Unit Tests

Run (inside Electron)

```
./scripts/test.[sh|bat]
```

All unit tests are run inside a electron-browser environment which access to DOM and Nodejs api. This is the closest to the environment in which VS Code itself ships. Notes:

- use the `--debug` to see an electron window with dev tools which allows for debugging
- to run only a subset of tests use the `--run` or `--glob` options
- use `yarn watch` to automatically compile changes

For instance, `./scripts/test.sh --debug --glob **/extHost*.test.js` runs all tests from `extHost` -files and enables you to debug them.

Run (inside browser)

```
yarn test-browser --browser webkit --browser chromium
```

Unit tests from layers `common` and `browser` are run inside `chromium`, `webkit`, and (soon'ish) `firefox` (using playwright). This complements our electron-based unit test runner and adds more coverage of supported platforms. Notes:

- these tests are part of the continuous build, that means you might have test failures that only happen with webkit on *windows* or *chromium* on linux
- you can run these tests locally via yarn `test-browser --browser chromium --browser webkit`
- to debug, open `<vscode>/test/unit/browser/renderer.html` inside a browser and use the `?m=<amd_module>` -query to specify what AMD module to load, e.g `file:///Users/jrieken/Code/vscode/test/unit/browser/renderer.html?m=vs/base/test/common/strings.test` runs all tests from `strings.test.ts`
- to run only a subset of tests use the `--run` or `--glob` options

Note: you can enable verbose logging of playwright library by setting a `DEBUG` environment variable before running the tests (https://playwright.dev/docs/debug#verbose-api-logs)

Run (with node)

```
yarn run mocha --ui tdd --run src/vs/editor/test/browser/controller/cursor.test.ts
```

Coverage

The following command will create a `coverage` folder in the `.build` folder at the root of the workspace:

OS X and Linux

```
./scripts/test.sh --coverage
```

Windows

```
scripts\test --coverage
```