

## 4. JavaScript性能

# JavaScript内存管理

- 1 当需要的时候分配内存
- 2 对内存进行读写操作
- 3 当上面分配的内存不再需要的时候，将他们释放掉

# 变量查找优化

- 变量声明带上var
- 慎用全局变量
- 缓存重复使用的全局变量
- 避免使用with

# 核心语法优化

- 通过原型优化方法定义
- 避开闭包陷阱
- 避免在循环中使用try-catch
- 使用for代替for...in...遍历数组
- 使用原始操作代替方法调用 `getElementById > $()`
- 数组操作取代操作字符串

# 脚本加载优化

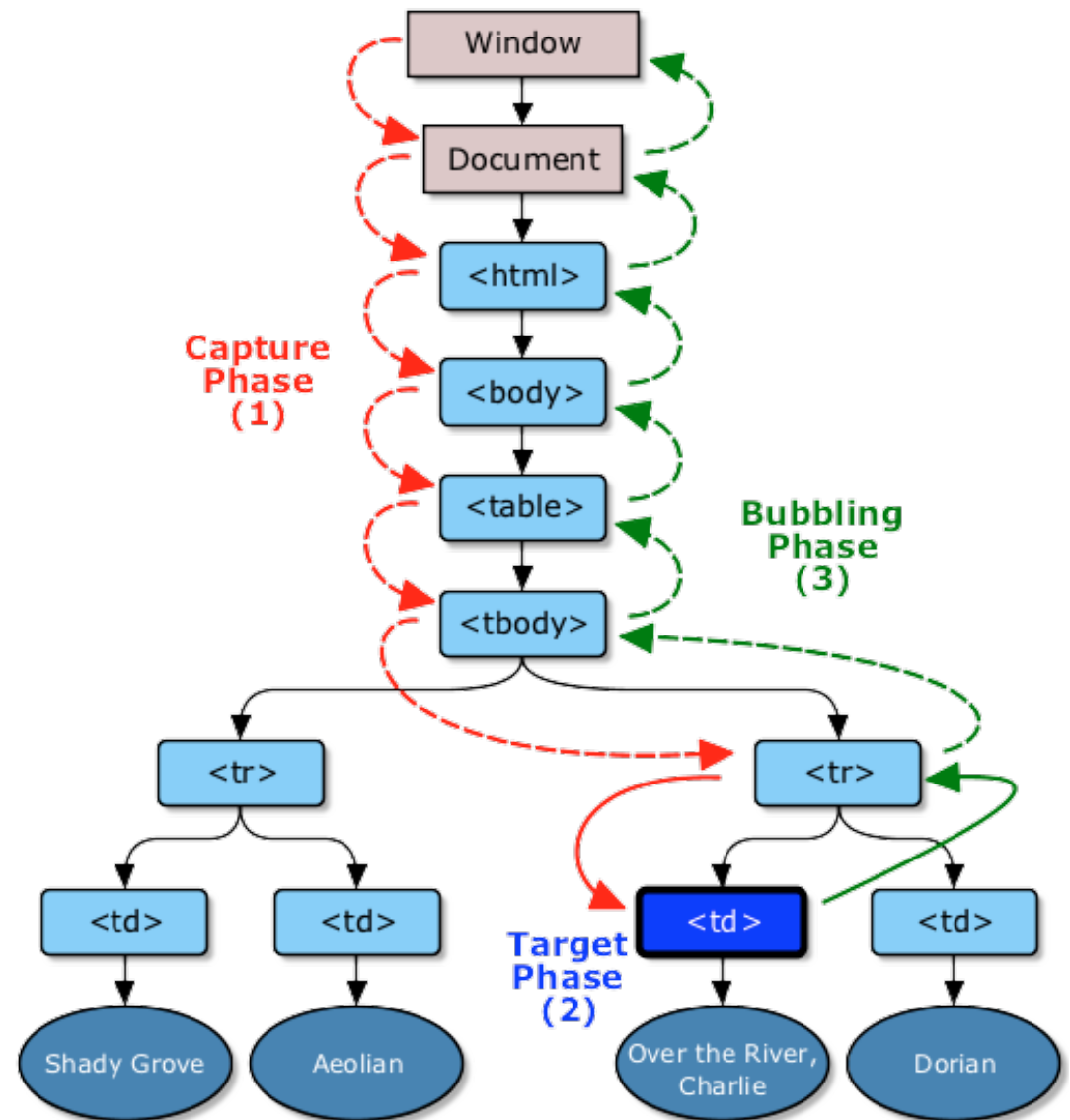
- 使用工具精简脚本
- 启用Gzip压缩
- 设置Cache-Control和Expires头
- 异步加载脚本

# DOM操作优化

- 减少DOM元素数量
- 优化CSS样式转换
- 优化节点添加
- 优化节点修改
- 减少使用元素位置操作
- 避免遍历大量元素

# 事件优化

- 使用事件代理



# 动画优化

- 设置动画元素为absolute或fixed
- 使用一个timer完成多个元素动画

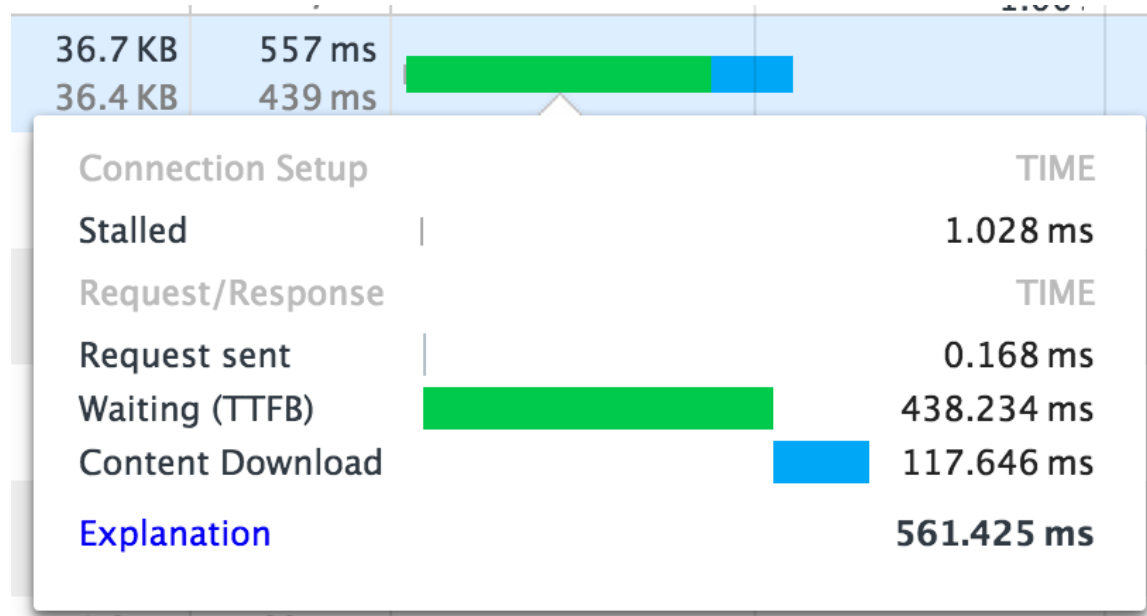


# 移动端JavaScript性能

- JavaScript放在底部
- 尽量用原生JavaScript
- 异步加载

# 知道如何对前端进行性能优化

- 网页加载最大的瓶颈在前端；
- 前端最大的瓶颈在HTTP。



# 前端要了解的HTTP

- header 字段
- 编码方式
- restful API
- 设置缓存，缓存失效
- .....