<https://flutter.dev/docs/cookbook/forms/focus>

# Build a form with validation

<https://flutter.dev/docs/cookbook/forms/validation>

# Focus and text fields

class \_SomeWidgetState extends State<SomeWidget> {

final FocusNode \_focusNode = FocusNode();

@override

void initState() {

super.initState();

\_focusNode.addListener(() {

print("Has focus: ${\_focusNode.hasFocus}");

});

}

@override

Widget build(BuildContext context) {

return TextField(focusNode: \_focusNode);

}

@override

void dispose() {

\_focusNode.dispose();

super.dispose();

}

}

Example 2

<https://stackoverflow.com/questions/51909798/using-focusnode-in-textformfield-flutter>

void main() {

runApp(MaterialApp(home: PassCodeExample()));

}

class PassCodeExample extends StatelessWidget {

FocusNode f1 = FocusNode();

FocusNode f2 = FocusNode();

FocusNode f3 = FocusNode();

FocusNode f4 = FocusNode();

@override

Widget build(BuildContext context) {

return Scaffold(

body: Center(

child: Padding(

padding: const EdgeInsets.all(8.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

crossAxisAlignment: CrossAxisAlignment.center,

children: [

Flexible(

child: TextField(

focusNode: f1,

keyboardType: TextInputType.number,

onChanged: (String newVal) {

if (newVal.length == 1) {

f1.unfocus();

FocusScope.of(context).requestFocus(f2);

}

},

decoration: InputDecoration(border: OutlineInputBorder()),

),

),

Flexible(

child: TextField(

focusNode: f2,

keyboardType: TextInputType.number,

onChanged: (String newVal) {

if (newVal.length == 1) {

f2.unfocus();

FocusScope.of(context).requestFocus(f3);

}

},

decoration: InputDecoration(border: OutlineInputBorder()),

),

),

Flexible(

child: TextField(

focusNode: f3,

keyboardType: TextInputType.number,

onChanged: (String newVal) {

if (newVal.length == 1) {

f3.unfocus();

FocusScope.of(context).requestFocus(f4);

}

},

decoration: InputDecoration(border: OutlineInputBorder()),

),

),

Flexible(

child: TextField(

focusNode: f4,

keyboardType: TextInputType.number,

decoration: InputDecoration(border: OutlineInputBorder()),

),

),

]),

),

),

);

}

}