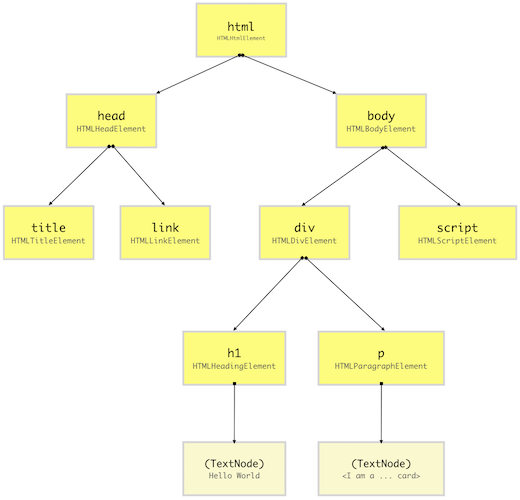
Parsing html

**Parsing** is the process of reading HTML content and constructing a DOM tree from it. Hence the process is also called **DOM parsing** and the program that does that is called the **DOM parser**.

Most browsers provide the DOMParser Web API to construct a DOM tree from the HTML code. An instance of DOMParser class represents a DOM parser and using the parseFromString prototype method, we can parse raw HTML text (code) into a DOM tree.

DOM tree



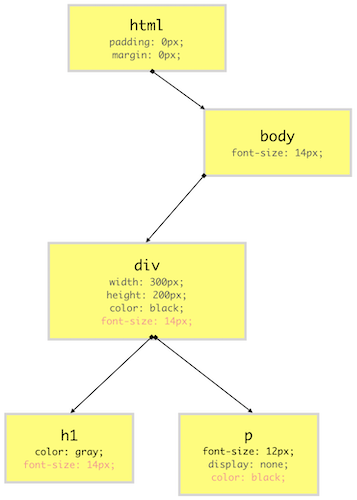
CSS

• The DOM and CSSOM trees are combined to form the render tree.

• Render tree contains only the nodes required to render the page.

• Layout computes the exact position and size of each object.

• The last step is paint, which takes in the final render tree and renders the pixels to the screen.



Layout

The first browser creates the layout of each individual Render-Tree node. The layout consists of the **size of each node** in pixels and where (position) it will be printed on the screen. This process is called **layout** since the browser is calculating the layout information of each node.

This process is also called **reflow** or **browser reflow** and it can also occur when you **scroll**, **resize** the window or **manipulate DOM** elements.

Paint

Until now we have a list of geometries that need to be printed on the screen. Since **elements** (or a sub-tree) in the Render-Tree can overlap each other and they can have CSS properties that make them frequently change the look, position, or geometry (such as animations), the browser creates a **layer** for it.Now that we have layers, we can combine them and **draw** them on the screen. Inside each layer, the browser fills the individual pixels for whatever visible property the element has such as border, background color, shadow, text, etc. This process is also called as **rasterization**. To increase performance, the browser may use different **threads** to perform rasterization.