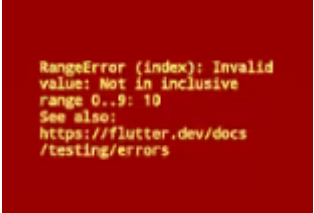


## פרויקט אנדרואיד 236271 - יבש 1

1. השורות שמאפשרות גלילה אין סופית:

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

במידה ונוריד שורות אלה, בשלב מסוים, נגיע למצב שהאינדקס איתו אנחנו ניגשים לרשימה (גדל ע"י גלילה) יעבור את אורך הרשימה suggestions, כיוון ששתי שורות אלה מאפשרות את ההגדלה החוזרת של הרשימה. ולכן ניגש למקום לא מוגדר בזיכרון (מחוץ לגבולות הרשימה – גישה לא חוקית לזיכרון).



```
RangeError (index): Invalid  
value: Not in inclusive  
range 0..9: 10  
See also:  
https://flutter.dev/docs  
/testing/errors
```

2. הדרך הנוספת להשתמש בdivider ולהפריד איברים בListView והוא באמצעות שימוש במתודה אחרת של ListView separated <= ListView אשר מקבל ארגומנט בשדה SeparatorBuilder נוסף של פונקציה אשר מחזירה Divider, כלומר מפרידה באמצעות פונקציה זו בין איברי הרשימה.

```
},  
separatorBuilder: (BuildContext context, int index) => const Divider(),  
);
```

זו שיטה עדיפה כי השיטה הראשונה נראית "מלאכותית", נעזר בפיצ'ר מובנה ועושה הכל באופן עצמאי. בשיטה הראשונה אנו עלולים להגיע למקרי קצה כלשהם בהם הסדר של האי זוגי וזוגי משתנה ודבר זה יפגע בסדר של ההפרדה ברשימה. בנוסף, בעת שימוש השיטה הראשונה אנו ממלאים את הList ב Divider מה שיקח מקום כאיבר ברשימה ולמשל ברשימה עם 100 איברים נוכל לשמור פחות מידע בגלל החוצצים שיתפסו מידע גם הם.

3. BuildRow מכיל את הקריאה ל SetState() מכיוון שכל לחיצה על האייקון הלב גוררת שינוי במצב (שינוי צבע האייקון/הוספה למועדפים), יש לעדכן ולבנות מחדש. כל שינוי גורם לבנייה מחדש של המשבצת ולכן יש להשתמש בפונקציה SetState() כדי לטעון את המצב החדש.

## תרגיל 2:

1. המטרה MaterialApp widget לשמש כ widget מעטפת, עבור מספר ויידגטים נפוצים שמשמשים לעיצוב האפליקציה.  
שלושה מאפיינים של MaterialApp וידגט:  
א. Color – הצבע המרכזי לאיפיון האפליקציה ואת הממשק שלה עם מערכת ההפעלה.  
ב. Home – הווידגט הדיפולטיבי (הראשי) המופיע בפתחת האפליקציה.  
ג. Actions – מאפשר לדרוס מיפוי פעולות דיפולטיבי של ה Material App ולספק חלופות כרצוננו.
2. השדה key הוא מאפיין עבור ויידגטים, אלמנטים. באופן זה הוא מאפשר לשלוט במחיקה, חלופה והוספה של כל אלמנט יחודי. מסיבה זו ה key הופך לחשוב ב dismissible במקרה שיש מחיקות רבות, יש להתייחס לכל ויידגט באמצעות המפתח הייחודי לו, כלומר Key שולט כיצד ויידגט מסוים יחליף ויידגט אחר ברשימה ואם נוסף ויידגט חדש עם Key שקיים ברשימה הוא יחליף את החדש בקיים במקום להוסיף אחד נוסף.