Script Profiling Cheat Sheet

FLAMEGRAPH Long bars = expensive script time Big flame gaps = long exclusive time Big timeline gaps = possible async code, fetches, etc. Lots of small repeated calls = possible excessive loop

Total time spent on line SOURCE VIEW 27 function test() { const prime = calculatePrime() }

Steps

- 1. Use a clean browser profile and open the Performance dev tool
- 2. Open a scratchpad and your test checklist/SOP
- 3. Close any intensive apps that might affect measurements
- 4. Profile the flow, running through your steps
 - to Profile
 - to Reload and Profile
- 5. Analyze the flamegraph
 - Record findings, but avoid diagnosing
- 6. Save profile in shared storage to refer to later

