

מבוא למדעי המחשב 67101

תרגיל 1 - Hello World and Hello Turtle

מועד ההגשה: יום חמישי, 6.11.2014 בשעה 21:00

יעדים

מטרת התרגיל היא הכרות עם סביבת העבודה ועם מספר פעולות בסיסיות, עבודה עם ה-shell וכתובת התוכנית הראשונה ב-Python.

חלק א' – יצירת תיקייה

יש ליצור תיקייה הנקראת intro2cs תחת תיקיית ה-safe שלכם (אשר נמצאת בתיקיית הבית שלכם):
~/safe. עליכם לכתוב את הפתרונות שלכם לתרגילים בקורס מבוא למדעי המחשב בתיקייה זו, כאשר לכל תרגיל תהיה תיקייה משלו.

תחת ~/safe/intro2cs צרו תיקייה הנקראת ex1.

זכרו לקבוע את ההרשאות לתיקיית intro2cs כך שהיא לא תהיה נגישה לאחרים. אם אתם נמצאים בתיקייה ~/safe, זה יכול להתבצע על ידי הפקודה `chmod 700 intro2cs`.

חלק ב' – הרצת תוכנית Python : Hello World

פתחו את העורך pluma ע"י הרצת הפקודה `pluma & shell`, וצרו קובץ הנקרא Hello.py תחת התיקייה ~/safe/intro2cs/ex1. הקובץ צריך להכיל את קוד ה-Python הבא:

```
#####  
# FILE : Hello.py  
# WRITER : your_name , your_login , your_id  
# EXERCISE : intro2cs ex1 2014-2015  
# DESCRIPTION:  
# A simple program that prints "Hello World!" to the standard  
# output (screen).  
#####  
print("Hello World!")
```

שימו לב – אל תעתיקו (copy&paste) את הקוד הנ"ל ממסמך זה, אלא הקלידו אותו מחדש בעצמכם (מכיוון שהעתקה כזו עלולה להכניס סימנים בעייתיים).

כפי שאתם יכולים לראות, תוכנית זו מדפיסה את המחרוזת "Hello World!" למסך.

עליכם לעדכן בקובץ את הפרטים שלכם (שם, login ותעודת זהות).

שמרו את הקובץ והריצו אותו מה-shell, כאשר אתם נמצאים בתיקייה ~/safe/intro2cs/ex1, בעזרת הפקודה python3 Hello.py. האם ראיתם את הפלט מודפס ב-shell? אם עשיתם טעות כלשהי בקובץ, השגיאות שלכם עשויות להיות מתוארות ב-shell. תקנו אותן, ווודאו שהפלט הנכון מודפס.

חלק ג' – כתיבת תוכנית Python : Hello Turtle

כעת ניצור תוכנית Python נוספת הנקראת HelloTurtle.

בפיתון אנחנו יכולים לייבא קוד שנכתב על ידי אחרים, בעזרת פקודת import. בהמשך הקורס, נדבר על החשיבות של import ועל איך פיתון מפענחת את הפקודה הזו ומוצאת את הקוד הרלוונטי. עבור תרגיל זה, אנו נשתמש במודול שנקרא Turtle. מודול זה (קובץ פיתון המכיל הגדרות ומימושים), מאפשר לנו לצייר אלמנטים גרפיים פשוטים.

פתחו את העורך pluma, וצרו קובץ חדש הנקרא HelloTurtle.py תחת התיקייה ~/safe/intro2cs/ex1. בתחילת הקובץ צריך להופיע קטע הקוד הבא:

```
#####
# FILE : HelloTurtle.py
# WRITER : your_name + your_login + your_id
# EXERCISE : intro2cs ex1 2014-2015
# DESCRIPTION:
# A program that draws some simple geometric shapes on the screen
# and prints "Hello Turtle!", using Turtle graphics.
#####
import turtle

# title for the display window
turtle.title("Fun with Turtle Graphics and Python")
turtle.up()           # lift the pen up, no drawing
turtle.goto(-100, -100) # Move turtle to the absolute position (-100,-100)
turtle.down()         # pen is down, drawing now
```

גם הפעם עליכם לעדכן בקובץ את הפרטים שלכם (שם, login ותעודת זהות), ולא להעתיק (copy&paste) את הקוד הנ"ל ממסמך זה, אלא להקליד אותו מחדש בעצמכם.

שמרו את הקובץ והריצו אותו מה-shell, כאשר אתם נמצאים בתיקייה ~/safe/intro2cs/ex1, בעזרת הפקודה python3 HelloTurtle.py.

שימו לב שהשורה import turtle מייבאת את המודול שנקרא Turtle, ולאחריה ניתן להשתמש בפעולות הממומשות במודול זה, על ידי קריאה לפונקציות הרלוונטיות של Turtle, בצורה הבאה:

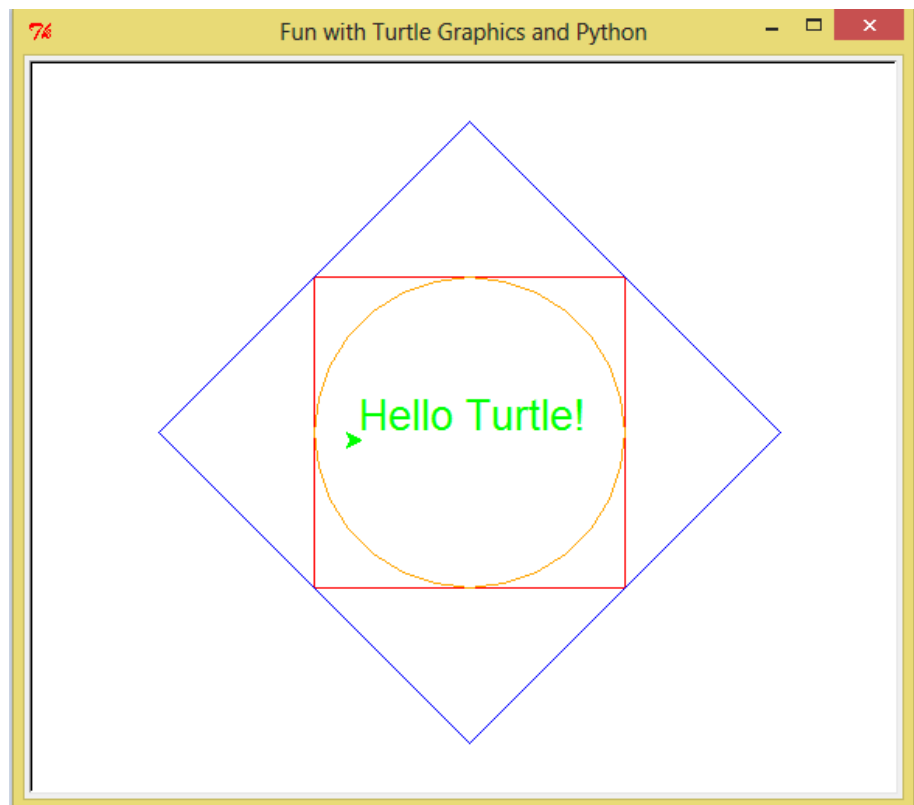
turtle.function_name(). הפונקציות הדרושות לכם לתרגיל זה מפורטות בהמשך.

לאחר קטע הקוד הנ"ל צריך להופיע הקוד שלכם, בו עליכם לצייר את האלמנטים הבאים:

1. ריבוע בצבע אדום, כאשר נקודת ההתחלה שהיא כאמור (-100,-100) מהווה את הפינה השמאלית התחתונה של הריבוע, וכל צלע בריבוע היא בגודל 200.
2. מעגל בצבע כתום בעל רדיוס 100, שמרכזו נמצא בראשית הצירים (0,0).

3. ריבוע גדול יותר בצבע כחול, המכיל את הריבוע הקודם שציירתם, כך שצלעותיו של הריבוע הגדול משיקות לצלעות הריבוע שציירתם קודם (ומרכזו בראשית הצירים).
(כלומר מרכזי הצלעות של הריבוע הגדול צריכים להשיק לפינות של הריבוע הקטן שציירתם קודם).
4. יש להגיע לנקודה (-70,-5), ולכתוב שם בצבע ירוק את המחרוזת "Hello Turtle!" בעזרת השורה הבאה :
`turtle.write("Hello Turtle!", font=("Arial", 20, "normal"))`
5. לבסוף יש לסיים בעזרת כתיבת השורה : `turtle.done()`

בסופו של דבר, הפלט של התוכנית שלכם צריך להראות כך :



לצורך ציור האלמנטים הנ"ל, אתם יכולים להשתמש בפונקציות הבאות של Turtle :

- `turtle.goto(x,y)` – מזיז את הצב למיקום המוגדר על ידי הקורדינטות `x,y`.
- `turtle.up()` – מרים את העט של הצב, כך שמעתה שהצב מפסיק לצייר.
- `turtle.down()` – מוריד את העט של הצב, כך שמעתה הצב מתחיל לצייר.
- `turtle.circle(r)` – מצייר מעגל על פי הרדיוס `r` הנתון. מרכז המעגל יהיה במרחק `r` שמאלה מהמיקום הנוכחי של הצב (כאשר הכיוון שמאלה נקבע ביחס לכיוון שאליו פונה הצב).
- `turtle.color("color")` – משנה את הצבע שבו מצייר/כותב להיות הצבע המוגדר ע"י המחרוזת הנתונה `"color"`. למשל `turtle.color("red")` קובע את הצבע שבו מצייר הצב להיות אדום.

עליכם לכתוב תיעוד לקוד, כלומר לכתוב הערות שמסבירות מה אתם עושים בכל חלק בקוד. לדוגמא, אם יש חלק בקוד שאחראי על ציור המעגל, יש להסביר זאת בהערה. כזכור, הערות בפייתון יש לכתוב בעזרת התו # (hash character) בתחילת ההערה.

שימו לב שבתרגיל זה נחשפתם כבר למספר עקרונות בסיסיים בתכנות :

1. **כתיבת הערות** – הערות נועדו להבהיר מה מתבצע בחלקים שונים של הקוד, כך שמי שמתסכל על הקוד יוכל להבין מה היתה מטרתו של המתכנת בכל חלק בקוד. ה-`interpreter` של פייתון מתעלם מהערות, ולכן הן לא משפיעות על התוכנית.
2. **import** – כפי שהוזכר קודם, בתרגיל זה ייבאנו את המודול `Turtle`, ולכן יכולנו להשתמש בפונקציות שמומשו בו כבר מראש על ידי אחרים, מבלי שהיינו צריכים לממש אותן מחדש בעצמנו.
3. **פונקציות** – פונקציות הינן קטעי קוד קצרים ועצמאיים שיכולים להיקרא מספר פעמים מבלי שצריך להקליד מחדש את הקוד שלהן כל פעם מחדש. אנו יכולים לזהות קריאה לפונקציה כאשר אנו רואים סוגריים: (). כפי שראינו, למודול `Turtle` יש מספר פונקציות ששייכות לו. כאשר אנו מבצעים את הקריאה `turtle.goto(100,100)`, אנו קוראים לפונקציה `"turtle.goto"` עם הפרמטרים 100 ו-100.
4. **פרמטרים** (או arguments) – הם הקלט של הפונקציה. למשל על ידי הקלדת השורה `turtle.goto(100,-100)`, אנו שולחים לפונקציה `turtle.goto` את הפרמטרים 100 ו-100-, ובכך מצווים על הצב לזוז למיקום (100,-100) על המסך. כפי שאתם זוכרים, למדתם בכתה על טיפוסים (types) שונים של משתנים בפייתון, כמו `string`, `float`, `int`, `(str)`. פונקציה יכולה לקבל טיפוסים שונים של משתנים בתור פרמטרים. למשל כפי שראיתם, הפונקציה `turtle.circle` מצפה לקבל פרמטר אחד שמהווה את הרדיוס של המעגל, ולכן היא מצפה לקבל מספר בתור פרמטר (`float` או `int`). נסו למשל לתת לפונקציה זו פרמטר שהוא `string` וראו מה קורה. כמו כן כפי שראיתם, הפונקציה `turtle.goto` מצפה לקבל שני פרמטרים שהם מספרים (`int` או `float`). גם כאן יקרה דבר דומה כשתנסו לתת לפונקציה זו פרמטר אחד או יותר שהוא `string`. לעומת זאת הפונקציה `turtle.title` כן יכולה לקבל פרמטר ראשון מטיפוס `string`, כפי שראינו בשורה: `turtle.write("Hello Turtle!", font=("Arial", 20, "normal"))`.

כעת פתחו את העורך `pluma` וצרו קובץ חדש שנקרא `ex1.txt`. את הקובץ שמרו בתיקייה `~/safe/intro2cs/ex1`. עליכם לכתוב את כל התשובות לשאלות הבאות בקובץ זה (גם את התשובות לחלק זה וגם לחלק ד'). יש לכתוב את כל התשובות **באנגלית**.

פתחו `shell` של פייתון (בעזרת הרצת הפקודה `python3` ב-`shell` הרגיל), והריצו בו את הפקודה `import turtle`.

1. לאחר מכן הקלידו `help(turtle.goto)` והקישו `enter`. מה קיבלתם?
2. כעת הקלידו רק `turtle.goto` והקישו `enter`. מה קיבלתם?
3. כעת השוו זאת עם הקלדה של השורה `turtle.goto(100,100)`, אשר תבצע קריאה לפונקציה ותחזיר את התוצאה של הרצת הפונקציה. מה התבצע כעת?

חלק ד' – סביבת UNIX

המטרה של חלק זה היא לוודא שהבנתם את מה שנלמד בתרגול הראשון. לכן תחילה עברו שוב על התרגול, ווודאו שאתם מבינים הכל (נסו לבצע בעצמכם באמצעות ה-shell את כל הפקודות שנלמדו בתרגול).

עליכם לענות גם לכל השאלות הבאות בקובץ ex1.txt.

שימו לב! עליכם לענות על כל השאלות במילים שלכם, לאחר שניסיתם בעצמכם להריץ את מה שדרוש והבנתם היטב את המשימה, ולא להעתיק מהמצגת של התרגול.

1. תארו בקצרה מה עושה כל אחת מהפקודות הבאות: `mkdir`, `rmdir`, `cd`, `cd ~`, `cd ~/`.
 2. תארו את המשמעות של התיקיות `'.'` (נקודה אחת) ו- `'..'` (שתי נקודות). למה הן משמשות?
 3. מהי כתובת יחסית? (relative path) ומהי כתובת מוחלטת? (absolute path)
 4. הסבירו איך ניתן להשתמש בסימנים `'*'` ו- `'?'` ב-shell. (מה המשמעות שלהם בפקודה `ls` ובפקודה `cp`?)
 5. מה המשמעות של הסימן `'&'` ב-shell? מה אתם יכולים לעשות אם שכחתם לכתוב אותו לאחר שהרצתם כבר אפליקציה כלשהי?
 6. בהינתן שם של קובץ, כיצד אתם יכולים לדעת מהן ההרשאות של הקובץ?
 7. השתמשו בפקודת `man` על מנת לגלות מה עושות הפקודות הבאות: `'grep'`, `'cat'`, `'diff'`, `'wc'`, `'cal'`.
- תנו דוגמא לשימוש בכל אחת מהפקודות האלה, והסבירו מה מתבצע.

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/~intro2cs>

בקשות אישיות בלבד (כמו בקשה לדחיה במועד ההגשה) יש לכתוב למייל הקורס: intro2cs@cs.huji.ac.il, על פי ההוראות המפורטות בקובץ נהלי הקורס.

נהלי הגשה

יצירת קובץ tar

בתרגיל זה התבקשתם ליצור את הקבצים הבאים:

1. Hello.py
2. HelloTurtle.py
3. ex1.txt
4. README (כפי שמפורט בקובץ נהלי הקורס)

כעת עליכם ליצור קובץ tar הנקרא ex1.tar המכיל בדיוק את ארבעת הקבצים הנ"ל, בעזרת פקודת ה-shell הבאה (כאשר אתם נמצאים בתיקייה ex0 שיצרתם):

tar cvf ex1.tar Hello.py HelloTurtle.py ex1.txt README

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

- זכרו את האזהרה מהתרגול הראשון – אם אתם שוכחים לכתוב את שם קובץ ה-tar שאתם רוצים ליצור, אתם תדרסו ותהרסו את הקובץ הראשון שאתם כותבים בפקודה הנ"ל (מיד אחרי ה-cvf'), וקובץ זה ישתנה ויהפוך להיות קובץ tar המכיל את הקבצים האחרים. למשל אם תכתבו את הפקודה:

tar cvf Hello.py HelloTurtle.py ex1.txt README

הקובץ Hello.py שכתבתם ידרס!

- מומלץ לבדוק את קובץ ה-tar שיצרתם על ידי העתקת התוכן שלו לתיקייה נפרדת ופתיחתו (extract) בעזרת ביצוע הפקודה: **tar xvf ex1.tar**, 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:

~intro2cs/bin/presubmit/ex1 ex1.tar

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 בעצמכם שהתוכנית שלכם פועלת כפי שדרוש.

הגשת קובץ tar

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

Enrol me in this course באתר הקורס בצד שמאל למטה.

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

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

הנכם רשאים 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.

הקובץ submission.pdf

לאחר שאתם מגישים את התרגיל באתר הקורס, תוך מספר שניות ייוצר קובץ שנקרא submission.pdf שמכיל בתוכו את כל התוכן של הקבצים שהגשתם, ותוצאות של חלק מהבדיקות האוטומטיות (הקובץ ייוצר בחלון ההגשה של התרגיל, במקום בו הגשתם את ex1.tar). עליכם לבדוק שהכל תקין בקובץ submission.pdf, ואם יש בעיה כלשהי בקבצים שלכם שבאה לידי ביטוי בקובץ ה-pdf עליכם לתקן אותה, גם אם לא נאמר שום דבר בפירוש בתרגיל לגבי זה, כך שקובץ ה-pdf שמיוצר מהתרגיל שלכם יהיה תקין לגמרי. וודאו שאין שורות ארוכות מדי בקוד שלכם שנחתכות בקובץ ה-pdf, ושהקובץ מסודר ורואים בו את הכל בצורה טובה וברורה. קובץ ה-pdf שנוצר הוא גם הקובץ שהבודקים יראו כאשר הם יבדקו את התרגיל שלכם.

הרצת פתרון תרגיל לדוגמא

על מנת להבין טוב יותר כיצד התרגיל צריך לרוץ, אתם יכולים להריץ את ה-"פתרון בית ספר" של התרגיל, ולראות כיצד הוא פועל. עשו זאת על ידי הקלדת הפקודות הבאות ב-shell:

```
~intro2cs/bin/ex1/Hello
```

```
~intro2cs/bin/ex1/HelloTurtle
```

- כל הקבצים שאתם מגישים בתרגיל צריכים להיכתב באנגלית בלבד.
- קראו היטב את קובץ נהלי הקורס לגבי הנחיות נוספות להגשת התרגילים.
- שימו לב - אין להגיש את התרגילים באיחור!

בהצלחה! ☺