

מבוא למדעי המחשב 67101 – סמסטר ב' 2020

תרגיל 1 - Hello Turtle

להגשה בתאריך **01/04/2020** בשעה 22:00

יעדים

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

הערות

בתרגיל זה אתן נדרשות להשתמש במודול turtle, בשביל להשתמש במודול turtle אל תשכחו להכניס את הפקודה הבאה בראש הקובץ hello_turtle.py שלכן:

```
import turtle
```

פונקציות חשובות שבהן תשתמשו בתרגיל הינן:

1. turtle.forward(number) - פונקציה זו מקבלת פרמטר number, ומזיזה את ראש הצב קדימה number צעדים.
2. turtle.left(deg) - פונקציה זו מקבלת פרמטר deg, ומזיזה את ראש הצב שמאלה בכ-deg מעלות.
3. turtle.right(deg) - פונקציה זו מקבלת פרמטר deg, ומזיזה את ראש הצב ימינה בכ-deg מעלות.
4. turtle.up() - פונקציה זו מרימה את ראש הצב (או ראש העיפרון), 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().
5. turtle.down() - פונקציה זו מורידה את ראש הצב, לאחר קריאה לפונקציה זו נוכל לחזור לצייר כרגיל.

בנוסף 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, הפונקציה אינה מקבלת פרמטרים. בנוסף, יש לתעד בתחילת הפונקציה את פעולתה. תיעוד הפונקציה יופיע מתחת לחותמת הפונקציה, בתוך גרשיים משולשות. לדוגמא:

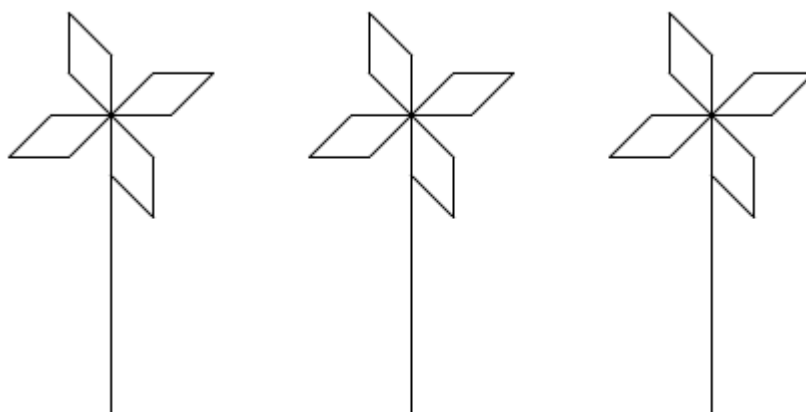
ממשו את הפונקציה **intro_example** אשר מדפיסה את המחרוזת hello.

פתרון השאלה:

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

תוצר סופי

אם תממשו את כל הפונקציות כראוי אתן תקבלו ציור בדומה לזה:



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

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

קבצים לדוגמא נמצאים במודל תחת התיקיה "Additional Files".

חלק א' - ציור של עלה כותרת

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

1. זוזו קדימה ב-30.
2. הסתובבו ימינה ב-45 מעלות.
3. זוזו קדימה ב-30.
4. הסתובבו ימינה ב-135 מעלות.
5. זוזו קדימה ב-30.
6. הסתובבו ימינה ב-45 מעלות.
7. זוז קדימה ב-30.
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. זוז קדימה ב-150.

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. ציור פרח (בעזרת קריאה לפונקציה **draw_flower**).
2. פנו ימינה ב-90 מעלות.
3. הרימו את ראש הצב.
4. התקדמו קדימה ב-150.
5. פנו ימינה ב-90 מעלות.
6. התקדמו קדימה בכ-150.
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. 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_and_advance**

בשביל לקרוא לפונקציות נשתמש בפורמט אותו למדתם בתרגול:

```
if __name__ == "__main__":
```

```
    call_function1()
```

```
    call_function2()
```

התנאי מופיע בקובץ שקיבלתם. תחת הבлок 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' - תרגול הדפסה למסך

בחלק זה אתן תתבקשו לרשום מספר פונקציות אשר יתרגלו אתכן בשימוש במודול המתמטיקה, בפונקציות תתבקשו להדפיס תוצאה של חישוב מתמטי כלשהו. שימו לב שניתן עקרונית להדפיס את התוצאה ישירות ללא חישוב (למשל אם הייתן מתבקשות להדפיס את ערך הפונקציה \sin בנקודה 30 הייתן יכולות 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 עליכן לרשום את השורה הבאה בראש הקובץ `math_print.py` שלכם:

```
import math
```

אתן מוזמנות לקרוא על מודול המתמטיקה -

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

לדוגמא, אם במשימה הייתן מתבקשות 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 ברשום למטה:

```
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 שנקראת `golden_ratio` שמדפיסה את יחס הזהב (ניתן לקרוא עליו [כאן](#)).
2. פונקציה שנקראת `six_squared` שמדפיסה 6 בחזקת 2.
3. פונקציה שנקראת `hypotenuse` שמדפיסה את אורך היתר במשולש ישר זווית אשר צלעותיו באורך 12 ו-5.
4. פונקציה שנקראת `pi` שמדפיסה את ערך המספר פאי.
5. פונקציה שנקראת `e` שמדפיסה את ערך המספר `e`.
6. פונקציה שנקראת `squares_area` שמדפיסה את שטחי ריבועים, באורך צלעות עולה מ-1 עד 10, עם תו רווח בין כל ערך וערך. לדוגמא, עבור המשימה "הדפיסו את היקף הריבועים עם אורך צלעות עולה באורך 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 קיראו לשש הפונקציות שמימשותן.

חלק ו - תרגול הרצת בדיקות

חלק זה של התרגיל נועד כדי 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`) בשני אופנים: א. על ידי הגשת התרגיל דרך המודל, ופתיחת קובץ ה `feedback` שמוחזר דרך המודל (ונשלח גם למייל). ב. על ידי הרצת הבדיקות במחשבי האוניברסיטה על ידי 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 ב, אלא רק על-ידי שיטה א בלבד.

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:

<http://www.cs.huji.ac.il/~intro2cs2>

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'.

כעת עליכן ליצור קובץ `zip` הנקרא `ex1.zip` המכיל בדיוק את ארבעת 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

(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 שיצרתם 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.

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.

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 בתיקיה בה נמצא הקובץ ex1.zip שייצרתם:

```
~intro2cs2/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! עליכן לוודא בעצמכן שהתוכנית שלכם פועלת כפי שדרוש.

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).

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.