## Sharing State Between Callbacks

* In some apps, you may have multiple callbacks that depend on expensive data processing tasks like making SQL queries, running simulations, or downloading data.

Rather than have each callback run the same expensive task, you can have one callback run the task and then share the results to the rest of the callbacks.

* *global* variables will break your app
  + Dash is designed to work in multi-user environments where multiple people may view the application at the same time and will have **independent sessions**.

→ If your app uses modified *global* variables, then one user's session could set the variable to one value which would affect the next user's session.

* + Dash is also designed to be able to run with **multiple python workers** so that callbacks can be executed in parallel.

$ gunicorn --workers 4 app:server

(app refers to a file named app.py and server refers to a variable in that file named server: server = app.server).

→ When Dash apps run across multiple workers, their memory is *not shared*. This means that if you modify a global variable in one callback, that modification will not be applied to the rest of the workers.

* There are three main places to store this data:

1 - In the user's browser session

2 - On the disk (e.g. on a file or on a new database)

3 - In a shared memory space like with Redis

Example 1 - Storing Data in the Browser with a Hidden Div