

[Home](#) > [Software Engineering](#) > [Questions](#) > [Frontend](#)

What are Reflow and Repaint? How to optimize?

February 3, 2023



Support Us

Your support will help us to continue to provide quality content. [👉 Buy Me a Coffee](#)

Reflow and **Repaint** are two important concepts in browser rendering process.

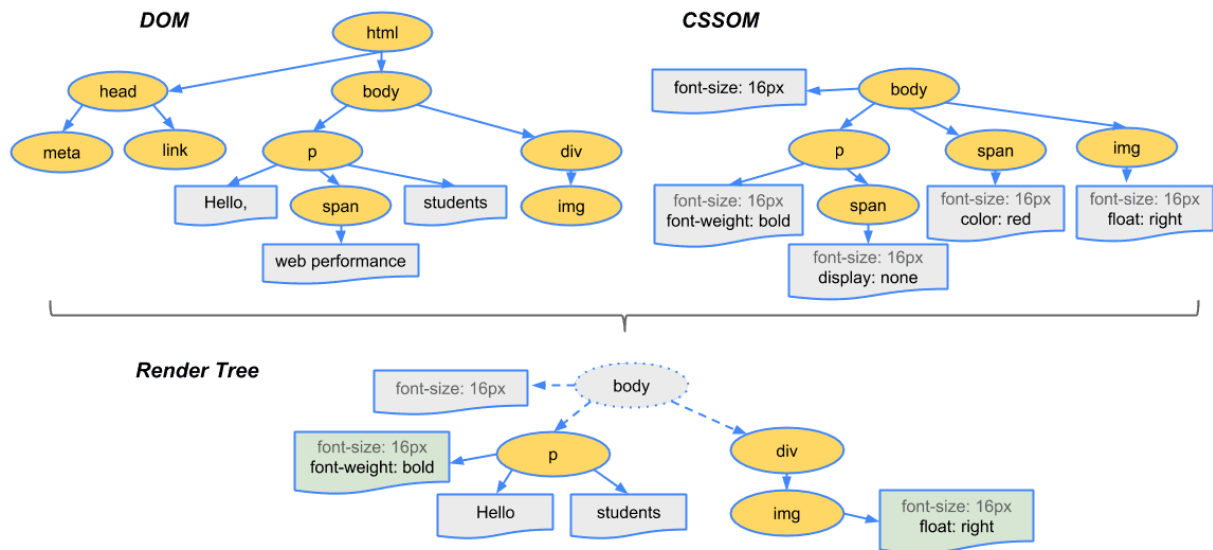
Reflow is the process of browser re-calculating the layout and position of elements on the page. Repaint is the process of browser re-drawing elements on the page. In this article, we will discuss the browser rendering process, Reflow and Repaint, and how to optimize.

Browser Rendering Process

The browser rendering process can be divided into several steps:

1. Parsing HTML and Style Calculation

The browser parses HTML into DOM, parses CSS into CSSOM, and merges DOM and CSSOM into **render tree**. See the following diagram:



DOM and CSSOM merged into render tree

<https://web.dev/critical-rendering-path-render-tree-construction/>

2. Layout

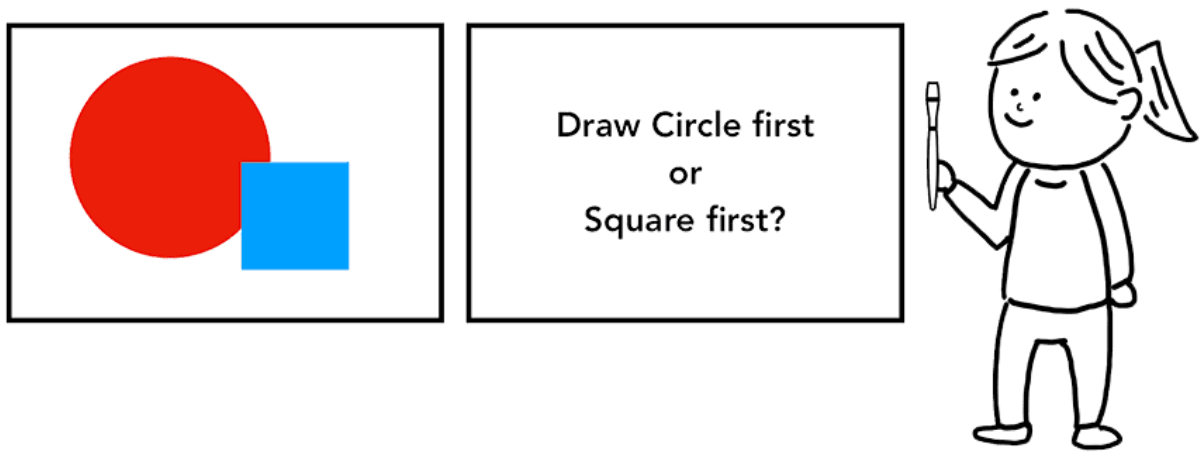
The **render tree** has the structure of DOM and the style of each node, but this is not enough to display the page. The browser also needs to calculate the size and position of each node on the screen. This process is called **layout** and produces a **layout tree**.

3. Paint

Having DOM, style and layout is still not enough to display the page, the browser still needs to determine the order of element painting. You can imagine this process as a painting process annotation (paint record), such as:

1. First, draw a background
2. Then draw text at (x,y,w,h)
3. Then draw a rectangle

As shown in the following diagram



Generate paint records

<https://developer.chrome.com/blog/inside-browser-part3/#paint>

4. Compositing

After the first three steps, the browser has obtained the information needed to render the page, but in order to improve the overall rendering efficiency, the browser will composite the information through **compositing** and render it to the screen. Compositing is a technique of dividing the page into layers (layers), and this technique will be executed on a separate thread called **compositor thread**. After this process is completed, a layer tree will be generated, and finally rendered to the screen.

Reflow and Repaint

After understanding the browser rendering process, let's go back to the question: what is Reflow and Repaint in the browser? And how to optimize?

Reflow and **Repaint** refer to the layout and paint steps of the rendering process. When we do something to change the layout or style, Reflow or Repaint will be triggered.

It should be noted that the browser rendering process actually has a cost, because in the rendering process, each step will use the result of the previous operation to create new data. For example: if the layout tree changes, it will need to be re-painted.

When does Reflow and Repaint happen?

When does Reflow happen?

Reflow is a very important factor that affects browser performance, because it may cause the entire or part of the page layout to be updated, and it may trigger the entire page Reflow because of a change in the size of a node. For example: change `width` , `height` , `font-size` and so on.

When does Repaint happen?

When an element on the page needs to change color or other properties that do not affect the layout, the browser will perform Repaint. Unlike Reflow, Repaint does not affect the layout of the page, but it will also affect the performance of the page. For example: change `outline` , `visibility` , `color` , `background-color` and so on.

Avoid Reflow and Repaint

The last process of the browser rendering process is **compositing**, in some cases, we can use some tricks to let the browser only perform **compositing**, avoiding Reflow and Repaint.

some ways to avoid Reflow and Repaint:

- Use `transform` to move and adjust elements
- Use `opacity` to change the visibility of elements
- If you need to frequently repaint or reflow the node, you can set it to a separate layer through `will-change` , because a separate layer can avoid the rendering behavior of this node affecting other nodes.

```
body > .sidebar {  
  will-change: transform;  
}
```

- Avoid frequent DOM operations with JavaScript



Support Us

Your support will help us to continue to provide quality content. [!\[\]\(74d4806277d7e73349d8e8c0897931e9_img.jpg\) Buy Me a Coffee](#)

Previous

← [What Is a Monorepo? Why Using a Monorepo?](#)

Next

[What is debounce? How to implement debounce function](#) →