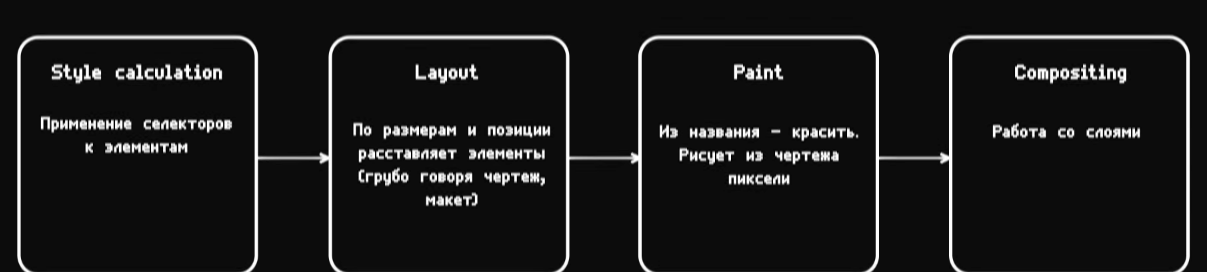
1. **Браузер**…………………………………………………………………………………….
   1. **Стадии** **рендера. DOM, CSSOM, Render tree, style calculation, layout, paint, composite**………………………………………………………………………………
   2. **Стадии** **рендера. DOM, CSSOM, Render tree, style calculation, layout, paint, composite**

Как отрисовывается страница, с которой пользователь взаимодействует в браузере. В какой очередности, по каким стадиям этот процесс происходит.



1. Калькуляций стилей. На этом этапе браузер применяет Style Rules к узлам DOM-дерева. Чем сложнее CSS селекторы и чем их больше – тем процесс калькуляции стилей будет дольше и тем больше нагрузка на железо.
2. Layout. На этом этапе браузер знает какие стили к какому элементу должны быть применены, поскольку до этого они были скалькулированы на предыдущем этапе. На этом этапе происходит составление макета, чертежа страницы. На выходе будет Layout-Tree (дерево-макета, дерево-расположения и т.п.). Содержит информацию о том, где элемент должен располагаться, виден он или не виден и т.п.
3. Paint. Применение подсчитанных стилей из первого этапа к макету со второго этапа. Строятся, так называемые, Paint Records.
4. Compositing (композиция). Браузер делит то, что было получено на предыдущем этапе на определенные слои – создает дерево слоев (layer-tree). На этом этапе браузер работает со слоями, задействует для этого видеокарту, например. Располагает элементы в зависимости от z-index и т.д.

DOM-дерево и дерево стилей строится чуток пораньше (как только мы получили HTML и CSS). В этот момент строится DOM, CSSOM и само деревео рендера (Render-tree). Вот таким образом работает всё при первой отрисовке. Вот полная схема:



Но как это работает в случае перерисовок? Например, нажали на кнопку, откроется модальное окно. В таком случае и стили меняются, и отображение элементов на странице меняется и DOM-дерево может меняться из-за модалки.

На самом деле новго ничего не происходит. JS триггерит, например, изменние каких-то стилей, и стади рендера начинаются по новой: перерисоывается **часть** DOM-дерева, обновляется некоторая часть CSSOM, строится новое Render-tree.

**Рендер вызывается:**1) Изменение размера окна

2) изменение шрифта

3) изменение контента

4) Добавление/удаление классов/стилей

5) Манипуляции с DOM

6) Изменение ориентации (альбом/книга)

7) Изменение размеров/позиции

8) Вычисление размеров/позиции