

O mecanismo de renderização ou motor de layout ou motor de renderização ou mecanismo de navegação

O motor de renderização¹ é um componente do navegador/browser que detém a função de renderizar, i.e., exibir os conteúdos solicitados na tela do navegador. Esse mecanismo é necessário, pois uma página da web não é uma entidade única que pode ser baixada e exibida na tela um pixel de cada vez. Em programação renderizar é o ato de compilar e obter o produto final de um processamento digital.

O navegador/browser lê, interpreta e exibe arquivos HTML conforme as especificações de HTML e CSS mantidas pelo W3C Consórcio World Wide Web, a organização que estabelece padrões para a web.

Os navegadores² de alto nível detém como principais componentes:

- (a) A interface do usuário - isso inclui a barra de endereços, botão voltar/avançar, menu de favoritos, etc. Todas as partes do navegador são exibidas, exceto a janela principal, onde você vê a página solicitada.
- (b) O mecanismo do navegador - a interface para consultar e manipular o mecanismo de renderização.
- (c) O mecanismo de renderização - responsável por exibir o conteúdo solicitado. Por exemplo, se o conteúdo solicitado for HTML, ele será responsável por analisar o HTML e o CSS e exibir o conteúdo analisado na tela.
- (d) Rede - usado para chamadas de rede, como solicitações HTTP. Possui interface independente de plataforma e implementações subjacentes para cada plataforma.
- (e) Backend de interface do usuário - usado para desenhar widgets básicos, como caixas de combinação e janelas. Ele expõe uma interface genérica que não é específica da plataforma. Por baixo, ele usa os métodos de interface do usuário do sistema operacional.
- (f) Interpretador de JavaScript. Usado para analisar e executar o código JavaScript.

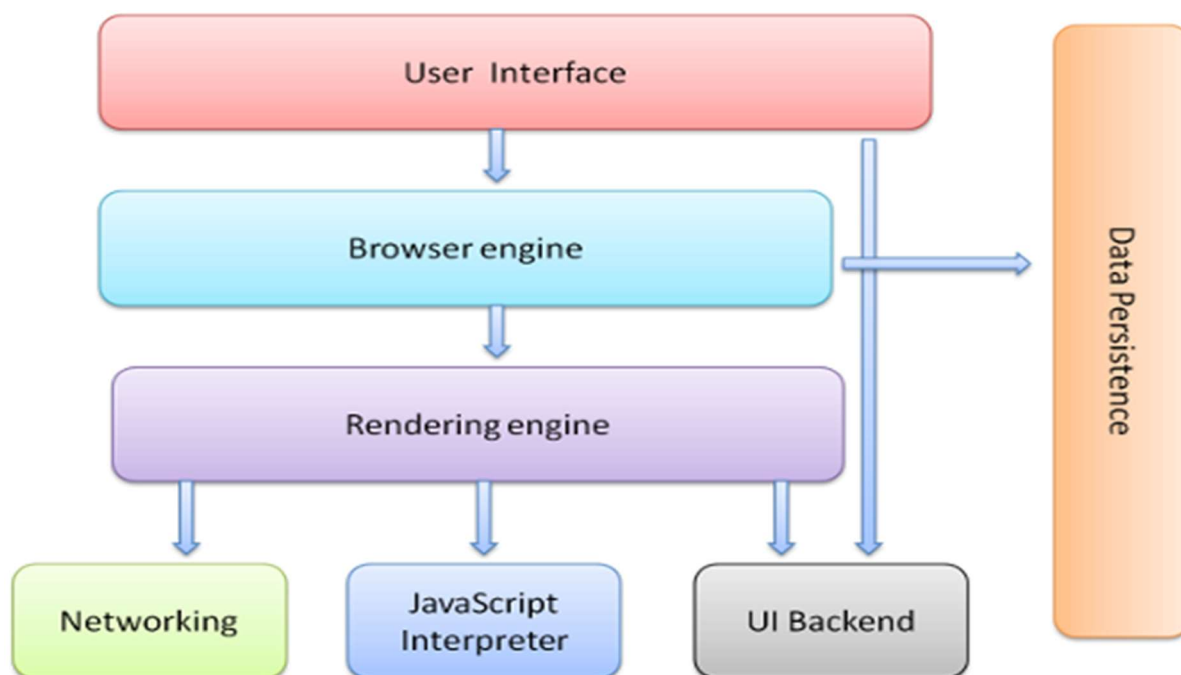
¹Baiadoconhecimento, Inc. © 2022. Disponível em: <https://baiadoconhecimento.com/biblioteca/conhecimento/read/418347-o-que-e-renderizacao-de-pagina>. Acesso em 23 de jul. de 2022.

²GROSSKURTH, Alan. Uma Arquitetura de Referência para Navegadores da Web. <http://grosskurth.ca/papers/browser-refarch.pdf>. Acesso em 23 de jul. de 2022.

CURSO TÉCNICO EM INFORMÁTICA PARA INTERNET

(g) Armazenamento de dados. Esta é uma camada de persistência. O navegador precisa salvar todos os tipos de dados no disco rígido, por exemplo, cookies. A nova especificação HTML (HTML5) define 'banco de dados da web' que é um banco de dados completo (embora leve) no navegador.

As imagens abaixo ilustram o conceito:



Importante salientar que o navegador/browser Chrome contém várias instâncias do mecanismo de renderização - uma para cada guia, ou seja, cada guia é um processo separado.

Fluxo básico do mecanismo de renderização³:



O mecanismo de renderização começará a analisar o documento HTML e transformará as tags em nós DOM em uma árvore chamada "árvore de conteúdo". Ele analisará os dados de estilo, tanto em arquivos CSS externos quanto em elementos de estilo. As informações de estilo junto com as instruções visuais no HTML serão usadas para criar outra árvore – a árvore de renderização .

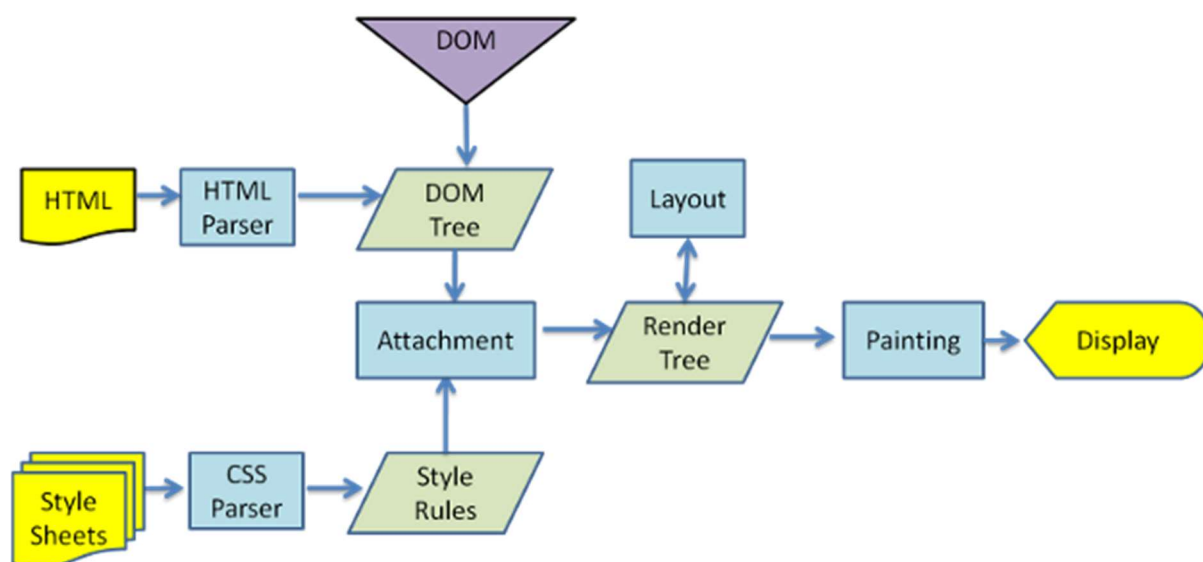
³GROSSKURTH, Alan. Uma Arquitetura de Referência para Navegadores da Web. <http://grosskurth.ca/papers/browser-refarch.pdf>. Acesso em 23 de jul. de 2022 e L. David Baron, Faster HTML e CSS: Layout Engine Internals for Web Developers (Google tech talk video). http://www.youtube.com/watch?v=a2_6bGNZ7bA.

A árvore de renderização contém retângulos com atributos visuais como cor e dimensões. Os retângulos estão na ordem certa para serem exibidos na tela.

Após a construção da árvore de renderização ela passa por um processo de "layout". Isso significa dar a cada nó as coordenadas exatas onde ele deve aparecer na tela. O próximo estágio é a pintura - a árvore de renderização será atravessada e cada nó será pintado usando a camada de back-end da interface do usuário.

É importante entender que este é um processo gradual. Para uma melhor experiência do usuário, o mecanismo de renderização tentará exibir o conteúdo na tela o mais rápido possível. Ele não esperará até que todo o HTML seja analisado antes de iniciar a construção e o layout da árvore de renderização. Partes do conteúdo serão analisadas e exibidas, enquanto o processo continua com o restante do conteúdo que continua vindo da rede.

Exemplo de diagrama de fluxo principal:



Alguns motores de renderização são Blink, EdgeHTML, Gecko, Goanna, KHTML, MARTHA, Prince, QtWebEngine, Servo, WebKit.

Vale ressaltar que os motores influenciam no comportamento dos layouts desenvolvidos, portanto torna-se importante entender quais os motores utilizados nos navegadores que os executarão.