



# פועלות

מצגת מס' 6

## מה למדנו בפרק הקודם?

- מהו ביצוע חוזר - לולאה
- מתי נשתמש בלולאות
- סוגי לולאות
- מבנה של לולאות
- שילוב רכיב שעון באפליקציה

## נושאי השיעור

- אתגרים מורכבים
- פעולות
- זימון פעולה עם ו בלי פרמטרים
- רכיב Notifier
- סנסור ACCELERATOR

## אתגרים מורכבים - ניהול זמנים

נניח ואתם עורכים שיפוץ בביתכם.

יש המונן חלקים בבית לצרכים לשיפוץ - חשמל, פרקט, דשא בגינה, שיפוץ המקלחת והשירותים, ארוןות מטבח והחלפת דלתות.

ניהול כל העבודות הינו קשה, במיוחד שלכל עבודה צריכים להגיע אנשי עבודה המתמחים בסוג העבודה המוסויים.

מה שנעשה הוא לפרק את כל השיפוץ לעבודות משנהיות בתחום השיפוץ.

air זה יראה ??



## אתגרים מורכבים - ניהול זמנים

כפי שאמרנו קודם - ניקח את כל עבודות השיפוץ במקשה אחת.



## אתגרים מורכבים - ניהול זמנים

נפרק את כל השיפורים הגדול לעבודות נפרדות, כל דבר יעשה בנפרד.



## אתגרים מורכבים ניהול זמנים

כאשר נפרק את כלל השיפוץ לעבודות נפרדות יהיה לנו קל יותר לנוהל את כלל השיפוץ אם זה בתכנון הזמניים וניהול אנשי המבצע.

לכל עבודה תהיה את הזמן שלה ואנשי העבודה שלה.

**בעולם התכניות - עבודה השיפוץ היא התכנית הראשית שלנו, בכל פעם התכנית שלנו(עבודת השיפוץ) תזמן עבודה אחרת לשיפוץ הבית (פועלה)**



## מהי פועלה?

לעתים, האפליקציה שאנו מפתחים מבצעת משימות מורכבות כך שלכל משימה יש תת-משימות.

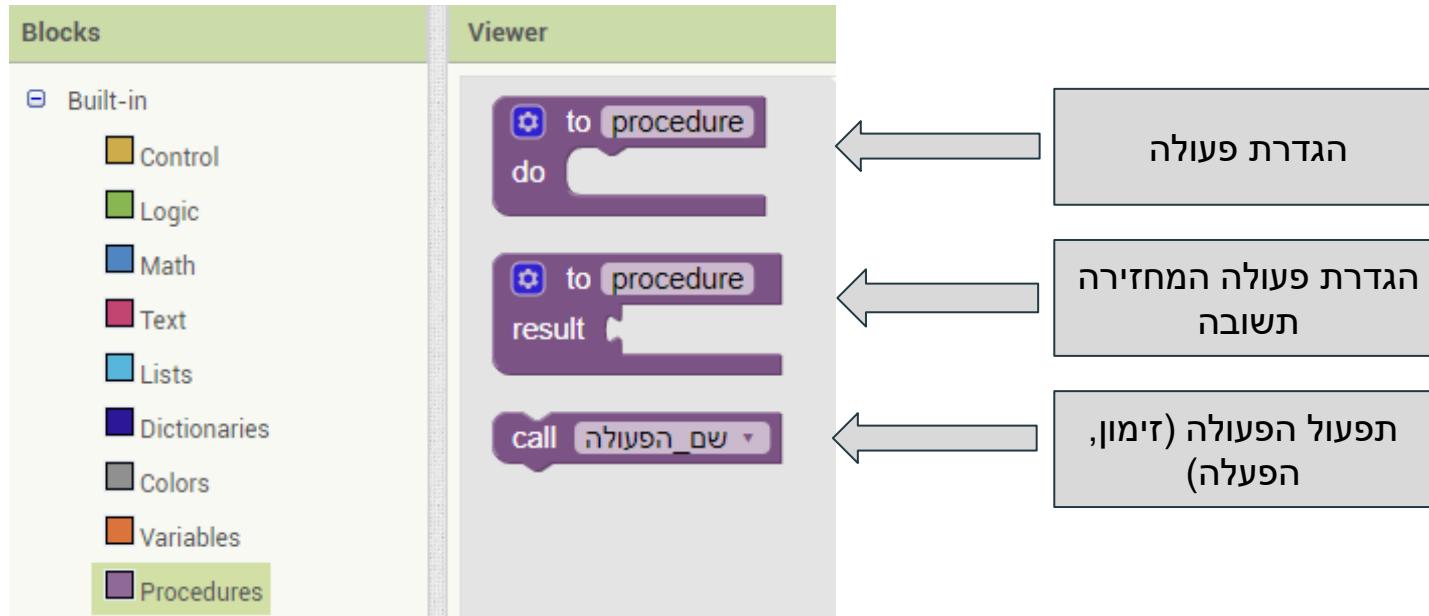
כדי להתמודד עם משימה מורכבת וגדולה נוכל לחלק כל משימה גדולה ומורכבת ולחולק אותה לחתת-משימות קטנות.

תתי המשימות נקראות **פעולות**.

לכל תת-משימה - פועלה יש מטרה, אופן ביצוע וזמן ביצוע (מתי תקרה).

דוגמא לפועלה: האם מספר מתחלק ב-5 ללא שארית (נעזר באופרטור מודולו).

על מנת להשתמש בפעולות נשתמש בתחום קטgorיה Procedure במסך ה-Blocks,  
ניצור פרוצדורה (פעולה) ונוכל לפעול אותה.



## הגדרת הפעולה



נגדיר פעולה עם שם ייחודי ובתוכה נגדיר מה מטרת הפעולה.

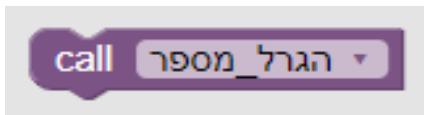
בדוגמה מטה, נרצה שבפעולה יוגרל מספר אקראי בין 1-50, שערך המספר שהוגרל יהיה פור לערך המשתנה.



## זמן הפעולה

בעת הגדרת הפעולה נגדיר מה מטרת הפעולה, על מנת להשתמש בפעולה נצטרך לזמן את הפעולה.

איך זמן פעולה?



נגרור את הבלוק call כדי לזמן את הפעולה, נגרור את הבלוק למקום בו נרצה שהפעולה תתרחש.

## תרגיל ביתה

צרו פרויקט שבו Button ו-Label שבתוכנת הטקסט שלו רשום "Procedure".  
רשמו בתוכנת הטקסט של ה-Button "Click Me".

מטרת הפרויקט להשתמש בפעולה שmagical\_number Akrai בטווח של 40-15 בעת לחיצת כפתור.

בנוסף, על המשימה לשנות את גודל הטקסט של ה-Label למספר שעלה בהגרלה!



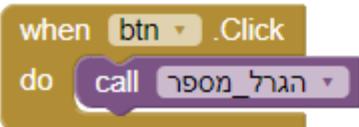
```
initialize global [num] to [0]
```

נאותחל משתנה עם ערך 0.



```
when green flag clicked
  [set [global\_מספר] to [0]
  do
    [set [global num] to [random integer from [15] to [40]]
    set [result] to [get [global num]]
    change [FontSize v] by [1]
    say [result] for [1] seconds
  end]
```

נגידר את הפעולה שלנו, הפעולה תגריל מספר בין 15-40 ותשנה את גודל הטקסט ב-Label למספר שעלה בהಗלה.



```
when [btn v].Click
  do
    [call [הgel_מספר v]]
```

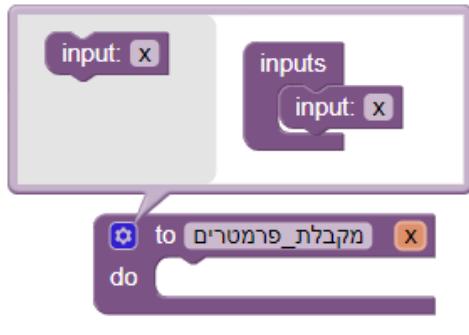
בכל פעם אשר נלחץ על הכפתור  
נקרא ל פעולה והיא תתבצע.

## פעולה המקבלת ערכים

עד בה ראיינו הגדרה ל פעולה אשר מבצעת מה שנכתב לה לבצע, אך היא לא מקבלת נתונים בעת הגדרתה.

נוכל להגיד פעולה אשר בעת הגדרתה תקבל נתונים/פרמטרים (טקסטים, מספרים וכו').  
לדוגמה, נוכל להגיד פעולה שתתקבל בהגדרתה את המספר 22 והמספר 7.

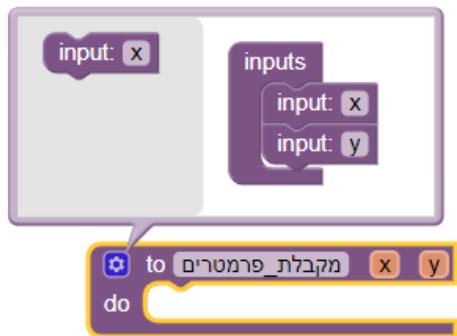
ה פעולה תוכל להשתמש או לא להשתמש במספרים, נהוג להשתמש בפרמטרים שמקבלים פעולה בעת הגדרת פעולה מקבלת נתונים.



וככל שהגדר פועלה עם כמה פרמטרים שנרצה.

על מנת להוסיף פרמטרים נוספים על גלגל השינויים וונגרור `input` לתוך `inputs`.

כל נתון ישמש כמשתנה באפליקציה שלנו.



## תרגום כיתה

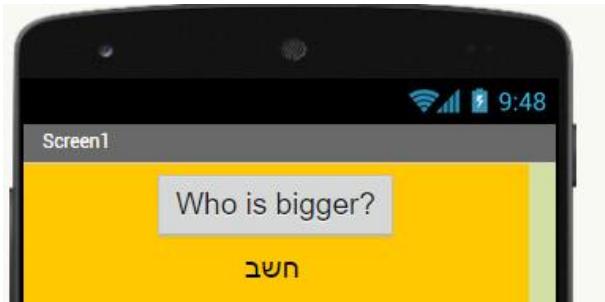
צרו פרויקט שבו 2 משתנים בעלי ערך כלשהו (אם בטור משתנים סטטיים או בקלט בתיבת טקסט).  
גררו למסך כפטור ושנו את שמו לשם `who is bigger`.

בנוסף, גררו `Label` שבו תכונת הטקסט ריקה.

לבסוף, גררו פעולה למסך `the-k-Block` וקרויה לה "חשב".

על הפעולה לקבל את 2 המשתנים, בעת לחיצת כפטור על הפעולה לבדוק איזה מι מהמשתנים גדול יותר ולרשום הודעה מתאימה.

**נסו בעצמכם!**



נבחר כפתרו + ליבל.  
נשנה את תכונות הטעסקט שלהם

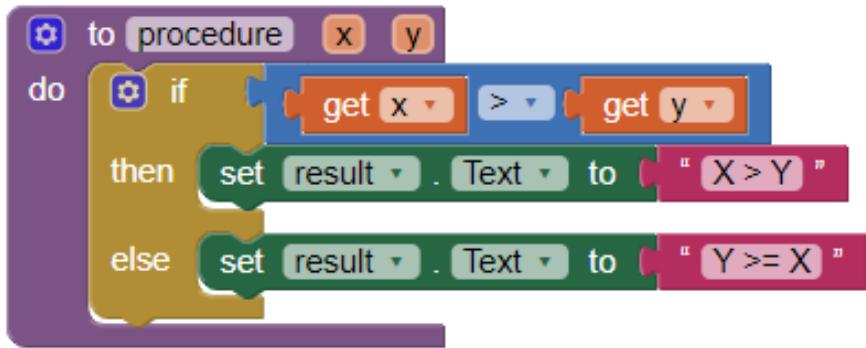
initialize global [x] to [22]

initialize global [y] to [18]

when [btn].Click  
do [call procedure [ ]]  
  [x get [global x]]  
  [y get [global y]]

נאחל שני משתנים סטיים וניתן להם  
ערך

בעת לחיצה על כפתרו נקרא לפועלה ונשלח  
אליה את X ו-Y.  
שים לב - לפני כן נוצרה להגדר את הפועלה.



נגיד את הפעולה עם משתנים X ו-Y.  
ນבדוק האם  $Y < X$  ונדפיס הודעה אם כן.  
אחרת נדפיס הודעה אחרת.

## פעולה המחזירה ערך

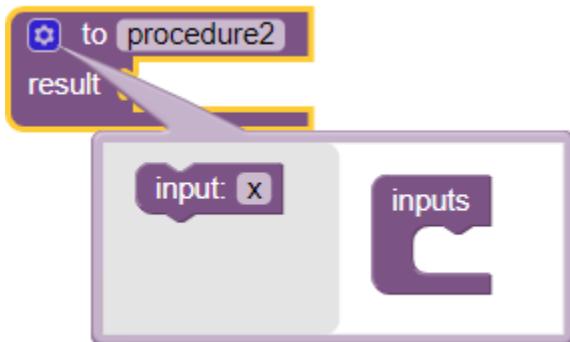
עד כה רأינו פעולות שמודדות ללא פרמטרים ועם פרמטרים.

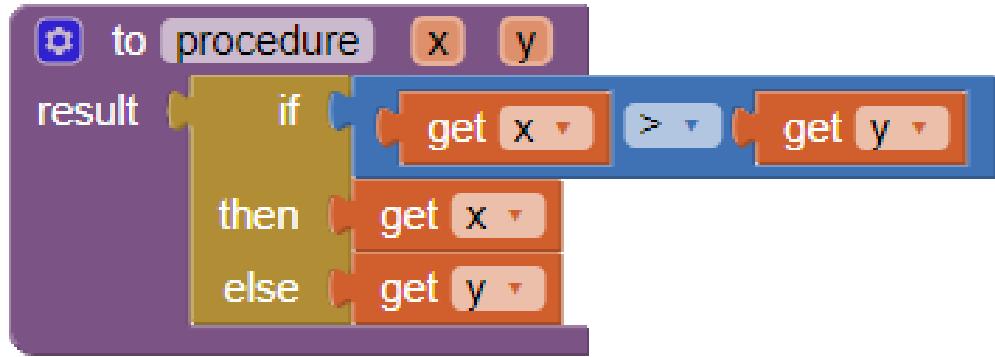
פעולה יכולה לקבל פרמטרים וגם להחזיר תשובה בסופה.

לדוגמא, ניקח את הדוגמא הקודמת, הפעולה קיבלה את המספרים 22 ו-7, תבדוק מי המספר הגדל מביניהם ותחזיר את המספר 22.

## איך נראהות פעלת המחזיר ערך?

ההבדל הוא שלעומת הפעולות הקודמות בפעולה זו קיבל תשובה - .result



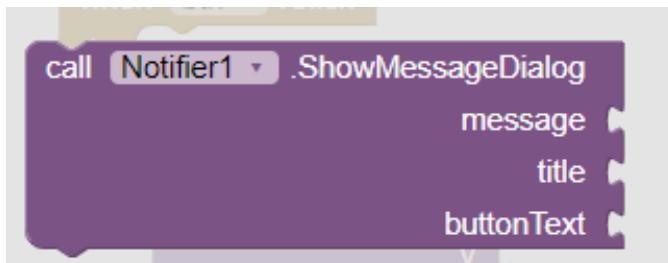


נבדוק מי הערך גדול יותר,  
ונחציר מהפעולה את הערך  
הנכון.

## רכיב Notifier



רכיב Notifier הוא רכיב המאפשר לנו לשלוח הודעה או אזהרה למסך. רכיב זה הוא רכיב בלתי נראה, כלומר אחראי שngror אותו למסך לא נראה אותו ויזואלית.



בדרך כלל נשתמש ברכיב עם תכונה זו. בתכונה זו הרכיב מורכב מההודעה עצמה, כותרת וכפתור. בכלל אחד מהם יהיה בתוכו טקסט.

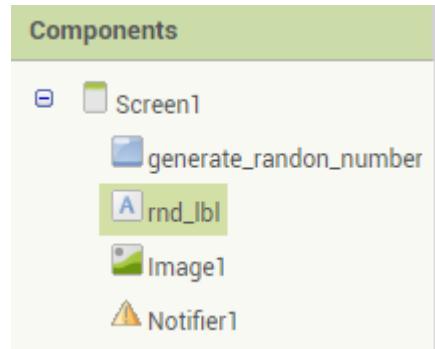
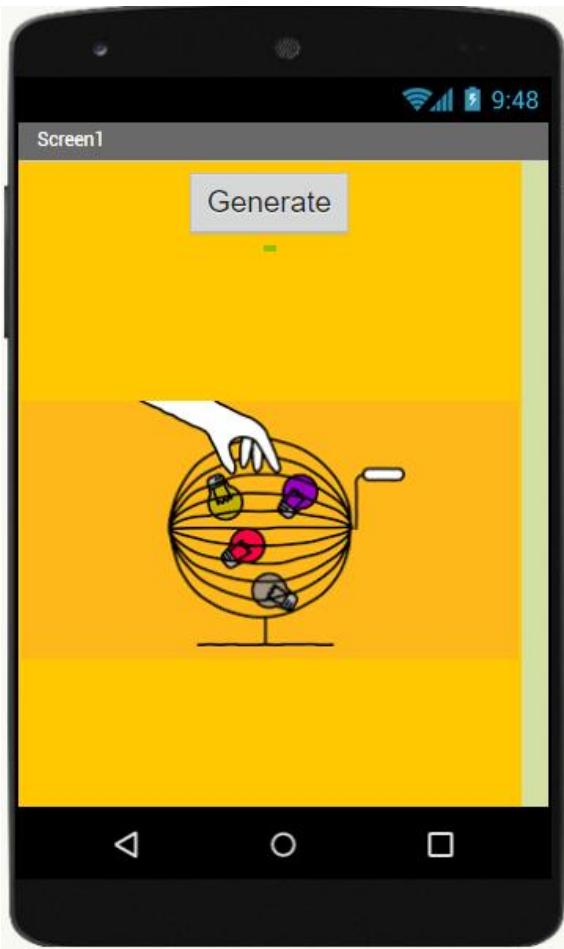
## תרגיל ביתה

צרו פרויקט שבו : Button, Label ו- Notifier .  
בעת לחיצה על כפתור יוגרל מספר אקראי בין 1-100.

על התוכנית להציג את המספר שהוגרל ב-Label .

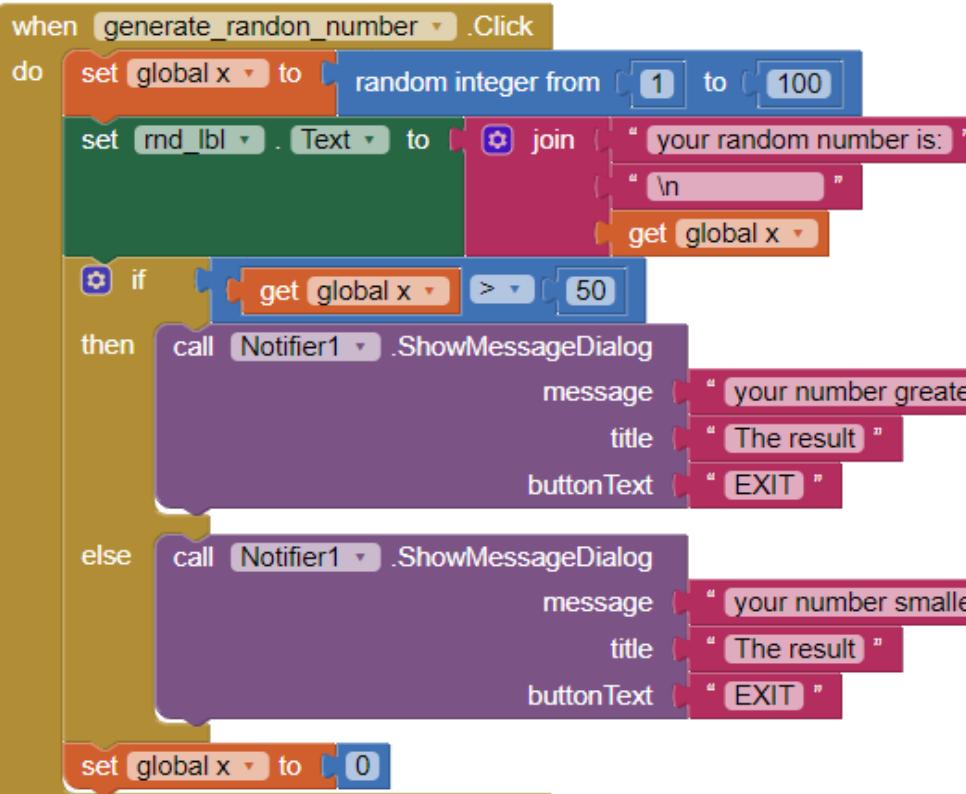
בנוסף, אם המספר שהוגרל גדול מ-50 רכיב ה-Notifier יציג הודעה שהמספר גדול מ-50, אחרת  
ייציא הודעה שהמספר קטן שווה מ-50 .

ניתן להוסיף תמונה לאפליקציה : ) .



נוסיף את הרכיבים הללו, נשנה את השמות ותכונות הטקסט.

נתחל משתנה ל-0

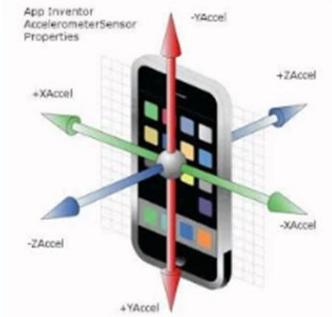


נraiיל מספר אקראי למשתנה X שלנו.

נציג את המספר שהוגרל באמצעות פעולה איחוד.

אם המספר גדול מ-50, רכיב ה-Notifier יקבע הודעה מתאימה למסר.

אחרת, כלומר המספר קטן שווה מ-50, רכיב ה-Notifier יקבע הודעה מתאימה למסר.



# סנסור ACCELERATOR

רכיב המאפשר לאפליקציה לחוש בתנועות ושינויים.

לדוגמה, הסנסור יכול לזהות Shaking של המסך.

. הרכיב נמצא תחת תת תפריט Sensors, הרכיב שנבחר למסך הוא : AccelerometerSensor .

הרכיב הוא רכיב בלתי נראה, בולם לאחר שנבחרו אותו למסך לא נראה אותו ויזואלית.

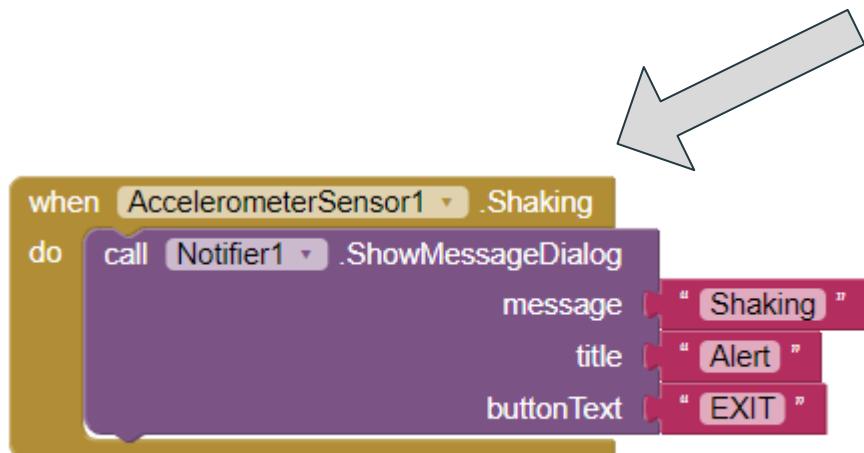
## Non-visible components



AccelerometerSensor1

## דוגמא

ניצור תכנית פשוטה שמזזה ניור ושינוי ציר Z , X של המבשיר שלנו.



## סיכום - מה למדנו?

- בעזרת פועלות נוכל לנהל את כל הקוד בצורה נקייה ומסודרת
- נוכל לזמן פעולה עם ובי פרמטרים בהתאם למה שנוצר
- נוכל לשלב רכיב Notifier שיאפשר לנו לשלוח הודעה או אזהרה למרס

## תרגיל מסכם

צרו פרויקט שמטרתו להיות "משחק הניחושים". מטרת הפרויקט להגריל מספר בין 1-50, על המשתמש לנחש מה המספר שהוגרל ב-5 ניחושים. בכל ניחוש, האפליקציה תודיע למשתמש (בעזרה רכיב `Notifier`) אם המספר שהוגרל שווה למספר שהוגרל, קטן ממנו או גדול ממנו. לאחר 5 ניסיונות ניחוש, אם המשתמש לא הצליח לנחש את המספר אז נגלה למשתמש מה המספר ונציג לו הודעה כישלון.

THAnK  
y ou