How to Monitor Performance on macOS: Discover How to Use Built-in Tools and Scripts

Monitoring performance is crucial for maintaining the health and efficiency of any computer system. For macOS users, understanding how to monitor system performance can help in diagnosing issues, optimizing resource usage, and ensuring smooth operation. This article will guide you through various methods and tools available in the macOS environment for monitoring performance, including built-in utilities and command-line tools.

Examples:

1. **Activity Monitor:** The most straightforward way to monitor performance on macOS is by using the Activity Monitor. This built-in utility provides real-time data on CPU, memory, energy, disk, and network usage.

   - **How to Open Activity Monitor:**

     1. Open Finder.
     2. Go to the Applications folder.
     3. Navigate to the Utilities folder.
     4. Double-click on "Activity Monitor."

   - **Using Activity Monitor:**

     - **CPU Tab:** Shows the processes consuming the most CPU resources.
     - **Memory Tab:** Displays memory usage and helps identify memory leaks.
     - **Energy Tab:** Useful for monitoring energy consumption, especially on laptops.
     - **Disk Tab:** Provides information on disk read/write operations.
     - **Network Tab:** Shows data sent and received over the network.

2. **Terminal Commands:** For more advanced users, macOS provides several command-line tools for performance monitoring:

   - **top:** Displays a dynamic, real-time view of system processes.

     ```
     top
     ```

   - **vm_stat:** Provides information about virtual memory usage.

     ```
     vm_stat
     ```

   - **iostat:** Reports CPU and I/O statistics.

```
    iostat
    ```

  - **netstat:** Shows network statistics and connections.

    ```
    netstat
    ```

3. **Custom Scripts:** You can create custom scripts to automate performance monitoring. Here's an example of a simple script that logs CPU and memory usage to a file:

   ```
   #!/bin/bash
   while true; do
      echo "$(date)" >> performance.log
      top -l 1 | head -n 10 >> performance.log
      vm_stat >> performance.log
      echo "-----------------------------" >> performance.log
      sleep 60
   done
   ```

   - Save the script as monitor_performance.sh.
   - Make it executable:

     ```
     chmod +x monitor_performance.sh
     ```

   - Run the script:

     ```
     ./monitor_performance.sh
     ```

   This script logs performance data every minute, which can be useful for later analysis.