1. Difference Between Stateless and Stateful Widgets

- **Stateless Widget**:
  - A widget that does not maintain any state.
  - It is immutable and cannot change once built.
  - Used when the UI remains constant throughout the widget's lifecycle.
- **Stateful Widget**:
  - A widget that can hold and update state.
  - It is mutable and can rebuild itself when the state changes.
  - Used when the UI changes dynamically based on user interaction or other factors.

---

2. Widget Lifecycle and State Management in Stateful Widgets

- The lifecycle of a **Stateful Widget** involves two main classes: the widget class and its associated state class.
- Key lifecycle methods include:
  - createState(): Creates the mutable state for the widget.
  - initState(): Called once when the state is initialized; used for setup.
  - didChangeDependencies(): Called when an inherited widget changes.
  - build(): Describes the widget’s UI; called every time the UI needs to be updated.
  - setState(): Notifies the framework that the internal state has changed and triggers a rebuild.
  - dispose(): Called when the widget is permanently removed; used for cleanup.

---

3. Five Common Flutter Layout Widgets

1. **Container**:
    - A versatile box widget used for styling, padding, margin, alignment, and background.
2. **Column**:
    - Arranges child widgets vertically in a single column.
3. **Row**:
    - Arranges child widgets horizontally in a single row.
4. **Expanded**:
    - Expands a child widget to fill the available space within a Row, Column, or Flex.
5. **Stack**:
    - Places widgets on top of each other, allowing overlapping of child widgets.