

# מבוא לתכנות מערכות תרגיל בית 2

סמסטר חורף  
2019-2018

תאריך פרסום: 9.12.2018

תאריך הגשה: 23.12.2018

משקל התרגיל: 5% מהציון הסופי (תקף)

מתרגל אחראי: מוחמד גריפאת ([mohamadg.mtm@gmail.com](mailto:mohamadg.mtm@gmail.com))

## 1 הערות כלליות

- שימו לב: לא יינתנו דחיות במועד התרגיל. תכננו את הזמן בהתאם.
- שאלות בנוגע לפרטי התרגיל יש לשאול בפורום התרגיל במודל. לשאלות פרטיות אפשר להגיע לשעת קבלה של מוחמד, להגיע לאחת מהסדנאות או לשאול במייל. לפני שליחת שאלה - נא וודאו שהיא לא נענתה כבר במודל, ושהתשובה אינה ברורה ממסמך זה, או מהדוגמא שפורסמה עם התרגיל.
- קראו מסמך זה עד סופו לפני שאתם מתחילים לממש. יתכן שתצטרכו להתאים את המימוש שלכם לחלק עתידי בתרגיל. תכננו את המימוש שלכם לפני שאתם ניגשים לעבוד. רצוי לעבור על הדוגמא שפורסמה לפני תחילת הפתרון.
- חובה להתעדכן בעמוד המודל של התרגיל - הכתוב שם מחייב.
- בתרגיל זה נלמד לעבוד עם פייתון. במהלך התרגיל נשתמש בחומר שמועבר בהרצאה 5 ובתרגולים 6+7.
- העתקות קוד בין סטודנטים תטופלנה בחומרה!
- מומלץ מאוד מאוד לכתוב את הקוד בחלקים קטנים, לקמפל כל חלק בנפרד על השרת, ולבדוק שהוא עובד באמצעות שימוש בטסטים על השרת CSL3.

## 2 התקנת Python

חלק זה לא דורש ידע מהרצאה/תרגול ויש לעשות אותו בהתחלה.

לשפת פייתון יש גרסאות. אנחנו נעבוד בגרסה 3.4.9 שמותקנת על השרת CSL3 (csl3.cs.technion.ac.il).

על מנת להריץ פייתון מתוך השרת יש לרשום python3.4 מתוך ה-SHELL של CSL3

כמו בדוגמה הזאת:

```
[muhamadg@csl3 ~]$ python3.4
Python 3.4.9 (default, Nov 27 2018, 14:18:12)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

בהינתן תוכנית פייתון שכתובה בקובץ program.py ניתן להריץ אותה ע"י הפקודה:

python3.4 program.py

או באופן כללי:

python3.4 <file name>

לנוחיותכם ניתן לכתוב ולבדוק חלקית את החלק של פייתון במחשב האישי שלכם (אבל ההגשה תהיה רק ביחס ל-CSL3).

כדי שתוכלו להריץ את הקוד על המחשב האישי שלכם יש להוריד ולהתקין את הגרסה הנ"ל ע"י ביצוע הפעולות הבאות:

1. ש להיכנס ל- <https://www.python.org/downloads/release/python-343> ולהוריד את קובץ ההתקנה המתאים (מסומן בתמונה עבור windows 64b)

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		4281ff86778db65892c05151d5de738d	19554643	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		7d092d1bba6e17f0d9bd21b49e441dd5	14421964	<a href="#">SIG</a>
<a href="#">Mac OS X 32-bit i386/PPC installer</a>	Mac OS X	for Mac OS X 10.5 and later	548f79e55708130c755bbd0f1ddd921c	24734803	<a href="#">SIG</a>
<a href="#">Mac OS X 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later	86b29d7dddc60b4b3fc5848de55ca704	23170148	<a href="#">SIG</a>
<a href="#">Windows debug information files</a>	Windows		b3d8752e74a502db97bd0c6ef30ac60f	36900012	<a href="#">SIG</a>
<a href="#">Windows debug information files for 64-bit binaries</a>	Windows		6c1be415ae552e190ef0fb06a5de9473	24301250	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		d5703787758eb1a674101ee2b0bc28be	7405996	<a href="#">SIG</a>
<a href="#">Windows x86-64 MSI installer</a>	Windows	for AMD64/EM64T/x64	f6ade29aca8fcd0463e69a6e7ccf87	25550848	<a href="#">SIG</a>
<a href="#">Windows x86 MSI installer</a>	Windows		cb450d1cc616bfc8f7a2d6bd88780bf6	24846336	<a href="#">SIG</a>

2. יש להריץ את קובץ ההתקנה ולעקוב אחרי ההוראות (ללחוץ next) עד לסיום ההתקנה

3. כרגע אפשר להריץ את תוכנית פייתון ולקבל את המסך SHELL הבא: (למשל ע"י פתיחת חלון CMD והקלדת פקודה python)

```
C:\Users\mgr99_000>python
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSC v.1600 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 5
>>> x = x + 10
>>> print(x)
15
>>>
```

למסך הזה אפשר להקליד פקודות פייתון ולקבל תחושה בסיסית לשפה.

---

לאחר ההתקנה אפשר לכתוב קוד פייתון בכל עורך טקסט (כמו notepad++) ולהריץ את הקוד על המחשב. אך קיימות סביבות פיתוח IDE - Integrated Development Environment שמאפשרות כתיבה יותר יעילה ונוחה לתוכניות גדולות. תוכנות אלו מספקות לדוגמא כתיבה בצבעים שונים, השלמות אוטומטיות של מילות קוד, גילוי אוטומטי של שגיאות ועוד. הסביבה שנדגים היא Pycharm.

התקנת Pycharm:

א. יש להיכנס ל- <https://www.jetbrains.com/pycharm/> ולהוריד את התוכנה למחשב.

ב. לסטודנטים ניתן רשיון בחינם למשך שנה אשר ניתן לחדשו כל שנה. יש להירשם לאתר כאן: <https://www.jetbrains.com/shop/eform/students> למלא בפרטים את כתובת המייל הטכניונית שלכם.

ג. לאחר גמר ההתקנה יש להריץ את התוכנה. בפתיחה הראשונה של התוכנה יש לבקש להפעיל רשיון סטודנט שם תתבקשו להזין את המייל הטכניוני שלכם וגם את הסיסמא שאיתה נרשמתם לאתר.

# 3 מימוש מערכת ניתוח נתוני בחירות ב-Python

בתרגיל זה נשתמש בפייתון כדי לקרוא ולפענח קובץ טקסט שמכיל נתונים לגבי הצבעות בבחירות. נכתוב קוד פייתון שמפענח נתונים אלה ויפעיל את תוכנית C שמימשנו בתרגיל בית 1 כדי לחשב תוצאות ממידע זה. לשם כך יש לבצע שתי משימות:

1. שימוש ב-SWIG כדי לייצר ממשק בין תוכנית פייתון לתוכנית C
  2. קריאה וניתוח לקובץ הנתונים ושליחת הפקודות דרך הממשק של שלב 1 לתוכנית C וקבלת תוצאות.
- מומלץ לפתור ולבדוק חלקית את שלב 2 לפני שלב 1 ורק אחרי בדיקה בסיסית שהקוד עובד לחבר את שלב 1. זה יאפשר לנו להתמקד רק בבעיות שנוצרות משימוש ב-SWIG ויקל על הדיבוג.

## 3.1 שימוש ב-SWIG כדי לייצר ממשק בין תוכנית פייתון לתוכנית C

חלק זה נלמד בהרצאה 5 ובמיוחד בתרגול 7 יש להשלים את קובץ הממשק הנתון `mtm_elections.i` כפי שנלמד בתרגול. ליצור קובץ מעטפת "`mtm_elections_wrap.c`" ואז להדר ולייצר ספרייה שניתן לטעון ל-python כך שיהיה ניתן לקרוא לפונקציות הבאות מתרגיל 1:

- 1.1. `mtmElectionsCreate`
- 1.2. `mtmElectionsAddCity;`
- 1.3. `mtmElectionsAddCitizen;`
- 1.4. `mtmElectionsAddCandidate;`
- 1.5. `mtmElectionsSupportCandidate;`
- 1.6. `mtmElectionsMayorOfCity;`
- 1.7. `mtmElectionsDestroy;`

על מנת לוודא שהקוד שלכם יעבור את הבדיקה שלנו יש להשתמש ב-SWIG שמותקנת על CSL3 כלומר:

- Python 3.4.9
- SWIG Version 2.0.10

השימוש ב-SWIG כפי שלומדים בתרגול 7 הוא ליצירת הקובץ:

`mtm_elections_wrap.c`

לאחר מכן יש לייצר את הספריה המשתופת shared library שנטענת לפייתון להלן דוגמה לשורות הקומפלציה.  
אתם צריכים להבין את השורות האלה ולבצע שינויים מתאימים אם יש בהתאם לפתרון שלכם:

שימו לב שורה 4 מייצרת את הספריה שנוכל לטעון לפייתון.

1. `gcc -std=c99 -fPIC -c mtm_elections_wrap.c -I/usr/local/include/python3.4m`
2. `ld -shared mtm_elections.o mtm_elections_wrap.o utilities.o  
uniqueOrderedList.o mtm_print.o city.o citizen.o -L/usr/local/include/python3.4m/  
-o _mtm_elections.so`

## 3.2 קריאה וניתוח לקובץ הנתונים ושליחת הפקודות דרך הממשק של שלב 1 לתוכנית C

חלק זה נלמד בתרגולים 6+7  
בחלק זה תוכנית פייתון מנתחת קובץ שמכיל נתונים לגבי הבחירות. תוכנית הפייתון תקרא את הקובץ ותנתח כל שורה כדי להפעיל את פונקציות C המתאימה מתוך הפונקציות של שלב 3.1. הקובץ כתוב בפורמט הבא:

city  
<city> one line. format: cityname, city\_id  
citizens  
<citizens> format: citizenName, citizenId, citizenAge, yearsOfEducation, cityId  
candidates  
<candidates> multiple lines format: candidateld, cityId  
support  
<support> multiple lines format: citizenId, candidateld, priority  
output  
<cityId>,<name of the file that should contain the output>

דוגמה לקובץ קלט בקובץ input.txt שפורסם יחד עם התרגיל.  
יש להניח שקובץ הקלט נמצא בתיקייה הנוכחית ושמו input.txt.

**הפלט** צריך להיות קובץ ששמו נקבע לפי מה שרשום בקובץ הקלט ומכיל את המועמד שזכה בבחירות.

הבהרה: הקובץ מיוצר ע"י הפעלת הפונקציה mtmElectionsMayorOfCity שהוגדרה בתרגיל בית 1.

דוגמה לקובץ פלט תקין מפורסמת יחד עם התרגיל בקובץ my\_output.txt.

### הטסטים שלנו יבדקו את תוכן קובץ הפלט ששמו מפורט בקובץ הקלט.

הערה: ניתן להניח קלט תקין ושאינן רווחים בכלל. כל השדות מופרדים ע"י " ".  
לפענוח קובץ הקלט מומלץ להשתמש בפונקציה split של פייתון שמפרקת מחרוזת למילים. ניתן לקרוא מידע נוסף על split ברשת. להלן דוגמה לשימוש:

```
>>> s = "aaa,bbb,ccc,ddd"
>>> s.split(",")
['aaa', 'bbb', 'ccc', 'ddd']
>>>
```

## הגשה

יש להגיש את הקבצים הבאים:

1. mtm\_elections.i – שהשלמתם את התוכן שלו.
2. hw2.py - קובץ פייתון שכתבתם בעצמכם.

חשוב: אין לבצע עריכה כלשהי לקבצי C או לקבצים שנוצרים אוטומטית ע"י SWIG. אנחנו נייצר קבצים אלה מחדש ולכן אין צורך להגיש אותם.

המלצות חשובות למימוש:

1. יש אפשרות למי שרוצה להגדיר ENUM (חפשו בגוגל את המחלקה Enum)
2. כדי לפענח את ערכי ההחזרה והשגיאות של פונקציות C מומלץ לכתוב פונקציה שממירה מספר שלם למחרוזת שמתארת את השגיאות.
3. כדי לקבל את המספרים שמיוצגים ב-C ע"י ENUM כללנו את ה-ENUM בקובץ הממשק mtm\_elections.i שנתנו לכם.
3. ב-C תו CHAR מיוצג בעזרת בית אחד (8 סיביות) לעומת זאת בפייתון 3 כל תו מיוצג ע"י 16 סיביות.
- אז יש להמיר כל מחרוזת לפני שליחתה לתוכנית C. ההמרה מתבצעת ע"י שימוש בפקודת פייתון str.
- דוגמה להפעלת פונקציות C מקוד פייתון, שימו לב להמרות השונות:  
result = mtm\_elections.mtmElectionsAddCitizen(me\_ptr,  
str(words[0]),  
int(words[1]),  
int(words[2]),  
int(words[3]),  
int(words[4]))
4. חשוב לבדוק את קוד הפייתון שקורא את הקובץ ומנתח אותו. אחרי שמנקים את כל הבאגים ממשיכים להריץ ולדבג את הקוד שמפעיל את תוכנית C. כי קשה מאוד לדבג את שני החלקים ביחד.
5. לנוחיותכם הוספנו הדפסה בכניסה לכל פונקציות C שמדפיסה את הפרמטרים, זה יעזור לכם בדיבוג החלק של SWIG כשאתם מפעילים את פונקציות C.

קובץ ממשק mtm\_elections.i נתון לכם **חלקית** ואתם נדרשים להשלים אותו. שימו לב לטיפול המיוחד בפונקציה שמקבלת מצביע להחזרת פלט. mtmElectionsMayorOfCity  
ולזה שיש לה **שני ערכי החזרה**. ולכן השימוש בה נראה כך:  
result, mayor\_id = mtm\_elections.mtmElectionsMayorOfCity(me\_ptr,  
int(words[0]),  
str(words[1]))