现代浏览器渲染过程

京程一灯
http://www.yidengxuetang.com

现代浏览器渲染过程

- 浏览器的进化过程
- 现代浏览器的特征与结构
- Chrome 浏览器架构
- Chrome 浏览器的渲染过程
- 初窥 WebKit

现代浏览器的进化

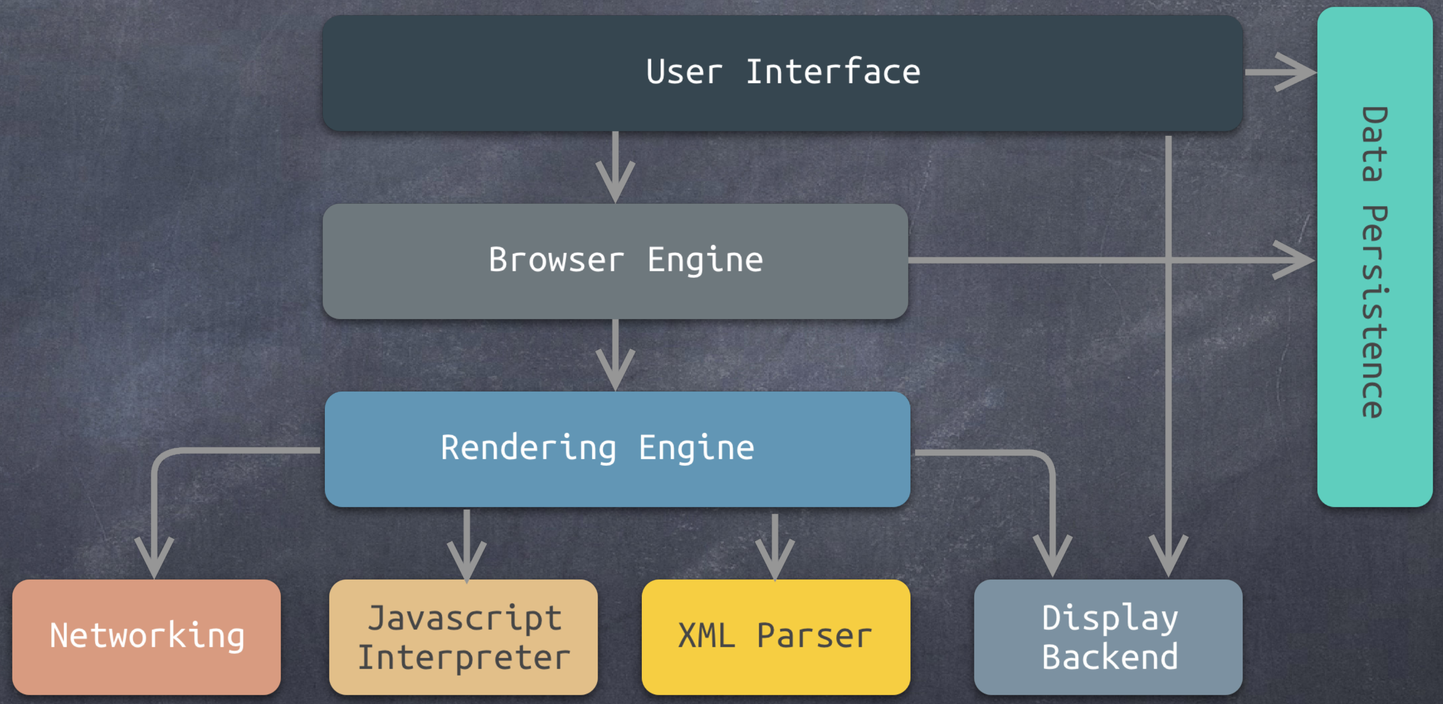
- 1990年，蒂姆·伯纳斯·李开发了第一个网页浏览器WorldWideWeb，后改名为Nexus。WorldWideWeb浏览器支持早期的HTML标记语言，功能比较简单，只能支持文本、简单的样式表、电影、声音、图片等资源的显示。
- 1993年，马克·安德森领导的团开发了一个真正有影响力的浏览器Mosaic，这就是后来世界上最流行的浏览器Netscape Navigator。
- 1995年，微软推出了闻名于世的浏览器Internet Explorer。第一次浏览器大战开始，持续两年
- 1998年，Netscape公司开放Netscape Navigator源代码，成立了Mozilla基金会。第二次浏览器大战开始，持续八年
- 2003年，苹果公司发布了Safari浏览器。
- 2004年，Netscape公司发布了著名的开源浏览器Mozilla Firefox
- 2005年，苹果公司开源了浏览器中的核心代码，基于此发起了一个新的开源项目WebKit（Safari浏览器的内核）。
- 2008年， Google公司已WebKit为内核，创建了一个新的浏览器项目Chromium。以Chromium为基础，谷歌发布了Chrome浏览器。至于这两者的关系，可以简单地理解为：Chromium为实验版，具有众多新特性；Chrome为稳定版。

现代浏览器的特征

- 网络
- 资源管理
- 网页浏览
- 多页面管理
- 插件和扩展
- 账户和同步
- 安全机制
- 开发者工具

现代浏览器的结构

- 用户界面 (User Interface)
- 浏览器引擎 (Browser Engine)
- 渲染引擎 (Rendering Engine)
- 网络 (Networking)
- XML解析器 (XML Parser)
- 显示后端 (Display Backend)
- 数据持久层 (Data Persistence)



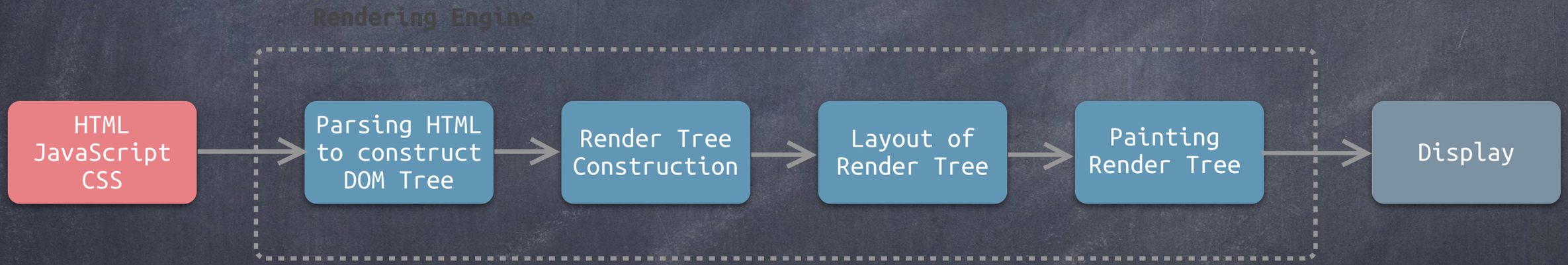
常见的渲染引擎

- 渲染引擎：能够能够将HTML/CSS/JavaScript文本及相应的资源文件转换成图像结果。
- 渲染引擎的种类：

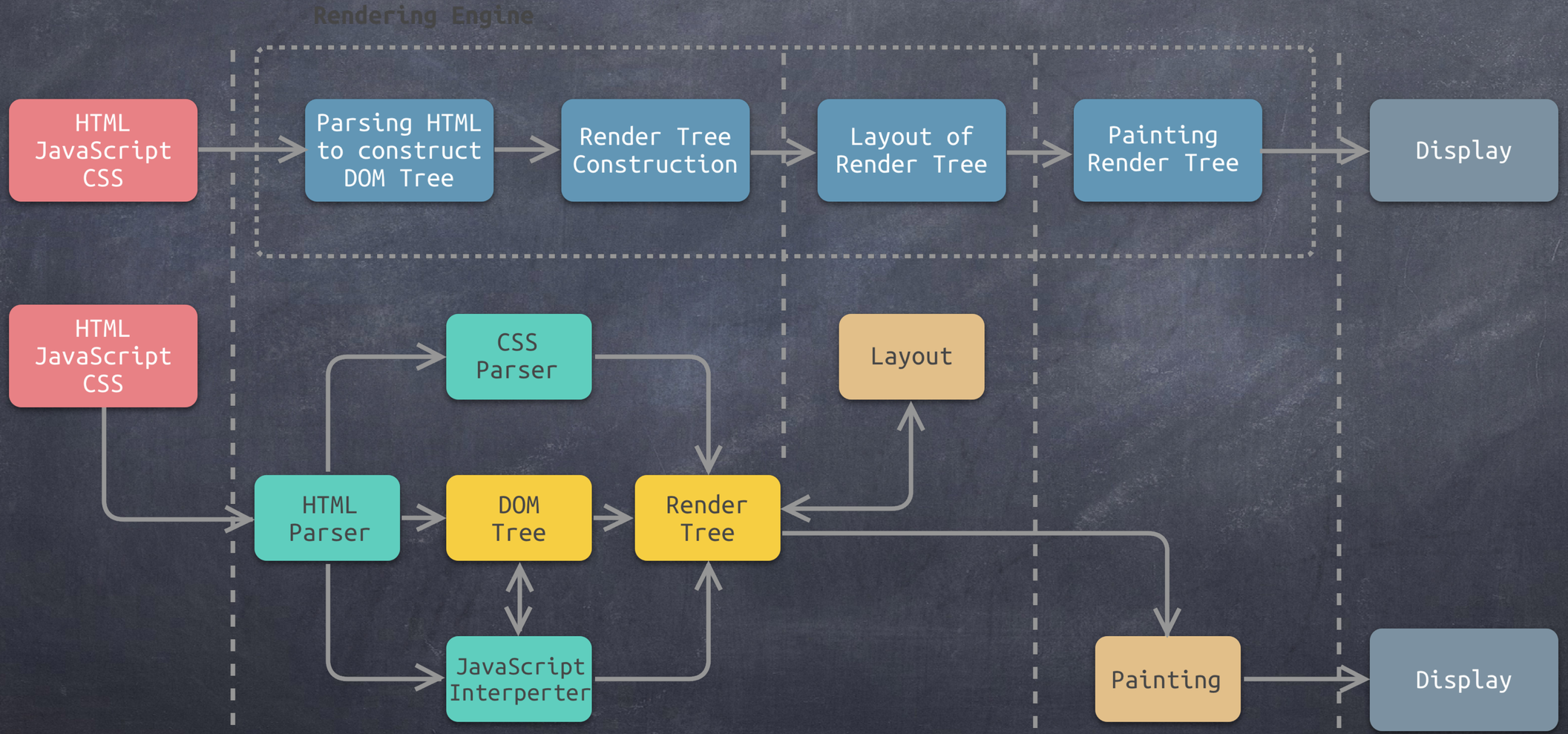
渲染引擎	浏览器
Trident	IE、Edge(旧)
Gecko	Firefox
WebKit	Safari
Blink(WebKit fork)	Chromium/Chrome，Opera，Edge(新)

渲染引擎结构与工作流程

- 以HTML/JavaScript/CSS等文件作为输入，以可视化内容作为输出

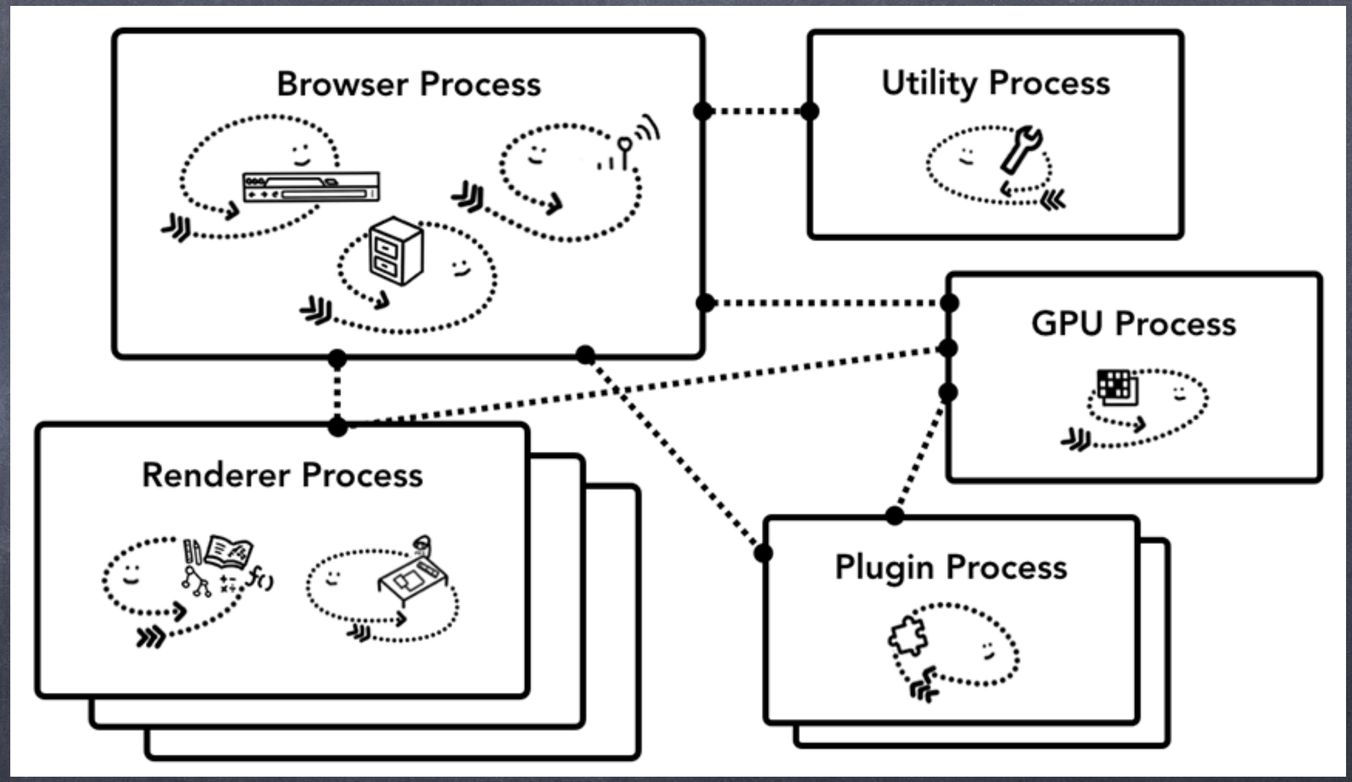


渲染引擎结构与工作流程



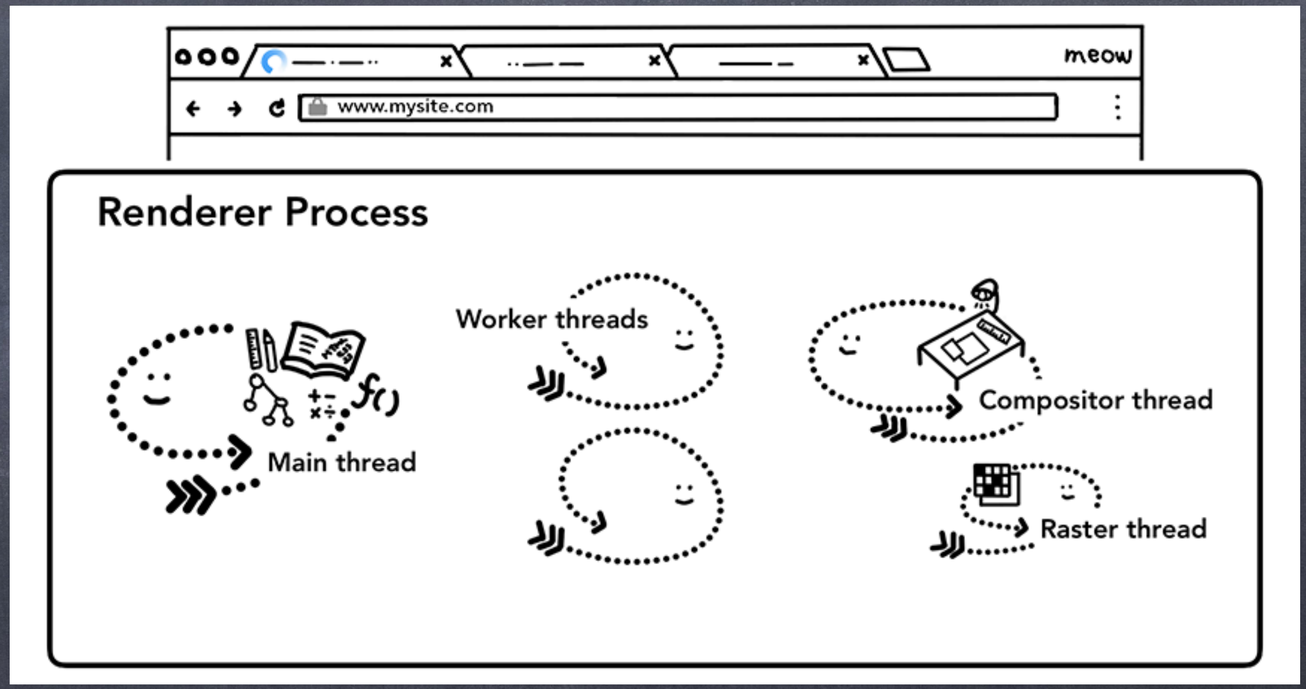
Chrome 架构

- Browser：控制程序的“chrome”部分，包括地址栏，书签，后退和前进按钮。还处理Web浏览器的不可见的，和特权部分，例如网络请求和文件访问。
- Renderer：负责显示网站的选项卡内的所有内容。
- Plugin：控制网站使用的所有插件，例如flash。
- GPU：独立于其他进程的GPU处理任务。 它被分成多个不同的进程，因为GPU处理来自多个程序的请求并将它们绘制在同一个面中。



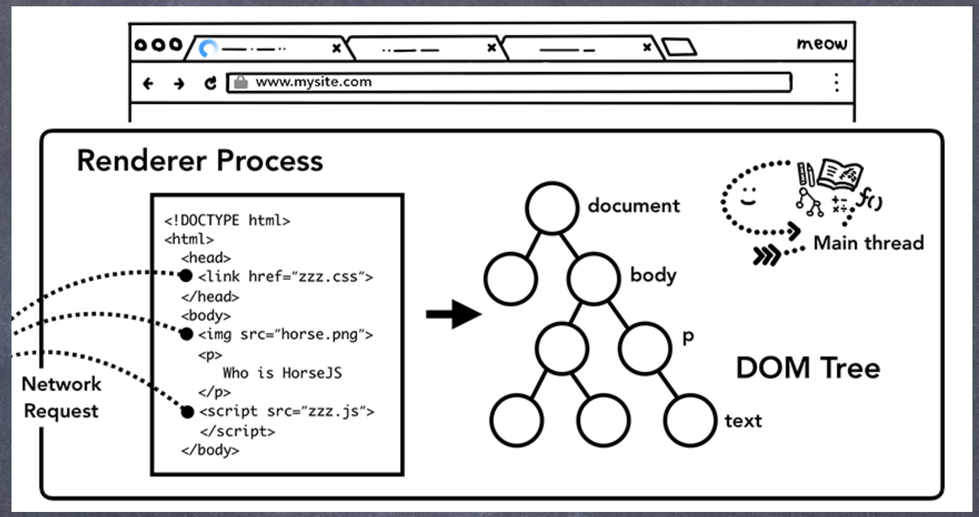
Chrome 渲染器进程

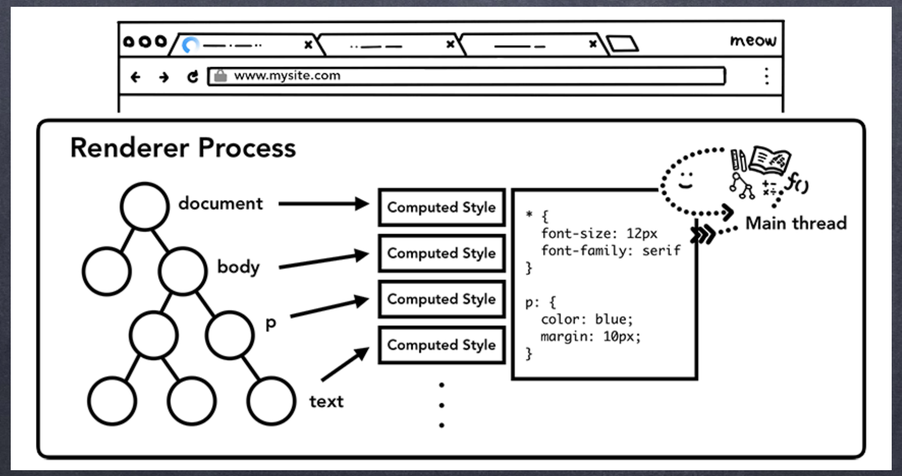
- 渲染器进程负责选项卡内发生的所有事情。 在渲染器进程中，主线程处理你为用户编写的大部分代码。
- 如果你使用了web worker 或 service worker，有时JavaScript代码的一部分将由工作线程处理。 排版和栅格线程也在渲染器进程内运行，以便高效、流畅地呈现页面。

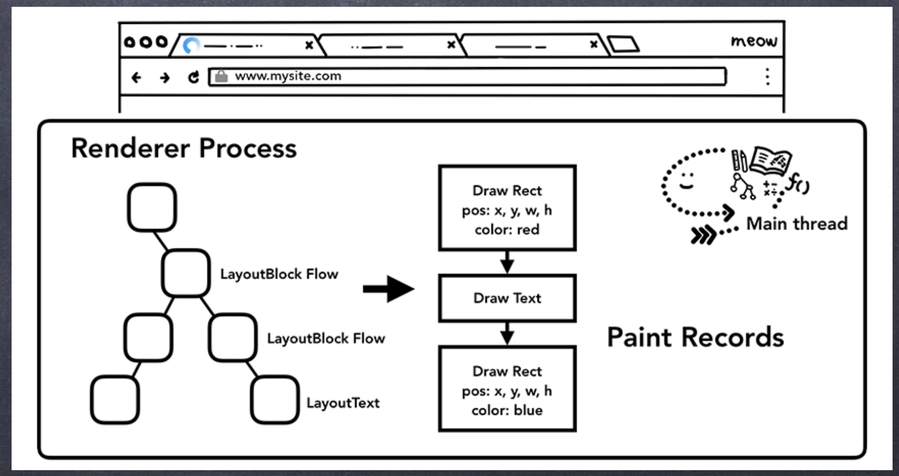


Chrome 渲染过程：解析部分

- 构建DOM
- 子资源加载
  - 注意JavaScript可以阻止解析
- 提示浏览器如何加载资源
- 样式表计算
- 布局
- 绘制

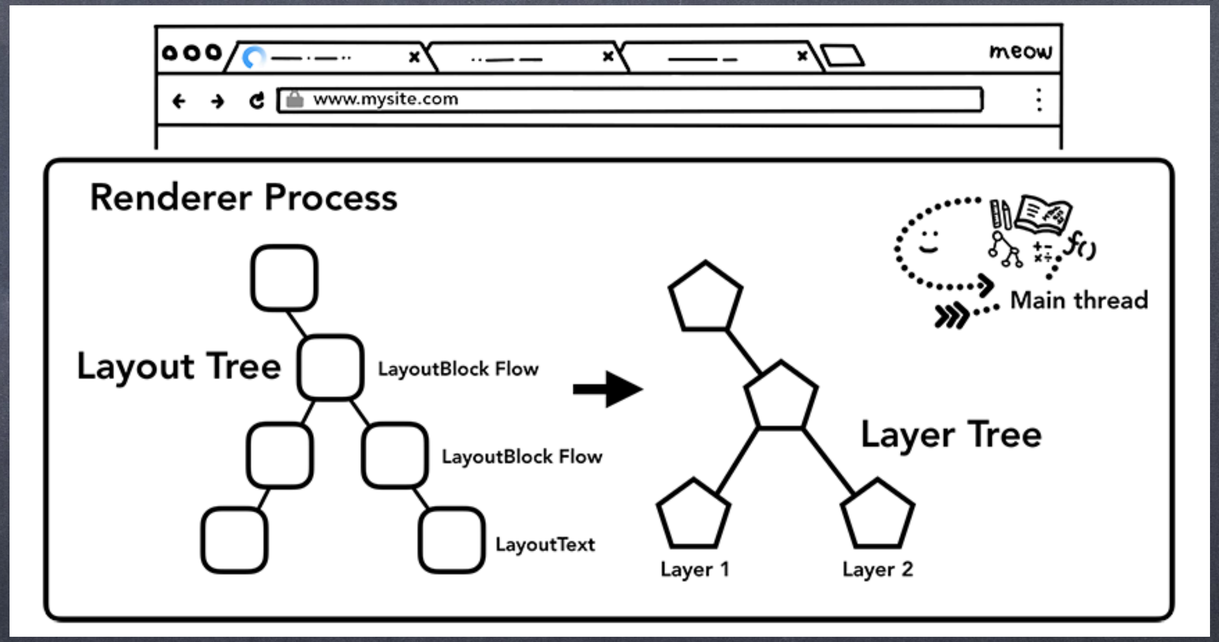






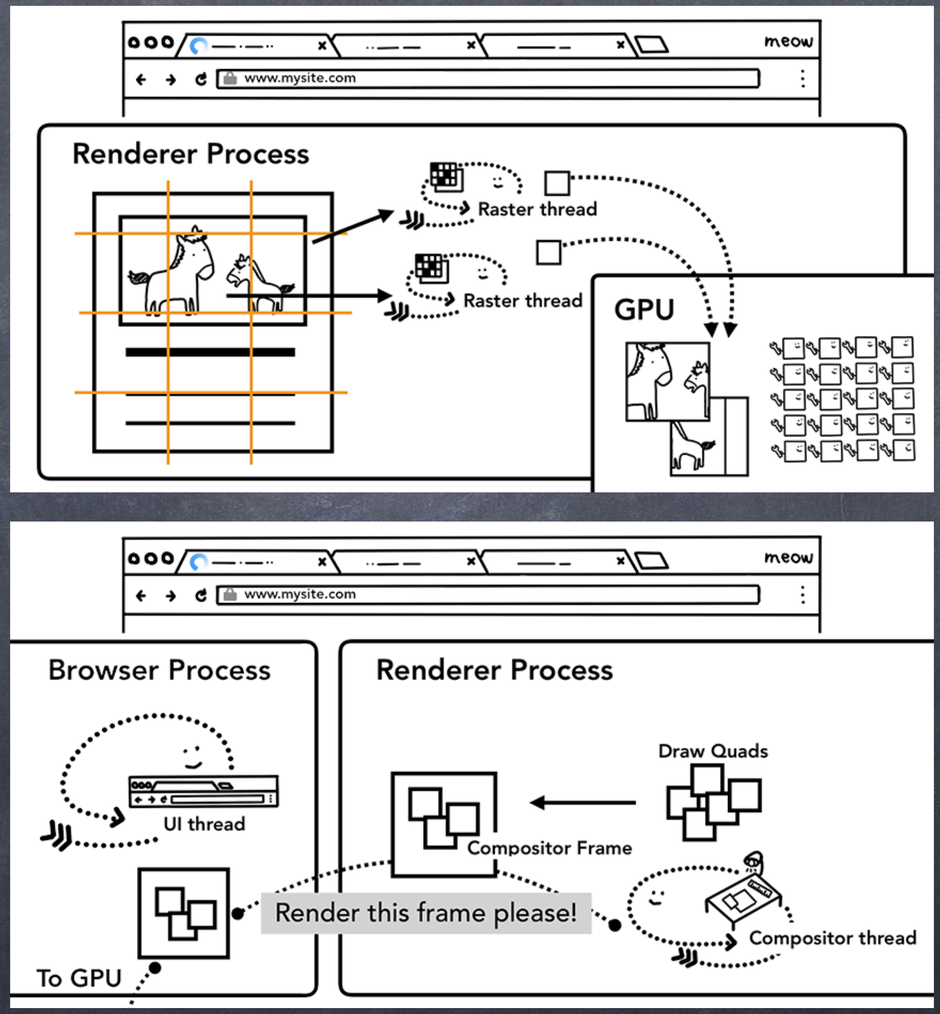
Chrome 渲染过程：合成部分

- 把文档的结构、元素的样式、几何形状和绘制顺序转换为屏幕上的像素称为光栅化。
- 合成是一种将页面的各个部分分层，分别栅格化，并在一个被称为合成器线程的独立线程中合成为页面的技术。



Chrome 渲染过程：GPU渲染

- 一旦创建了层树并确定了绘制顺序，主线程就会将该信息提交给合成器线程。 合成器线程然后栅格化每个图层。 一个图层可能像页面的整个长度一样大，因此合成器线程会将它们分成图块，并将每个图块发送到光栅线程。 栅格线程栅格化每一个tile并将它们存储在GPU内存中。
- 通过IPC将合成器帧提交给浏览器进程。这时可以从UI线程添加另一个合成器帧以用于浏览器UI更改，或者从其他渲染器进程添加扩充数据。 这些合成器帧被发送到GPU用来在屏幕上显示。 如果发生滚动事件，合成器线程会创建另一个合成器帧并发送到GPU。
- 合成的好处是它可以在不涉及主线程的情况下完成。 合成线程不需要等待样式计算或 JavaScript 执行。 这就是合成动画是平滑性能的最佳选择的原因。 如果需要再次计算布局或绘图，则必须涉及主线程。



初窥WebKit

- WebKit 官网：https://webkit.org/
- Blink 是未来
- Blink官方文档：http://www.chromium.org/blink
-