

Fiche de Référence : Travailler avec Flutter

Interaction et formulaires dans Flutter

Type	Description	Exemple de code
ElevatedButton	Déclenche des actions lors de l'interaction de l'utilisateur	<pre>ElevatedButton(onPressed: () { print('Button Pressed'); }, child: Text('Press Me'));</pre>
TextField	Collecte des entrées textuelles des utilisateurs	<pre>TextField(decoration: InputDecoration(hintText: 'Enter your username'));</pre>
Validation de formulaire	Assure que l'entrée répond à certains critères	<pre>Form(child: TextFormField(validator: (value) { if (value.isEmpty) return 'Cannot be empty'; }));</pre>
GestureDetector	Gère des gestes plus complexes comme les glissements	<pre>GestureDetector(onTap: () { print('Tapped!'); }, child: Container(color: Colors.blue, width: 100, height: 100));</pre>

Routage dans Flutter

Type	Description	Exemple de code
Navigation de base	Gère les transitions d'écran au sein d'une application	<pre>Navigator.push(context, MaterialPageRoute(builder: (context) => NewScreen()));</pre>
Routes nommées	Utilise des routes nommées pour la navigation, ce qui simplifie la gestion des routes	<pre>Navigator.pushNamed(context, '/details');</pre>
Passer des données entre les routes	Permet d'envoyer des données entre les écrans	<pre>Navigator.push(context, MaterialPageRoute(builder: (context) => DetailScreen(data: 'Data from previous screen')));</pre>

Mise en œuvre des styles dans Flutter

Type	Description	Exemple de code
Stylisation du widget Text	Personnalise l'apparence des éléments de texte	<pre>Text('Hello Flutter', style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold, color: Colors.blue));</pre>
Styliser les boutons	Modifie l'apparence des boutons pour s'aligner avec le design de l'application	<pre>ElevatedButton(style: ElevatedButton.styleFrom(primary: Colors.green), onPressed: () {}, child: Text('Submit'));</pre>

Navigation dans Flutter

Type	Description	Exemple de code
Navigation par pile	Gère une pile d'écrans	<pre>Navigator.push(context, MaterialPageRoute(builder: (context) => SecondScreen()));</pre>
Navigation par onglets	Organise le contenu à travers différents onglets	<pre>AppBar(tabs: [Tab(icon: Icon(Icons.directions_car)), Tab(icon: Icon(Icons.directions_transit))]);</pre>
Barre de navigation inférieure	Fournit une navigation entre les vues de premier niveau d'une application via une série d'onglets en bas de l'écran	<pre>BottomNavigationBar(items: <BottomNavigationBarItem>[BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'), BottomNavigationBarItem(icon: Icon(Icons.business), label: 'Business'),], onTap: (index) { /* Handle tap */ },);</pre>

);
Navigation par tiroir	Propose un menu coulissant depuis le bord de l'écran pour des fins de navigation	<pre> Drawer(child: ListView(children: <Widget>[DrawerHeader(child: Text('Header')), ListTile(title: Text('Item 1'), onTap: () {}),],),); </pre>

Widgets Flutter

Type	Description	Exemple de code
DropDownButton	Permet aux utilisateurs de sélectionner dans une liste d'éléments dans un menu déroulant	<pre> 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(),) </pre>
Widget flexible	Permet aux widgets enfants de s'adapter à l'espace disponible dans une ligne ou une colonne	<pre> Row(children: [Flexible(child: Container(color: Colors.red, height: 50)), Flexible(child: Container(color: Colors.blue, height: 50))]) </pre>
ListView.builder	Construit une vue de liste de manière dynamique avec une fonction de création d'éléments	<pre> ListView.builder(itemCount: items.length, itemBuilder: (context, index) { return ListTile(title: Text(items[index])); },) </pre>
SnackBar	Fournit un retour léger sur une opération en affichant un message bref	<pre> final snackBar = SnackBar(content: Text('Yay! A SnackBar!')); ScaffoldMessenger.of(context).showSnackBar(snackBar); </pre>
Dialogues	Affiche des dialogues en design matériel	<pre> 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'),),],); },); </pre>
Widget de remplissage	Applique un remplissage à son widget enfant	<pre> Padding(padding: EdgeInsets.all(8.0), child: Text('Hello World!'),) </pre>
GridView	Crée une grille défilante de widgets disposés en lignes et en colonnes	<pre> GridView.count(crossAxisCount: 2, children: List.generate(20, (index) { return Center(child: Text('Item \$index'),); })),) </pre>
Panneau d'expansion	Fournit un moyen d'étendre et de réduire le contenu	<pre> 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(),) </pre>

)
Peinture personnalisée	Fournit une toile sur laquelle dessiner pendant la phase de peinture	<pre>CustomPaint(painter: MyPainter(),)</pre>
Données de thème	Définit les valeurs de couleur et de typographie pour une application Material	<pre>ThemeData(primarySwatch: Colors.blue, visualDensity: VisualDensity.adaptivePlatformDensity,)</pre>



Skills Network