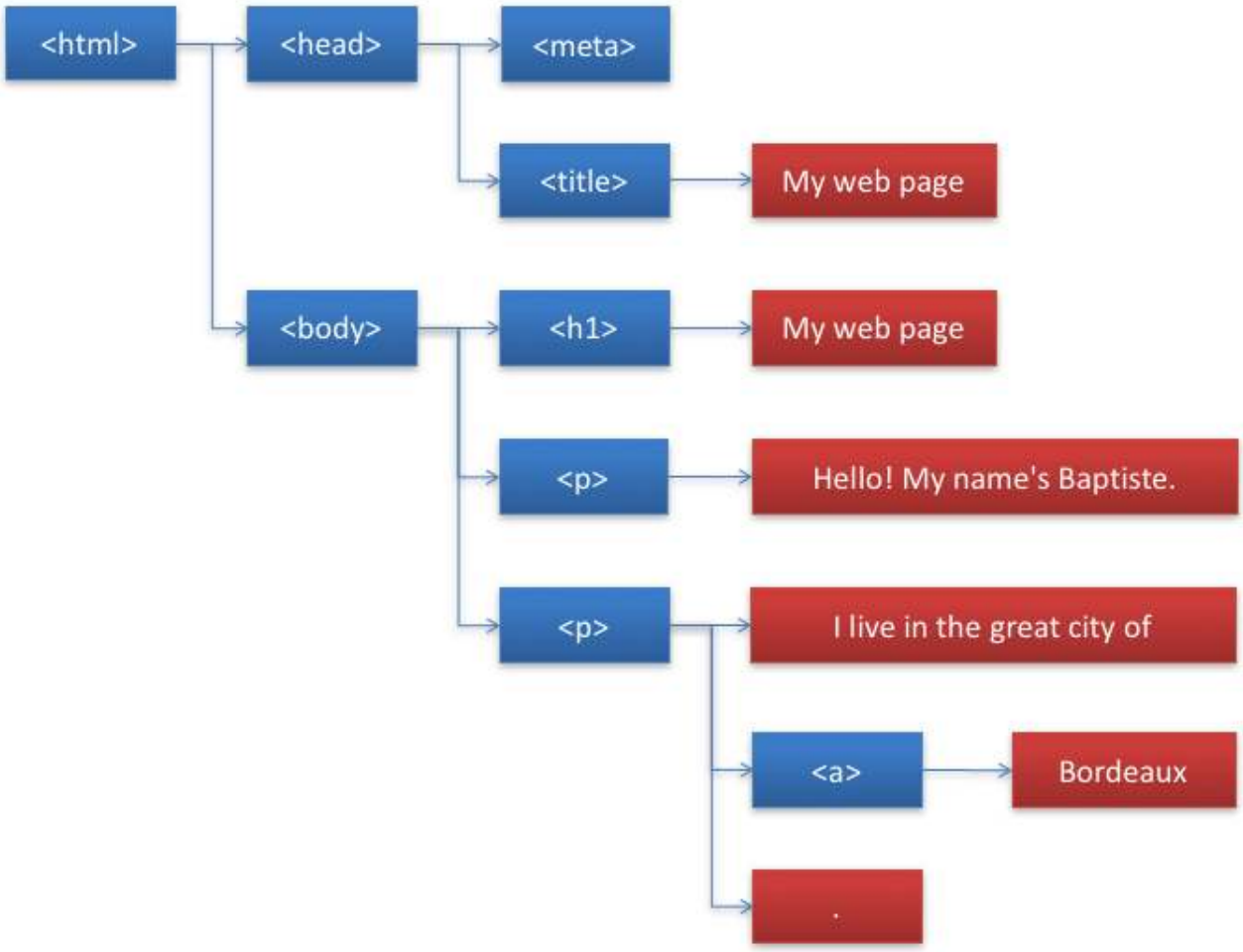
The structured representation of a web page is called **DOM**, short for Document Object Model. The DOM defines the structure of a page and a way to interact with it. This means it's a programming interface, or **API** (Application Programming Interface). JavaScript is the language of choice for interacting with the DOM.

> At the dawn of the Web, each browser was using its own DOM, giving headaches to JavaScript developers trying to code web pages. These hard times are over. Through a World Wide Web Consortium (W3C) effort, the first version of a unified DOM was created in 1998. Nowadays, all recent browsers use a standardized DOM.

Web Page Structure

A web page is a set of nested tags. You can represent a web page in a hierarchical form called a *tree*. The `<html>` element sets up your document as HTML and contains two sub-elements, `<head>` and `<body>`, which themselves contain several sub-elements.

Here is the tree corresponding to our example HTML page.



Each entity in the tree is called a *node*. There are two types of nodes:

Those (in blue here) that correspond to HTML tags like `<body>` or `<p>`. These nodes are called element nodes and they can have subnodes, called *child nodes* or children. Those (in red) that match the textual content of the page. These nodes are called *text nodes* and do not have children.