

יעדים

מטרת התרגיל היא עבודה עם פונקציות ועם המודול turtle שהכרתן בכיתה. בנוסף תתרגלו שימוש בפונקציה print.

הערות

בתרגיל זה אתן נדרשות להשתמש במודול turtle. לשם כך, יש לייבא את הספרייה ע"י הוספת הפקודה הבאה בראש הקובץ hello_turtle.py שלכן:

```
import turtle
```

פונקציות חשובות בהן תשתמשו בתרגיל הינן:

1. turtle.forward(number) - פונקציה המקבלת פרמטר בודד, number, ומזיזה את ראש הצב קדימה number צעדים.
2. turtle.left(deg) - פונקציה המקבלת פרמטר בודד, deg, ומזיזה את ראש הצב שמאלה ב-deg מעלות.
3. turtle.right(deg) - פונקציה המקבלת פרמטר בודד, deg, ומזיזה את ראש הצב ימינה ב-deg מעלות.
4. turtle.up() - פונקציה זו מרימה את ראש הצב (או ראש העיפרון) מהמסך. כל פעולה שתבצע לאחר קריאה לפונקציה זו לא תוצג על המסך. כדי להמשיך לצייר על המסך, יש להחזיר חזרה את הראש למטה ע"י קריאה לפונקציה turtle.down().
5. turtle.down() - פונקציה זו מורידה את ראש הצב אל המסך. דבר זה מאפשר ציור על המסך כשמתבצעת תזוזה.

בתרגיל עליכן לכתוב תיעוד לקוד, כלומר לכתוב הערות שמסבירות מה עושה כל חלק בקוד. לדוגמא, אם יש חלק בקוד שאחראי על התקדמות קדימה של הצב ולאחר מכן רצף פקודות הגורמות לציור של משולש אז 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 (לא לשנות את השם). הפונקציה אינה מקבלת פרמטרים. בנוסף, יש לתעד בתחילת הפונקציה את פעולתה. תיעוד הפונקציה יופיע מתחת לחותמת הפונקציה, בתוך גרשיים משולשות. לדוגמא:

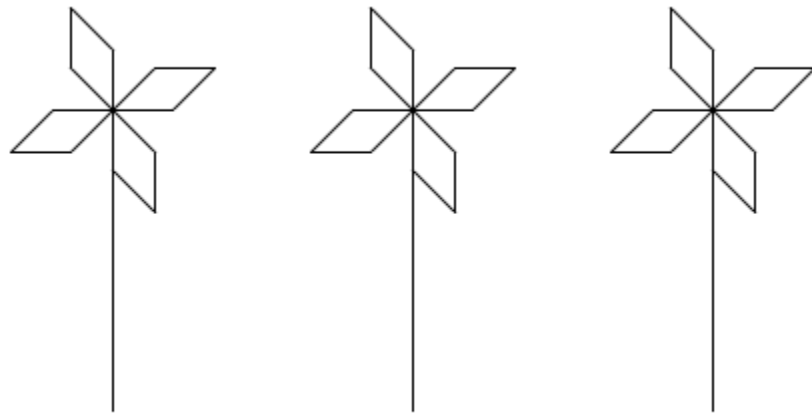
ממשו את הפונקציה `intro_example` אשר מדפיסה את המחרוזת `hello`.

פתרון השאלה:

```
def intro_example():  
    """This is only an example method for printing hello"""  
    print("hello")
```

תוצר סופי

אם תממשו את כל הפונקציות כראוי, תקבלו את הציור הבא:



את חלקים א'-ד' יש לכתוב בקובץ בשם `hello_turtle.py`.

את חלק ה' יש לכתוב בקובץ בשם `math_print.py`.

קבצים לדוגמא נמצאים בMoodle תחת התיקיה "**Additional Files**".

חלק א' - ציור של עלה כותרת

ממשו את הפונקציה `draw_petal` אשר מציירת עלה כותרת בודד. כדי לצייר עלה כותרת עליכן לבצע את רצף הפעולות הבאות:

1. זוזו קדימה 30 צעדים.
2. הסתובבו ימינה ב-45 מעלות.
3. זוזו קדימה 30 צעדים.
4. הסתובבו ימינה ב-135 מעלות.

5. זוזו קדימה 30 צעדים.
6. הסתובבו ימינה ב-45 מעלות.
7. זוזו קדימה 30 צעדים.
8. הסתובבו ימינה ב-135 מעלות.

חלק ב' - ציור של פרח

ממשו את הפונקציה **draw_flower** אשר מציירת פרח בודד. בכדי לצייר פרח עליכן לבצע את רצף הפעולות הבאות:

1. פנו שמאלה ב-45 מעלות.
2. ציירו עלה כותרת (בעזרת קריאה לפונקציה **draw_petal**).
3. פנו שמאלה ב-90 מעלות.
4. ציירו עלה כותרת (בעזרת קריאה לפונקציה **draw_petal**).
5. פנו שמאלה ב-90 מעלות.
6. ציירו עלה כותרת (בעזרת קריאה לפונקציה **draw_petal**).
7. פנו שמאלה ב-90 מעלות.
8. ציירו עלה כותרת (בעזרת קריאה לפונקציה **draw_petal**).
9. פנו שמאלה ב-135 מעלות.
10. זוזו קדימה 150 צעדים.

שימו לב כי קריאה לפונקציה זו עלולה לצייר לכן פרח הפוך. שכן ציור הפרח תלוי בזווית בה התחלתן את ציורו. לאחר שתשלימו את חלק ד' הפרחים יציירו כהלכה.

חלק ג' - ציור של פרח וקידום הצב

ממשו את הפונקציה **draw_flower_and_advance** אשר מציירת פרח בודד אך גם מזיזה את ראש הצב בשביל לאפשר ציור של פרחים נוספים. בכדי לצייר פרח ולבצע את התזוזה עליכן לבצע את רצף הפעולות הבאות:

1. ציירו פרח (בעזרת קריאה לפונקציה **draw_flower**).
2. פנו ימינה ב-90 מעלות.
3. הרימו את ראש הצב.
4. זוזו קדימה 150 צעדים.
5. פנו ימינה ב-90 מעלות.
6. זוזו קדימה 150 צעדים.
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 עליכן לבצע את רצף הפעולות הבאות:

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

בשביל לקרוא לפונקציות נשתמש בפורמט אותו למדתם בתרגול:

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

התנאי מופיע בקובץ שקיבלתם. תחת הבלוק של התנאי:

קראו לפונקציה `draw_flower_bed`

קראו לפונקציה `turtle.done`

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

חלק ה' - שימוש באופרטורים מתמטיים ובמודול `math`

בחלק זה אתן תתבקשו לרשום מספר פונקציות אשר יתרגלו אתכן בשימוש באופרטורים מתמטיים ובמודול המתמטיקה (`math`). בפונקציות תתבקשו להדפיס תוצאה של חישוב מתמטי כלשהו. שימו לב שעל אף שניתן עקרונית להדפיס את התוצאה ישירות ללא חישוב (למשל עבור ערך הפונקציה \sin ב-30 ניתן להדפיס ישירות את התוצאה -0.9880316240928618 מבלי להשתמש בפונקציית הסינוס בקוד), פתרון זה לא יקבל ניקוד כלל (יכשל בבדיקה הידנית ויאבד את הנקודות שניתנו ע"י הבדיקה האוטומטית).

את הפתרון לתרגיל יש לממש בקובץ `math_print.py`. כדי להשתמש בפונקציות המתמטיות עליכן לרשום את השורה הבאה בראש הקובץ `math_print.py`:

```
import math
```

אתן מוזמנות לקרוא על מודול המתמטיקה -

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

לדוגמא, במשימה בה אתן מתבקשות להדפיס את:

(1) הערך של הפונקציה \sin עבור 30 רדיאנים בפונקציה הנקראת `sin_30`

(2) הערך של הפונקציה \tan עבור 50 רדיאנים בפונקציה הנקראת `tan_50`

(3) הערך של הפונקציה \cos עבור 86 רדיאנים בפונקציה הנקראת `cos_86`

הפתרון הוא כדלהלן:

```
def sin_30():  
    print(math.sin(30))  
  
def tan_50():  
    print(math.tan(50))  
  
def cos_86():  
    print(math.cos(86))
```

הפונקציות שעליכן לממש:

1. פונקציה הנקראת `golden_ratio` שמדפיסה את יחס הזהב (ניתן לקרוא עליו [כאן](#)).
 2. פונקציה הנקראת `six_squared` שמדפיסה את התוצאה של 6 בחזקת 2.
 3. פונקציה הנקראת `hypotenuse` שמדפיסה את אורך היתר במשולש ישר זווית אשר צלעותיו באורך 12 ו-5.
 4. פונקציה הנקראת `pi` שמדפיסה את ערך המספר פאי.
 5. פונקציה הנקראת `e` שמדפיסה את ערך המספר e .
 6. פונקציה הנקראת `squares_area` שמדפיסה את שטחי הריבועים, מאורכי צלעות 1 עד 10 (בסדר עולה), עם תו רווח בין כל שני ערכים.
- לדוגמא, עבור המשימה "הדפסו את היקף הריבועים מאורכי צלעות 1 עד 3" הפתרון יהיה:

```
def squares_area():  
    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 קראו לשש הפונקציות שמימשותן.

חלק ו - תרגול הרצת בדיקות

סקריפט קדם-הגשה (Pre submit script): זהו סקריפט לבדיקה בסיסית של קבצי ההגשה של התרגיל.

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

סקריפט קדם ההגשה נועד לוודא רק תקינות בסיסית ביותר ומעבר של בדיקות הסקריפט לא מבטיח את תקינותה של התוכנית! עליכן לוודא בעצמכן שהתוכנית שלכם פועלת כפי שדרוש.

חלק זה של התרגיל נועד לוודא שאתם מסתכלים על תוצאות הבדיקות שמורצות עם הגשת התרגיל.

לאחר הגשת התרגיל בפעם הראשונה, תקבלו משוב בו כתוב שנכשלתם באחת הבדיקות האוטומטיות. כדי

להשלים את חלק זה עליכם לפתוח קובץ חדש בשם שיופיע בבדיקות האוטומטיות ולהכניס אליו פונקציה

בשם `secret_function` המדפיסה מחרוזת מסוימת. חפשו במשוב את המחרוזת אותה אתם צריכים

להדפיס והשלימו את הפונקציה. שימו לב שתו ירידת השורה '\n' המופיע בסוף המחרוזת בקובץ המשוב

מתווסף באופן אוטומטי על-ידי הפונקציה `print` ואין צורך לרשום אותו במפורש.

ניתן לקרוא את המשוב (המכונה גם `pre-submission script`) באחד משני האופנים הבאים:

1. על ידי הגשת התרגיל דרך Moodle, ופתיחת קובץ `feedback` שמוחזר דרך Moodle (ונשלח גם למייל).
2. על ידי הרצת הבדיקות במחשבי האוניברסיטה (או פיסית, או כאשר מתחברים מרחוק) על ידי הקלדת הפקודה הבאה:

```
~intro2cs2/bin/presubmit/ex1 ex1.zip
```

כאשר `ex1.zip` הוא שם הקובץ אותו אתם רוצים להגיש.

הערה: שימו לב שבאופן חריג, המחרוזת אותה אתם אמורים לחפש במקרה זה לא תופיע במשוב שתקבלו

בעזרת שיטה 2, אלא רק על-ידי שיטה 1 בלבד.

שאלות ופניות

ראשית קראו היטב את קובץ נהלי הקורס! בקובץ זה מפורטים כל הנהלים, ובפרט כיצד ניתן לפנות לצוות בכל שאלה, נהלי הגשת תרגילים ועוד.

לכל תרגיל קיים פורום מיועד ב Moodle בו ניתן לפנות בשאלות בנושא.

בקשות אישיות בלבד (כמו בקשה לדחיה במועד ההגשה) יש לכתוב למייל הקורס: intro2cs2@cs.huji.ac.il, על פי ההוראות המפורטות בקובץ נהלי הקורס.

נהלי הגשה

יצירת קובץ zip

בתרגיל זה התבקשתן ליצור את הקבצים הבאים:

1. hello_turtle.py
2. math_print.py
3. הקובץ המכיל את הפונקציה secret_function אותו יצרתן עבור חלק ו'.

עליכן 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 שיצרתן):

zip ex1.zip hello_turtle.py math_print.py

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

- זכרו להוסיף לפקודה את שם הקובץ המכיל את הפונקציה הסודית של חלק ו'.
- זכרו את האזהרה מהתרגול הראשון – אם אתן שוכחות לכתוב את שם קובץ ה-zip שאתן רוצות ליצור, אתן תדרסו ותהרסו את הקובץ הראשון שאתן כותבות בפקודה הנ"ל, וקובץ זה ישתנה ויהפוך להיות קובץ zip המכיל את הקבצים האחרים. למשל אם תכתבו את הפקודה:

zip hello_turtle.py math_print.py

הקובץ hello_turtle.py שכתבתם ידרס!

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

בנוסף, רשמו בתחילת הקובץ hello_turtle.py בהערה (כלומר עם הסימן # בתחילת השורה, כפי שהוסבר קודם לכן) את שמות הסטודנטים איתם התייעצתן במהלך העבודה על התרגיל, במידה ויש כאלו.

הגשת קובץ התרגיל (קובץ zip)

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

שימו לב שהגשת תרגיל דורשת שתהיו מחוברות עם ה-user והסיסמא שלכן (שנרשמתן איתן למערכת CS).

וודאו שסקריפט קדם ההגשה (pre-submission script) עבר בהצלחה.

הנכן רשאיות 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 שהתכוונתן להגיש וכי הקוד עובד על פי ציפיותיכן.

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

שימו לב - יש להגיש את התרגילים בזמן!

בהצלחה!

* התרגיל כתוב בלשון נקבה אך מיועד לכלל תלמידי ותלמידות אינטרו. בהצלחה בקורס ובתואר.