

Stateless vs Stateful in Dart (Flutter)

In Flutter, everything is a widget. Widgets describe how your UI should look, but some widgets need to change during the app's lifetime while others remain constant. That's where Stateless and Stateful widgets come in.

1. Stateless Widget

Definition: A widget that does not change once it's built. It is immutable – meaning, its properties can't change after creation.

Characteristics:

- No internal state to manage.
- Renders UI based only on the input data (constructor parameters).
- Fast and lightweight.

When to use:

- Static UI elements (e.g., text labels, icons, static images).
- No need to update UI during runtime.

2. Stateful Widget

Definition: A widget that can change its appearance in response to events, user interaction, or data updates.

It maintains a State object that stores mutable data.

Characteristics:

- Has two classes: the widget itself and its associated State class.
- Can rebuild UI when `setState()` is called.
- Used for dynamic content.

When to use:

- Forms, animations, counters, user interactions.
- UI that changes over time.

3. Main Differences Table

Feature	StatelessWidget	StatefulWidget
Mutability	Immutable	Mutable (via State)
Performance	Faster, lightweight	Slightly heavier
Lifecycle	Single build	Multiple rebuilds
Use case	Static UI	Dynamic, interactive UI

4. Summary:

- Stateless → Best for UI that never changes.
- Stateful → Best for UI that changes during the app lifecycle.

- Good Flutter apps mix both, using Stateless widgets where possible for performance and simplicity.