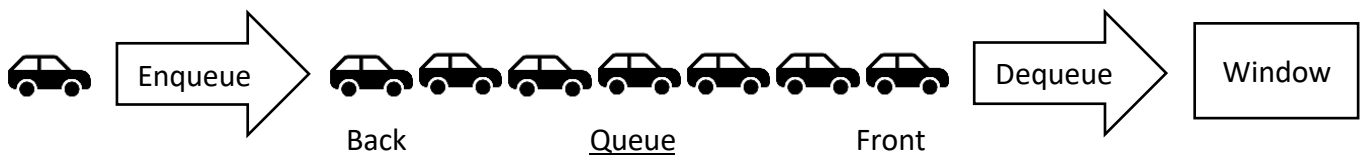
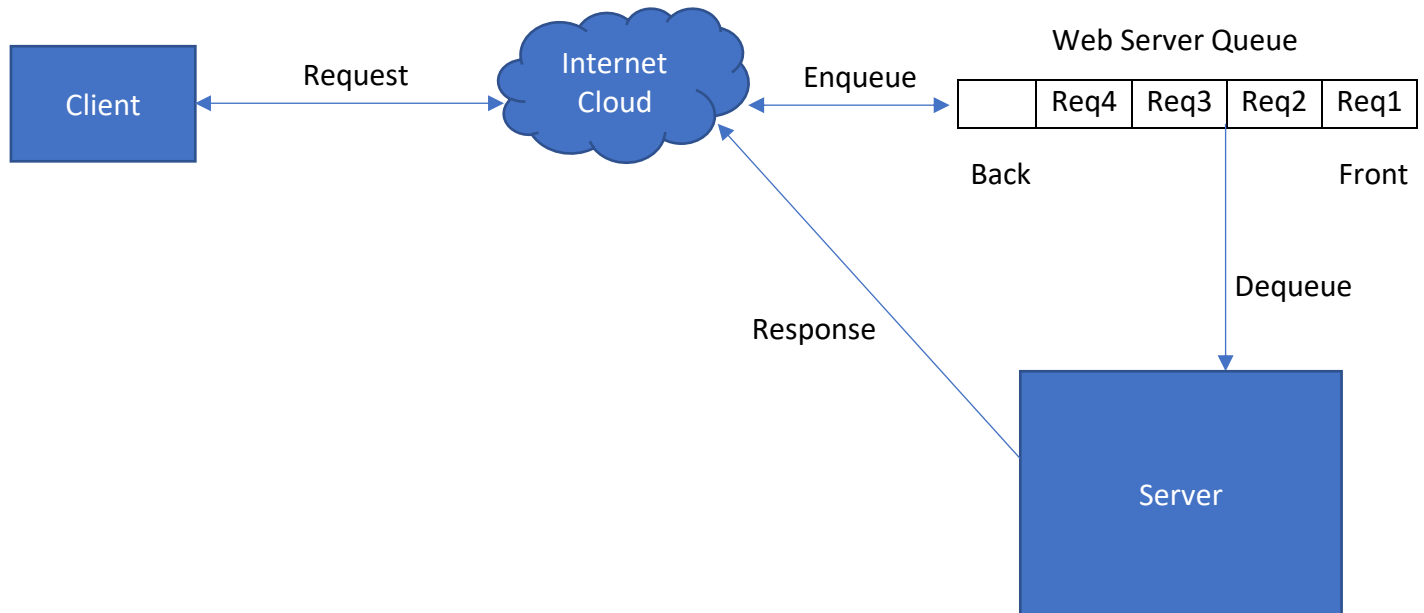


## Final Project: Query Tutorial

- Introduction
  - A query is a simple 2-dimensional data structure which resembles a list, or a string of data items. What makes it special is that for queries, data is always processed in the order in which it was received.
- Drive-Through Queues
  - This type of queue resembles a drive-through for a fast-food restaurant. In a drive-through there is a line of cars waiting to get their food, each one waiting their turn. It follows a first come-first-serve basis, with the first car to arrive being the first to be served, the second car to be the second to be served, and so on. When a car enters the line, they are **enqueued** and sent to the **back** of the line. The car at the window receiving its food is at the **back** of the line, and when they leave, they are **dequeued**. This is like how data is processed and organized in a Drive-Through Queue.



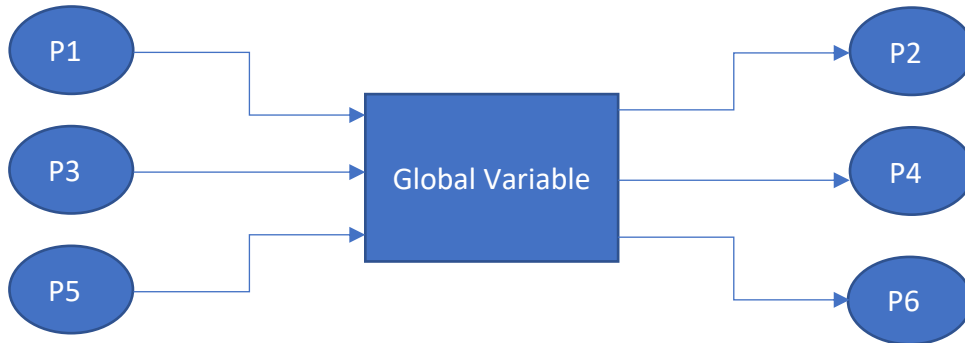
- Web Server Queue
  - Web servers receive requests from clients. Since processing and returning data to clients takes time, and web servers can receive thousands of requests every minute, web servers cannot respond to every request immediately. To organize the requests and make sure each one is handled properly, they are placed into a queue and answered in the order they are received.



- Reader/Writer Queue
  - Sometimes a program will have multiple processes trying to read or write to a global (shared throughout the entire program) variable at the same time. If not managed correctly, this can lead to confusion and errors with editing of the variable. To create an orderly way of editing a shared variable, we use Reader/Writer Queues.

Writing Processes

Reading Processes



- To sort through the various items which want to read/write to the global variable, requests to read/write are placed in a queue. A process is enqueued when requesting to read/write to the global variable. It is dequeued from the list when it arrives to the front of the queue and then given permission to read/write; after it has finished, the next process is dequeued, and so on.
- Example with Python code (in code files)
  - Problem to solve (in code files)