

Misc tips

pyright is not dealing with import paths correctly for functional2

I don't know why it's doing this but it seems to be assuming that the functional2 test suite is relative to the workspace root.

You can jam the following into `:CocLocalConfig` for vim, and probably the workspace config for VSCode and it should fix it.

```
{
  "python.analysis.include": ["tests"],
  "python.analysis.extraPaths": ["tests"],
}
```

buildbot user style to make the pulsing pills bearable

```
@keyframes pulse_animation {
  0% { transform:scale(.9) }
  50% { transform:scale(1) }
  to { transform:scale(.9) }
}

.pulse {
  animation-duration: 10s !important;
}
```

FIXME: someone should PR this, now that we [have the ability to patch buildbot](#)

run all lix vm tests locally

```
tests=$(
  nix eval --json --impure \
```

```

--apply '
  let f = n: t:
    if __isAttrs t
    then (if t.type or "" == "derivation"
          then (if t.system == __currentSystem
                then [ n ]
                else [])
          else __concatMap (m: f "${n}.${m}" t.${m}) (__attrNames t))
    else [];
  in f ".#hydraJobs.tests"
' \
.#hydraJobs.tests \
| jq -r '.[0]'
)

nix build --no-link -L ${tests[@]}

```

check out current patchset of a cl by git alias

put this in a gitconfig that can configure aliases:

```

[alias]
cocl = !\
ps=${\
ssh $(git config remote.origin.gerriturl) \
gerrit query --format=json --current-patch-set $1 \
| jq -sr .[0].currentPatchSet.ref \
) && git fetch origin $ps && git checkout FETCH_HEAD && true

```

then run as `git cocl <cl-number>`. needs a `git config remote.origin.gerriturl gerrit.linux.systems`, or some other url that ssh can connect to. (could've extracted it from the remote url but we didn't want to do that much shell)

git stuff

git-revise

[git-revise](#) is a cool tool for splitting and shuffling commits in-memory without breaking your working tree. it's great.

It also has some broken stuff with respect to gerrit commit-msg hooks. However, this can be fixed (this is opt-in because some commit-msg hooks make unsound assumptions but the gerrit one should be fine):

```
# allow gerrit git hooks to run on git-revise  
[revise "run-hooks"]  
    commit-msg = true
```

Making `git clean` clean the stuff that isn't removed by `make clean`

If you don't want to use `git clean -x` to remove all git-ignored stuff, but want to remove things that are generated in Lix's build process but aren't removed by `make clean`, apply this patch: [no-ignore-not-cleaned.patch](#)

Revision #4

Created 25 March 2024 04:06:35 by jade

Updated 24 January 2025 22:29:54 by jade