

Simple Hello World App

```
import
'package:flutter/material.dart';
import the material component
package
void main() => runApp(MyApp());
create and run the app
class MyApp extends StatelessWidget { the main widget which
is launched
  @override
  Widget build(BuildContext
context) { overridden build
method which holds the content
    return MaterialApp(
      title: 'Welcome to
Flutter', title
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome
to Flutter'), set the appbar-
text
        ),
        body: Center(
          child: Text('Hello
World'), set the body to center
and add text
        ),
      );
    }
  }
}
```

Infinite scrolling listview

```
class RandomWordsState extends
State<RandomWords> {
  final _suggestions = <WordP-
air>[]; adds a suggestion list
  final _biggerFont = const
TextStyle(fontSize: 18.0); incr-
eases the font size
  @override
  Widget build(BuildContext
context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Startup Name
Generator'),
      ),
      body: _buildSuggestions(),
    );
  }
}
Widget _buildSuggestions() {
  return ListView.builder(
    padding: const EdgeInsets-
s.all(16.0),
    itemBuilder: (context, i)
{ creates the items
    if (i.isOdd) return
Divider(); adds a divider
between items
    final index = i ~/ 2;
calculates number of word
parings in the list without
counting dividers
    if (index >= _suggesti-
ons.length) {
      _suggestions.addAll(g-
enerateWordPairs().take(10));
add ten more words when the end
of the list is reached
    }
  }
}
```

Infinite scrolling listview (cont)

```

}
return _buildRow(_sugg-
estions[index]); build the row
using the widget below
Widget _buildRow(WordPair pair)
{
  return ListTile(
    title: Text(
      pair.asPascalCase, **sets
the word as text
      style: _biggerFont,
changes the fontsize
    ),
  );
}
});
}
```

External packages

```
edit pubspec.yaml, to add
english_words
dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^0.1.2
+ english_words: ^3.1.5
after that, run flutter pub get
to get the new dependencies
and then add the new dependency
to lib/main.dart
import 'package:english_words/e-
nglish_words.dart';
```

Generate random words

```
class MyApp extends
StatelessWidget {
  @override
  Widget build(BuildContext
context) {
+ final wordPair = WordPair.ran-
dom(); create a new wordpair
  return MaterialApp(
    title: 'Welcome to
Flutter',
    home: Scaffold(
@@ -16,7 +18,7 @@
      title: Text('Welcome
to Flutter'),
    ),
    body: Center(
- child: Text('Hello World'),
remove the hello world text
+ child: Text(wordPair.asPasca-
lCase), show random word as text
    ),
  );
}
```

Stateful widgets

```
class RandomWordsState extends
State<RandomWords> {
  @override
  Widget build(BuildContext
context) {
    final wordPair = WordPair.r-
andom();
    return Text(wordPair.asPa-
scalCase);
  }
}
```

Stateful widgets (cont)

this creates a "state" which saves the generated word and displays it as text

```
class RandomWords extends
StatefulWidget {
  @override
  RandomWordsState createState()
=> RandomWordsState();
}
add the state to the statefull widget
class MyApp extends Stateless-
Widget {
  @override
  Widget build(BuildContext
context) {
- final wordPair = WordPair.ran-
dom();
    return MaterialApp(
      title: 'Welcome to
Flutter',
      home: Scaffold(
@@ -18,8 +17,8 @@
        title: Text('Welcome
to Flutter'),
      ),
      body: Center(
- child: Text(wordPair.asPasca-
lCase),
+ child: RandomWords(),
    ),
  );
}
remove the previous random word generation and replace it with the stateful widget
```



By [deleted]

cheatography.com/deleted-96251/

Not published yet.

Last updated 4th March, 2020.

Page 2 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>