Lighthouse Web page quality audits

CONCEPTS

> Default run is mobile emulation with simulated slow 4G and a slowed CPU, so scores are lower than what a fast laptop feels. Use --preset=desktop for desktop.

Gather load page in Chrome, collect artifacts
↓ artifacts
Audit run audits on artifacts
↓ scores
Categories performance, a11y, SEO ...
↓ reporters
Report html / json / csv

> Scores vary between runs. Close other tabs, run several times and compare medians before trusting a small change.

> Needs Node 22+ and an installed Chrome or Chromium. Point CHROME_PATH at a specific binary.

RUNNING

`lighthouse <url>`
Audit a URL, write an HTML report to the cwd

`lighthouse <url> --view`
Open the HTML report when done

`lighthouse --url-file=<file> --output-path=<dir>`
Audit every URL listed in a file

OUTPUT

`--output=<html|json|csv>`
Pick the report format

`--output=json --output=html`
Write several formats in one run

`--output-path=<file|dir|stdout>`
Where to write the report

`--view`
Open the HTML report in a browser

CATEGORIES & AUDITS

`--only-categories=<a,b>`
Run only these categories

`--only-audits <id> ...`
Run only the named audits

`--skip-audits <id> ...`
Run everything except these audits

`--preset=<name>`
Built-in config: perf, desktop, experimental

`--extra-headers <json|file>`
Send extra HTTP headers (cookie, auth)

EMULATION & THROTTLING

`--preset=desktop`
Desktop viewport, desktop metrics scoring

`--throttling-method=<method>`
Throttling: simulate, devtools, provided

`--throttling.cpuSlowdownMultiplier=<n>`
Set CPU slowdown

CHROME

`--chrome-flags="<flags>"`
Pass flags to the launched Chrome

`--chrome-flags="--headless"`
Run without a visible window

`--chrome-flags="--no-sandbox"`
Needed in many containers and CI

`CHROME_PATH=<path> lighthouse <url>`
Use a specific Chrome binary

GATHER & AUDIT SEPARATELY

> For CI budgets and history across commits, use Lighthouse CI (GoogleChrome/lighthouse-ci) instead of scripting the CLI.