ECOM 5416

Parallel and Distributed Systems

Python + Memcached

When writing Python applications, caching is important. Using a cache to avoid recomputing data or accessing a slow database can provide you with a great performance boost.

What is Memcached?

Free & open source, high-performance, distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load.

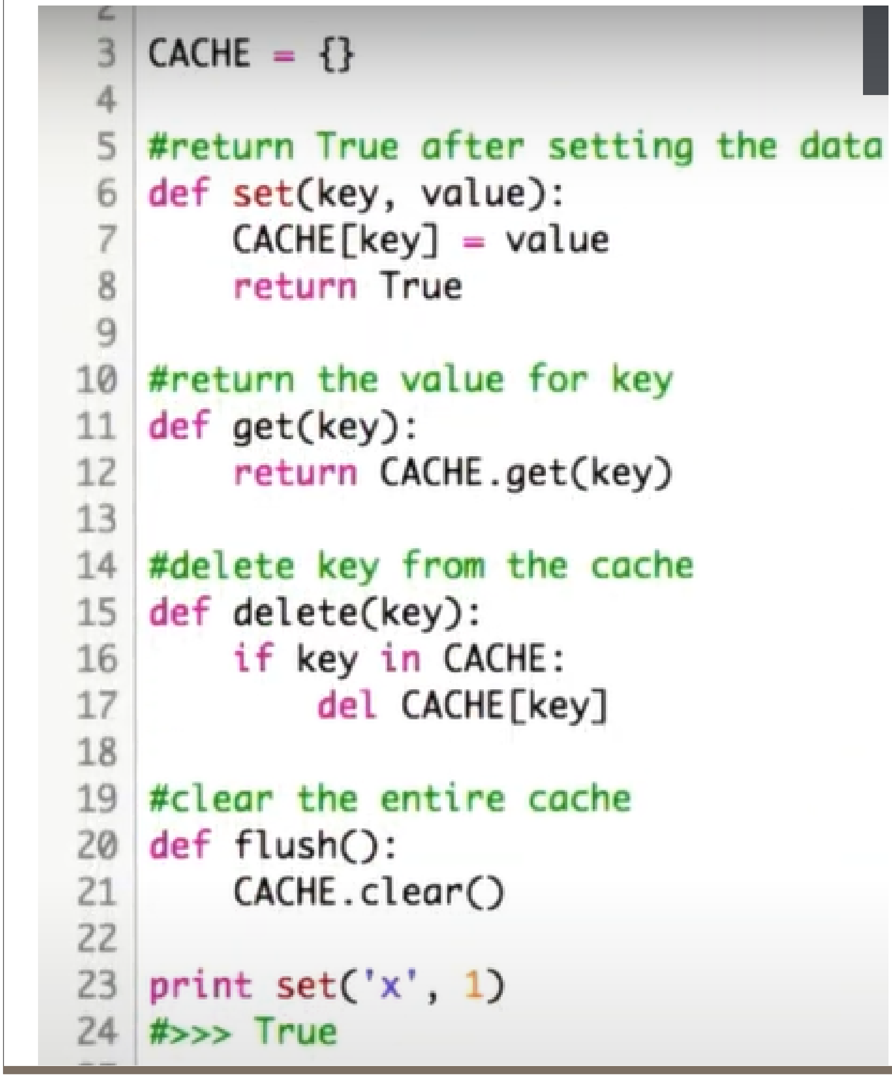
Memcached is an in-memory key-value store for small chunks of arbitrary data (strings, objects) from results of database calls, API calls, or page rendering.

Memcached is simple yet powerful. Its simple design promotes quick deployment, ease of development, and solves many problems facing large data caches. Its API is available for most popular languages.

Python + Memcached

The pseudocode below represents a typical memcache request:

```
def get_data():
    data = memcache.get('key')
    if data is not None:
        return data
    else:
        data = query_for_data()
        memcache.add('key', data, 60)
    return data
```



```
CACHE = {}

#return True after setting the data
def set(key, value):
    CACHE[key] = value
    return True

#return the value for key
def get(key):
    return CACHE.get(key)

#delete key from the cache
def delete(key):
    if key in CACHE:
        del CACHE[key]

#clear the entire cache
def flush():
    CACHE.clear()

print set('x', 1)
#>>> True
```