

# How Does a Browser Render a Page?

---

(🔗) [freecodecamp.org/learn/front-end-development-libraries-v9/lecture-understanding-performance-in-web-applications/how-does-a-browser-render-a-page](https://freecodecamp.org/learn/front-end-development-libraries-v9/lecture-understanding-performance-in-web-applications/how-does-a-browser-render-a-page)



Have you ever wondered what happens behind the scenes when you enter a URL and press enter? How does the browser transform code into the interactive pages we see?

Understanding how a browser renders a web page is essential for developers aiming to optimize performance and enhance user experience.

In this lesson, we'll demystify the rendering process, breaking it down into clear, digestible steps.

First the browser parses the HTML and builds the DOM.

The process begins with the browser fetching the HTML content of the page. It parses this HTML to construct the Document Object Model, or DOM — a tree-like structure representing the page's content.

Consider this simple HTML snippet:

```
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>Welcome to our website.</p>
  </body>
</html>
```

The browser parses this to create a DOM tree with `html` as the root, containing `body`, which in turn contains `h1` and `p` elements.

Next, the browser processes the CSS, constructing the CSS Object Model, or CSSOM. This is another tree structure that dictates how elements should be styled.

Given this CSS:

```
h1 {  
    color: blue;  
}  
p {  
    font-size: 16px;  
}
```

The CSS Object Model specifies that `h1` elements are blue and `p` elements have a font size of 16 pixels.

The browser then combines the DOM and CSS Object Model to form the render tree. This tree includes only the nodes needed to render the page, each paired with its corresponding styles.

For our example, the render tree consists of the `h1` and `p` elements, styled appropriately.

With the render tree in hand, the browser calculates the exact position and size of each element in a process called layout or reflow. This ensures that elements appear in the correct location and dimensions on the screen.

Finally, the browser paints the pixels to the screen, rendering each element based on the calculated styles and layout. In complex pages, this might involve multiple layers that are composited together to form the final visual output.

In conclusion, from parsing HTML and CSS to painting pixels on the screen, the browser's rendering process is a sophisticated sequence of steps.

## Questions

---

### What does the browser do first?

The browser processes the CSS, constructing the CSSOM.

The browser parses the HTML and builds the DOM.

Correct!

The browser combines the DOM and CSSOM to form the render tree.

The browser paints the pixels to the screen.

## What does the CSSOM stand for?

The Custom CSS System Object Model.

The CSS Operating Model.

The Custom CSS System Operating Model.

The CSS Object Model.

Correct!

## What does constructing the CSSOM do?

Creates a DOM tree with `html` as the root.

Dictates how elements should be styled.

Correct!

Fetches the HTML content of the page.

Paints the pixels to the screen.

Navigated to How Does a Browser Render a Page?