Debugging in Visual Studio Code

Overview

- Configure debugging for various project types.
- Utilize VS Code’s debugging interface effectively.
- Debug Python applications in diverse environments.

Configuring Debugging

Simple Scripts

1. Open your script in VS Code.
2. Create a launch configuration:
    - Open the **Run and Debug** tab.
    - Click **Create a launch.json file**.
    - Select the environment (e.g., Python).
3. Customize the `launch.json` as needed.

Django or Flask Applications

1. Install extensions:
    - **Python** (by Microsoft).
    - **Django** or **Flask** support, if needed.

VS Code’s Debugging Interface

Key Features

- **Breakpoints**: Pause execution at specific lines.
- **Variables**: Inspect variable states dynamically.
- **Call Stack**: Navigate the order of function calls.
- **Watch**: Monitor expressions in real-time.
- **Debug Console**: Execute commands during debugging.

Setting Breakpoints

1. Click in the margin next to a line number.
2. Conditional breakpoints:
    - Right-click a breakpoint.
    - Add conditions like `i == 5`.

Attaching to Running Processes

Steps

1. Identify the process ID (PID):
    - Use `ps` (Linux/Mac) or Task Manager (Windows).
    - Alternatively, use a debugger tool.
2. Create a launch configuration:
    - Set `processId` in `launch.json`.
3. Start debugging:
    - Open **Run and Debug**.
    - Attach to the running process.

Debugging Python Applications

Local Environment

1. Ensure Python is installed.
2. Configure the virtual environment:
    - Activate it.
    - Install dependencies (e.g., Flask, Django).

Remote Debugging

1. Install **ptvsd** or **debugpy** for remote access.
2. Configure the target application to accept debugger connections.
3. Attach to the remote debugger in VS Code.

Summary

Key Takeaways

- Configure debugging for simple scripts, Django, and Flask.
- Utilize VS Code’s debugging tools:
  - Breakpoints
  - Attach to processes
  - Debug across environments.
- Streamline Python debugging locally and remotely.

Questions?

Thank you!