# web性能优化

TooBug 2017-03

# 前端性能关注点

# 速度







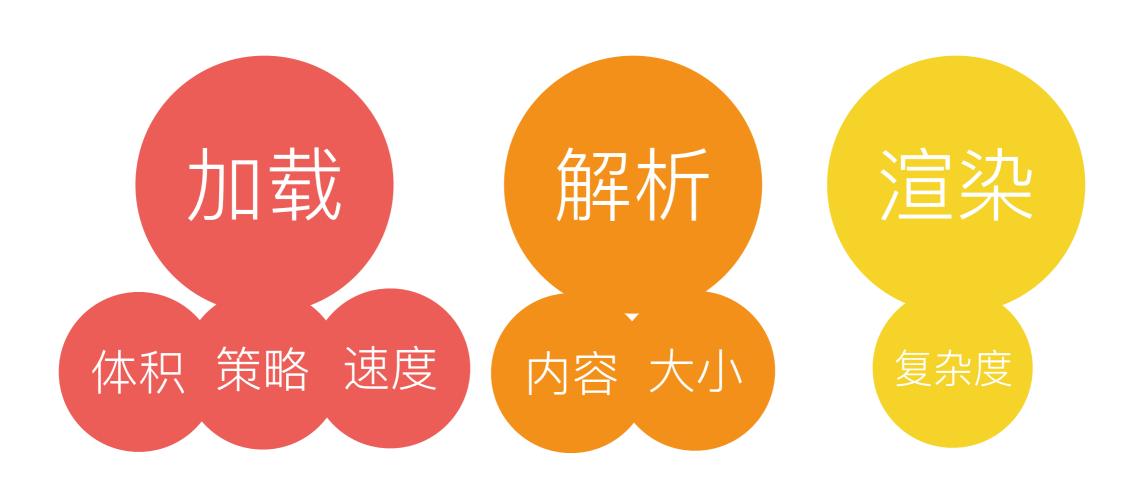
# 什么速度







### 感观速度由什么决定







#### 程序体积



没用的别引入 能不用别乱用 体积小要优先



图片使劲压(有损) 选择合适的格式 文本也要压(js/css)





#### 传输体积



极限缓存



压缩传输(gzip)



## 极限缓存

- 无脑长缓存(版本号的来历)
- 自建缓存机制 (本地存储)
- 程序离线
- 接口离线(缓存/service worker)
- 预加载







### 加载策略

布局 优先

结构和对应样式先行 隐藏结构样式置后 行为置后

首屏 优先

结构+样式





### 速度



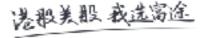
就近交付 高质量网络 cookie-free 突破并发限制



预加载 HTTP keep-alive chunked编码 HTTP/2 HTTPS优化







### 解析和渲染

- 不同格式解码速度不一样
- 图像重新取样需要花时间
- 画幅越大性能越差
- DOM越复杂越慢
- 样式越复杂越慢
- 特定样式非常耗性能







# 体验







## 不流畅是什么体验

- 慢卡
- 没反应
- 不知道点到了没有





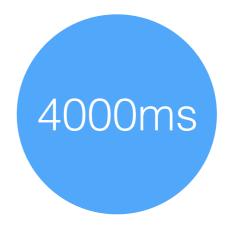


## 时间

100ms

可感知时间





反应时间 产生情感时间







# 时间



帧时间





- 帧运算时间不超过16ms
- requestAnimationFrame()







- 及时响应 给出交互反馈
- 计算逻辑拆开 分片运行
- 使用worker
- 干掉300ms点击延迟







- 明确的信息反馈
- loading态
- 控制动画时长







- 时间预期
- 进度条





The End