Infinite Scroll & Larger Text

We have two goals we want to achieve

1. **Scrolling**

   We'd like to be able to scroll indefinitely

2. **Larger Text**

   We'd like to be able to read the text (make it BIGGER and **bolder**)

Easy _getSentence()

We want a simpler way of storing a list of sentences to work with.

- Create a String list for storing sentences
- And a function to generate them

```
final _sentences = <String>[];
final _biggerFont = const TextStyle(fontSize: 14.0);

String _getSentence() {
  final noun = new WordNoun.random();
  final adjective = new WordAdjective.random();
  return  "The programmer wrote a ${adjective.asCapitalized}"
          " app in Flutter and showed it"
          " off to his ${noun.asCapitalized}";
}
```

- We've also added a constant **_biggerFont** that will store the size of the font for each sentence

Item Builder List View

```
Widget _buildSentences() {
  return new ListView.builder(
    padding: const EdgeInsets.all(16.0),
    // An ItemBuilder callback is used to add sentences and dividers
    // based on when the user scrolls down the view.
    itemBuilder: (context, i) {
      // Adds a pixel divider before each row
      if (i.isOdd) return new Divider();
      // Divides i by 2 then returns the value into index
      // This counts how many sentences are in the ListView
      final index = i ~/ 2;
      // Checks to see if we've hit the end of the sentence list
      if (index >= _sentences.length) {
        // If we have then we generate another 10
        for (int x = 0; x < 10; x++) {
          _sentences.add(_getSentence());
        }
      }
      return _buildRow(_sentences[index]);
    },
  );
}
```

We'll write this function in a second!

https://flutter.io/get-started/codelab/#step-4-create-an-infinite-scrolling-listview

_buildRow and build()

Clean up the **build()** function and add **_buildRow()**

```
Widget _buildRow(String sentence) {
  return new ListTile(
    title: new Text(
      sentence,
      style: _biggerFont,
    ),
  );
}

@override
Widget build(BuildContext context) {
  return new Scaffold(
    appBar: new AppBar(
      title: new Text('Word Game'),
    ),
    body: _buildSentences(),
  );
}
```

Here's some logic we've pulled out of **MyApp** to handle here instead.

I'll show you in the next slide

Clean up MyApp()

Back up in **MyApp**, change the **build()** function to simply call **RandomSentences()**

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: 'Word Game',
      home: new RandomSentences(),
    );
  }
}
```

Rerun the app and scroll indefinitely!