What is a DOM?

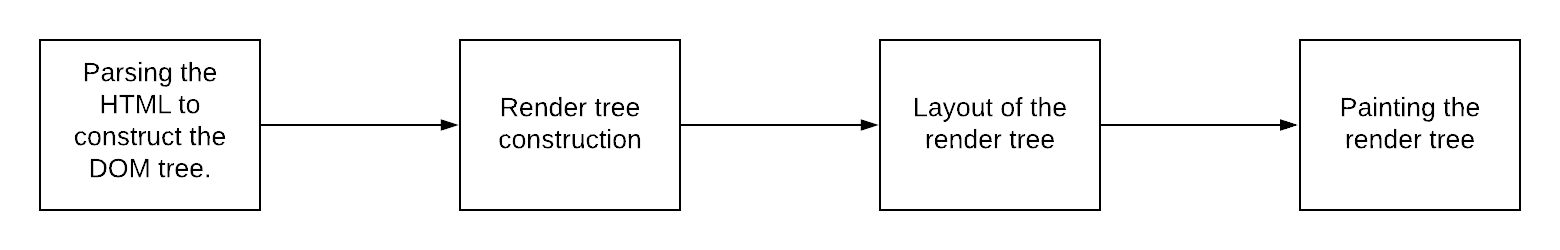
DOM stands for Document Object Modeling

* Standardized API that represents an HTML or XML document as a tree structure.
* Each node on the tree is an object representing an HTML element.
* Connects JavaScript to HTML
* DOM nodes can be manipulated directly



Figure: Example of a DOM hierarchy tree

Web Browser workflow



* The DOM tree itself is constructed by the rendering engine (Webkit) from parsing the HTML document
* Parses CSS and applied CSS to the HTML creating a render tree
* Render tree is painted on to the browser

What is a Virtual DOM?

* Lightweight representation of the DOM
* Exists in memory; is never actually rendered
* It is a tree data structure of plain JavaScript objects

What problems does the virtual DOM solves?

* Instead of rendering all these changes to the real DOM, we apply them first to the virtual DOM
* Virtual DOM does not get rendered in real life (IRL) so changes are cheap
* Batch changes together for efficiency

How do you render the virtual DOM tree?

* JSX tells template compiler how to build an in-memory DOM tree
* The render function creates virtual DOM tree of React elements
* The in-memory DOM is rebuilt completely when the application is updated
* Two virtual DOM always exists in-memory. React compares the changes and paints the tree only once.

Two assumptions React is based on:

1. Two elements of different types will produce different trees

* React will not attempt to differ them, but replace them completely
* Breath-First Traversal
* If any React element changes then React will first mark them dirty, then re-render the entire subtree

1. Child elements may be hinted as stable across renders with a key. To optimize performance, keys should be stable unique and predictable.

Things to note:

Find the minimum modifications required is an O(n^3) complexity problem. However, React uses a diffing/reconciliation algorithm to accomplish the task in O(n). React’s diffing algorithm uses the above two assumptions.