## 每日一题 - 页面注入50万个li怎么做提升性能

### 信息卡片

* 时间：2019-07-26
* tag：性能优化 开放问题

### 题目描述

页面注入50万个li怎么做提升性能？

### 思路

50万个li是基本不可能同时在页面上展现的。 这个时候我们需要具体地限定一下问题范围。

我们不妨假设一个li的宽度为100%， 高度为20px; 屏幕高度为1000px. 那么也就说理论上，一个屏幕所能 容纳的最大li个数为为1000 / 20 = 50个。

因此一个简单且直观的想法是，在必要的时候我们再去DOM中插入。 我们可以利用“视口检测”技术判断当前是否应该去插入li。但是这里有个问题， 就是如果用户滚动太快怎么办？ 不在视口的DOM需不需要回收？ 如果需要回收，我们应该怎么回收？ 这些都需要大家去思考。大家可以查一下相关的资料。

由于插入50个li，如果一个一个插入肯定会不断重排，性能会很差。 我们可以考虑使用[createDocumentFragment](https://developer.mozilla.org/zh-CN/docs/Web/API/Document/createDocumentFragment) 来减少重排的次数。

我们创建50个li的过程是否会涉及到后端请求，如果涉及到了，我们一次请求多少比较合适？ 我们需不需要进行预先请求。 如果本地网络IO很差的话，如何尽可能提高用户体验？

其实性能是一个相当主观的东西， 比如React使用了fiber架构，它的总运行时间 甚至比以前还要长，但是给用户的感觉是“性能好”。因此性能优化其实还有很多 讨巧的点可以考虑，大家不妨发散一下脑洞。

### 扩展1

假如创建li的过程非常耗时，我们应该怎么去应对？

扩展： 前端如何知道本机计算性能很差？

这个时候，我们可以考虑使用webworker，将计算量比较大的任务放到别的线程去执行， 然后利用线程通信，获取返回的结果，从而避免由于JS运算导致主线程阻塞。

### 扩展2

如果本机计算性能很差，我们怎么应对？

这个时候，我们可以考虑使用类似RIC(requestIdleCallback) 或者RAF(requestanimationframe) 这样的API。 保证用户正常交互的同时，完成我们的插入任务。

上面的API在低版本浏览器不存在，我们如何hack？