

Collapsing Functions

What is a "Collapsing Function"?

A collapsing function is a function whose behavior changes depending upon the circumstances under which it's run. Function collapsing is useful when you find yourself repeatedly checking a variable whose value never changes.

How do I make a function collapse?

Function collapsing requires some static feature in the environment. A common example is a script that gives the user the option of having "verbose" output.

```
#!/bin/bash

[[ $1 = -v || $1 = --verbose ]] && verbose=1

chatter() {
    if [[ $verbose ]]; then
        chatter() {
            echo "$@"
        }
        chatter "$@"
    else
        chatter() {
            :
        }
    fi
}


echo "Waiting for 10 seconds."
for i in {1..10}; do
    chatter "$i"
    sleep 1
done
```

How does it work?

The first time you run `chatter()`, the function redefines itself based on the value of `verbose`. Thereafter, `chatter` doesn't check `$verbose`, it simply is. Further calls to the function reflect its collapsed nature. If `verbose` is unset, `chatter` will echo nothing, with no extra effort from

the developer.

More examples

 Add more examples!

```
# Somewhat more portable find -executable
# FIXME/UNTESTED (I don't have access to all of the different version
s of find.)
# Usage: find PATH ARGS -- use find like normal, except use -executab
le instead of
# various versions of -perm /+ blah blah and hacks
find() {
    hash find || { echo 'find not found!'; exit 1; }
    # We can be pretty sure "$0" should be executable.
    if [[ $(command find "$0" -executable 2> /dev/null) ]]; then
        unset -f find # We can just use the command find
    elif [[ $(command find "$0" -perm /u+x 2> /dev/null) ]]; then
        find() {
            typeset arg args
            for arg do
                [[ $arg = -executable ]] && args+=(-perm /u+x) || args+=("$ar
g")
            done
            command find "${args[@]}"
        }
    elif [[ $(command find "$0" -perm +u+x 2> /dev/null) ]]; then
        find() {
            typeset arg args
            for arg do
                [[ $arg = -executable ]] && args+=(-perm +u+x) || args+=("$ar
g")
            done
            command find "${args[@]}"
        }
    else # Last resort
        find() {
            typeset arg args
            for arg do
                [[ $arg = -executable ]] && args+=(-exec test -x {} \; -prin
t) || args+=("$arg")
            done
            command find "${args[@]}"
        }
    fi
    find "$@"
}
```



```
#!/bin/bash
# Using collapsing functions to turn debug messages on/off

[ "--debug" = "$1" ] && dbg=echo || dbg=:

# From now on if you use $dbg instead of echo, you can select if mess
ages will be shown

$dbg "This message will only be displayed if --debug is specified at
the command line"
```

Discussion

 [howto/collapsing_functions.txt](#)  Last modified: 2015/08/08 20:26 by bill_thomson

This site is supported by Performing Databases - your experts for
database administration

Bash Hackers Wiki



Except where otherwise noted, content on this wiki is licensed under the following license:
GNU Free Documentation License 1.3