



Ödev 4-2

☰ Açıklama	
🕒 Created	@November 26, 2021 1:20 PM
🔗 Link	https://youtu.be/BZfClqxRbAI

Uygulamanın youtube linki : <https://youtu.be/BZfClqxRbAI>

Kodlar :

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

void main(){
  runApp(MaterialApp(home: MyApp()));
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Startup',
      home: RandomWords(),
    );
  }
}

class RandomWords extends StatefulWidget{
  @override
  _RandomWordsState createState() => _RandomWordsState();
}

class _RandomWordsState extends State<RandomWords> {
  final _suggestions = <WordPair>[];
  final _biggerFont = TextStyle(fontSize: 18);

  @override
  Widget build(BuildContext context) {
    return Scaffold (
      appBar: AppBar(
        title: Text('Startup Adı Oluşturucu'),
      ),
      body: _buildSuggestions(),
    );
  }

  Widget _buildSuggestions() {
    return ListView.builder( // her satır için çalış ...
      padding: const EdgeInsets.all(16),
      itemBuilder: (BuildContext _context, int i) { //i satır index'ini tutar.
        /*
        index 0'dan başlar ve her çalıştırılmada artar.
        Her sözcük eşleştirmesi için iki kez artar: bir kez ListTile için ve bir kez Divider() için.
        Böylece, kullanıcı kaydirdıkça önerilen liste büyümeye devam eder
        */

        if (i.isOdd) { // eğer index tekse (örn: 0-> ListTile , 1-> Divider , 2-> ListTile , 3-> Divider . . . )
          return Divider(); // Divider (çizgi) koy
        }
        final int index = i ~/ 2;
        if (index >= _suggestions.length) { // index , listeden uzunsa ( kullanıcı ekran kaydırıyorsa )
          _suggestions.addAll(generateWordPairs().take(10)); // yenilerini ekle
        }
      }
    );
  }
}
```

```

        return _buildRow(_suggestions[index]);
    }
    );
}

Widget _buildRow(WordPair pair) { // Satırlar oluşturulur.
    return ListTile(
        title: Text(
            pair.asPascalCase,
            style: _biggerFont,
        ),
    );
}
}

```

Ekran görüntüleri ve açıklamalar

- Öncelikle english_words paketini import ettik.
- MyApp sınıfını çalıştırdık.
- Burada MaterialApp widget'ını kullandık ve RandomWords sınıfımızı çağırdık.

```

1  import 'package:flutter/material.dart';
2  import 'package:english_words/english_words.dart';
3
4  void main(){
5      runApp(MaterialApp(home: MyApp()));
6  }
7
8  class MyApp extends StatelessWidget {
9      @override
10     Widget build(BuildContext context) {
11         return MaterialApp(
12             title: 'Startup',
13             home: RandomWords(),
14         ); // MaterialApp
15     }
16 }
17
18 class RandomWords extends StatefulWidget{
19     @override
20     _RandomWordsState createState() => _RandomWordsState();
21 }
22
23 class _RandomWordsState extends State<RandomWords> {
24     final _suggestions = <WordPair>[];
25     final _biggerFont = TextStyle(fontSize: 18);
26

```

- Bu sınıf StatefulWidget'tan miras almaktadır. State durumu ise _RandomWordsState sınıfında tutulur.
- Bu sınıfta bir liste tutan _suggestions ile fontSize bilgisini tutan _biggerFont değişkenleri var. Liste kelime çiftlerini tutar ve import edilen paket içerisinden alınır.

- build metodunda Scaffold widget'ını çağırdık. Scaffold'un body bölümünde kendi widget'ımızı çağırdık.
- Burada ListView.builder yapılandırıcısı çağırılır.
- itemBuilder parametresine context ve index bilgisini verdik.
- ListView.builder metodu her satır için çalışacak.

```

27  @override
28  Widget build(BuildContext context) {
29    return Scaffold (
30      appBar: AppBar(
31        title: Text('Startup Adı Oluşturucu'),
32      ), // AppBar
33      body: _buildSuggestions(),
34    ); // Scaffold
35  }
36
37  Widget _buildSuggestions() {
38    return ListView.builder( // her satır için çalış ...
39      padding: const EdgeInsets.all(16),
40      itemBuilder: (BuildContext _context, int i) { //i satır index'ini tutar.
41        /*
42         index 0'dan başlar ve her çalıştırılmada artar.
43         Her sözcük eşleştirmesi için iki kez artar: bir kez ListTile için ve bir kez Divider() için.
44         Böylece, kullanıcı kaydırdıkça önerilen liste büyümeye devam eder
45        */
46
47        if (i.isOdd) { // eğer index tekse (örn: 0-> ListTile , 1-> Divider , 2-> ListTile , 3-> Divider ...)
48          return Divider(); // Divider (çizgi) koy
49        }
50        final int index = i ~/ 2;
51        if (index >= suggestions.length) { // index listeden uzunsa (kullanıcı ekran kaydırdıysa)

```

- Eğer index numarası tekse araya bir Divider yerleştirecek. Örneğin ilk başta 0 olduğu için bir ListTile (kelime) , daha sonra 1 'de Divider , ardından 2 'de tekrar Divider ve bu şekilde devam edecek.
- Eğer index , listemizdeki kelime sayısından (liste uzunluğu) büyük veya eşit olursa listeye 10 eleman daha ekleyecek. Bunu da paketdeki metodlar sayesinde yapacak.
- Listedeki bu elemanları _buildRow metoduna vererek döndürecek.
- _buildRow kendi yazdığımız bir widget'tır. ListTile döndürür. Buradaki yazları PascalCase tipinde yazdırır. Font bilgisini ise sınıf içinde tanımladığımız _biggerFont değişkeninden alır.

```

50         final int index = i ~/ 2;
51         if (index >= _suggestions.length) { // index , listeden uzunsa ( kullanıcı ekran kaydırıyorsa )
52             _suggestions.addAll(generateWordPairs().take(10)); // yenilerini ekle
53         }
54         return _buildRow(_suggestions[index]);
55     }
56     ); // ListView.builder
57 }
58
59 Widget _buildRow(WordPair pair) { // Satırlar oluşturulur.
60     return ListTile(
61         title: Text(
62             pair.asPascalCase,
63             style: _biggerFont,
64         ), // Text
65     ); // ListTile
66 }
67
68 }
69
70

```

