Priority Queue in Python

A priority queue is an abstract data type similar to a regular queue, but with an additional feature: each element in the queue has a priority associated with it. Elements with higher priority are dequeued before elements with lower priority.

Common Operations in a Priority Queue:

1. Insertion (enqueue): Add an element with an associated priority.
2. Deletion (dequeue): Remove and return the element with the highest priority.
3. Peek: Return the element with the highest priority without removing it from the queue.

Python Implementation of a Priority Queue:

```
import heapq

class PriorityQueue:
    def __init__(self):
        self._queue = []
        self._index = 0

    def push(self, item, priority):
        heapq.heappush(self._queue, (-priority, self._index, item))
        self._index += 1

    def pop(self):
        return heapq.heappop(self._queue)[-1]

    def peek(self):
        if self._queue:
            return self._queue[0][-1]
        return None

    def is_empty(self):
        return len(self._queue) == 0
```

Example usage:

```
if __name__ == '__main__':
    pq = PriorityQueue()
    pq.push("Task 1", priority=3)
    pq.push("Task 2", priority=1)
    pq.push("Task 3", priority=2)

    print(f"Top Priority Task: {pq.peek()}")  # Task 1

    while not pq.is_empty():
        print(pq.pop())  # Task 1, Task 3, Task 2
```

Explanation:

1. PriorityQueue Class:
   - __init__: Initializes an empty priority queue.
   - push(item, priority): Adds an item to the priority queue with a specified priority.

- pop(): Removes and returns the item with the highest priority.
- peek(): Returns the item with the highest priority without removing it.
- is_empty(): Checks if the priority queue is empty.

2. Heapq Module:

   - Python's heapq module provides an implementation of the heap queue algorithm, also known as the priority queue algorithm.

Real-World Applications of Priority Queues:

- Task Scheduling: Operating systems often use priority queues for managing processes.
- Dijkstra's Algorithm: A priority queue is used in graph algorithms to find the shortest path.
- A* Algorithm: Priority queues are used in pathfinding algorithms for exploring nodes based on their cost.