

# HW1 - Dry Part

## Exercise 1

### Question 1

The following lines are responsible for the infinite scrolling:

```
if ( index >= _suggestions.length) {  
  _suggestions.addAll(generateWordPairs().take(10));  
}
```

If we remove those lines an exception will be thrown when we reach the end of the list, indicating that the requested index is out of range.

```
RangeError (index): Invalid value:
```

### Question 2

The `ListView.seperated` constructor will create a list with dividers. It uses 2 builders - one for items and one for dividers. However, that constructor requires a `itemCount` param which limits the length of the list, so it cannot be used in our original case of infinite list.

In my opinion, in finite lists `ListView.seperated` is better since it allows separation of different logics - what is an item vs what is a divider. That makes the code cleaner and simpler.

### Question 3

To reflect the change in the state in the UI.

Using `setState()` notifies the framework that the state of the widget has changed and triggers a call to the state's `build()` method which updates the UI.

## Exercise 2

### Question 1

I used the method `Navigator.push(route)`.

I could also have used `Navigator.pushReplacement(route)` but then I couldn't return from the login screen back to the main screen with the back arrow icon.

### Question 2

I used the method `ScaffoldMessenger.showSnackBar(SnackBar)`.

It requires the widget `SnackBar`.