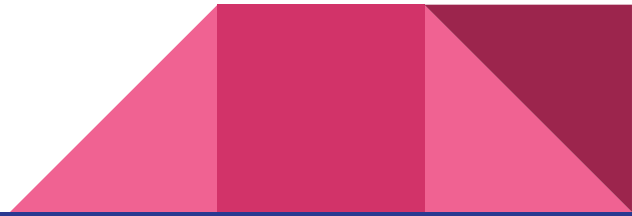
Programmazione Mobile

Nicola Noviello
nicola.noviello@unimol.it

Corso di Laurea in Informatica
Dipartimento di Bioscienze e Territorio
Università degli Studi del Molise
Anno 2023/2024

Lezione: Flutter e Dart - Funzionalità avanzate (parte prima)

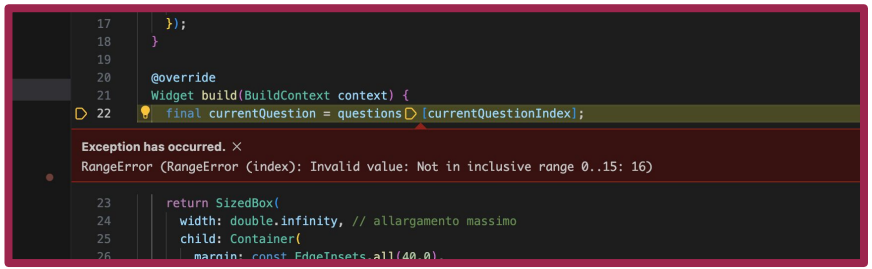
- Rendering dei contenuti in base a criteri condizionali
- Approfondimento del ciclo di vita di un Widget Stateful
- Espressioni ternarie e operatori condizionali
- Data Model
- Stili, Allineamento, Margini e Padding
- Integrazione con Package di terze parti
- Contenuti Scrollabili



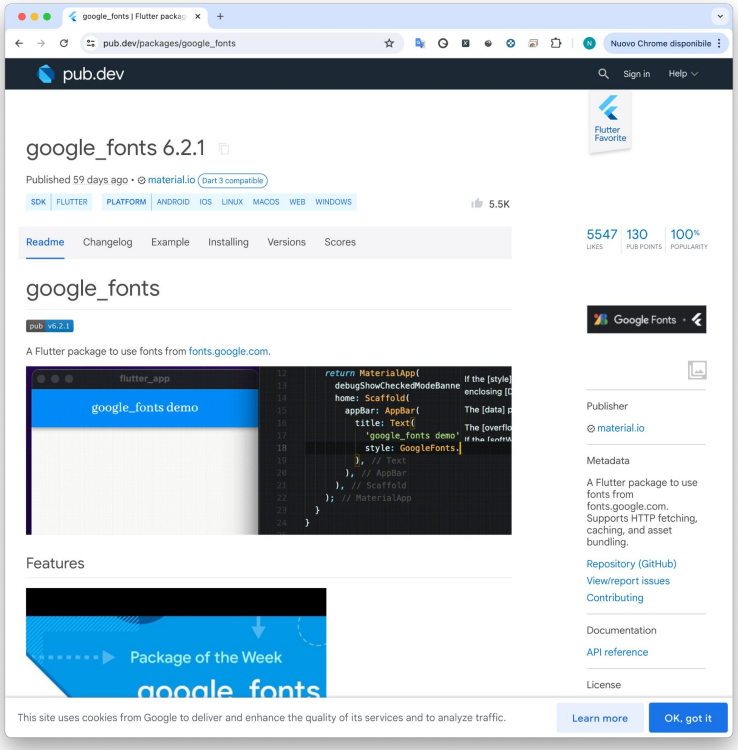
Dove eravamo

Integro una funzione in grado di cambiare domanda

```
            fontSize: 15.0,
            fontWeight: FontWeight.bold,
            color: Colors.white), // TextStyle
        textAlign: TextAlign.center,
    ), // Text
    const SizedBox(height: 30.0),
    ...currentQuestion
        .getShuffledAnswers()
        .map((answer) => AnswerButton(
              answerText: answer,
              onTap: answerQuestion,
            )), // AnswerButton
    // AnswerButton(
```



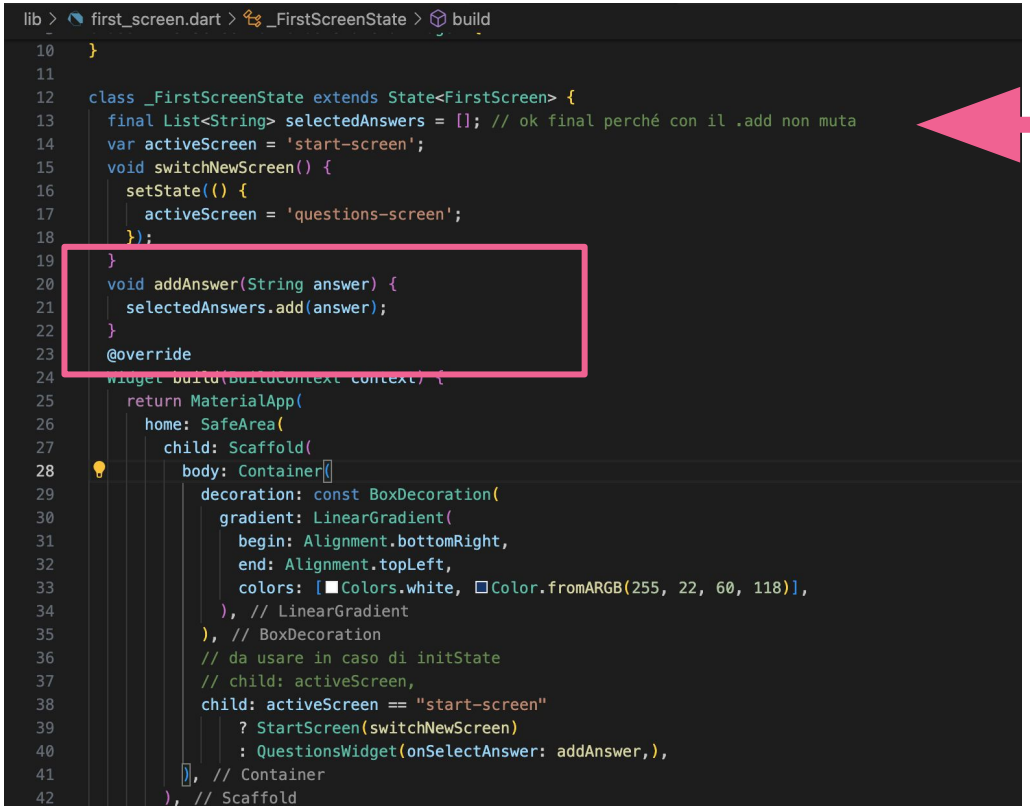
Integrare package esterni - pub.dev



https://pub.dev/

Memorizzare le risposte

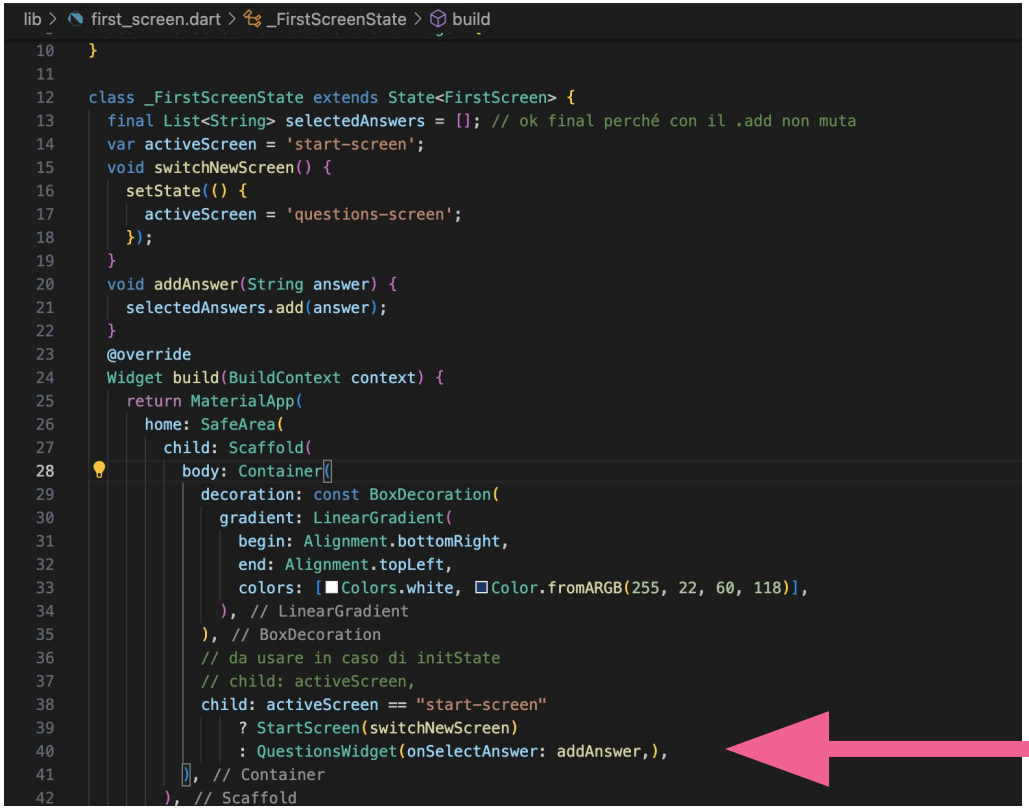
Lifting up state



Aggiorno il widget con le domande

```dart
import 'package:lezione7/answer_button.dart';
import 'package:lezione7/data/questions.dart';
import 'package:google_fonts/google_fonts.dart';

class QuestionsWidget extends StatefulWidget {
  const QuestionsWidget({
    super.key,
    required this.onSelectAnswer,
  });
  final void Function(String answer) onSelectAnswer;

  @override
  State<QuestionsWidget> createState() => _QuestionsWidgetState();
}

```

Lifting up state



Il valore deve essere passato all'interno della State Class

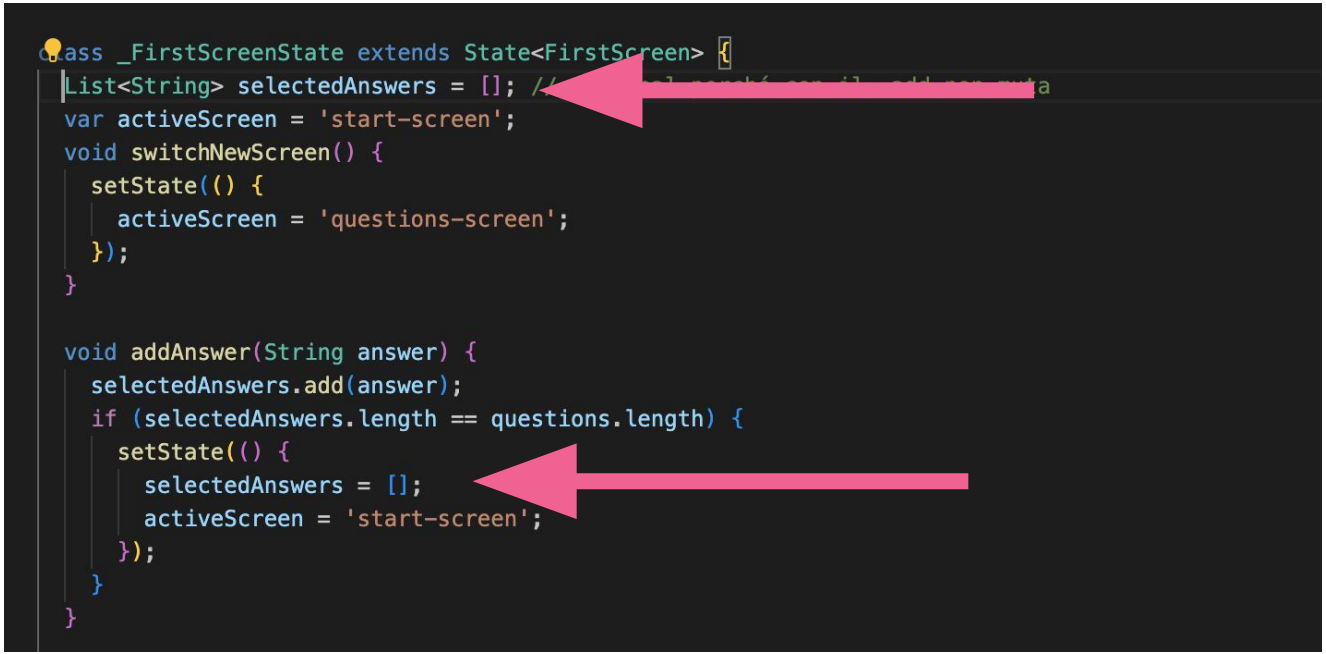
Passa la risposta e aggiorna il contatore

Evitare di uscire fuori dall’index delle domande

Redirect in home quando raggiungiamo la “fine”

```
void addAnswer(String answer) {
  selectedAnswers.add(answer);
  if (selectedAnswers.length == questions.length){
    setState(() {
      activeScreen = 'start-screen';
    });
  }
}
```

Perfetto, quindi funziona?


class _FirstScreenState extends State<FirstScreen> {
List<String> selectedAnswers = [];
var activeScreen = 'start-screen';
void switchNewScreen() {
setState(() {
activeScreen = 'questions-screen';
});
}
void addAnswer(String answer) {
selectedAnswers.add(answer);
if (selectedAnswers.length == questions.length) {
setState(() {
selectedAnswers = [];
activeScreen = 'start-screen';
});
}
}

Mostriamo i risultati

Schermata desiderata

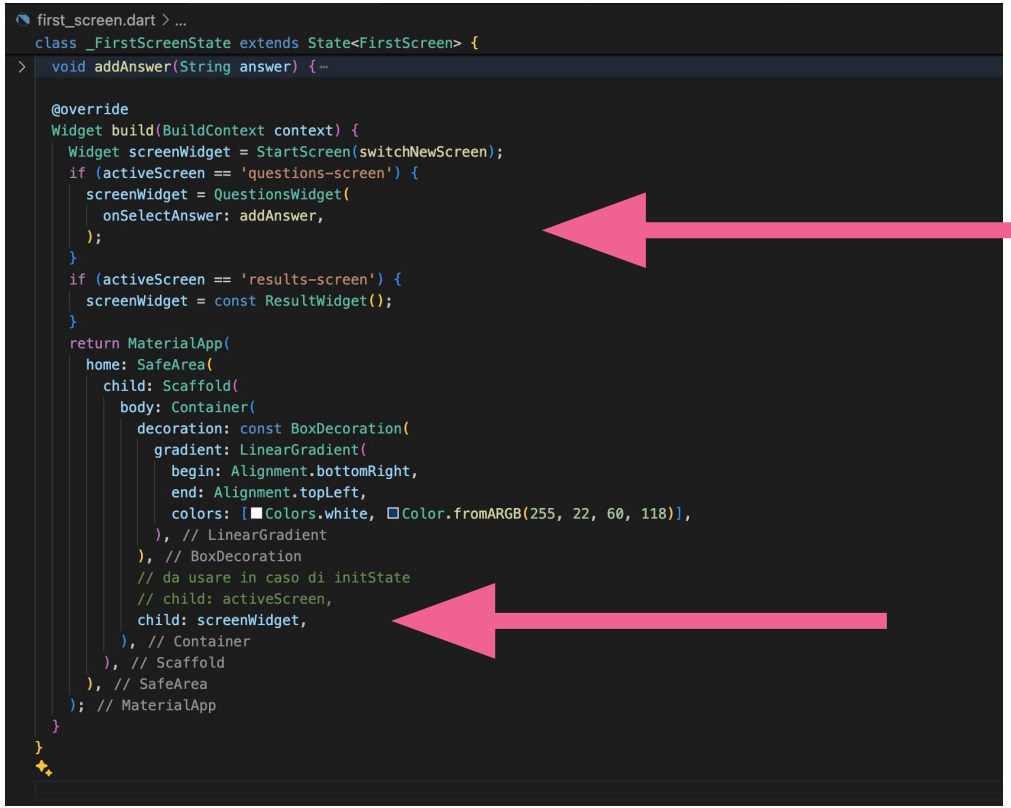
La schermata con il risultato sarà un Widget Stateless o Stateful?

ResultWidget

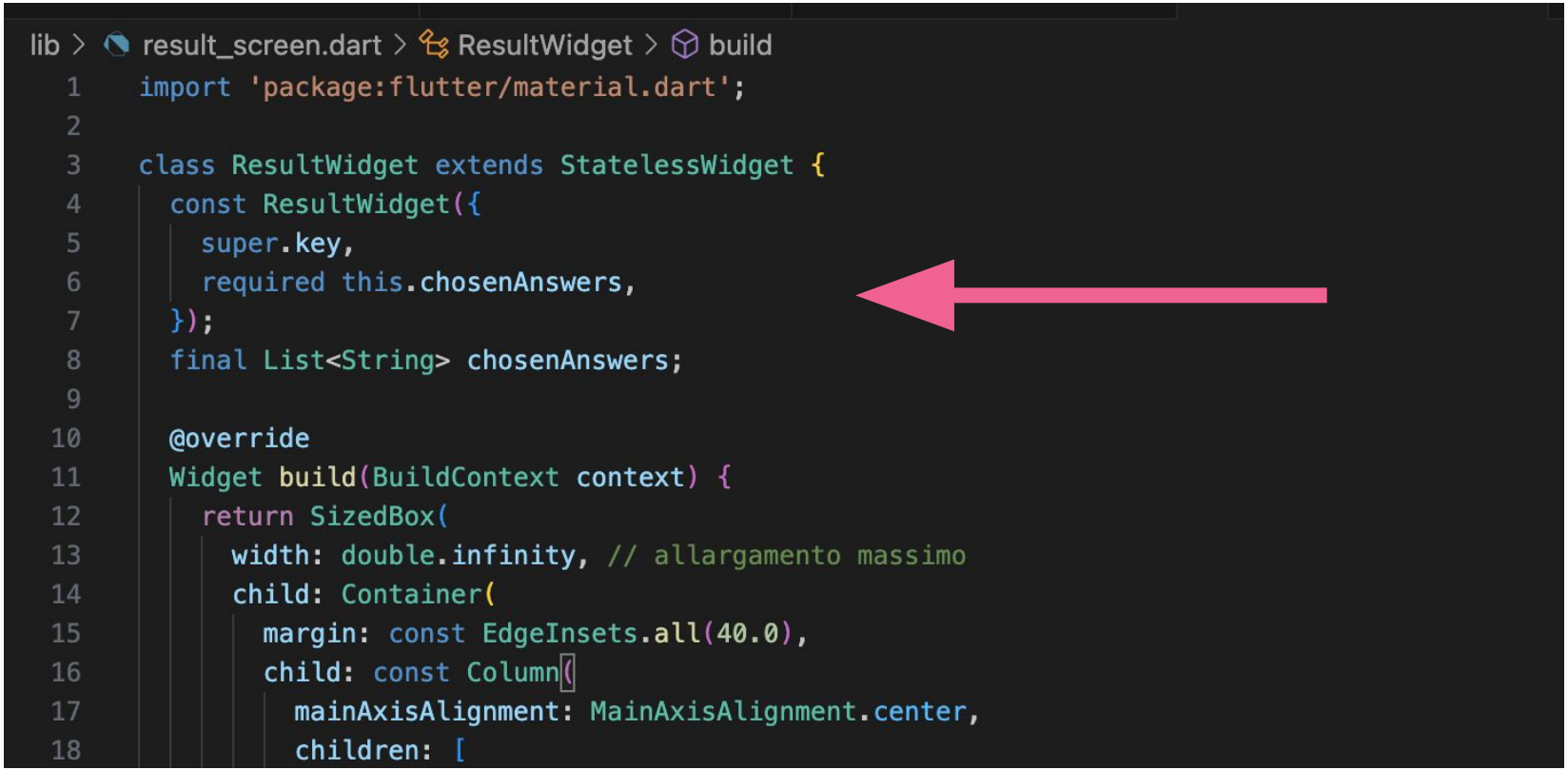
result_screen.dart > ...

```dart
import 'package:flutter/material.dart';

class ResultWidget extends StatelessWidget {
  const ResultWidget({super.key});

  @override
  Widget build(BuildContext context) {
    return SizedBox(
      width: double.infinity, // allargamento massimo
      child: Container(
        margin: const EdgeInsets.all(40.0),
        child: const Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text('Hai risposto correttamente a X domande su Y'),
            SizedBox(
              height: 30,
            ), // SizedBox
            Text('Lista di domande e risposte'),
            SizedBox(
              height: 30,
            ), // SizedBox
            TextButton(
              onPressed: null,
              child: Text('Ricomincia'),
            ), // TextButton
          ],
        ), // Column
      ), // Container
    ); // SizedBox
  }
}
```

Integriamo nel flusso delle schermate il nuovo Widget



Passare le informazioni a ResultWidget



Passare le informazioni a ResultWidget

```
class _FirstScreenState extends State<FirstScreen> {
  List<String> selectedAnswers = [];
  var activeScreen = 'start-screen';
  void switchNewScreen() {
    setState(() {
      activeScreen = 'questions-screen';
    });
  }

  void addAnswer(String answer) {
    selectedAnswers.add(answer);
    if (selectedAnswers.length == questions.length) {
      setState(() {
        selectedAnswers = [];
        activeScreen = 'results-screen';
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    Widget screenWidget = StartScreen(switchNewScreen);
    if (activeScreen == 'questions-screen') {
      screenWidget = QuestionsWidget(
        onSelectAnswer: addAnswer,
      );
    }
    if (activeScreen == 'results-screen') {
      screenWidget = ResultWidget(chosenAnswers: selectedAnswers,);
    }
    return MaterialApp(
```

Controlliamo le risposte

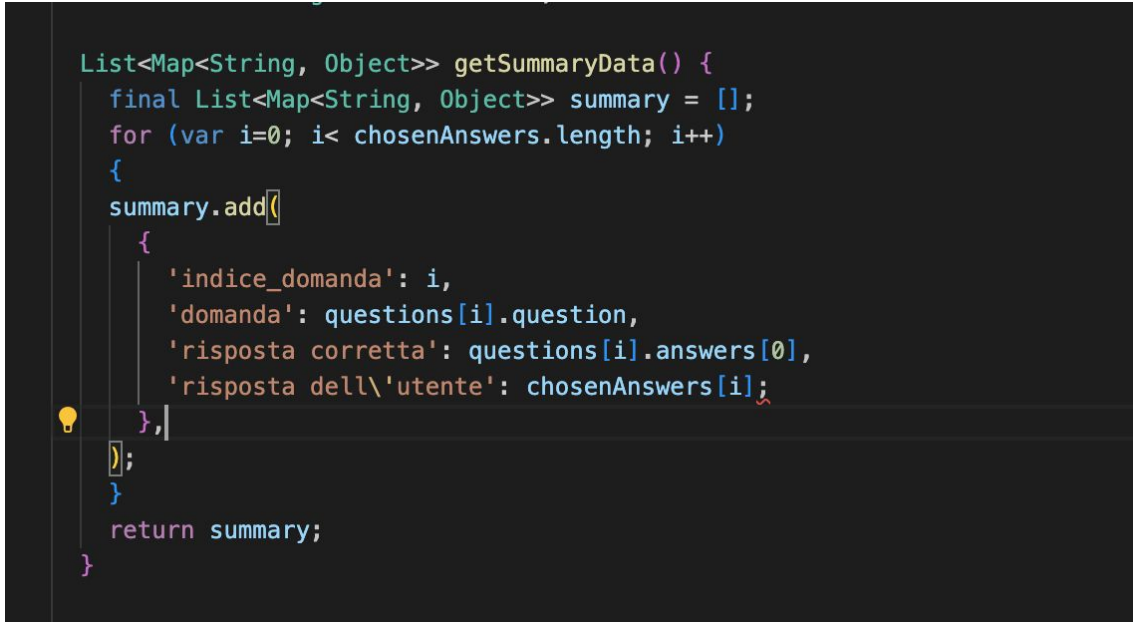
result_screen.dart

lib > result_screen.dart > ResultWidget > build

```dart
import 'package:flutter/material.dart';

class ResultWidget extends StatelessWidget {
  const ResultWidget({
    super.key,
    required this.chosenAnswers,
  });
  final List<String> chosenAnswers;

  List<Map<String, Object>> getSummaryData() {
    final List<Map<String, Object>> summary = [];

    // qui il codice per controllare i risultati
    return summary;
  }
```

Una Map è una raccolta di **coppie chiave-valore**, dove ogni chiave è associata ad un valore.
Le **chiavi devono essere uniche** all'interno della mappa, ma i **valori possono essere duplicati**.

Controlliamo le risposte

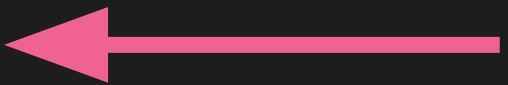


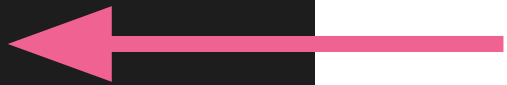
```
List<Map<String, Object>> getSummaryData() {
  final List<Map<String, Object>> summary = [];
  for (var i=0; i< chosenAnswers.length; i++)
  {
  summary.add(
    {
      'indice_domanda': i,
      'domanda': questions[i].question,
      'risposta corretta': questions[i].answers[0],
      'risposta dell\'utente': chosenAnswers[i];
    },
  );
  }
  return summary;
}
```

Widget specifico per le risposte

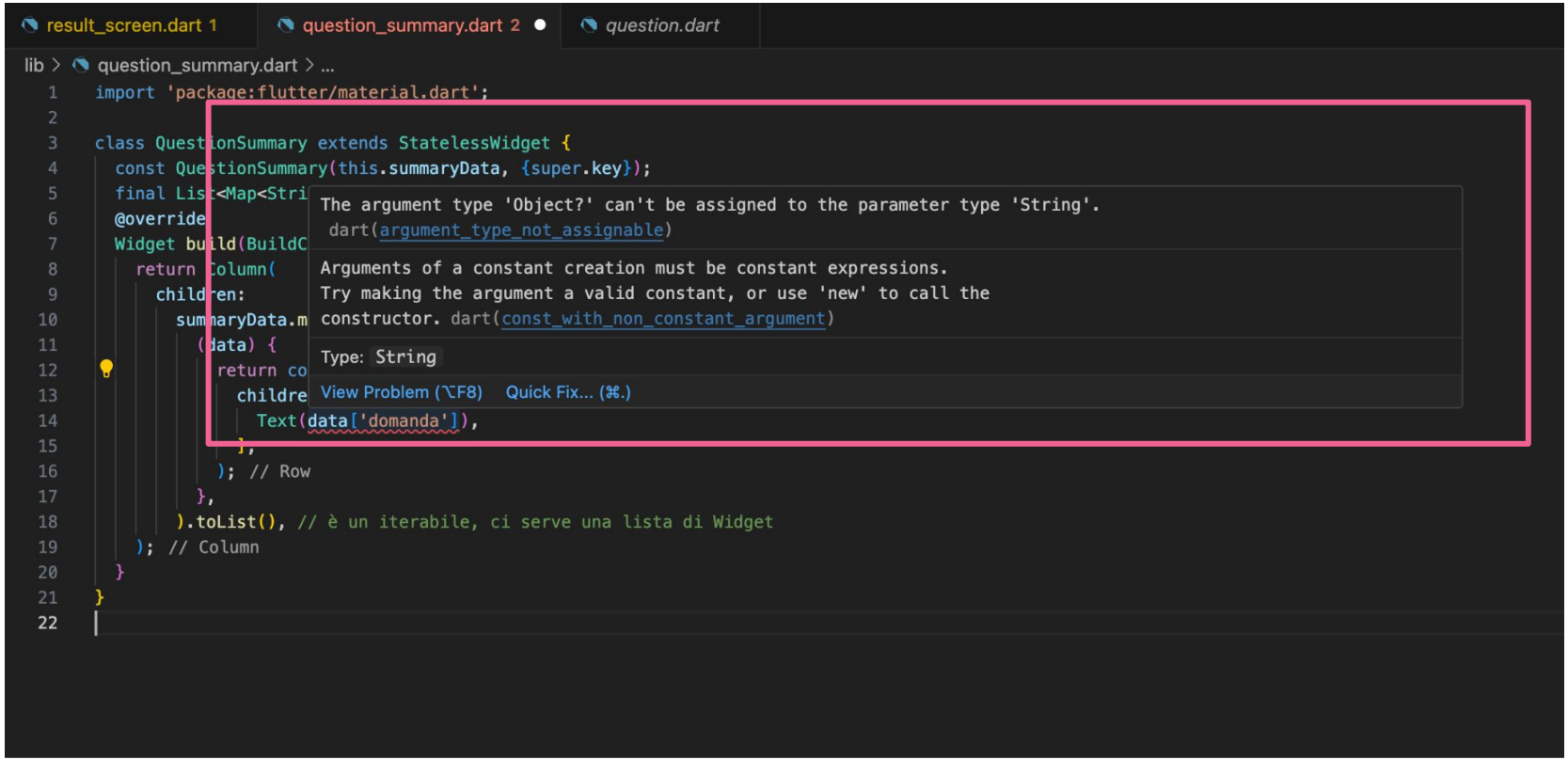
lib > question_summary.dart > ...

```dart
import 'package:flutter/material.dart';

class QuestionSummary extends StatelessWidget {
  const QuestionSummary(this.summaryData, {super.key});
  final List<Map<String, Object>> summaryData;
  @override
  Widget build(BuildContext context) {
    return Column(
      children:
        summaryData.map( // crea dei widget popolandoli con i dostri dati
          (data) {
            return const Row(
              children: [
                Text(data['domanda']),
              ],
            ); // Row
          },
        ).toList(), // è un iterabile, ci serve una lista di Widget
    ); // Column
  }
}

```





Cast di una variabile



Integriamo le risposte

lib > question_summary.dart > ...

```dart
import 'package:flutter/material.dart';

class QuestionSummary extends StatelessWidget {
  const QuestionSummary(this.summaryData, {super.key});
  final List<Map<String, Object>> summaryData;
  @override
  Widget build(BuildContext context) {
    return Column(
      children: summaryData.map(
        // crea dei widget popolandoli con i dostri dati
        (data) {
          return Row(
            children: [
              Text(
                ((data['indice_domanda'] as int) + 1).toString(),
              ), // Text
              Column(
                children: [
                  Text(data['domanda'] as String),
                  const SizedBox(height: 10),
                  Text(data['risposta corretta'] as String),
                  Text(data['risposta dell\'utente'] as String),
                ],
              ), // Column
            ],
          ); // Row
        },
      ).toList(), // è un iterabile, ci serve una lista di Widget
    ); // Column
  }
}
```

Aggiungiamo il Widget dinamico alla schermata di riepilogo

Funziona?

result_screen.dart ● | question_summary.dart | *questions_screen.dart*

lib > result_screen.dart > ...

```dart
class ResultWidget extends StatelessWidget {
  List<Map<String, Object>> getSummaryData() {
  }

  @override
  Widget build(BuildContext context) {
    return SizedBox(
      width: double.infinity, // allargamento massimo
      child: Container(
        margin: const EdgeInsets.all(40.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const Text('Hai risposto correttamente a X domande su Y'),
            const SizedBox(
              height: 30,
            ), // SizedBox
            QuestionSummary(getSummaryData()),
            const SizedBox(
              height: 30,
            ), // SizedBox
            const TextButton(
              onPressed: null,
              child: Text('Ricomincia'),
            ), // TextButton
          ],
        ), // Column
      ), // Container
    ); // SizedBox
  }
}

```

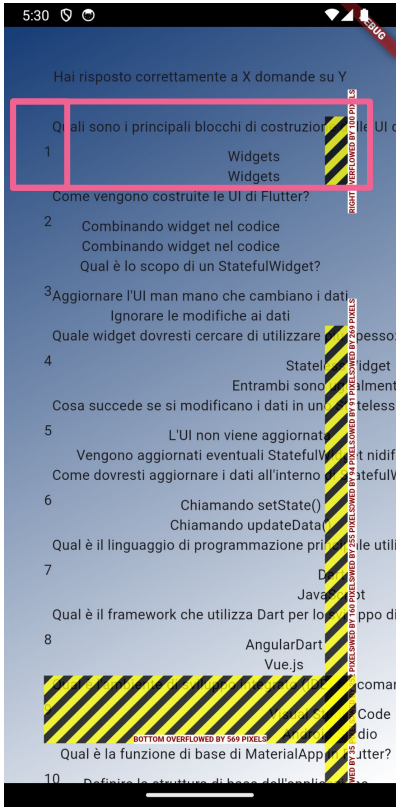
Aggiungiamo il Widget dinamico alla schermata di riepilogo


```
void addAnswer(String answer) {
  selectedAnswers.add(answer);
  if (selectedAnswers.length == questions.length) {
    setState(() {
      selectedAnswers = [];
      activeScreen = 'results-screen';
    });
  }
}
```

Funziona?
No

Aggiungiamo il Widget dinamico alla schermata di riepilogo

Funziona?
Alla grande

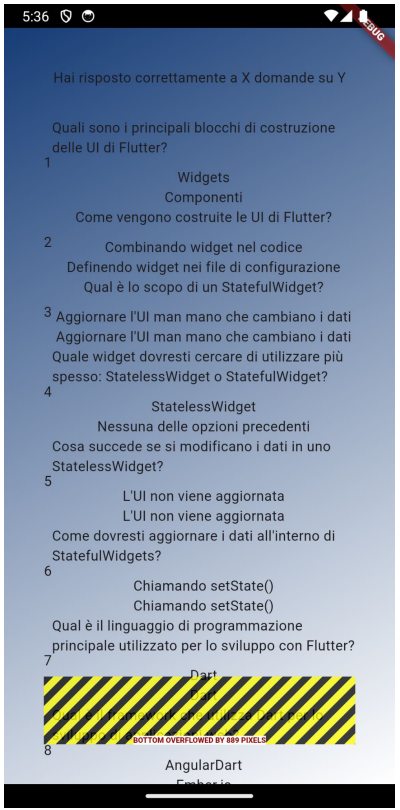


Out of Boundary: ci aiuta Expanded

result_screen.dart | question_summary.dart

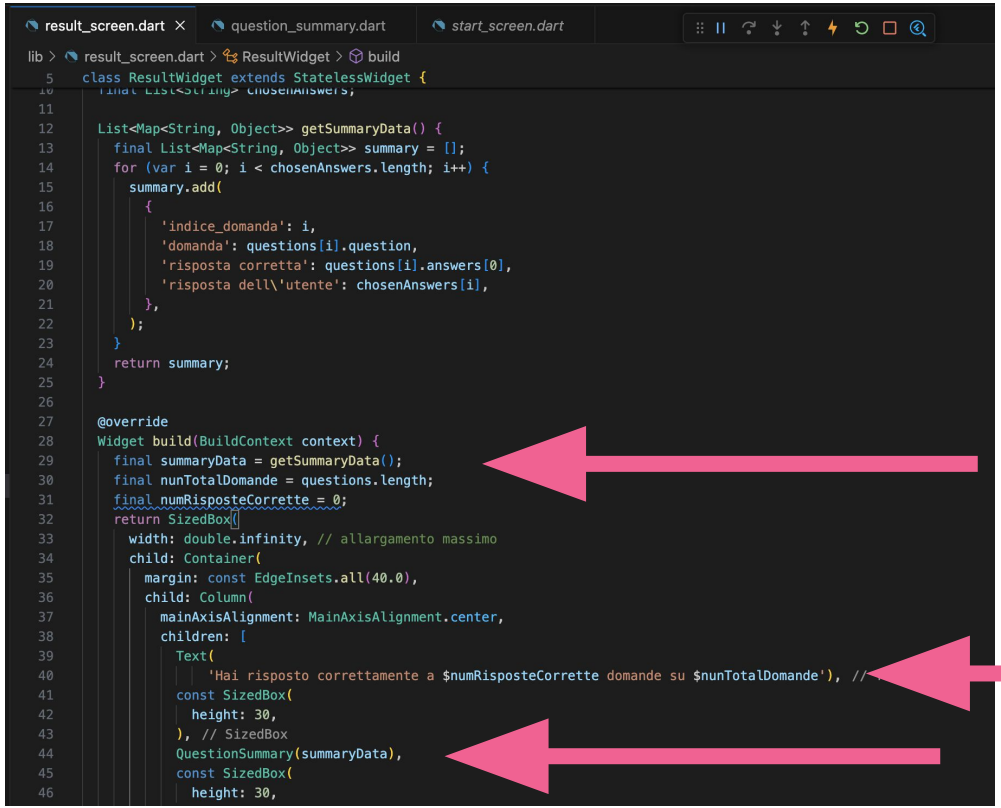
lib > question_summary.dart > QuestionSummary > build

```dart
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';

class QuestionSummary extends StatelessWidget {
  const QuestionSummary(this.summaryData, {super.key});
  final List<Map<String, Object>> summaryData;
  @override
  Widget build(BuildContext context) {
    return Column(
      children: summaryData.map(
        // crea dei widget popolandoli con i dostri dati
        (data) {
          return Row(
            children: [
              Text(
                ((data['indice_domanda'] as int) + 1).toString(),
              ), // Text
              Expanded(
                child: Column(
                  children: [
                    Text(data['domanda'] as String),
                    const SizedBox(height: 10),
                    Text(data['risposta corretta'] as String),
                    Text(data['risposta dell\'utente'] as String),
                  ],
                ), // Column
              ), // Expanded
            ],
          ); // Row
        },
      ).toList(), // è un iterabile, ci serve una lista di Widget
    ); // Column
  }
}

```



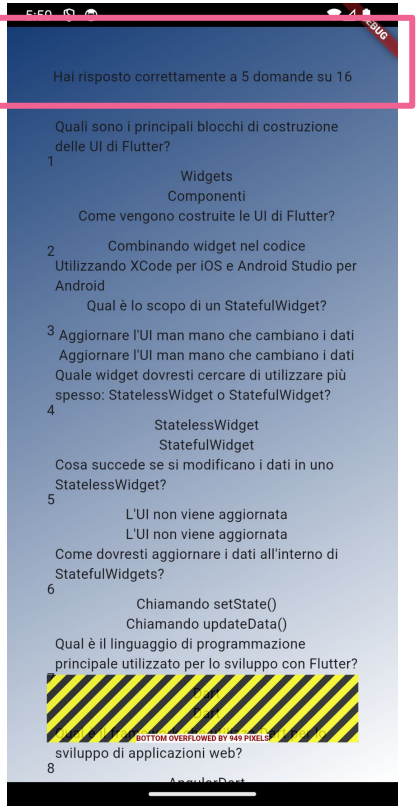
Adatta il contenuto allo spazio disponibile!

Aggiungiamo il numero di risposte corrette

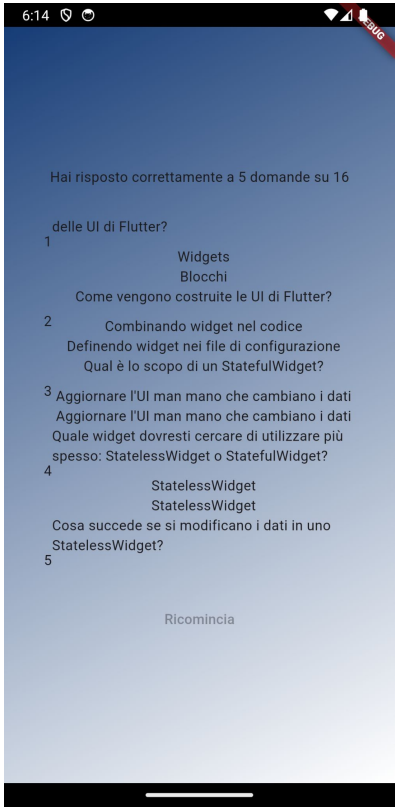


Aggiungiamo il numero di risposte corrette

```dart
@override
Widget build(BuildContext context) {
  final summaryData = getSummaryData();
  final nunTotalDomande = questions.length;
  final numRisposteCorrette = summaryData
      .where((element) =>
          element['risposta corretta'] == element['risposta dell\'utente'])
      .length;
  return SizedBox(
```



Contenuti scrollabili: SingleChildScrollView



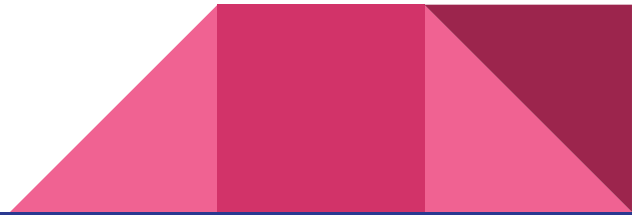

```dart
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';

class QuestionSummary extends StatelessWidget {
  const QuestionSummary(this.summaryData, {super.key});
  final List<Map<String, Object>> summaryData;
  @override
  Widget build(BuildContext context) {
    return SizedBox(
      height: 350,
      child: SingleChildScrollView(
        child: Column(
          children: summaryData.map(
            // crea dei widget popolandoli con i dostri dati
            (data) {
              return Row(
                children: [
                  Text(
                    ((data['indice_domanda'] as int) + 1).toString(),
                  ), // Text
                  Expanded(
                    child: Column(
                      children: [
                        Text(data['domanda'] as String),
                        const SizedBox(height: 10),
                        Text(data['risposta corretta'] as String),
                        Text(data['risposta dell\'utente'] as String),
                      ],
                    ), // Column
                  ), // Expanded
                ],
              ); // Row
            },
          ).toList(), // è un iterabile, ci serve una lista di Widget
        ), // Column
      ), // SingleChildScrollView
    ); // SizedBox
  }
}
```

Esercizio

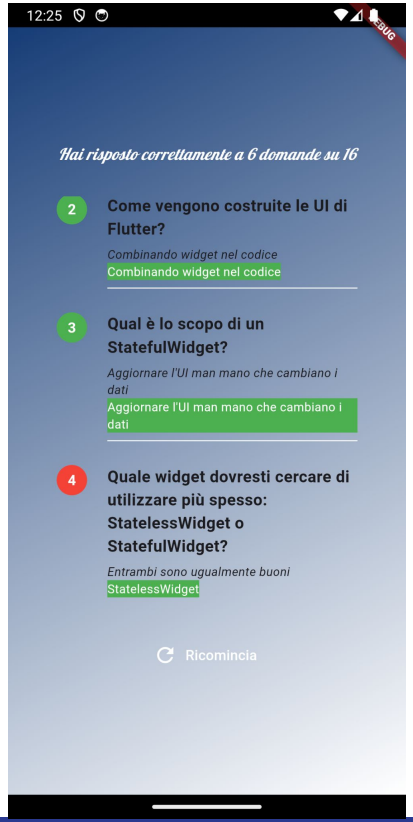
Esercizio: Finalizzazione del progetto svolto nelle ultime due lezioni

L'obiettivo di questo esercizio è quello di verificare le conoscenze acquisite fino a questo punto del corso.

Scaricate lo zip con il nome esercizio ed integrate quello che manca per completare il progetto: pulsante per ritorno alla home ed uno o più Widget che permettano di rappresentare graficamente le risposte come da screen mostrato nella slide successiva.



Esercizio: Finalizzazione del progetto svolto nelle ultime due lezioni



Esercizio: Finalizzazione del progetto svolto nelle ultime due lezioni

Consegna:

- **Inviare il progetto in uno zip a nicola.noviello@unimol.it utilizzando come oggetto della e-mail “Lezione8: Nome Cognome Matricola”**; In alternativa è possibile usare GitHub, garantendomi l’accesso al progetto, ma è sempre necessario mandarmi una mail, nel caso, con il link al progetto;
- Da completare **tassativamente** nelle ore di lezione residue della giornata odierna, dopo aver completato la lezione teorica;
- Non obbligatorio, ma valutato positivamente ai fini dell’esame.