CSuite: Local benchmarking help for V8 performance analysis

CSuite helps you make N averaged runs of a benchmark, then compare with a different binary and/or different flags. It knows about the "classic" benchmarks of SunSpider, Kraken and Octane, which are still useful for investigating peak performance scenarios. It offers a default number of runs, by default they are:

- SunSpider - 100 runs
- Kraken - 80 runs
- Octane - 10 runs

Usage

Say you want to see how much optimization buys you:

```
./csuite.py kraken baseline ~/src/v8/out/d8 -x="--noopt"
./csuite.py kraken compare ~/src/v8/out/d8
```

Suppose you are comparing two binaries, and want a quick look at results. Normally, Octane should have about 10 runs, but 3 will only take a few minutes:

```
./csuite.py -r 3 octane baseline ~/src/v8/out-master/d8
./csuite.py -r 3 octane compare ~/src/v8/out-mine/d8
```

You can run from any place:

```
../../somewhere-strange/csuite.py sunspider baseline ./d8
../../somewhere-strange/csuite.py sunspider compare ./d8-better
```

Note that all output files are created in the directory where you run from. A `_benchmark_runner_data` directory will be created to store run output, and a `_results` directory as well for scores.

For more detailed documentation, see:

```
./csuite.py --help
```

Output from the runners is captured into files and cached, so you can cancel and resume multi-hour benchmark runs with minimal loss of data/time. The -f flag forces re-running even if these cached files still exist.