**Get vs TOP vs POP in Python and C++**

It depends on which data structure and operation you’re referring to. In Python, the term "get" is used in different contexts and doesn’t always behave like C++’s "pop" method. Here are a few examples:

1. **Python’s PriorityQueue vs. C++’s std::priority\_queue:**
   * In Python’s queue.PriorityQueue, the method get() removes and returns the smallest item from the queue.
   * In C++’s std::priority\_queue, the pop() method removes the top element but does **not** return it—you must use top() to inspect the element before calling pop().
2. **Python List Methods:**
   * Python lists have a pop() method (e.g., my\_list.pop()) which removes and returns an element (by default, the last element). This is closer in behavior to C++’s pop\_back() for containers like std::vector (when used as a stack).
   * Python lists do not have a get() method.
3. **Python Dictionary Methods:**
   * Python dictionaries have a get(key, default) method that retrieves a value for a key if it exists (without removing it). There isn’t a direct equivalent in C++ STL maps. In C++, you might use find() to check for a key, then access its value, but that operation does not remove the key from the map.
   * Dictionaries also have a pop(key, default) method that removes the key from the dictionary and returns its value, which is closer to what you might think of as “popping” from a map (though C++ STL maps don’t have a built-in pop method).

**In summary:**

* **get() in a Python PriorityQueue** removes and returns an element (which is somewhat similar to combining C++’s top() and pop() from std::priority\_queue, but with the key difference that C++’s pop() does not return the element).
* **get() in a Python dictionary** is used to retrieve a value without removing it.
* **pop() in Python lists** removes and returns an element, which is more directly comparable to C++ container pop operations.

So, there isn’t a one-to-one equivalence between Python’s get() and C++’s pop()—they serve different roles depending on the context.