

# 浏览器渲染原理

—任科

## 浏览器地址栏输入网址，敲下回车后，发生了哪些事情？

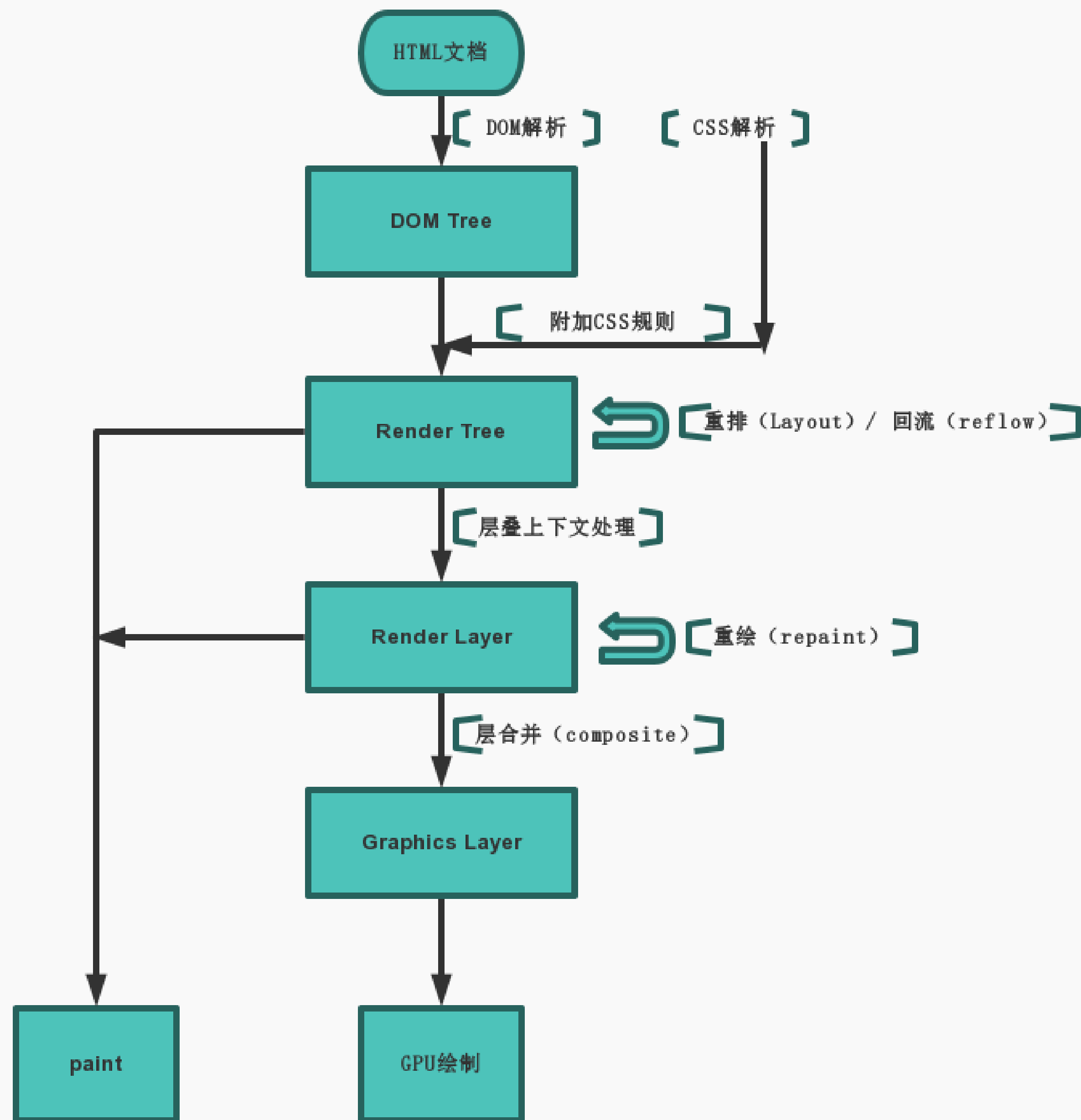
- 1、浏览器发出请求，向DNS服务器（先查找缓存）查找输入URL对应的IP地址。
- 2、DNS服务器返回对应的IP地址。
- 3、浏览器根据IP地址与目标服务器建立TCP连接，请求服务器。
- 4、服务端代码运行，返回HTML文档。
- 5、浏览器渲染页面。



## 浏览器渲染引擎类别（浏览器内核）

- IE (Trident)
- Chrome (Blink)
- Firefox (Gecko)
- Safari (Webkit)
- QQ浏览器/微信webview (X5/Blink)

# 浏览器渲染过程





# DOM解析

把HTML文档解析为DOM树的过程

- 遇到<script>标签则停止解析，先执行js
- DOM解析完成后触发DOMContentLoaded事件
- 此时图片资源并未加载完成

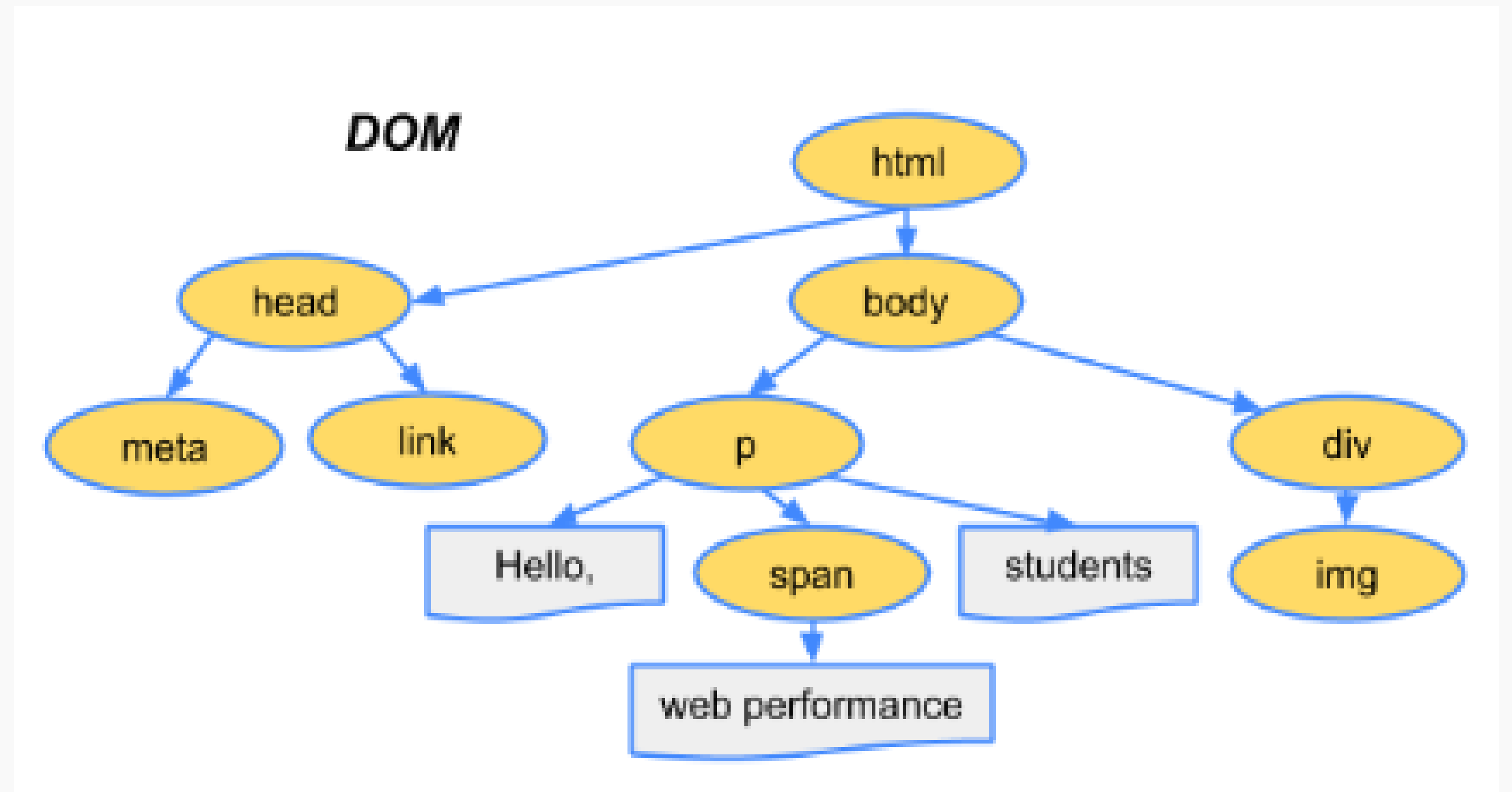
## DOM Tree

DOM树结构与HTML标签一一对应

display: none的元素也在DOM树中

<script>标签也在DOM树中

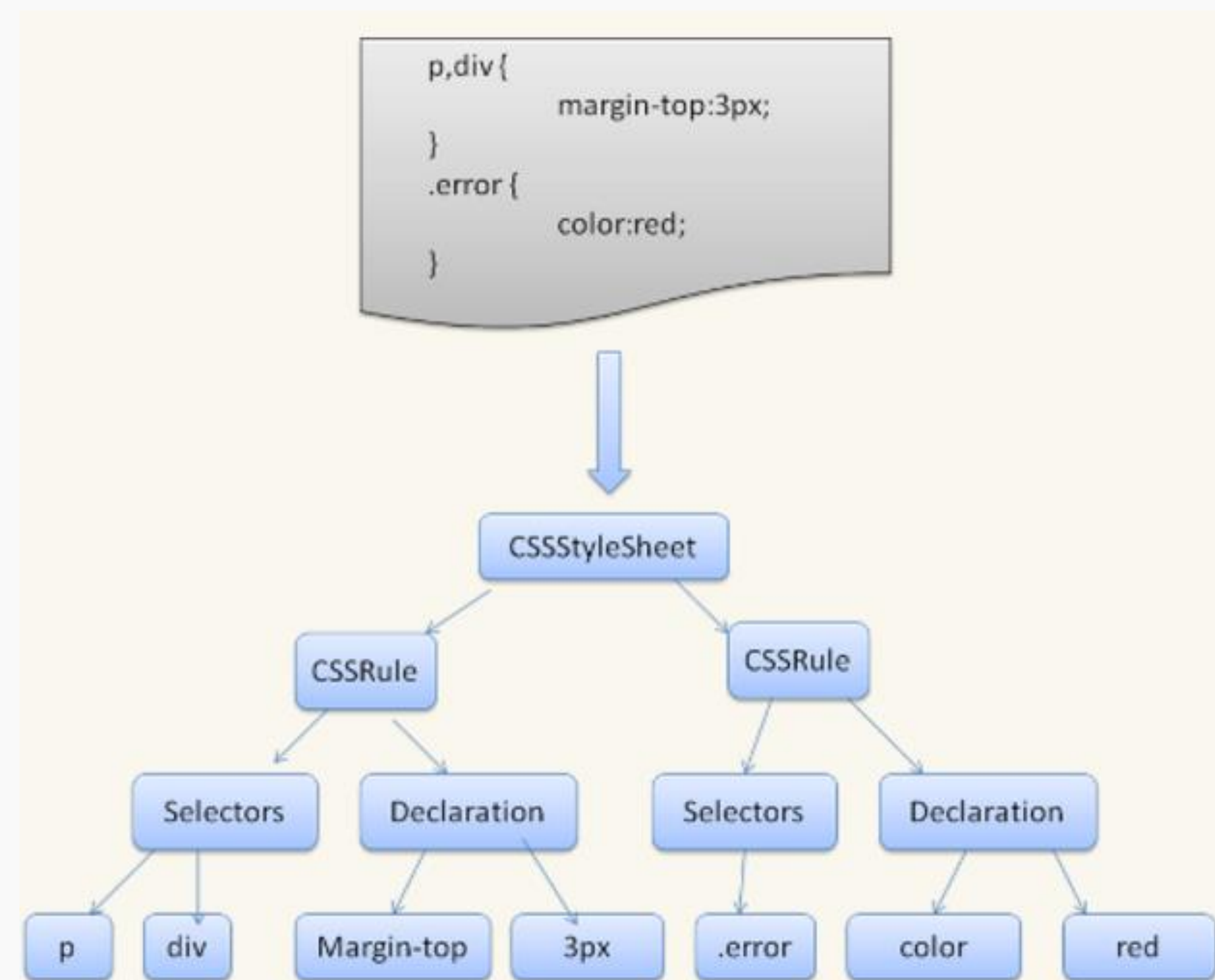
注释也在DOM树中



# CSS解析

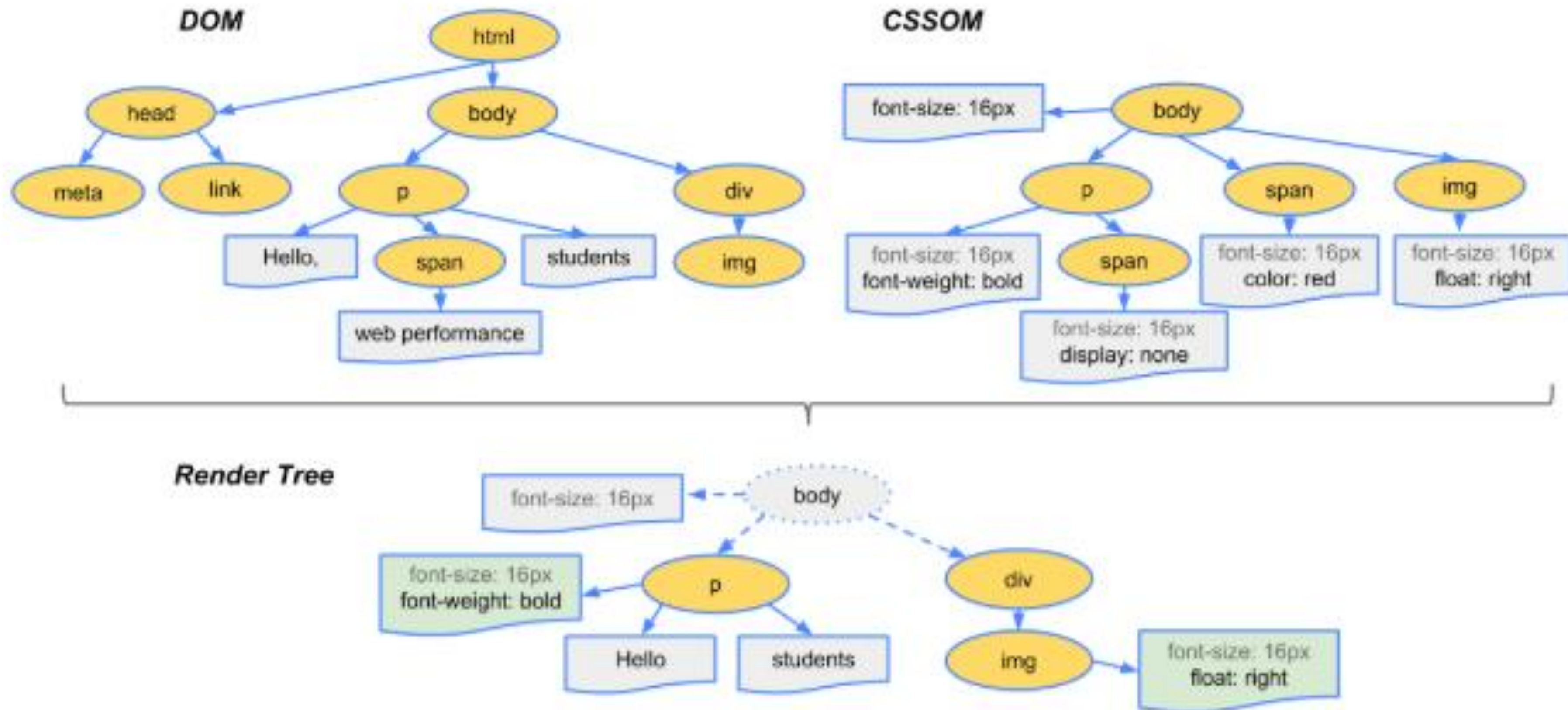
将CSS代码解析为CSS规则树的过程

- 与DOM解析同步进行
- 与script的执行互斥
- Webkit内核进行了script执行优化





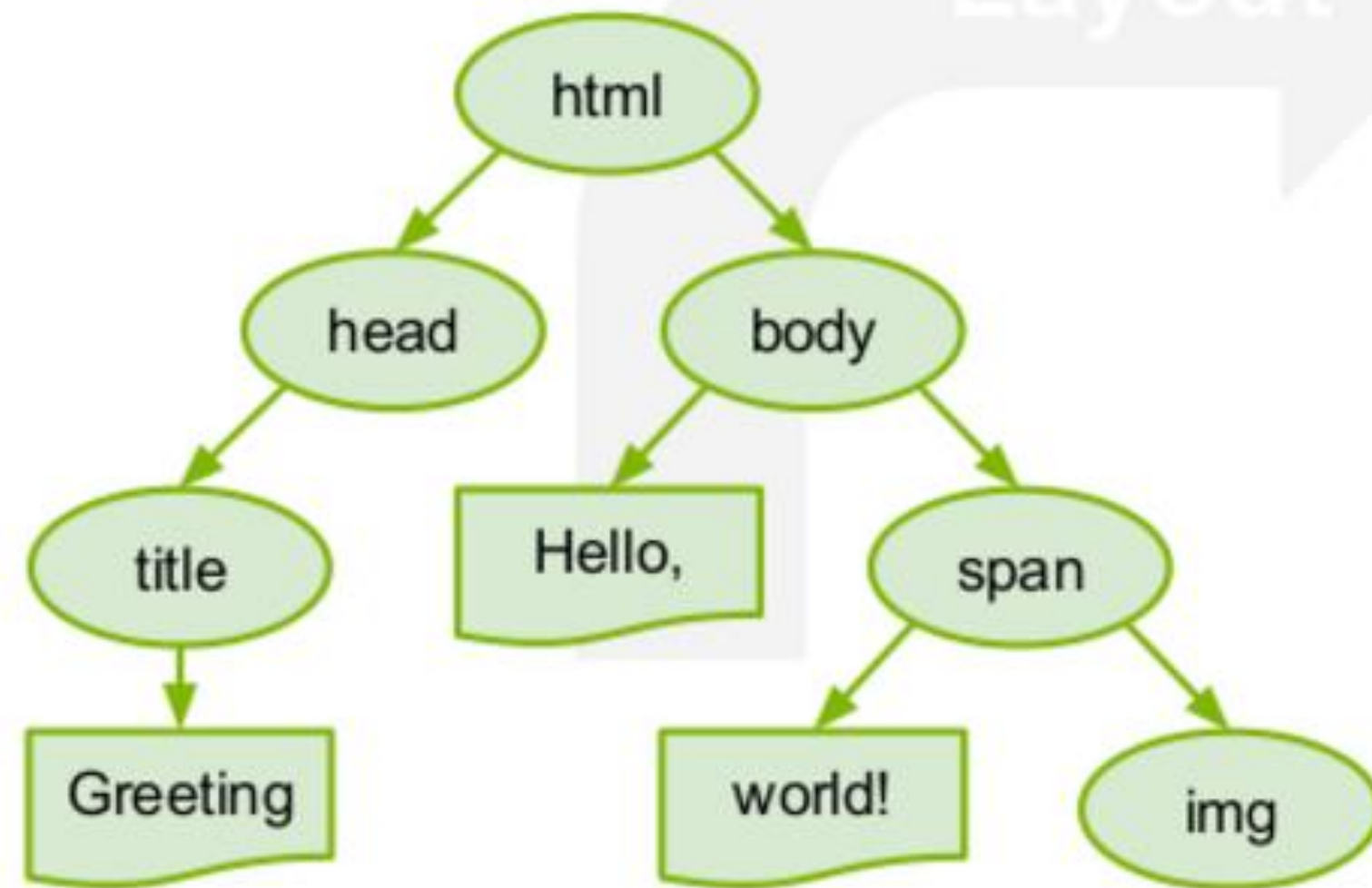
# Render Tree



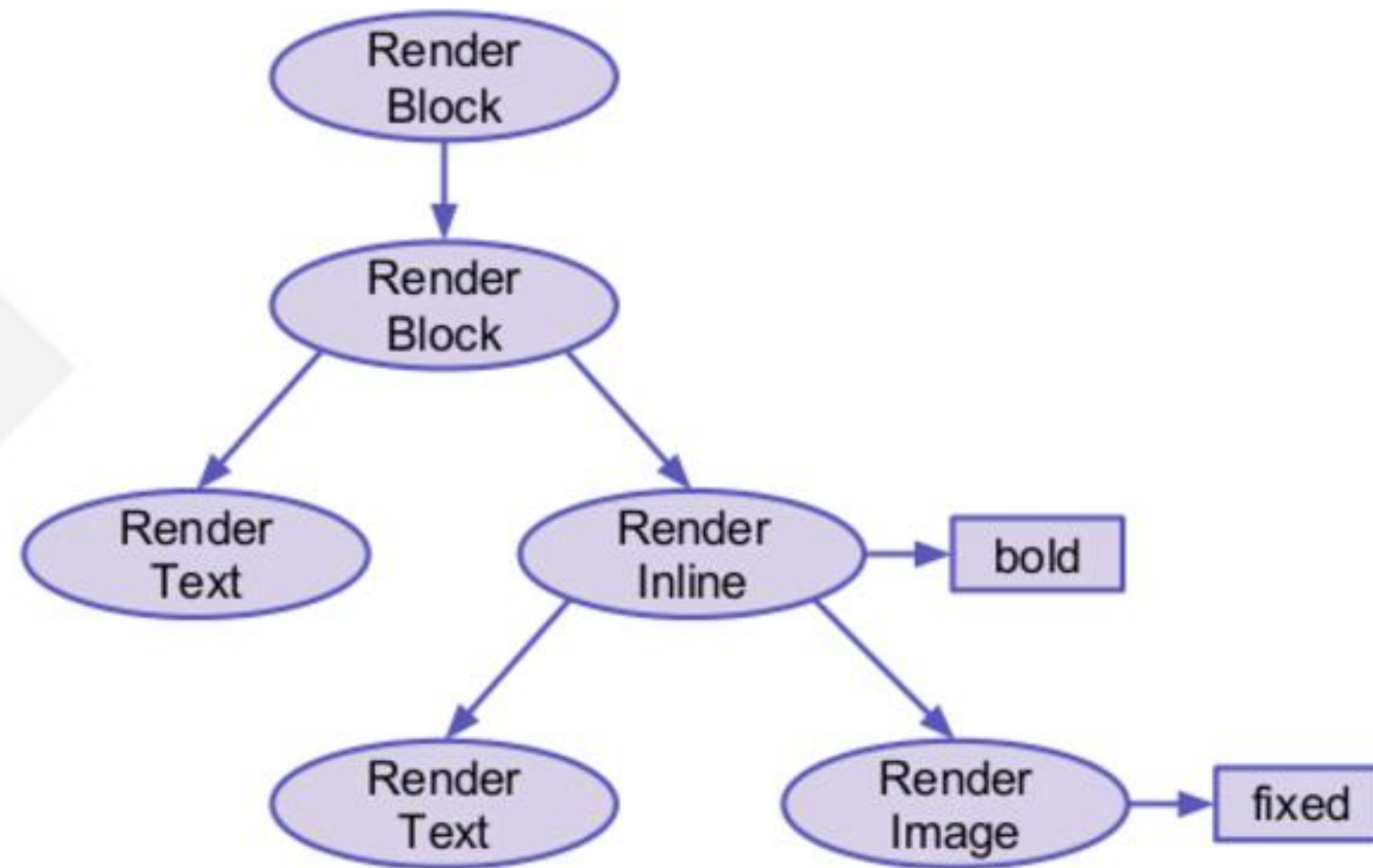


# Render Tree

```
#footer { position: fixed; bottom: 0; left: 0 }  
body > span { font-weight: bold; }
```



Layout





# Render Tree

DOM Tree + CSS Rules = Render Tree

- 每个节点为一个Render Object对象，包含宽高、位置、背景色等样式信息
- 宽高和位置是通过Layout（重排）计算出
- Render Tree和DOM Tree不完全对应
- display: none的元素不在Render Tree中
- visibility: hidden的元素在Render Tree中
- float元素、absolute元素、fixed元素会发生位置偏移
- 常说的脱离文档流，就是脱离Render Tree



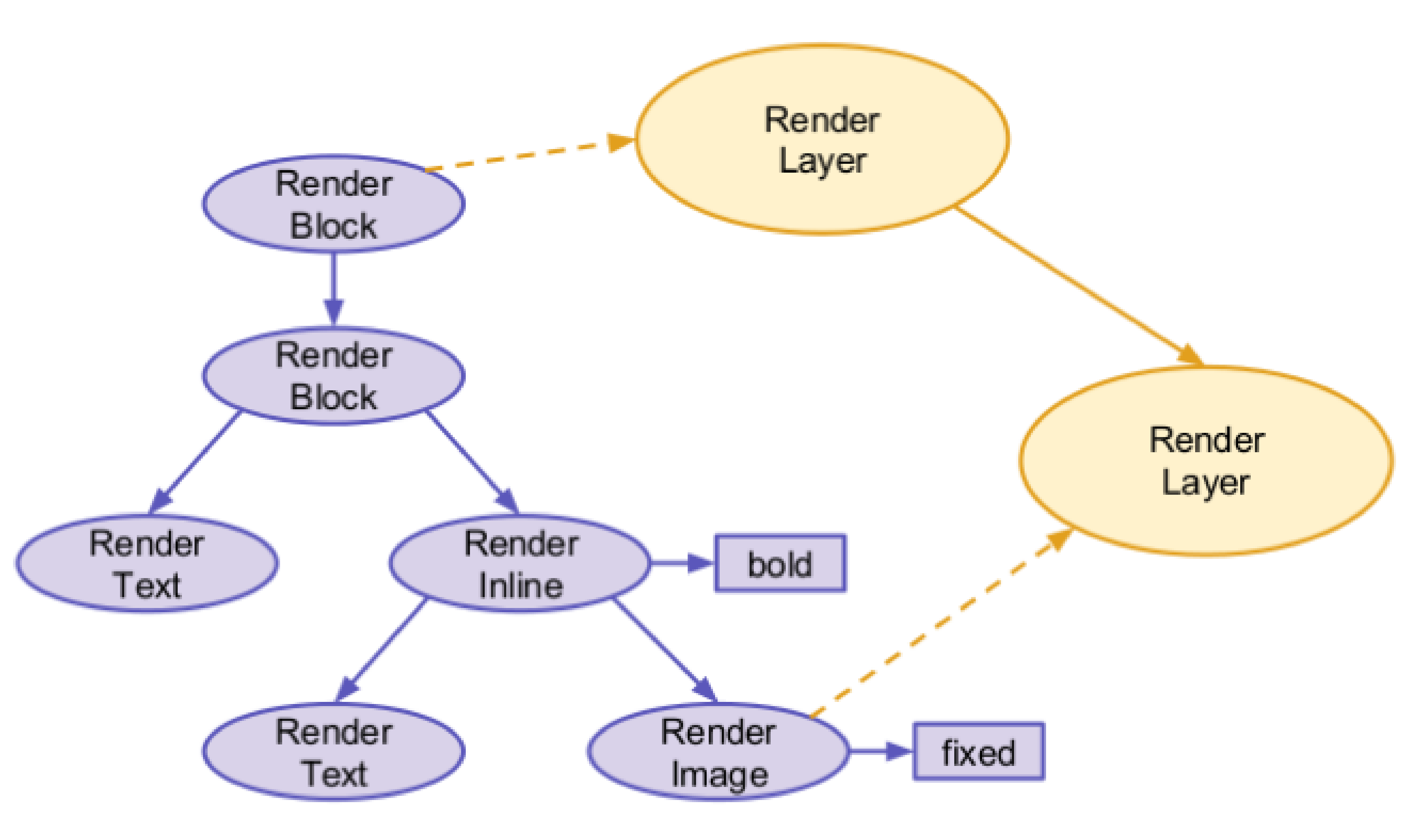
## 重排（Layout）/回流（reflow）

- 当修改元素的位置、大小时，引起浏览器的重排
- 对一个元素的重排，可能影响到其父级元素和相邻元素

如何避免重排？

- 用transform做形变和位移
- 通过绝对定位，脱离当前层叠上下文（即形成新的Render Layer）

# 生成Render Layer



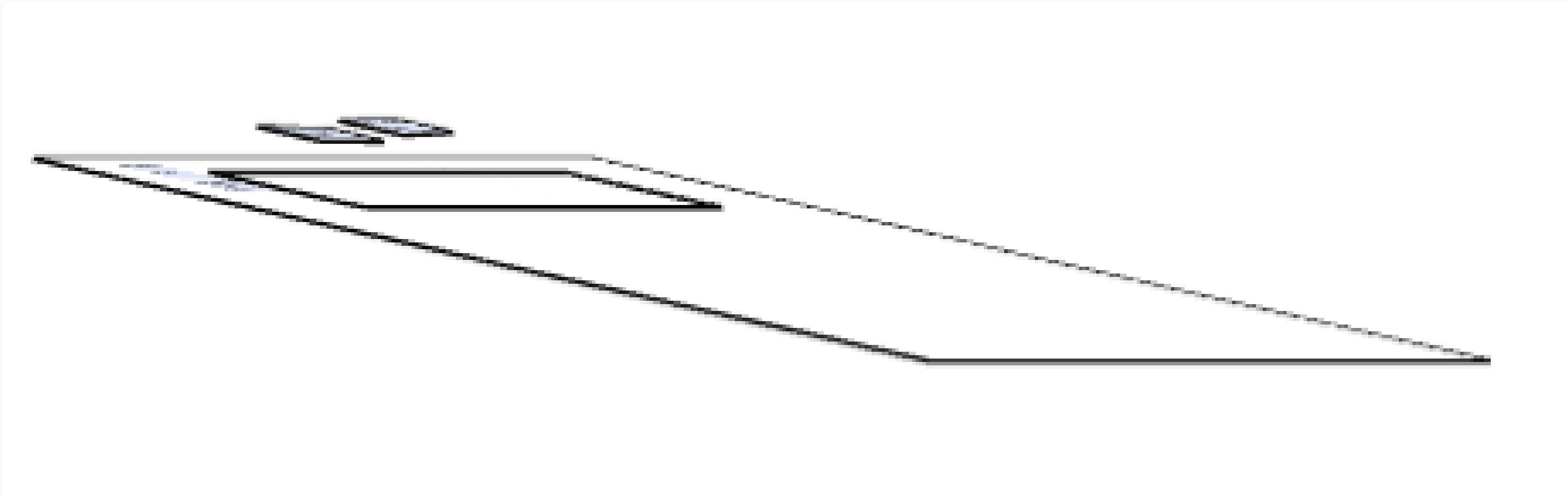


# 生成Render Layer

将Render Tree上的某些节点提升到同一个Layer的过程

- 根元素 (HTML)
- 有明确的定位属性 (relative、fixed、sticky、absolute)
- 透明的 (opacity 小于 1)
- 有 CSS 滤镜 (filter)
- 有 CSS mask 属性
- 有 CSS transform 属性 (不为 none)
- backface-visibility 属性为 hidden
- overflow 不为 visible

生成Graphics Layer





## 生成Graphics Layer

将Layer Tree上的某些节点进一步提升与合并

- video、canvas元素，flash插件
- 拥有perspective、CSS3D变形的元素
- backface-visibility 为 hidden
- 对 opacity、transform、filter、backdropfilter 应用了 animation 或者 transition
- 设置了will-change属性的元素





## 生成Graphics Layer

- 没有Graphics Layer的元素与父元素共属同一个
- 过多的合成层会造成GPU传输的压力
- 优势：
- GPU直接渲染，快于CPU
- 当需要 repaint 时，只需要 repaint 本身，不会影响到其他的层
- 对于 transform 和 opacity 效果，不会触发 layout 和 paint