

# המדריך הישראלי המלא – פיתוח אפליקציות לאנדרואיד

TAL Gnainsky

Microsoft



## שם הפרק

## עמוד

- 3..... ראשית דבר - הקדמה, התקנת התוכנה ושאלות נפוצות
- 6..... - שיעור ראשון - יצירת פרויקט ראשון
- 10..... - שיעור שני - עיצוב Activity
- 13..... - שיעור שלישי - הודעה בלחיצת כפתור
- 15..... - שיעור רביעי - הקוד של לחיצה על כפתור
- 17..... - שיעור חמישי - הוספת Activity חדש ומעבר אליו
- 19..... - שיעור שישי - הוספת תמונה
- 21..... - שיעור שביעי - משיכה מטקסט לטקסט
- 22..... - שיעור שמיני - שמירת נתונים
- 24..... - שיעור תשיעי - הצגת מידע ברשימה
- 27..... - שיעור עשירי - בדיקות
- 29..... - שיעור אחד-עשר - מסך פתיחה
- 32..... - שיעור שתיים-עשרה - שאלה דיאלוגית
- 35..... - שיעור שתיים-עשרה - התראות
- 37..... - שיעור שלוש-עשרה - מניפסט והגדרות
- 41..... נספח 1: אנימציות
- 44..... נספח 2: צד שרת
- 47..... נספח 3: גוגל פליי

## ראשית דבר - הקדמה, התקנת התוכנה ושאלות נפוצות

מי לא חווה את הרגע הזה בחייו שהוא יושב עם עצמו ואומר **"איך לא המציאו אפליקציה כזו עדיין?"**. אחרי כמה שבועות והיא יוצאת לאוויר ואתם עדיין על הספה. בשביל זה המדריך הזה קיים. בשביל זה אני כאן. במהלך השיעורים הבאים, אסביר לכם איך לפתח את האפליקציה שתמיד חלמתם עליה – גם אם אתם חסרי מושג בתכנות ובפיתוח. המדריך הכי פשוט וקל שיש ברשת לפיתוח אנדרואיד – נמצא כאן.

המדריך מלווה בסרטוני הסברה ביוטיוב. אז **בואו נצא לדרך עם התקנת התוכנה.**

### מהו אנדרואיד?

- אנדרואיד (Android) זו מערכת הפעלה המבוססת על ליבת לינוקס ומיועדת לסמארטפונים, טאבלטים, טלוויזיות ועוד.
- אנדרואיד הפכה למערכת הנפוצה ביותר בעולם לטלפונים חכמים, כשעפ"י הסקרים היא מחזיקה בנתח של מעל 80% מכלל שוק הטלפונים החכמים העולמי.

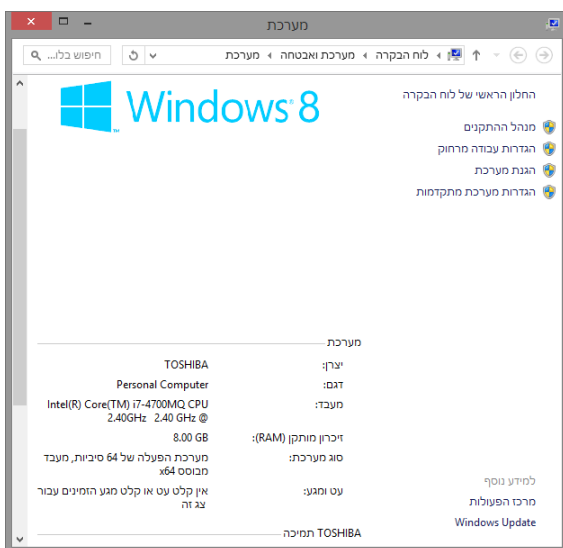
### באיזה שפה מפתחים אפליקציות לאנדרואיד?

על מנת לפתח את האפליקציה הגדולה שתמיד חלמתם עליה – יש לדעת ידע בסיסי בכתיבת ג'אווה. לכן מומלץ ללמוד באתר את חלק פיתוח הג'אווה, אולם - בקורס כאן נתעסק יותר בפיתוח האפליקציות בדרך פשוטה וקלה יחסית ללא צורך בידע קודם על ג'אווה.



## התקנת ג'אווה

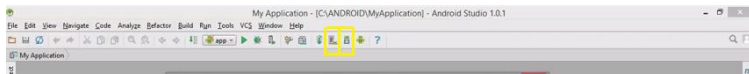
ג'אווה הינה שפת-על שמיועדת לפיתוח יישומים ופריסתם על גבי פלטפורמות שונות. על הרבה מחשבים, הג'אווה כבר מותקנת עם מערכת ההפעלה, אך במידה ולא – עליכם להתקין אותה לפני שאנחנו בכלל נכנסים לעולם האנדרואיד. ההתקנה מותאמת עפ"י מס' הסיביות. 32 או 64. היכנסו למחשב שלי ובידקו.לאחר מכן – הורידו מכאן את הג'אווה הרלוונטית אליכם.



## התקנת אנדרואיד סטודיו

התוכנה בה כתבו עד לא מזמן וחלק מחברות ההייטק עדיין משתמשות בה, היא תוכנת הפיתוח בג'אווה Eclipse. גוגל יצרו תוכנת פיתוח ביתית וטובה בשם "אנדרואיד סטודיו". היא נכנסת חזק יותר לעולם פיתוח האפליקציות ועליה אסביר לכם במדריך. הורידו את האנדרואיד סטודיו והתקינו את כל הקבצים הנדרשים

בכונן C. לאחר ההתקנה, הפעילו את התוכנה.  
ממליץ להיעזר בסרטון שצירפתי על מנת להבין טוב יותר.



לפני תחילת השימוש בתוכנה – יש את הסרגל העליון –  
לחצו בו על האייקונים המסומנים הבאים:

- האנדרואיד עם החץ הכחול התחתון – יריץ על המחשב שלכם את כל ההתקנים והתוספות של מערכת האנדרואיד. לרוב יופעל עם הכניסה הראשונה לתוכנה.
  - האנדרואיד עם הסמארטפון הסגול – דרכו ניתן להגדיר אימולטורים בגדלים שונים. כך, ניתן לראות איך האפליקציה עובדת על נקסוס, טאבלט, גלאקסי, טלוויזיה וכו'.
- במידה ויש לכם מכשיר אנדרואיד, חברו אותו למחשב עם הכבל. היכנסו ללוח הבקרה. היכנסו למנהל ההתקנים. התקינו את המכשיר.

בכך למעשה סיימנו את התקנת סביבת הפיתוח לאנדרואיד על המחשב שלנו. במידה ויש לכם בעיות שצצו במהלך התקנת הג'אווה או האנדרואיד, ניתן ליצור קשר עם טל או להעתיק מילה במילה את ה error הכתוב בחיפוש ברשתות האינטרנט ולגלות את התשובה.

**שימו לב, הספר אינו תחליף למדריכים המוקלטים או לשיעורים  
אלא תוסף ליווי חיוני.**

## - שיעור ראשון - יצירת פרויקט ראשון

הרעיון הגדול שלכם לאפליקציה הבאה מתחיל ממש כאן.... לאחר שהתקנתם את הג'אווה ואת האנדרואיד סטודיו, זה הזמן להפעיל את התוכנה ולפלל שהכל עובד לכם כמו שצריך. במידה ונתקעתם בפתיחת התוכנה או באחד משלבי ההסבר שלי כאן – שרשרו תגובה כאן או בהודעה בפרטית ואשמח לעזור לכם. בשיעור היום, נלמד איך ליצור את הפרויקט הראשון שלנו בצורה הכי פשוטה וקלה שיש. גם אם אין לכם שום מושג בג'אווה או בפיתוח של דברים.

### הוספת המכשיר ואימולטור



לפני שנתחיל ליצור את האפליקציה, בשיעור המבוא הסברתי איך להתקין אימולטור שדרכו תוכלו לראות איך האפליקציה מוצגת על כל מכשירי האנדרואיד למיניהם. במידה ויש לכם מכשיר אנדרואיד, מצוין. חברו אותו למחשב, התקינו אותו דרך לוח הבקרה ובכל פעם שנריץ את האפליקציה תוכלו לראות את האפליקציה גם עליו.

### הקמת פרויקט ראשון

במידה והגעתם למצב בו התוכנה עלתה לכם, הכל הותקן ומעודכן – סימן שאפשר להתחיל.



בסרגל העליון, לחיצה על "FILE" לאחר מכן, לחיצה על "NEW PROJECT" ואנחנו מתחילים. נפתח חלון בו תמלאו את שם הפרויקט שלכם, שם סביבת העבודה שלכם ושם התיקיה בה יותקנו הקבצים. חשוב שהשם של

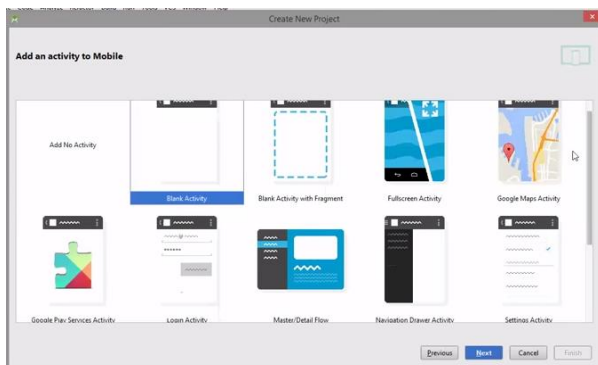
סביבת העבודה יהיה מקורי משלכם ולא example.com כפי שמגיע עם ההתקנה, מפני שגוגל לא מקבלים את זה ביום בו תרצו להעלות את האפליקציה לגוגל סטור.

לאחר מכן, נגיע לחלון גדול בו יש אפשרויות שונות לבחירת ה Activity שייפתח לכם את האפליקציה. נבחר בהתחלה ב"Blank Activity" ולא ב-Activity מפה, Activity-טאבים או כל Activity אחר.

אני אומר הרבה את המילה "**Activity**" מיד אסביר לכם מהי.

לאחר שבחרנו ב"Blank Activity", ניגש לבחירת שם לאותו Activity ראשון. במקרה הזה, ממליץ להישאר כך - "MainActivity". כלומר, ה Activity הראשי שייפתח לכם את האפליקציה.

לאחר שסיימתם, לחצו על "FINISH" והאפליקציה תיבנה לכם כחצי דקה של עבודה עצמאית של התוכנה. היכנסו לסרטון שלי ועברו שלב שלב איתי ביחד בדיוק כמו שפירטתי כאן.



## מהו Activity?

Activity זהו המסך אותו ייראה המשתמש. ברגע שאתם פותחים את אפליקציית Whatsapp, Activity הפותח הוא BlankActivity עם רשימה נגללת. ברגע שאתם פותחים את אפליקציית Waze, Activity הפותח הוא MapActivity עם מפת הדרכים שלכם. כלומר, Activity זהו הדף עליו נציג את האפליקציה. לכל אפליקציה יהיו כמה וכמה Activityים עליה נציג את המידע שלנו. נתחיל בתור התחלה, עם הראשון.

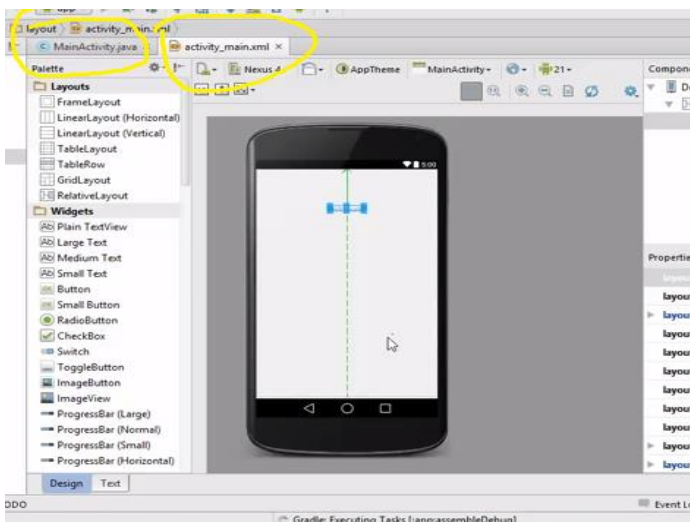
## Activity הראשון

ברגע שהתוכנה סיימה להתקין את האפליקציה שלנו עם Activity הראשי שהגדרנו לה, היא פתחה שני קבצים. האחד `MainActivity.java`, בו ניגע בהמשך. השני, `activity_main.xml`. עליו נעבוד כרגע. ברגע שנפתח אותו, נראה את המסך הלבן שבחרנו. לא מסך מפה, לא מסך כזה או אחר. Blank Activity – כפי שבחרנו – יוצר מסך לבן ונקי. עם טקסט אחד קטן עליו.

## "Hello World!"

ברגע שנלחץ בסרגל העליון על החץ הירוק, הוא יסרוק אילו אימולטורים או מכשירי אנדרואיד מחוברים למחשב עליהם הוא יכול להריץ את האפליקציה החדשה שלנו. במידה והתקנתם כפי שציינתי בשיעור הקודם ובתחילת השיעור הזה, הוא לא יעשה בעיות ויצג את האימולטורים ו/או המכשירים שמכיר. ביחרו אחד מהם, תריצו אותו והמתינו בסבלנות. תוך כמה רגעים והופ – האפליקציה הראשונה נוסדה לה. כן, בהחלט, קצת מעפנה. רקע לבן, בלי אפקטים, כיתוב לא רציני וגם הלוגו מעפן ביותר. אבל הי, התחלה!





בשיעור הבא, נלמד איך לגרום ל־Activity הזו לחיות קצת. נוסיף לו אלמנטים שונים כמו כפתורים, טקסטים ותמונות.

## - שיעור שני - עיצוב Activity

---

אחרי שבשיעור הקודם הבנו איך לפתוח פרויקט חדש עם Activity ראשון כדף פתיחה, בשיעור היום, נלמד איך לגרום לו להיות יותר "חי" ולעצב אותו.

### הוספת אלמנטי View שונים

**בסרגל הנפתח בצד שמאל** יש את כל האלמנטים השונים שאפשר לגרור ולהוסיף. כפתור, טקסט, תיבת טקסט, תמונה ועוד.

נסו לגרור אחד מהם לתוך הActivity שלכם ובעזרת הסרגל בצד ימין תעצבו אותו לבחירתכם.

### כתיבת אלמנטים בXML

כל מה שכתבתם וגררתם נכתב אוטומטית לXML של האנדרואיד. למשל, גררנו תיבת טקסט חדשה דרך הסרגל השמאלי. הגדרנו שהיא בגודל 30dp בצבע כחול דרך הסרגל הימני.

כעת, ניכנס לתוך קובץ הXML ונוכל לראות את הקוד כתוב בו. כמובן שאפשר גם לשנות ולהזיז, אולם למתחילים שבינינו מומלץ להשתמש באפשרות הגרירה הפשוטה שיצרו גוגל עבורנו.

בשיעור הבא, נגרום לכל האלמנטים שהוספנו לעשות פעולה מסוימת דרך קוד הג'אווה.

**ניתן לקרוא אודות כתיבה בXML במקורות הבאים –**



## תירגול

בלי להעביר את קטע הקוד לאנדרואיד סטודיו, ענו על השאלה הבאה:  
1. שרטטו על דף כיצד ייראה קטע הקוד הנ"ל.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <Button
        android:id="@+id/btn"
        android:text=""
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    >
    <ImageView
        android:src="@drawable/androideity"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
</LinearLayout>
```



## עיצוב מתקדם

ניתן לעצב ברמות מתקדמות הרבה יותר מאשר רק לזרוק כפתורים, תמונות, טקסטים. בסרגל השמאלי יש לנו הרבה מאוד אופציות.

נסו לבדוק בעצמכם תוך כדי שאתם **מנסים וטועים** מה יש לאנדרואיד סטודיו להציע.

# ניסוי וטעייה

## Layout

כעת, נבצע תירגול על אלמנטי layout השונים ונסביר כל אחד מהם.

הlayout זוהי משפחה של אלמנטים הנותנים לנו אפשרות "לזרוק" את האלמנטים השונים (הכפתור, הטקסט וכו') בצורה מסודרת ונוחה יותר. בצורה של טבלאות עבודה מסודרות. תמיד יהיה הlayout ברירת המחדל שהוא Relative Layout. ניתן לשנות אותו ולהחליף אותו לכל אחד מהבאים במשפחתו שלו. הכל על מנת שנוכל לשים את כל האלמנטים בצורה מסודרת יותר ומתאמת לכל מכשיר שהוא.

## - שיעור שלישי - הודעה בלחיצת כפתור

אחרי שהקמנו את הפרויקט עם קובץ XML יפה ומעוצב, עכשיו ניגש לתכנות. כפי שציינתי עם תחילת המדריך, גם מי שאין לו ידע בג'אווה – יוכל להצליח במידה ויעבור שלב שלב במדריך שלי. היום, נתחיל פרקטיקה וניגש לג'אווה עצמה.

### פונקציות ב-JAVA

ג'אווה זו שפת תכנות הפועלת עפ"י קלאסים (קבצי ג'אווה טקסטניים עם סיומת java). שמכילים פונקציות. ברגע שאתם פותחים פרויקט חדש "בלאנק Activity", האנדרואיד סטודיו יוצר עבורכם קובץ קלאס ג'אווה ראשי שמכיל פונקציות שונות. כל פונקציה מפעילה דבר אחר באותו העמוד באפליקציה שלכם. כל פונקציה נפתחת בשם שלה ומיד לאחר השם סוגריים מסולסלים שבהם נמצאים כל הפקודות והקודים השונים. לבינתיים, נתרכז אך ורק בפונקציה הראשית "onCreate" ואת שתי הפונקציות הנוספות שהקים עבורינו האנדרואיד סטודיו נמחק.

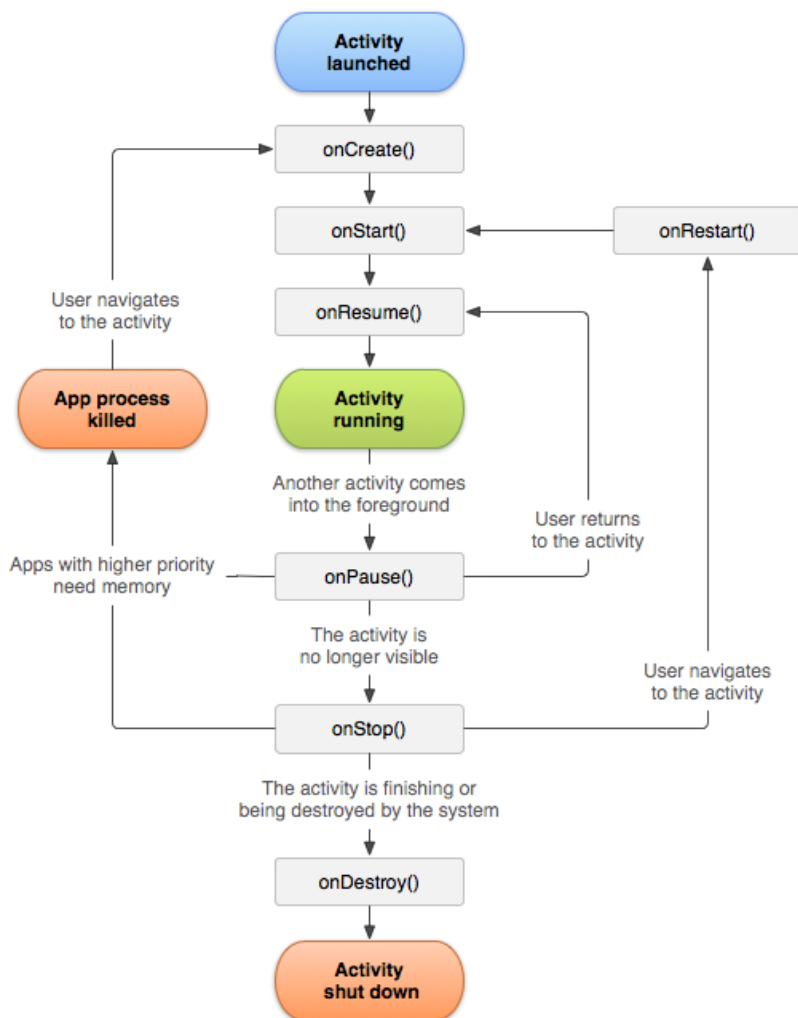
### מעגל החיים של Activity

ברגע שאנחנו יוצרים Activity חדש, הדבר הראשון שהוא מחפש זו פונקציה בשם "onCreate". לאחריה, באות עוד 5 פונקציות המעידות על "התחלה", "התנתקות", "הפסקה", "יציאה", "מחיקה מהזיכרון".

"onCreate" זו הפונקציה הראשונה ב"מעגל החיים" של Activity. עליה נעבוד לבינתיים ובה נכתוב את כל הפקודות. כל מה שיהיה כתוב בה, יקרה מיד עם פתיחתו של Activity שלנו.

במידה ואתם רוצים ליצור באפליקציה שלכם פונקציות ופעולות שונות ברגעים אחרים במעגל החיים של Activity, לדוגמת: פקודה ברגע שאתם יוצאים מהActivity/מפסיקים אותו/מוחקים

אותו מהזיכרון ועוד, אנא קראו על כך בגוגל. לא מורכב.



מצורף – מעגל החיים של הActivity.

## - שיעור רביעי - הקוד של לחיצה על כפתור

---

על מנת לקשר אלמנט מסוים ללחיצה, יש קוד ברור ופשוט. כל שעליכם לעשות הוא לדעת מהו ה-ID של אותו VIEW עליו אתם רוצים לעבוד. למשל, ה-ID של הTEXTVIEW שלנו הוא `.text_tal`.

כלומר נכתוב בתוך ה`onCreate` את המלל הבא:

```
findViewById(R.id.text_tal).setOnClickListener(new  
    View.OnClickListener() {  
        @Override  
        public void onClick(View v) {}
```

### *Toast*

בתוך הסוגריים המסולסלים יבוא הקוד הבא:

```
Toast.makeText(MainActivity.this,"Your  
    ;()Text!!",Toast.LENGTH_LONG).show
```

כך תוקפץ להם הודעה קטנה בדף למטה עם המלל שכתבתם בתוך הגרשיים.

במקרה הזה – Your Text!!

בשיעור הבא, נלמד איך מקשרים את הכפתור הזה ל-Activity חדש שנפתח.

1. כיתבו קוד פשוט בג'אווה בתוך הפונקציה onCreate() עם הודעה קופצת בעל המלל בעברית: "נלחצתי!"
2. כיתבו קוד פשוט בג'אווה בתוך הפונקציה onCreate() עם הודעה קופצת הסופרת כמה פעמים הכפתור נלחץ באפליקציה.
3. הקימו את כלל הפונקציות השונות של "מעגל החיים" (ראה איור עמוד 23). בכל פונקציה כיתבו טואסט חדש. בידקו במכשיר מתי כל טואסט קופץ לו.



## - שיעור חמישי - הוספת Activity חדש ומעבר אליו

---

אחרי שגרמנו לכפתור שלנו להילחץ ולעשות פעולה מסוימת בזמן לחיצה, נשמח לגרום לו להגיע ל-Activity חדש. בשביל זה, צריך קודם כל להקים אחד כזה חדש...

### הוספת Activity חדש

בלחיצת כפתור שמאלי בעכבר על התיקיה בה נמצאים קצבי הג'אווה, ייפתח תפריט. נבחר באופציה של יצירת Activity חדש => בלאנק Activity.

אותו התפריט שהיה לנו בהתחלה, עם היווצרות ה-Activity הראשון, נחשף גם עכשיו וניתן לבחור שם ל-Activity והגדרות נוספות. לאחר מכן, ברגע שאישרתם, "אנדרואיד סטודיו" יוצר עבורכם גם את קובץ ה-XML (בו נעצב את ה-Activity החדש עכשיו) וגם את קבצי הג'אווה (בו נכתוב את שורות הקוד).

### Intent

Intent זהו הכלי המקשר בין ה-Activityים ודרכו אפשר לשלוח ממקום אחד למקום שני. בואו נראה איך זה עובד בקוד: מי שהקוד נראה לו ג'יבריש, נא לחזור רגע לשיעור הקודם.

```
findViewById(R.id.text_tal).setOnClickListener(new
    View.OnClickListener() {

        @Override

        public void onClick(View v) {

            Intent intent = new Intent(MainActivity.this,
                MainActivity2.class);
            startActivity(intent);
```

}

MainActivity – זהו שם קובץ הג'אווה של ה-Activity הראשי והראשון. ממנו אני שולח את הבקשה להעביר דף. MainActivity2 - זהו שם קובץ הג'אווה החדש שיצרתי. כמובן שאת השמות אפשר לשנות לבחירה. Intent. זהו הכלי דרכו מעבירים מידע. מומלץ לצפות בסרטון ולהבין איך הכל עובד שלב אחר שלב.

ישנה קריאת קוד לסיום המסך בשם finish(). נשלב אותה גם בלחיצה על אלמנט כלשהו כדי לצאת מהמסך של האפליקציה.

## תירגול

התאימו בין כל הגדרה את קריאת הקוד הרלוונטית שנלמדה (משכו קו):

findViewById	מעבר בין מסכים
Intent	סיום מסך
Toast	קריאה לאלמנט מהXML
Finish()	הודעה קופצת

## - שיעור שישי - הוספת תמונה

אין אפליקציה מצליחה שהיא לא מעוצבת. גם אם זה אומר שאתם בעצמכם מעצבים אותה או משלמים למעצב חיצוני, חובה לעצב אותה ואת כל התמונות לשמור על גבי האפליקציה. אז איך עושים את זה?

### הוספת תמונה לתיקית האפליקציה

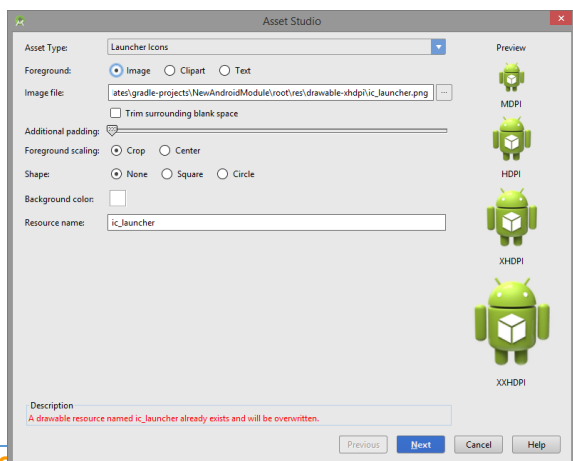
בקליק ימני על העכבר באחת מתיקות הסרגל השמאלי, ייפתח תפריט ובו אפשרות של יצירת תמונה.

ייפתח חלון ובו ניתן להכניס תמונה. ניתן להכניס תמונה מהמחשב, דרך מאגר תמונות של האנדרואיד סטודיו וגם דרך מלל.

במידה ואתם רוצים להוסיף דמויות ורקעים לאפליקציה, תבחרו באפשרות של ייבוא תמונה מהמחשב. תעלו את התמונה ותראו שהאנדרואיד סטודיו מסדר את התמונות לפי 4 גדלים ומתאים כל מכשיר שיוריד את האפליקציה שלכם לפי הגודל שנקבע.

### שינוי לוגו

באותו החלון שנפתח יש שורה בתחתית. בה נקרא לתמונה בשם. במידה ונרשום ic\_launcher התמונה שניצור תהפוך ללוגו.



בתוך הXML שלנו, נוכל לרשום בצורת קוד או לגרור תמונה מהסרגל השמאלי.  
הsrc מהווה את המיקום של התמונה. כאשר תמיד נרשום @drawable:/ ורק אז את השם של התמונה שלנו.

```
<ImageButton
```

```
    "android:layout_width="wrap_content
```

```
    "android:layout_height="wrap_content
```

```
    "android:src="@drawable/myImage
```

```
    />"android:id="@+id/btnImg
```

## - שיעור שביעי - משיכה מטקסט לטקסט

---

זה השיעור הראשון מתוך שניים המסבירים על טופס הרשמה באפליקציה.  
אחד הדברים שנהיה הכי מעוניינים באפליקציה שלנו זה בדיקת שם משתמש וסיסמא.  
בשיעור זה, נלמד למשוך את הטקסט שנכתב על ידי המשתמש בשדה ההרשמה ולהציג אותו בToast.

### משיכה לטקסט

לאחר שהגדרנו בקובץ הXML שיהיה לנו תיבת טקסט וכפתור – ניגש לקובץ הג'אווה שלנו.  
על מנת לקבל את המידע שהמשתמש רשם בתוך תיבת הטקסט, נרשום את הקוד הבא:

```
EditText edit = (EditText) findViewById(R.id.editText);  
edit.getText().toString();
```

שתי הפעולות של ("getText") ו("toString") הן קריאות בJAVA למשיכת הטקסט.

### הופעה על טקסט

לאחר שמשכנו את הטקסט נרצה להציג אותו בToast קופץ לצורך הדוגמא:

```
Toast.makeText(MainActivity.this,edit.getText().toString(),Toast.LENGTH_LONG).show
```

בשיעור הבא, נלמד במקום להציג בToast קופץ – ממש לשמור את הנתון שכתב המשתמש לתוך המכשיר ובכך ליצור טופס הרשמה שייזכור תמיד שם משתמש וסיסמא.

בשיעור זה, נלמד איך לבצע שמירת נתונים שהמשתמש מזין לאפליקציה בתוך המכשיר שלו. כמו, שם משתמש וסיסמא והצגה שלהם בעת הצורך.

### Sharedpreferences

"Sharedpreference" זהו השם של שיטת העבודה לשמירת נתונים על המכשיר. השמירה הינה שמירה אישית, בתוך המכשיר עצמו. היא יכולה לשרת אותנו באפליקציות של חידונים אישיים, הרשמה אישית בתוך המכשיר, משחק שנרצה לשמור בו שלבים. יש את המשחק "Venus" שבניתי והעלתי לגוגל סטור. בשלב זה אני בהחלט מאמין שאתם יכולים להתנסות עליו. זהו משחק שבניתי לפני מספר חודשים על מנת להציג שמירת נתונים על כל מכשיר בלי קשר לצד שרת וחיבור לאינטרנט.

המטרה ב"Sharedpreferences" היא שהמשתמש יזין את השם שלו לצורך העניין וכשיחזור לאפליקציה פעם נוספת השם שלו יופיע בגדול ויברך אותו.

### שמירה אל

ראשית, נכניס בתוך ה"onCreate()" שלנו את הקוד הבא:  
צפו בסרטון ההסבר.

```
SharedPreferences sp = getSharedPreferences("Test", 0);
```

```
SharedPreferences.Editor sedt = sp.edit();
```

בתוך הסוגריים הראשונות יש ביטוי בתוך מרכאות. בו נכתוב את שם האפליקציה שלנו או השם של ממסד הנתונים. עדיף לעשות את זה פשוט ולרשום שם אחד בכל האפליקציה. בכל Activity חדש, שימו את הקוד הנ"ל למעלה ב"onCreate()".

כעת, לאחר שהמשתמש הכניס שם משתמש וסיסמא לתוך 2 תיבות טקסט נמשוך את הנתונים, בדיוק כפי שעשינו בשיעור הקודם. נא להיזכר.

נשמור גם מספר (putInt) ונשמור גם מלל (putString). חשוב להבין את ההבדל כדי לא לקבל errors.

```
sedt.putString("User_Name", value2.getText().toString());  
;sedt.putInt("Password", value1.getText().toString())  
;sedt.commit
```

את הקוד הנ"ל נשים בתוך onClick של הכפתור שלנו, כך - רק ברגע שיילחצו על הכפתור הנתונים יישמרו. נציג את הנתונים בתוך טקסט שיש לנו:

```
TextView txt = (TextView) findViewById(R.id.txt);  
txt.setText(sp.getString("User_Name", ""));
```

זוה הכל. כעת, בכל פעם שהמשתמש ייכנס לאפליקציה הוא ייראה את מה שהזין בתיבת הטקסט.

עלינו כמה רמות בכתיבת קוד ובשיעור הבא נמשיך לעלות ברמה.



נסו לבנות אפליקציה שומרת נתונים פשוטה משלכם. נסו לבנות משהו דומה ל"Venus" שלי.



## - שיעור תשיעי - הצגת מידע ברשימה

---

בשיעור זה נלמד - הצגת מידע ונתונים בצורה הרבה יותר מסודרת לעין – הצגת המידע ברשימה.

### רשימה

ניגש לקובץ הXML שלנו ונזין בו מספר שורות:

```
<ListView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
        android:id="@+id/listView"
```

```
        android:layout_below="@+id/text2"
```

```
        android:layout_centerHorizontal="true" />
```

ניתן כמובן גם לגרור מהסרגל השמאלי רשימה וקובץ הקוד יכתב אוטומטית כרגיל.



Adapter בתרגום חופשי לעברית זה "מתאם".  
אכן כן, האדפטר הוא המתאם בין הרשימה שגדרנו בתוך הXML לבין המלל שנרצה להכניס.  
לכן בג'אווה נכתוב את הקוד הבא:

```
ArrayAdapter adapter = new ArrayAdapter(this,  
    android.R.layout.simple_list_item_1);
```

```
ListView list = (ListView) findViewById(R.id.listView);
```

```
    adapter.add("one");
```

```
    adapter.add("two");
```

```
    adapter.add("three");
```

```
list.setAdapter(adapter);
```

כך שבסופו של דבר נקבל רשימה עם שלושה מספרים כתובים –  
one, two, three. האחד אחר השני.

## לחיצה קצרה על אייטם

על מנת לקשר ללחיצה על אייטם נוסיף את הפונקציה הבאה:

```
list.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {

        @Override

        public void onItemClick(AdapterView<?> parent, View
            view, int position, long id) {

            Intent intent = new Intent(this, Activity3.class);

                startActivity(intent);

            }

        });
```

## - שיעור עשירי - בדיקות

---

בשיעור זה נשלים את הטופס שלנו ונבצע בו בדיקה. האם שם המשתמש והסיסמא תקינים? במידה וכן, ייכנס לאפליקציה. במידה ולא, נקפיץ Toast ושהמשתמש יזין מחדש.

### השוואה

על מנת לערוך השוואה בין שני נתונים נשתמשים במושג "equals".  
לצורך העניין המשתנה text הוא משתנה מסוג String. נבדוק האם text שווה למילה tal.

```
if (text.equals("tal")){  
  
    //What?}
```

### בדיקה פשוטה

בבדיקה הפשוטה ביותר נוציא לדרך שתי אפשרויות. אחת – אכן המשתנה שווה לערך שכתבנו. השניה – המשתמש טעה.

```
if (text.equals("tal")){  
  
    //What?  
  
    }  
  
    else {  
    }
```

## בדיקה מורכבת

עם זאת, נוכל לבצע בדיקה בתוך בדיקה ובדיקה לגבי מספר אלמנטים. כלומר -

```
if (text.equals("tal")){
```

```
    if (text2.equals("tal")){
```

```
        //What?
```

```
    }
```

```
}
```

לא לשכוח לשים else !

בנוסף, אפשר גם מספר אלמנטים. כלומר –

```
if (text.equals("tal") && text.equals("gal")){
```

```
    //&&
```

```
}
```

&& יסמן לנו ביחד. רק במידה וטקסט יהיה שווה גם לטל וגם לגל, אז הקוד ירוץ.

רוצים או טל או גל ? גם אפשרי. במקום && כיתבו || .

### מסך פתיחה

מסך פתיחה משמש בהרבה אפליקציות לשני דברים. האחד – מסך נוח ופשוט המציג ברכה קלה ללקוח עם לוגו החברה/האפליקציה. השני – טעינת כל האפליקציה בצורה חלקה יותר. כעת, נתרכז בכתיבת מסך פתיחה למטרת נוחות ללקוח. נכתוב קוד בו ניצור Activity שיוצג במשך 5 שניות ולאחר מכן ייסגר ויפתח אוטומטית ה Activity הראשי שלכם.

### קוד מסך הפתיחה

לאחר שיצרנו Activity חדש וקיבלנו גם קובץ JAVA וגם קובץ XML, נכניס ל XML את כל העיצוב הרלוונטי אליכם. הוסיפו תמונה נחמדה שתעצבו, כיתבו ב TEXTVIEW וכו'.

ניגש לקוד עצמו בקובץ ה JAVA:

בתוך הפונקציה onCreate נזין את הקוד הבא:

```
new Thread(new Runnable() {  
  
    @Override  
  
    public void run() {
```

```

        try {
            synchronized (this) {
                wait(2000);

                Intent intent = new Intent
                (MainActivity.this,Activity2.class);

                startActivity(intent);

                finish();

            }

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

    }.start();

```

נראה בתוך כל הסוגריים בפנים, Intent המוביל מהActivity הראשון של מסך הפתיחה לActivity החדש שלכם. בתוך הסוגריים של הפקודה wait, נשנה את כמות השניות בהן המשתמש יחכה. כרגע – 2 שניות.

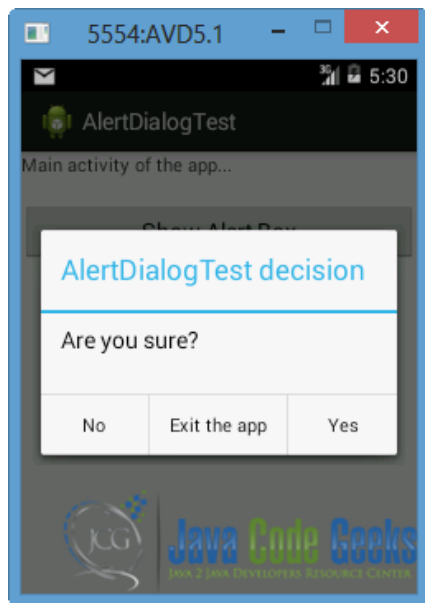
מסך פתיחה הוא רלוונטי לנוחות המשתמש ולכניסה רכה  
לאפליקציה שלכם, אף על פי שכרגע נכון למתחילים שבינינו אין  
צורך במסך פתיחה מהבחינה של טעינת האפליקציה

## - שיעור שתיים-עשרה - שאלה דיאלוגית

בהרבה מאוד אפליקציות משתמשים בשאלות המופנות כלפי הגולש. השאלות מוצגות בצורת תיבת טקסט קופצת עם 2-3 אפשרויות לבחירת הגולש. בואו נראה איך עושים את זה.

### שאלה דיאלוגית

השאלה הדיאלוגית תיראה בצורה הבאה:



ונכתוב אותה בקוד הג'אווה בתוך הonClick כך:

```
AlertDialog = new  
AlertDialog.Builder(this).create();
```



```

        alertDialog.setTitle("Dialog Button");

        alertDialog.setMessage("This is a three-
                               button dialog!");

        alertDialog.setButton(AlertDialog.BUTTON_POSI
                               TIVE, "Button 1 Text", new
        DialogInterface.OnClickListener() {

            public void onClick(DialogInterface
        dialog, int id) {

                //...

            } });

        alertDialog.setButton(AlertDialog.BUTTON_NEGA
                               TIVE, "Button 2 Text", new
        DialogInterface.OnClickListener() {

            public void onClick(DialogInterface
        dialog, int id) {

                //...

            } });

        alertDialog.setButton(AlertDialog.BUTTON_NEUT
                               RAL, "Button 3 Text", new

```

```

DialogInterface.OnClickListener() {
    public void onClick(DialogInterface
        dialog, int id) {
        //...
    }
});

```

ישנה אפשרות לוותר על אחד משלושת הכפתורים או אפילו על שניים. ניתן גם לוותר על שלושתם למען האמת, אולם אז האפקטיביות תיעלם לכם. לכן, מומלץ על שתיים. אחת "אישור" ואחת "ביטול/חזור למסך הקודם".

### ביטול תיבת השאלה עם הופעתה

- במידה ונלחץ על כפתור האנדרואיד שבתחתית המסך "חזור" נוכל לבטל את התיבה ולחזור חזרה. ניתן להוסיף פונקציה בתחתית המסך עם הקוד הבא, דבר שיבטל את מקש החזור ובלחיצה עליו לא יקרה דבר והתיבה לא תיעלם לגולש

```

@Override:

public boolean onKeyDown(int keyCode, KeyEvent
    event) {

    if ((keyCode == KeyEvent.KEYCODE_BACK)) {

    }

    return false;
}

```

## - שיעור שתיים-עשרה - התראות

---

כל אפליקציה מרובת גולשים והורדות, תאהב להשאיר את משתמשיה ולקוחותיה תמיד מחוברים לאפליקציה. בנוסף, תרצו להראות שהאפליקציה מבצעת עדכונים מדי פעם והיא עדיין חיה, נושמת ובוועטת.

בשביל הדברים האלה – יש את המושג "notification" או בעברית "התראות".  
על מנת לבצע התראה משתנה כל זמן מה, נצטרך ידע בצד שרת. בקורס זה אלמד אתכם אודות צד הלקוח בלבד, ללא עזר ושימוש בענן או צד שרת כלשהו.  
עדיין, נלמד איך לכתוב קוד ייצור להתראה קופצת ברגע שהלקוח ייצא מהאפליקציה שלכם.

### התראה קופצת

בכל פעם שהלקוח ייצא מהאפליקציה, תהיה הודעה קופצת עם מלל של "אנא ממך חזור במהרה" ונוסיף גם את אייקון האפליקציה שלכם ליד.

```
final NotificationManager mgr =
```

```
(NotificationManager)
```

```
Main.this.getSystemService(Context.NOTIFICATION_S  
SERVICE);
```

```
Notification note = new  
Notification(R.drawable.ic_launcher,
```

```
"Adventure castle",
```

```

System.currentTimeMillis());

PendingIntent i =
PendingIntent.getActivity(Main.this, 0,
new Intent(Main.this, Main.class),
0);

note.setLatestEventInfo(Main.this, "Even if you
went during the game -",
"He saved you from the last. Back soon!", i);

mgr.notify(1337, note);

```

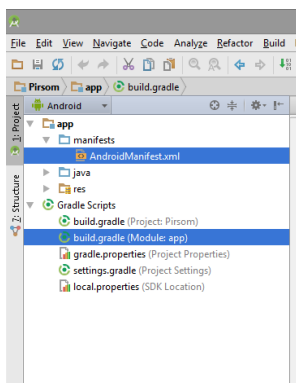
את כל הקוד הזה נסו לשתול בדיוק כמו שהוא ב onCreate של מסך הסיום שלכם/מסך הפתיחה לשם הניסוי הראשוני ושימו לב להודעה הקופצת בראש המסך.

הדבר נחמד ביותר, מוסיף טעם לאפליקציה ותמיד מזכיר לכל לקוח שאתם עדיין קיימים.

\*ניתן גם לבצע התראות שוטפות. כלומר, דרך צד שרת להקפיץ הודעה לכל מי שיש בטלפון החכם שלו את האפליקציה שלכם. נילמד על כך בנספח ב' צד שרת.

## - שיעור שלוש-עשרה - מניפסט והגדרות

ישנם שני קבצים עיקריים אותם עליכם להכיר בעת בניית האפליקציה.  
האחד שמו מניפסט. בתרגום חופשי לעברית: מנשר או מניפולציה.  
קובץ טקסטי המתאר את הActivityים השונים שיש באפליקציה,  
מה לוגו האפליקציה ועוד הגדרות נלוות.  
השני שמו gradle והוא מתאר את הגדרות המשתמש כגון מה  
שם תיקיית האפליקציה, מה המקס' והמינ' של SDK של  
האפליקציה, מהם הקבצים הנלווים לאפליקציה וכו'.



לאט ובזהירות נשתלט על שני  
הקבצים הללו.

שימו לב, כך ניתן להיכנס אליהם  
דרך הסרגל השמאלי:

### המניפסט

בתוך הקובץ הראשון שלנו – המניפסט – נתעסק במה הלקוח  
ייראה בעת הגלישה באפליקציה.  
אילו Activityים יש בה? מה העיצוב שלכם ומה הלוגו. וכדומה.

כל Activity שלכם שלא יהיה רשום שם, פשוט לא יופיע לגולש.  
ואף יותר מכך – האפליקציה תקרוס לכם. היא תחפש את  
Activity דרך המניפסט ולא תמצא.

ניתן גם להוסיף כמה שורות קוד במניפסט, לדוגמא:

```
uses-permission >  
android:name="android.permission.WRITE_EXTERNAL_STORAGE"  
</ "L_STORAGE
```

```
uses-permission >  
</ "android:name="android.permission.INTERNET
```

```
uses-permission >  
android:name="android.permission.ACCESS_NETWORK_STATE"  
</ "RK_STATE
```

כל שורה כזו היא "הרשאה". נסו לשלב את השורות הללו באפליקציה שלכם ועכשיו תהיה תמיכה באינטרנט. רישמו בגוגל uses-permission ותקבלו רשימה של כל ההרשאות. יש דברים מיוחדים כמו למשל, תמיכה ברטט באפליקציה שלכם ושליטה במצלמה של הלקוח בכל עת שתרצו.

הרשאות יכולות להיות דבר מסוכן לגולשים...

בנוסף, ישנה שורה אחת:

```
" android:theme="@style/Theme.NoTitleBar
```

נסו למחוק את כל מה שכתוב מאחורי סימן השווה וללחוץ על `ctrl` ורווח יחד.

יש לכם עשרות אפשרויות של סגנונות עיצוב שונים וטובים. נסו לבדוק. אנחנו ממליצים על הכתוב למעלה.

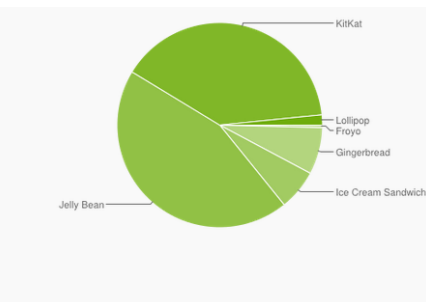
הקובץ השני הוא gradle. הוא אחראי על מאחורי הקלעים. נניח והאפליקציה שלכם בעלת אנימציות? לא תוכלו להציג אותה בגוגל סטור בפני כל משתמשי האנדרואיד לדורותיהם. מפני האנימציות נכנסו לתוקף רק מגרסא 9. אם יש לכם אפליקציה פתוחה ואתם רוצים לפתוח אותה לכולם – בקובץ הXML של Gradle נשנה את הכתוב:

```
defaultConfig {  
  
    applicationId "com.tal.pirsom"  
  
    minSdkVersion 1  
  
    targetSdkVersion 21  
  
    versionCode 2  
  
    versionName "1.1"  
  
}
```

מינימום 1 , מקסימום 21. כך כולם יוכלו להנות מהאפליקציה שלכם.

**שימו לב בשנית**, הרבה פקודות לא יעבדו והאפליקציה שתתמוך בכל המכשירים תהיה עם ממש מעט פונקציות. ממליץ על גרסא 9 כמינימלית. לפניכם רשימת הגרסאות:

Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	7.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	6.4%
4.1.x	Jelly Bean	16	18.4%
4.2.x		17	19.8%
4.3		18	6.3%
4.4		19	39.7%
5.0	Lollipop	21	1.6%



מלבד זאת, ניתן גם לכתוב:

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

כלומר מהם הקבצים הנלווים לאפליקציה. תוכלו לחקור על כך בגוגל.  
 לצורך העניין אתן לכם דוגמא, אם נוסף את השורה הבאה בתוך הסוגריים המסולסלים:  
`'+.compile 'com.parse.bolts:bolts-android:1`  
 ובנוסף, גם נשים את קובץ ה-`*.jar` של `parse.com` בתיקיה של `gradle` - נקבל צד שרת באפליקציה!  
 קצת מורכב? לא עם המדריכים שלי.



בפרקי הנספחים מומלץ לצפות קודם ורק לאחר מכן מומלץ לקרוא ולהיעזר בנספח.

ראשית כל, נקרא לתמונה/טקסט/כפתור בשמו:

```
ImageView aniView = (ImageView)
;findViewById(R.id.imageView1)
```

לאחר מכן, נגדיר לאותו aniView תזוזה בציר ה Y ו־FADEIN. זהו הקוד:

```
ObjectAnimator mover =
ObjectAnimator.ofFloat(aniView, "translationY", 400f,
0f);
```

```
mover.setDuration(3500);
```

```
ObjectAnimator fadeIn =
ObjectAnimator.ofFloat(aniView, "alpha", 0f, 1f);
```

```
fadeIn.setDuration(3500);
```

```
AnimatorSet animatorSet = new AnimatorSet();
```

```
animatorSet.play(fadeIn).with(mover);
```

```
animatorSet.start();
```

במידה וברצונכם לשנות את ציר ה Y לציר ה X רישמו :  
translation.  
במידה ואתם רוצים לשנות את ה FADEIN ל FADEOUT –  
החליפו בין 0f ל 1f.  
במידה ואתם רוצים אך ורק דבר אחד מבין השניים, הסירו אחת  
משתי שורות הקוד.

## החלפה כל שתי שניות

סוג אנימציה נוסף זו החלפה בכל שתי שניות של התמונה  
עצמה/הרקע שלה.  
כך יכולה להיווצר הרגשה של תזוזה על המסך. הקוד הוא להלן,  
שלבו גם אותו כרגיל ב onCreate:

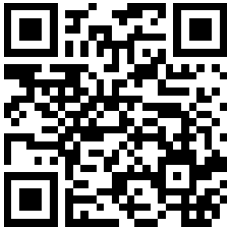
```
boolean whichColor = true;  
  
;Button btn = (Button) findViewById(R.id.btn)  
  
new Thread(new Runnable() {  
  
    public void run() {  
  
        while (true) {  
  
            try {  
  
                Thread.sleep(100);  
  
            } catch (Exception e) {  
  
                e.printStackTrace();  
  
            }  
  
        }  
  
    }  
  
})
```

```
        }  
        if (whichColor)  
            btn.setBackgroundColor(Color.BLACK);  
        else  
            btn.setBackgroundColor(Color.BLUE);  
        whichColor = !whichColor;  
    }  
    }  
}).start();
```

## נספח 2: צד שרת

### בפרקי הנספחים מומלץ לצפות קודם ורק לאחר מכן מומלץ לקרוא ולהיעזר בנספח.

בפרק זה טל מלמד אודות צד השרת. כיצד ניתן למשוך סטרינג מסוים וכיצד ניתן לשלוח סטרינג מסוים.

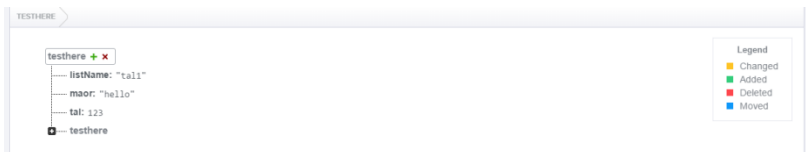


איך ניתן לבצע שם משתמש וסיסמא. לאחר מכן בדיקה לנתונים.

בעבר, השתמשנו בפרק זה בשימוש חנימי של פייסבוק שנקרא parse.com, אולם הוא ניסגר. לאחר מכן התקדמנו ל-firebase.

כיום, אין אפליקציה שלא יכולה להרשות לעצמה שלא להתחבר לצד שרת ולמשוך נתונים שונים. למשל, כל מנצח רושם את עצמו ברשימה ארצית.

ניתן לבצע זאת בקלות דרך firebase.



נלחץ על ה(+) ונוסיף אלמנטים שונים. בעזרת בדיקת if קלה (ראה פרק "בדיקות").

```

13 import com.firebase.client.DataSnapshot;
14 import com.firebase.client.Firebase;
15 import com.firebase.client.FirebaseError;
16 import com.firebase.client.ValueEventListener;
17
18 public class MainActivity extends AppCompatActivity {
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.content_main);
24
25         Firebase.setAndroidContext(this);
26         Firebase ref = new Firebase("https://testthere.firebaseio.com/");
27
28         ref.addValueEventListener(new ValueEventListener() {
29             @Override
30             public void onDataChange(DataSnapshot snapshot) {
31                 if (snapshot.child("maor").getValue().equals("hello")){
32                     // OK
33                 }
34             }
35             @Override
36             public void onCancelled(FirebaseError firebaseError) {
37             }
38         });
39
40         ref.child("listName").setValue("tall"); // add
41
42
43     }
44 }

```

1. בשורה 25 זיין את שורת ההוראה.
2. בשורה 26 זיין את ה Link של כתובת השרת שניתן לנו ב firebase.
3. נבצע בשורות 28-38 בדיקה בתוך ליסטנר. נוכל במקום שורת הקוד בהערה //OK לרשום Toast לצורך העניין או Intent לעבור הלאה. נוכל להחליף את הסטרינג "maor" בסטרינג של editText (ראה פרק משיכה מטקסט לטקסט).
4. בשורה 40 נוסיף עמודה לצד השרת מתוך האפליקציה.

כמובן, שזאת לאחר הוספת שורת gradle הרלוונטית ושורת  
ההרשאה (פרמיישן) הרלוונטית. (ראה פרק 'מניפסט והגדרות').

### בפרקי הנספחים מומלץ לצפות קודם ורק לאחר מכן מומלץ לקרוא ולהיעזר בנספח.

לאחר שאנחנו מרגישים מוכנים ניתן לבצע העלאת APK מהאפליקציה לגוגל פליי. ישנו תהליך מאוד מסודר, אנא צפו ראשית בפרק האינטרנטי.

1. לחיצה על לחצן הסרגל העליון BUILD.
2. לחיצה על 'Generate Signed APK'.
3. יצירת key store חדש.
4. לחיצה על שלושת הנקודות בצד בחלונית החדשה שנפתחה, ויצירת JAR.
5. מילוי הפרטים והרצה.
6. הרשמה לגוגל פליי (חיפוש בגוגל: "publish google play") והעלאת האפליקציה הראשונה שלכם.



ימין

