Flutter Gesture Detection: Advanced Touch Interactions

Subtitle: Learn how to detect and respond to user gestures like swipe, drag, pinch, and tap in Flutter apps.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app

Introduction

User interactions define app usability. Flutter provides **gesture detection widgets and listeners** to handle **taps, swipes, drags, and multi-touch gestures**, enabling highly interactive UI.

Step 1: Using GestureDetector

```
GestureDetector(
  onTap: () {
    print('Screen tapped');
  },
  onDoubleTap: () {
    print('Double tap detected');
  },
  onLongPress: () {
    print('Long press detected');
  },
  child: Container(
    color: Colors.blue,
    width: 200,
    height: 200,
  ),
)
```

Exercise: Detect single, double, and long presses on a colored box.

Step 2: Detect Swipes

```
GestureDetector(
  onPanUpdate: (details) {
```

```
    if(details.delta.dx > 0) print('Swipe right');
    if(details.delta.dx < 0) print('Swipe left');
  },
  child: Container(
    color: Colors.green,
    width: 200,
    height: 200,
  ),
)
```

Exercise: Detect horizontal swipe directions and log them.

Step 3: Pinch & Zoom Gesture

```
GestureDetector(
  onScaleUpdate: (details) {
    print('Scale: ${details.scale}');
  },
  child: Image.asset('assets/sample_image.png'),
)
```

Exercise: Allow a user to pinch-zoom on an image.

Step 4: Drag & Drop

```
Draggable(
  data: 'Flutter',
  feedback: Container(color: Colors.red, width: 100, height: 100),
  child: Container(color: Colors.yellow, width: 100, height: 100),
)
```

Exercise: Drag a widget and drop it in a target area.

Step 5: Mini Project

Build a **Flutter Interactive UI App**:

- Tap, double-tap, and long press on multiple widgets
- Pinch to zoom an image
- Drag and drop shapes into target areas
- Bonus: Combine gestures with animations for responsive UI

- Bonus: Combine gestures with animations for responsive UI

Key Takeaways

- Gesture detection enables **interactive and responsive apps**
- Use `GestureDetector` and `Draggable` for rich touch interactions
- Pinch, swipe, and drag gestures improve **UX for mobile apps**
- Mastering gestures is essential for **game-like or interactive Flutter apps**

Skipping gesture detection limits your app to **static, non-responsive interfaces**, reducing user engagement.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: haas.dev
Website Link: https://dev-roast-app.vercel.app
