**1. Linting syntactic error**

* Linting is the process of automatically checking your code for errors, bugs, or suspicious patterns.
* Example: It can warn you if you forgot a semicolon, used an undefined variable, or wrote unused code.  
  Helps improve **code quality**.

**2. Debugging**

* Debugging allows you to run your code step by step and find where it’s going wrong.
* You can set **breakpoints**, watch variables, and analyze errors.  
  Helps in **fixing bugs efficiently**.

**3. Autocompletion**

* Also called **IntelliSense**, this feature suggests code while you type.
* Example: If you type pri, it suggests print().  
   Saves time and reduces **typing mistakes**.

**4. Code Formatting**

* Automatically arranges your code in a neat and consistent style.
* Example: Proper indentation, spacing, and alignment.  
  Improves **readability and maintainability** of code.

**5. Unit Testing**

* A way to test small parts (units) of your program separately.
* Example: Testing if a function gives the right output for given inputs.  
   Ensures **correctness and reliability** of code.

**6. Code Snippets**

* Small reusable pieces of code you can insert quickly.
* Example: Typing for and hitting Tab could insert a complete for-loop template.  
  Speeds up coding with **predefined structures**.