Download and install windows 11 at https//www.microsoft.com. create a bootable pen drive using rufus . Insert the bootable USB drive into your PC.. Restart your computer and boot from the USB drive (usually by pressing a key like F12 or Esc during startup to access the boot menu). Follow the prompts to begin the installation process. Choose your language, time, and keyboard preferences. Select **Install now**. Enter your product key if prompted (Windows 11 may activate automatically if your PC was previously running a genuine copy of Windows 10). Follow the on-screen instructions to complete the installation.

Install Development Tools: **Integrated Development Environment (IDE)**

Download and install vscode. Install Git for version control. Configure your Git username and email globally using:arduino

Copy code

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

**Command Line Tools:**

Install essential command-line tools i.e Chocolatey on Windows, apt on Ubuntu.

Programming Language Setup:

**Install Required Programming Languages:**

Use language-specific package managers (e.g., npm for Node.js, pip for Python, RubyGems for Ruby) or official installers.

Set environment variables such as PATH for executable binaries.

**Dependency Management:**

Set up package managers like npm, pip, Maven, or others relevant to your stack.

Create virtual environments or use containers (e.g., Docker) as needed.

Database Setup:

Download and Install MySQL.

Set up databases with initial configurations (root password, initial schema).

**Database Clients:**

Install database clients (e.g., MySQL Workbench, pgAdmin, Robo 3T) for GUI-based management.

Development Environment Customizations:

**IDE/Editor Settings:**

Customize IDE/editor settings, themes, and key bindings according to your preferences.

Configure linting, code formatting, and debugging settings.

**Shell Customizations:**

Customize your shell (e.g., Bash, Zsh) with aliases, prompts, and plugins (e.g., Oh My Zsh).

Security and Permissions:

**Firewall and Security Settings:**

Configure firewall settings to allow necessary ports for development tools and servers.

**User Permissions:**

Adjust file and directory permissions to ensure access for development tasks.

**Troubleshooting and Customizations:**

**Path and Environment Variables:**

Ensure correct setup of PATH and other environment variables.

Troubleshoot issues related to path conflicts or missing dependencies.

**Dependency Management Issues:**

Resolve dependency conflicts using version pinning or virtual environments.

Update package versions to fix bugs or security vulnerabilities.

**Networking and Firewall Issues:**

Troubleshoot network-related issues affecting package downloads or server connectivity.

Adjust firewall settings to allow local development server access.

**IDE/Editor Plugin Conflicts:**

Resolve conflicts between plugins causing performance issues or crashes.

Update plugins to their latest versions compatible with your IDE/editor.

**Version Control Problems:**

Address Git errors such as merge conflicts, detached HEAD state, or repository corruption.

Use Git commands and GUI tools to resolve conflicts and manage branches effectively.