

# דף עבודה – פועלות (Procedures)

שם התלמיד/ה: \_\_\_\_\_

תאריך: \_\_\_\_\_

בכל תרגיל:

1. צרו פרויקט חדש בשם המבוקש.
  2. תכנו את המסר ב-**Designer**.
  3. כתבו את הלוגיקה ב-**Blocks**.
  4. הריצו ובדקו שהכל עובד.
- 

## תרגיל 1 – פועלה שמחילה טקסט

שם הפרויקט: ProcHello

1. במסמך **Designer**:

- הוסיף **Label** עם טקסט התחלתי: "לפנִי פועלה".
- הוסיף **Button** עם טקסט: "הפעל פועלה".

2. במסמך **Blocks**:

- צרו פועלה **sayHello** ללא פרמטרים בשם **sayHello**.
- בתוכה – שנו את הטקסט של ה-**Label** ל: "שלום מהפועלה!".
- באירוע **Click** של **Button** – קראו לפועלה **sayHello** **when Click**.

שאלה בדיקה:

מה יופיע ב-**Label** אחרי לחיצה על הכפתור?

תשובה: \_\_\_\_\_

---

## תרגיל 2 – פועלה שmagila מספר ומפנה גודל טקסט

(דומה לתרגיל הכתיבה במצגת, בגרסה פשוטה)

מצגת 6 - פועלות

שם הפרויקט: RandomSizeProc

1. במסמך **Designer**:

- הוסיף **Procedure** בשם **RandomSize** עם טקסט: "RandomSize".
- הוסיף **Button** בשם **Click Me** עם טקסט: "Click Me".

2. במסמך **Blocks**:

- צרו משתנה גלובלי `num` עם ערך התחלתי 0.
  - צרו פעולה ללא פרמטרים `:setRandomSize`
  - הפעולה מגירה מספר שלם בין 15 ל-40.
  - שומרת את המספר ב-`num`.
  - משנה את `FontSize` של ה-`Label` לערך `num`.
  - באירוע `Button.Click` – קראו לפעולה `.setRandomSize`.
- 

## תרגיל 3 – פעולה עם פרמטר אחד (ברכת שלום)

שם הפרויקט: `HelloNameProc`

- :`Designer` .1
    - `TextBox – Hint` – "הכנס שם".
    - `Button – "ברך."`
    - `Label` ריק.
  - :`Blocks` .2
    - צרו פעולה בשם `sayHelloTo` עם פרמטר אחד (למשל `name`).
    - בתוך הפעולה:
      - הציגו ב-`Label` את הטקסט: "שלום " + הפרמטר `name`.
    - באירוע `Button.Click` – קראו לפעולה `sayHelloTo` ושלחו לה את הטקסט מתוך ה-`TextBox`.
    - קראו לפעולה `sayHelloTo` ושלחו לה את הטקסט מתוך ה-`TextBox`.
- 

## תרגיל 4 – פעולה עם שני פרמטרים (מי גדול יותר)

(גרסה קלה של תרגיל "who is bigger" "who" במצגת)

מצגת 6 - פעולות

שם הפרויקט: `WhoIsBiggerProc`

- :`Designer` .1
  - שני `TextBox` (מספר ראשון, מספר שני – `NumbersOnly`).
  - `Button – "Who is bigger"`
  - `Label` ריק לתוצאה.
- :`Blocks` .2
  - צרו פעולה `checkBigger` עם שני פרמטרים: `x, y`.
  - בתוך הפעולה:
    - אם `y > x` – כתבו ב-`Label`: "הראשון גדול יותר".
    - אחרת אם `x > y` – "השני גדול יותר".
    - אחרת – " שניהם שוויים".

- בAIROU :Button.Click
  - קרו לפעולה checkBigger ושלחו לה את שני המספרים מה-TextBox-ים. ■
- 

## תרגיל 5 – פעולה שמחזירה ערך (סכום פשוט)

שם הפרויקט: SumReturnProc

- :Designer .1
  - שני TextBox (מספר 1, מספר 2 – TextBox.(NumbersOnly
  - צרו פעולה עם החזרת ערך בשם sumTwoNumbers עם שני פרמטרים: a, b.
  - בתוך הפעולה:
    - החזירו (result) את a + b.
  - בAIROU :Button.Click
  - קרו לפעולה sumTwoNumbers עם שני המספרים מה-TextBox-ים.
  - שמרו את התוצאה במשתנה מקומי או שירות ב-Label.
  - הציג ב-Label: "הסכום הוא: " + התוצאה.
- :Blocks .2

## תרגיל 6 – שימוש ב-Notifier (הודעה על מספר גדול/קטן מ-50)

(בהרשמה התרגיל במצב על Notifier)

מצגת 6 - פעולות

שם הפרויקט: NotifierRandom

- :Designer .1
  - "הගREL מספR" – Button
  - Label לתצוגת המספר.
  - ריכיב בלתי נראה Notifier (מקטגורית Notifier).
- :Blocks .2
  - בלחיצה על הכפתור:
    - הגירו מספר בין 1 ל-100 ושמרו אותו במשתנה.
    - הציג את המספר ב-Label.
    - אם המספר גדול מ-50 – הציג הודעה Notifier: "המספר גדול מ-50".
    - אחרת – הודעה Notifier: "המספר קטן או שווה ל-50".

---

## תרגיל 7 – תרגיל קל בלי “גלאלי עזר”

**שם הפרוייקט:** (תבחרו שם בעצמכם)

בתרגיל זהה אין הוראות צעד-אחר-צעד. רק דרישות כלליות:

1. השתמשו לפחות בפעולה אחת עם פרמטר.
2. השתמשו לפחות פעם אחת ב-**Notifier**.
3. האפליקציה שלכם יכולה להיות כל דבר שתרצו (משחק קטן, בדיקה של ציון, מחשבון, ברכות וכו').

תכנו לבד:

- אילו רכיבים יהיו ב-**Designer**.
- מה תהיה הפעולה, אילו פרמטרים היא מקבלת ומה היא עשויה.
- מתי אתם מזמינים את הפעולה ואיפה משתמשים ב-**Notifier**.

בסוף, כתבו במילים קצנות מה האפליקציה שלכם עשו:

---

---