Cheat Sheet: Working with Flutter

Interaction and forms in Flutter

Type	Description	Code example
ElevatedButton	Triggers actions on user interaction	`ElevatedButton( onPressed: () { print('Button Pressed'); }, child: Text('Press Me') );`
TextField	Collects textual input from users	`TextField( decoration: InputDecoration(hintText: 'Enter your username') );`
Form validation	Ensures input meets certain criteria	`Form( child: TextFormField( validator: (value) { if (value.isEmpty) return 'Cannot be empty'; } ) );`
GestureDetector	Handles more complex gestures like swipes	`GestureDetector( onTap: () { print('Tapped!'); }, child: Container(color: Colors.blue, width: 100, height: 100) );`

Routing in Flutter

Type	Description	Code example
Basic navigation	Manages screen transitions within an app	`Navigator.push( context, MaterialPageRoute(builder: (context) => NewScreen()) );`
Named routes	Uses named routes for navigation, which simplifies route management	`Navigator.pushNamed(context, '/details');`
Passing data between routes	Allows data to be sent between screens	`Navigator.push( context, MaterialPageRoute( builder: (context) => DetailScreen(data: 'Data from previous screen') ) );`

Implementing styles in Flutter

Type	Description	Code example
Text widget styling	Customizes the appearance of text elements	`Text( 'Hello Flutter', style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold, color: Colors.blue) );`
Styling buttons	Modifies the look of buttons to align with the app design	`ElevatedButton( style: ElevatedButton.styleFrom(primary: Colors.green), onPressed: () {}, child: Text('Submit') );`

Navigation in Flutter

Type	Description	Code example
Stack navigation	Manages a stack of screens	`Navigator.push( context, MaterialPageRoute(builder: (context) => SecondScreen()) );`
Tab navigation	Organizes content across different tabs	`TabBar( tabs: [Tab(icon: Icon(Icons.directions_car)), Tab(icon: Icon(Icons.directions_transit))] );`
Bottom navigation bar	Provides navigation between top-level views of an app through a series of tabs at the bottom of the screen	`BottomNavigationBar( items: <BottomNavigationBarItem>[ BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'), BottomNavigationBarItem(icon: Icon(Icons.business), label: 'Business'), ], onTap: (index) { /* Handle tap */ },`

		); `
Drawer navigation	Offers a slide-in menu from the edge of the screen for navigation purposes	Drawer(child: ListView(children: <Widget>[DrawerHeader(child: Text('Header')), ListTile(title: Text('Item 1'), onTap: () {}),],),);

Flutter widgets

Type	Description	Code example

DropdownButton	Allows users to select from a list of items in a dropdown menu	DropdownButton<String>(
		value: dropdownValue,
		onChanged: (String newValue) {
		setState(() {
		dropdownValue = newValue;
		});
		},
		items: <String>['One', 'Two', 'Three', 'Four']
		.map<DropdownMenuItem<String>>((String value) {
		return DropdownMenuItem<String>(
		value: value,
		child: Text(value),
		);
		}).toList(),
		)

Flexible widget	Allows child widgets to flex within available space in a row or column	Row(children: [Flexible(child: Container(color: Colors.red, height: 50)), Flexible(child: Container(color: Colors.blue, height: 50))])

ListView.builder	Constructs a list view dynamically with an item builder function	ListView.builder(
		itemCount: items.length,
		itemBuilder: (context, index) {
		return ListTile(title: Text(items[index]));
		},
		)

SnackBar	Provides lightweight feedback about an operation by showing a brief message	final snackBar = SnackBar(content: Text('Yay! A SnackBar!')); ScaffoldMessenger.of(context).showSnackBar(snackBar);

Dialogs	Displays material design dialogs	showDialog(
		context: context,
		builder: (BuildContext context) {
		return AlertDialog(
		title: Text('Alert!'),
		content: Text('You have been alerted.'),
		actions: <Widget>[
		TextButton(
		onPressed: () {
		Navigator.of(context).pop();
		},
		child: Text('Close'),
		),
		],
		);
		},
		);

Padding widget	Applies padding to its child widget	Padding(padding: EdgeInsets.all(8.0), child: Text('Hello World!'),)
GridView	Creates a scrollable grid of widgets arranged in rows and columns	GridView.count(crossAxisCount: 2, children: List.generate(20, (index) { return Center(child: Text('Item $index'),); }),)

ExpansionPanel	Provides a way to expand and collapse content	ExpansionPanelList(
		expansionCallback: (int index, bool isExpanded) {
		setState(() {
		items[index].isExpanded = !isExpanded;
		});
		},
		children: items.map<ExpansionPanel>((Item item) {
		return ExpansionPanel(
		headerBuilder: (BuildContext context, bool isExpanded) {
		return ListTile(title: Text(item.headerValue));
		},
		body: ListTile(title: Text(item.expandedValue)),
		isExpanded: item.isExpanded,
		);
		}).toList(),

		)
Custom paint	Provides a canvas on which to draw during the paint phase	`CustomPaint( painter: MyPainter(), )`
Theme data	Defines the color and typography values for a Material app	`ThemeData( primarySwatch: Colors.blue, visualDensity: VisualDensity.adaptivePlatformDensity, )`

Skills Network