

## יעדים

מטרות התרגיל הן עבודה עם פונקציות ועם המודול turtle שנלמדו בכיתה. בנוסף תתרגלו שימוש בפונקציה print.

## הערות

בתרגיל זה תדרשו להשתמש במודול [turtle](#). בשביל להשתמש במודול turtle אל תשכחו להכניס את הפקודה הבאה בראש הקובץ hello\_turtle.py :

```
import turtle
```

פונקציות חשובות שבהן תשתמשו בתרגיל הינן:

```
1. turtle.forward(number)
```

פונקציה זו מקבלת פרמטר בודד, number, ומזיזה את ראש הצב קדימה number צעדים.

```
2. turtle.left(deg)
```

פונקציה זו מקבלת פרמטר בודד, deg, ומסובבת את ראש הצב שמאלה deg- מעלות.

```
3. turtle.right(deg)
```

פונקציה זו מקבלת פרמטר בודד, deg, ומסובבת את ראש הצב ימינה deg- מעלות.

```
4. turtle.circle(radius, extent)
```

פונקציה זו מקבלת שני פרמטרים : radius, extent, ומזיזה את ראש הצב במעגל ברדיוס radius, עבור החלק במעלות מתוך המעגל אותו נרצה לצייר -extent.

```
5. turtle.setheading(to_angle)
```

פונקציה זו מקבלת פרמטר בודד, to\_angle, ומסובבת את ראש הצב כך שימוקם בזווית to\_angle.

```
6. turtle.up()
```

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

פונקציה זו מרימה את ראש הצב (או ראש העפרון), כלומר כל פעולה שתבצע לאחר קריאה לפונקציה זו לא תוצג במסך, בכדי להחזיר את הראש למטה (דבר שיאפשר לנו לצייר) יש לקרוא לפונקציה `turtle.down`.

```
7. turtle.down()
```

פונקציה זו מורידה את ראש הצב, לאחר קריאה לפונקציה זו נוכל לחזור לצייר כרגיל.

בנוסף תיידרשו לכתוב תיעוד לקוד, כלומר לכתוב הערות שמסבירות מה התפקיד של כל חלק בקוד. לדוגמא, אם יש חלק בקוד שאחראי על התקדמות קדימה של הצב ולאחר מכן רצף פקודות הגורמות לציור של משולש, צריך לרשום מעל חלק זה הערה מתאימה. למשל:

```
# These next lines draw a triangle
```

כזכור, הערות בפייתון יש לכתוב בעזרת התו `#` (hash character) בתחילת ההערה.

## מבנה התרגיל

התרגיל מחולק למספר תתי סעיפים, בכל סעיף תתבקשו לממש פונקציה אשר שמה יופיע מתחת לשם הסעיף, הפונקציות אינן מקבלות פרמטרים. בנוסף, יש לתעד בתחילת כל פונקציה את פעולתה. תיעוד הפונקציה יופיע מתחת לחותמת הפונקציה, בתוך גרשיים משולשים. לדוגמא:

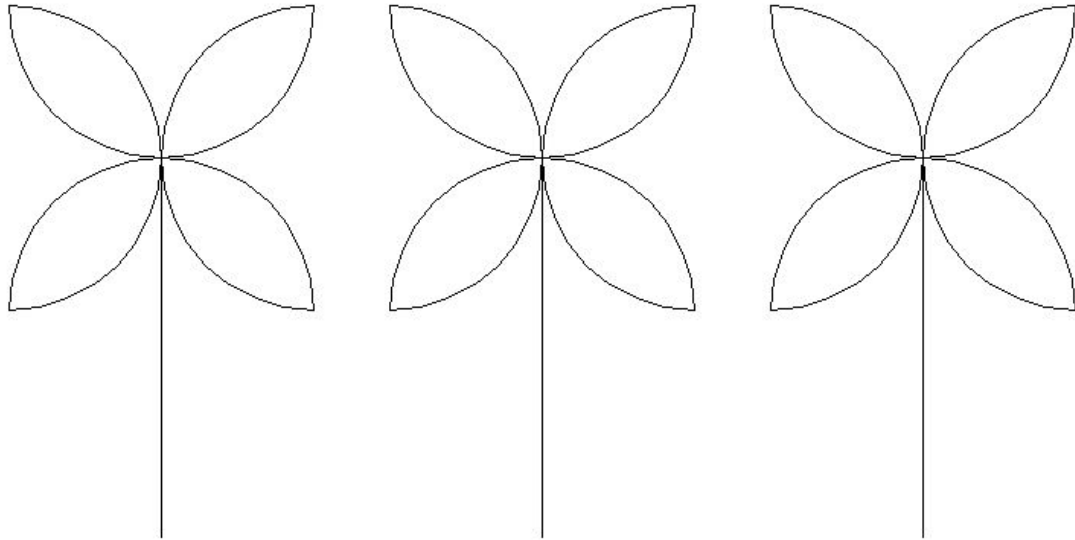
ממשו את הפונקציה **intro\_test** אשר מדפיסה את המחרוזת `hello`.

פתרון השאלה:

```
def intro_test():  
    """This is only a test method for printing hello"""  
    print("hello")
```

## תוצר סופי

אם תממשו את כל הפונקציות כראוי, תקבלו ציור הדומה לזה:



**את חלקים א'-ד' יש לכתוב בקובץ בשם `hello_turtle.py`.**  
**את חלק ה' יש לכתוב בקובץ בשם `math_print.py`.**  
**אתם מתבקשים להוריד את שני הקבצים מאתר הקורס.**

### **חלק א' - ציור של עלה כותרת**

ממשו את הפונקציה `draw_petal` אשר מציירת עלה כותרת בודד. בכדי לצייר עלה כותרת בצעו את רצף הפעולות הבאות:

1. ציירו חלק ממעגל. רדיוס המעגל צריך להיות 100, והחלק אותו נרצה לצייר יהיה 90 מעלות.
2. הסתובבו שמאלה ב-90 מעלות.
3. ציירו חלק ממעגל. רדיוס המעגל צריך להיות 100, והחלק אותו נרצה לצייר יהיה 90 מעלות.

### **חלק ב' - ציור של פרח**

ממשו את הפונקציה `draw_flower` אשר מציירת פרח בודד. בכדי לצייר פרח בצעו את רצף הפעולות הבאות:

1. מקמו את ראש הצב לזווית 0.
2. ציירו עלה כותרת (בעזרת קריאה לפונקציה `draw_petal`).
3. מקמו את ראש הצב לזווית 90.
4. ציירו עלה כותרת (בעזרת קריאה לפונקציה `draw_petal`).
5. מקמו את ראש הצב לזווית 180.

6. ציירו עלה כותרת (בעזרת קריאה לפונקציה `draw_petal`).
7. מקמו את ראש הצב לזווית 270.
8. ציירו עלה כותרת (בעזרת קריאה לפונקציה `draw_petal`).
9. מקמו את ראש הצב לזווית 270.
10. זוזו קדימה ב-250.

## חלק ג' - ציור של פרח מתקדם

ממשו את הפונקציה `draw_flower_advance` אשר מציירת פרח בודד אך גם מזיזה את ראש הצב בשביל לאפשר ציור של פרחים נוספים. בכדי לצייר פרח ולבצע את התזוזה, בצעו את רצף הפעולות הבאות:

1. ציירו פרח (בעזרת קריאה לפונקציה `draw_flower`).
2. פנו ימינה ב-90 מעלות.
3. הרימו את ראש הצב.
4. התקדמו קדימה ב-250.
5. פנו ימינה ב-90 מעלות.
6. התקדמו קדימה בכ-250.
7. פנו שמאלה ב-90 מעלות.
8. הורידו חזרה את ראש הצב.

## חלק ד' - ציור של גינת פרחים

ממשו את הפונקציה `draw_flower_bed` אשר מציירת שלושה פרחים. הרצת פונקציה זו צריכה לתת תוצאה זהה לזו אשר מופיעה בתחילת התרגיל. בכדי לצייר את גינת הפרחים, בצעו את רצף הפעולות הבאות:

1. הרימו את ראש הצב.
2. התקדמו קדימה בכ-200.
3. פנו שמאלה ב-180 מעלות.
4. הורידו חזרה את ראש הצב.
5. ציירו שלושה פרחים בעזרת שלוש קריאות לפונקציה `draw_flower_advance`.

**בשביל לקרוא לפונקציות נשתמש בפורמט אותו למדתם בתרגול :**

```
if __name__ == "__main__":  
    call_function1()  
    call_function2()
```

**התנאי מופיע בקובץ שקיבלתם. תחת הבלוק של התנאי :**

קיראו לפונקציה `draw_flower_bed`  
קיראו `turtle.done()`

## מימוש נכון של חלקים א-ד יוביל לציור דומה לציור המופיע בתחילת הקובץ

### חלק ה' - תרגול הדפסה למסך

בחלק זה תתבקשו לרשום מספר פונקציות אשר ישמשו לתרגול של מודול המתמטיקה, בפונקציות תתבקשו להדפיס תוצאה של חישוב מתמטי כלשהו. שימו לב שניתן עקרונית להדפיס את התוצאה ישירות ללא חישוב (למשל אם נרצה להדפיס את ערך הפונקציה  $\sin$  בנקודה 30 ניתן להדפיס ישירות את הערך 0.5), פתרון זה אמנם יעבור את הבדיקות האוטומטיות אך לא יקבל ניקוד כלל בבדיקה הידנית וגם יאבד את הנקודות שניתנו ע"י הבדיקה האוטומטית.

את הפתרון לתרגיל יש לרשום בקובץ `math_print.py` ובשביל להשתמש בפונקציות המתמטיות צריך לרשום את השורה הבאה בראש הקובץ `math_print.py` שלכם:

```
import math
```

ניתן לקרוא על מודול המתמטיקה -

<https://docs.python.org/3/library/math.html>

לדוגמא, אם במשימה נרצה להדפיס את:

(1) הערך של הפונקציה  $\sin$  עבור ערך של 30 רדיון בפונקציה הנקראת `sin_30`

(2) את הערך של הפונקציה  $\tan$  עבור ערך של 50 רדיון בפונקציה הנקראת `tan_50`

(3) את הערך של הפונקציה  $\cos$  עבור ערך של 86 רדיון בפונקציה הנקראת `cos_86`

הפתרון היה צריך להיות כרשום למטה:

```
def sin_30():
    print(math.sin(30))

def tan_50():
    print(math.tan(50))

def cos_86():
    print(math.cos(86))
```

הפונקציות שצריך לממש מתוארות בהמשך. שימו לב (!) השמות של הפונקציות חייבים להיות זהים לשמות שכתובים כאן:

1. פונקציה שנקראת `golden_ratio` שמדפיסה את יחס הזהב (ניתן לקרוא עליו [כאן](#)).
2. פונקציה שנקראת `six_cubed` שמדפיסה את המספר 6 בחזקת 3.

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

3. פונקציה שנקראת `hypotenuse` שמדפיסה את אורך היתר במשולש ישר זווית אשר צלעותיו באורך 3 ו-5.
4. פונקציה שנקראת `pi` שמדפיסה את ערך המספר פאי.
5. פונקציה שנקראת `e` שמדפיסה את ערך המספר `e` (ניתן לקרוא עליו [כאן](#)).
6. פונקציה שנקראת `triangular_area` שמדפיסה את שטחי המשולשים ישרי זווית שזווי השוקיים שאורך ניצביהם עולה מ-1 עד 10. נרצה שבהדפסה ערכי התוצאות השונות יופרדו ע"י רווח. לדוגמא, עבור המשימה "הדפסו את היקף הריבועים עם אורך צלעות עולה באורך 1 עד 3", הפתרון יהיה:

```
def squares_circumference():  
    print(4*1, 4*2, 4*3)
```

**בשביל לקרוא לפונקציות נשתמש בפורמט אותו למדתם בתרגול :**

```
if __name__ == "__main__" :  
    call_function1()  
    call_function2()  
    ...
```

**התנאי מופיע בקובץ שקיבלתם. תחת הבלוק של התנאי קיראו לשש הפונקציות שמומשו.**

## שאלות ופניות

ראשית קראו היטב את [קובץ נהלי הקורס](#)! בקובץ זה מפורטים כל הנהלים לגבי איך יש לפנות לצוות בכל שאלה, הנהלים של הגשת תרגילים ועוד.

שימו לב! כל שאלה הקשורה לתרגיל יש לשאול בפורום המיועד לתרגיל זה, הנמצא באתר הקורס:

<https://moodle2.cs.huji.ac.il/nu18/course/view.php?id=6710101>

בקשות אישיות בלבד (כמו בקשה לדחיה במועד ההגשה) יש לכתוב למייל הקורס: `intro2cs1@cs.huji.ac.il`, על פי ההוראות המפורטות בקובץ נהלי הקורס.

## נהלי הגשה

### יצירת קובץ zip

בתרגיל זה יצרתם את הקבצים הבאים:

1. `hello_turtle.py`
2. `math_print.py`
3. `README` (כפי שמפורט בקובץ נהלי הקורס)

כעת תצטרכו ליצור קובץ zip הנקרא ex1.zip המכיל בדיוק את שלושת הקבצים הנ"ל.

בווינדוס בחרו את הקבצים ולחצו מקש ימני, לאחר מכן בחרו ב-send to ובחרו באפשרות של "Compressed (zipped) folder".

בלינוקס ניתן לעשות זאת בעזרת פקודת ה-shell הבאה (כאשר אתם נמצאים בתיקייה ex1 שיצרתם):

**zip ex1.zip hello\_turtle.py math\_print.py README**

(ראו במצגת של התרגול הראשון הסבר לגבי קבצי zip).

- זכרו את האזהרה מהתרגול הראשון – אם שוכחים לכתוב את שם קובץ ה-zip שתוצא ליצור, הקובץ הראשון שיהיה כתוב בפקודה הנ"ל ידרס ויהרס. קובץ זה יהפוך להיות קובץ zip המכיל את הקבצים האחרים. למשל אם תכתבו את הפקודה:

zip hello\_turtle.py math\_print.py README

הקובץ hello\_turtle.py שכתבתם ידרס!

- מומלץ לבדוק את קובץ ה-zip שנוצר על ידי העתקת התוכן שלו לתיקייה נפרדת ופתיחתו (extract) בעזרת ביצוע הפקודה: **unzip ex1.zip**, ולאחר מכן יש לבדוק באמצעות הפקודה **ls -h** שכל הקבצים הדרושים קיימים שם ולא ריקים.

**סקריפט קדם-הגשה (Pre submit script):** זהו סקריפט לבדיקה בסיסית של קבצי ההגשה של התרגיל. על מנת להריץ את הסקריפט לתרגיל 1 יש להשתמש במחשבי בית הספר (או פיסית, או כאשר מתחברים מרחוק) הקלידו את הפקודה הבאה בתיקייה בה נמצא הקובץ ex1.zip שייצרתם:

~intro2cs1/bin/presubmit/ex1 ex1.zip

הסקריפט מייצר הודעת הצלחה במקרה של מעבר כל הבדיקות הבסיסיות והודעות שגיאה רלוונטיות במקרה של כישלון בחלק מהבדיקות.

שימו לב, סקריפט קדם ההגשה נועד לוודא רק תקינות בסיסית ביותר ומעבר של בדיקות הסקריפט לא מבטיח את תקינותה של התוכנית! עליכם לוודא שהתוכנית שלכם פועלת כנדרש.

הגשת קובץ התרגיל (קובץ zip)

אם לא עשיתם זאת עדיין, הירשמו למערכת הציונים של הקורס, על ידי לחיצה על:

Enrol me in this course באתר הקורס בצד שמאל למטה.

הגישו את הקובץ ex1.zip בקישור ההגשה של תרגיל 1, על ידי לחיצה על "Upload File".

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

שימו לב שהגשת תרגיל דורשת שתהיו מחוברים עם ה-user והסיסמא שלכם (שנרשמתם איתם למערכת CS).

ניתן להגיש תרגילים דרך מערכת ההגשות באתר הקורס מספר רב של פעמים. ההגשה האחרונה בלבד היא זו שקובעת ושתיבדק.

לאחר הגשת התרגיל, ניתן ומומלץ להוריד אותו ולוודא כי הקבצים המוגשים הם אלו שהתכוונתם להגיש וכי הקוד עובד על פי ציפיותיכם.

קראו היטב את קובץ נהלי הקורס לגבי הנחיות נוספות להגשת התרגילים.

שימו לב - יש להגיש את התרגילים בזמן!

**בהצלחה!**

בהצלחה בקורס ובתואר.