Critical resources

Critical bytes

browser rendering pipeline.

Style

HTML -> DOM

=> Render tree

CSS -> CSSOM

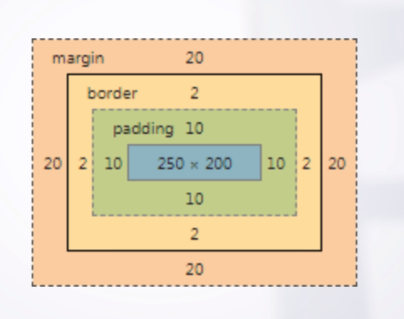
Layout

it's to calculate the position and dimension of each element.

All of the relative measurements like percentages ems, or rems are converted into absolute units

- the pixels.

The output of this layout processes is box model.



Paint

Each element will be drawn on the screen one by one.

compositing layers

Your application can have multiple layers

The browser can paint each of the layers individually.

The process of handling these layers is called a compositing layers.

1. Style

HTML parsing algorithm

it has 4 steps: conversion, tokenizing, lexing and building the DOM tree.

Conversion

Bytes => Characters

Tokenizing

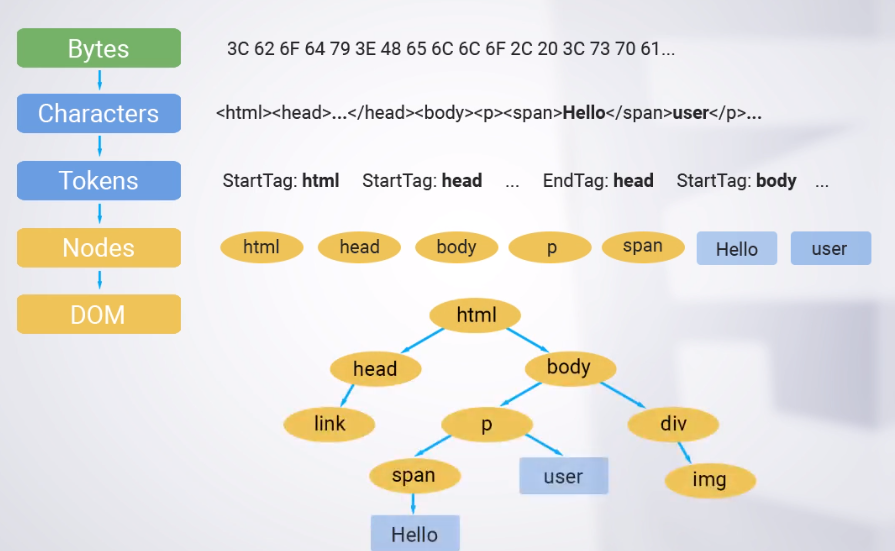
=> Tokens

Lexing

=> Nodes

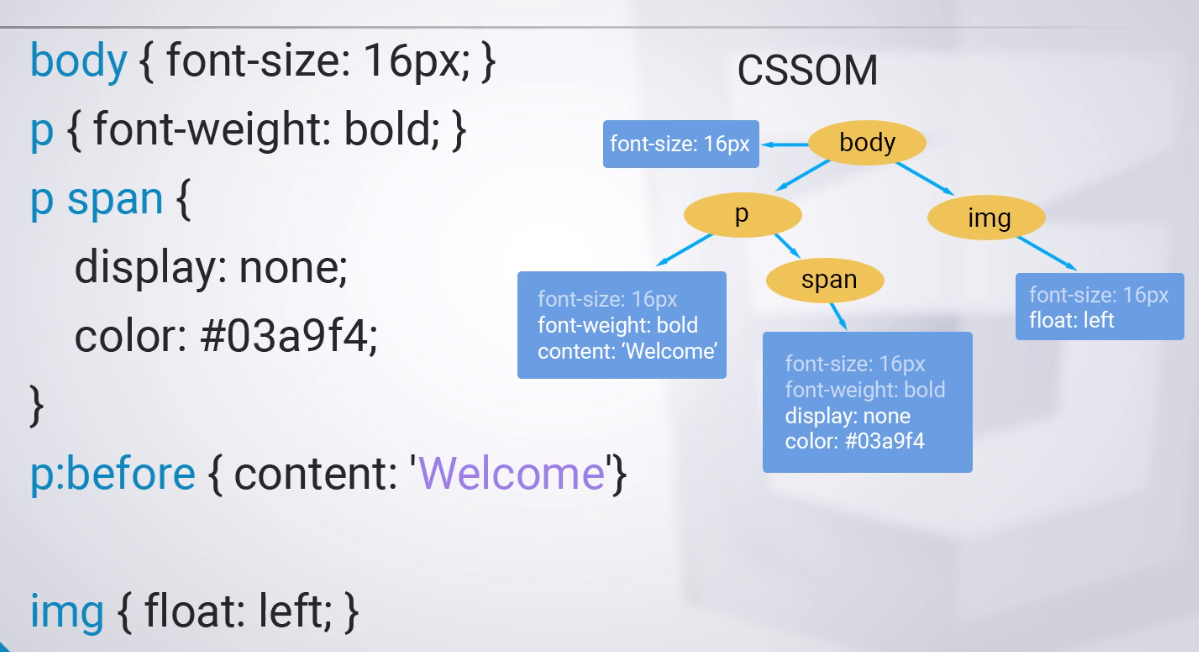
DOM tree

=> DOM



Parse CSS

Bytes => Characters => Tokens => Nodes => CSSOM (CSS Object Model)

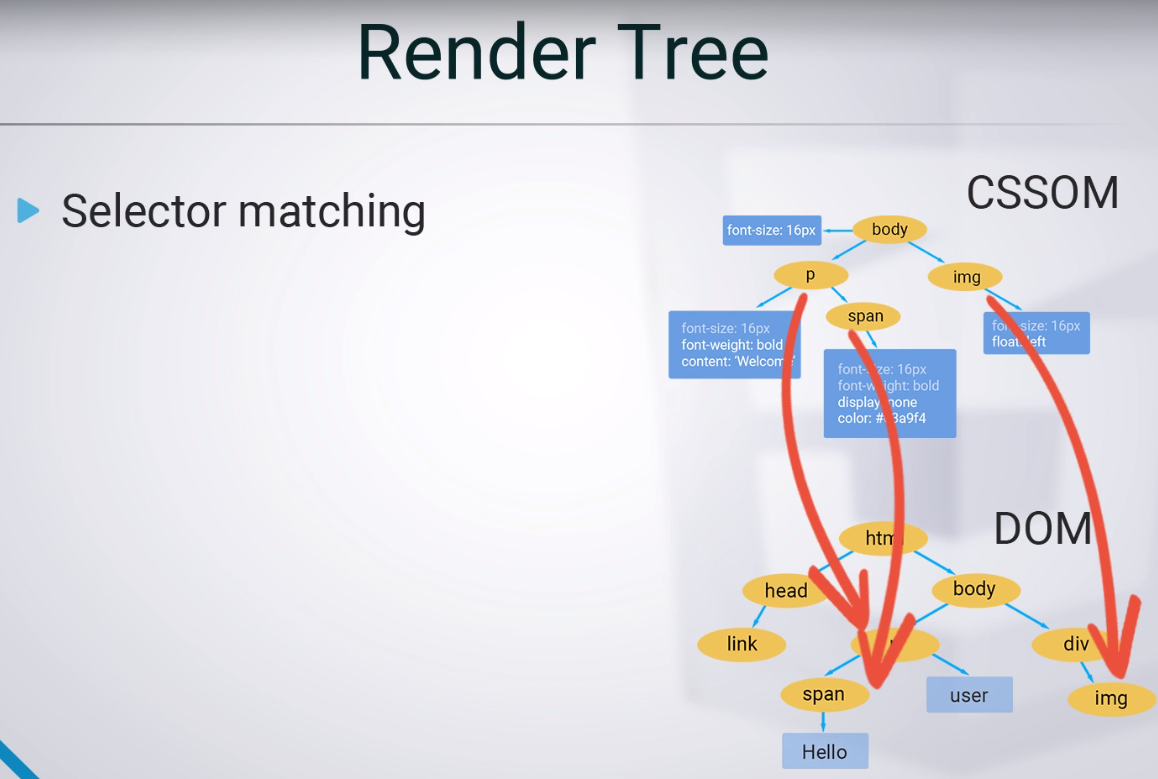


While DOM contains all the content of the page, CSSOM contains all the styles of the page

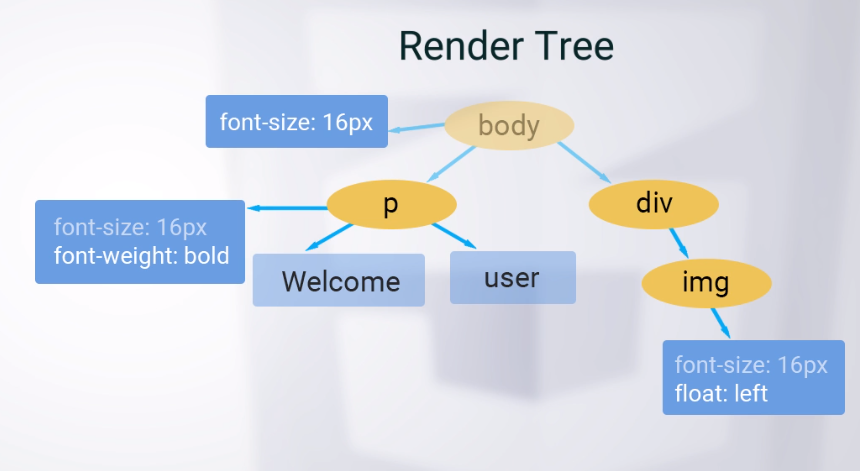
Render tree

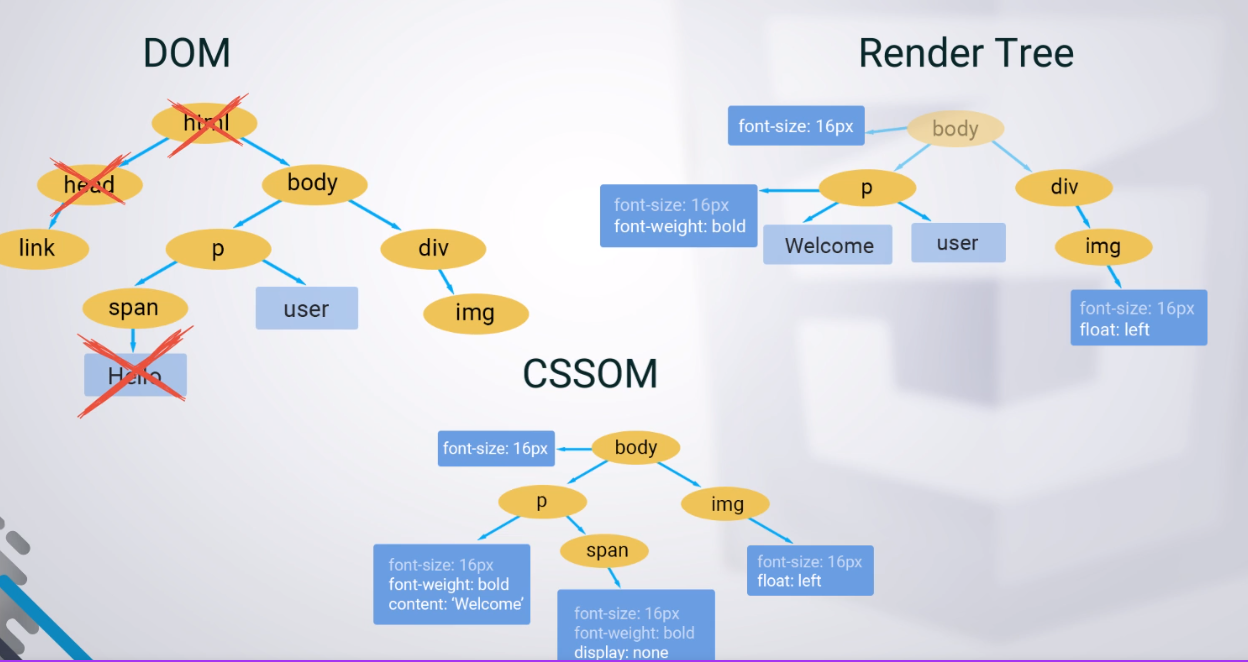
The next part of the process is to combine DOM and CSSOM.

Selector matching



The second part of the process involves taking all the style rules from the matching selectors and figuring out what final styles the element has.





Composite Layers

. Update Layer Tree

. Composite Layers

You can do a lot of things to optimize the web application: optimize your server, use caching, optimizing the javascript, use lazy loading, reloading and many other things. But we will talk only about CSS.

Selector matching

Writing simple selectors is a good idea.

More specific rule is more expensive because it has to travel more nodes in the DOM.

But in general the browser does selector matching very quickly. So it’s not worth spending your time on rewriting your old selectors. But you should keep in mind, that some selectors are faster than others.

Paint

MoreTools -> Rendering -> Paint Flashing

Creating new layer

will-change: transform;

Or transform: translateZ(0); // older browsers

Request Animation Frame

. Guarantee that your javascript will run at the start of a frame

. That gives the browser as much time as possible to run other activities

. Don’t use setTimeout or setInterval for animations

Improve donwload time

we need the critical resources

A Critical Resource is a resource that could block initial rendering of the page.

we need to minimize the number of ciritical resources and the number of critical byes.

Critical Bytes are the total number of bytes required to complete the initial render of the page.

Whats the minimum amount of resources that we can ship to render the page?

Today it seems impossible to render a page in 1 or 2 seconds, because the browser needs to load HTML….. .

But we need to show the user the most important content in the first secoond or two.

We don’t want to wait for all files to be downloaded before we can show something on the screen.

Parser blocking javascript

The main blockers for accelerating rendering are style sheets and scripts.

JavaScript is parser blocking by default.

When HTML parser finds a script tag, it pauses the parsing of the HTML document and has to load, parse, and execute the Javascript code.

The browser waits for the script to be fetched from the server, which can add thousands of milliseconds of delay to the critical rendering path. Therefore synchronous scripts in HEAD of the HTML, blocks the entire page from rendering until they finish loading.

To eliminate unnecessary Javascript from the critical rendering path, one option is to make your javascript “async”.

This allows the browser to continue to construct the DOM and let the script execute when it’s ready.

You can use it only if your javascript doeesn’t change the DOM structure.

Of course, the script will be downloaded in both cases, but the script is no longer parser blocking.

There is a common rule to put script tag just before closing “body”.We still block rendering, but hopefully by that point, we have already constructed most of the page.

What happens if you have a large image?

The good thing is that images do not block the render of the page. Browser will render a page even if images never arrive at all.

We reduce the number of critical resources by 1.

Javascript -/

images

CSS

Render Blocking CSS

CSS is a render blocking resource.

The browser will not start creating CSSOM without fully loaded CSS.

CSS is critical for rendering a page. It will not do Layout or paint.

What’s the minimum amount of resources that we can ship to render the page?

Do we really need a print media to render the page?

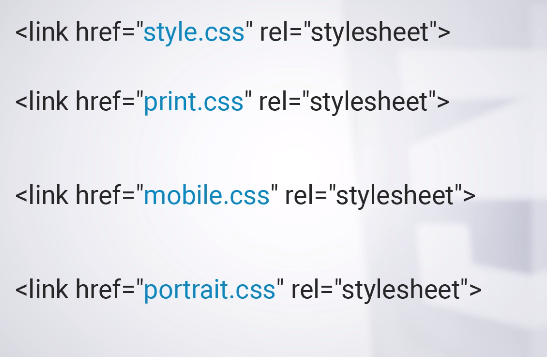
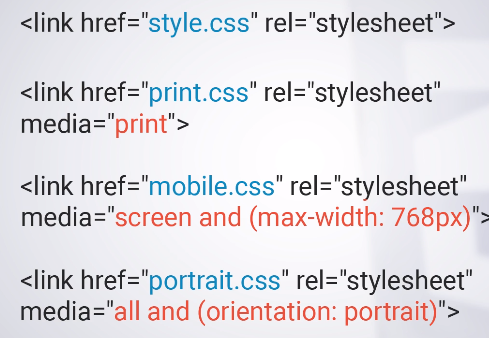
If we are not printing the page, there is no reason t o block rendering.

It makes sense to split the style.

So instead of single file we have multiple smaller files.

The browser assumes that each style sheet is render blocking.

However, we can also tell the browser when the stylesheet should be aplied.

Cutting down unused files Make your website

More reliable

Easier to maintain

Simpler to debug

Reduce the number of critical bytes

Optimize critical rendering path

>> More Tools > Coverage

Base54

Don’t inlcude image in CSS as base64.

It means we turned hundres of kilobytes of

non-blocking resourcs into blocking ones!!!

Network speed is the bottleneck of any web application.

It’s important to keep file sizes small, because nothing will render until all render blocking assets download.