# שיפור זמן ריצה

לפייתון כבר יש תדמית של שפת תכנות חזקה, גמישה, ורסטילית ופשוטה.   
זה מה שהופך אותה לאחת השפות הפעילות יותר בשוק והשימוש בשפה רק צומח עם הזמן.   
אך עם כל כוחה של השפה, העיצוב שלה- השימוש במפרש במקום במהדר, משתנים דינאמיים ולא סטטיים ועוד, גורמים לשפה להיות איטית יותר משפות מכונה מקוריות כמו c/c++ .  
במהלך השנים מתכנתים הביאו כמה כלים לשיפור זמן הריצה של השפה, למשל האפשרות לכתוב קוד שלם ב- c או c++ ולחבר אותו לקוד קיים בפייתון, כך החלק הכבד של הקוד ירוץ במהירות המרבית של שפת מכונה ותוך כדי יהיה ניתן להשתמש בספריות של פייתון, ובאמת יצא לנו לראות כמה מספריות הפייתון שמשתמשות ב-c כבסיס לתוכנית , למשל nupy או pandas.  
במסמך הבא נסקור כמה כלים שיכולו לעזור לנו לשפר את זמן ריצת התוכנית.

**cython** –

הוא כלי תכנות לפייתון שמאפשר לנו להפוך את פייתון לסוג של שפה סטטית ובכך הוא משפר את זמן הריצה של התוכנית, ועוזר למנוע שגיאות תכנות נפוצות בשפות דינמיות.  
cython הוא שילוב כוחות בין פייתון לשפת c ,מה שנותן לנו את האופציה לכתוב קוד פייתון שניתן להעביר לתוכניות ב-c או c++.   
התקנה: בשביל להשתמש ב-cython צריך ,מלבד שיהיה לנו פייתון במחשב, גם קומפיילר מותקן ל-c. מערכות הפעלה של לינוקס אמורות להגיע כבר עם קומפיילר, ואפשר להתקין תוסף ל-visual studio code למי שמשתמש.   
בנוסף צריך להוריד את הספרייה עם pip:

pip install cython

החלק המסובך ביותר בתהליך הוא להבין כיצד לקמפל קוד פייתון לקוד c ולהפך.  
קבצי cython הם קבצים עם הסיומת .pyx'' לאחר שכתבנו את הקוד ניצור כמין קובץ makefile שכתוב בפייתון.   
הקובץ setup הזה אמור להיכל :

from setuptools import setup

from Cython.Build import cythonize

לצורך הדוגמא הקובץ שלנו הוא helloworld.pyx והוא מכיל רק פונקציה אחת והיא הדפסה של "hello world" :

print('Hello world')

אז בקובץ setup , מתחת לייבוא הספריות, נצטרך לעשות להשתמש בפונקציה setup() כדי להמיר את הקוד ל-c או ל-python כך:

from setuptools import setup

from Cython.Build import cythonize

setup(

    ext\_modules = cythonize("hello.pyx")

)

נשמור את הקובץ, למשל כ-setup.py ונקמפל אותו:

python setup.py build\_ext --inplace

עכשיו אמור להיווצר קובץ חדש- בלינוקס הוא אמור להיקרא helloworld.so ובווינדוס helloworld.pyd.  
נפעיל את המצב האינטראקטיבי ונראה שנוכל לייבא כעת את הקובץ שייצרנו:

>>> import helloworld

Hello world

סינטקס:  
יש שתי משפחות טיפוסי נתונים ב- cython.   
הראשונה היא cdef – והם כל אותם טיפוסי נתונים שמוכרים ב-c כולל מצביעים, מערכים, double ועוד ,שאינם קיימים בפייתון :

cdef int i, j, k

cdef float f, g[42], \*h

או אבייקטים ב-c מתקדמים יותר כמו מבנה או union:

cdef struct Grail:

int age

float volume

cdef union Food:

char \*spam

float \*eggs

cdef enum CheeseType:

cheddar, edam,

camembert

cdef enum CheeseState:

hard = 1

soft = 2

runny = 3

המשפחה השנייה היא משפחת האובייקט פייתון- אותם אובייקטים שלא הגדרנו אותם כמשתני c, אז הקומפיילר מקמפל אותם כאובייקטים מיוחדים של פייתון.  
חוץ מהאובייקטים יש גם שלושה סוגי פונקציות: סוג אחד הן פוקציות c שמוגדרות כ-cdef ,ניתן להשתמש בהן חופשי בתוך קובצי pyx או לקרוא להן בקובץ c , אבל אי אפשר להשתמש בהן בקובץ פייתון.  
סוג נוסף הוא פונקציית פייתון רגילה שמוגדרת כ-def, ויש עוד סוג שמשלב בין השתיים והוא פונקציית cpdef, שאומנם היא לא מהירה כמו cdef אך היא מהיר יותר מפונקציית def רגילה של פייתון.   
מומלץ להתייחס לפונקציות cdef כמו לפונקציות פרטיות, כך שמתי שאנחנו משתמשים בפונקציה cdef נשתמש בה בתוך המודול כדי לחשב דברים מתוך המודול, אבל עבור פונקציות למשתמש נעדיף להשתמש בפונקציות cpdef ולא cdef.  
לסויים נראה דוגמא לזמן להבדל בין זמן ריצה של תוכנית פייתון רגילה לתוכנית ב- cython.   
יצרנו שני קבצים שכל מה שהם עושים זה לחשב סכום אינטרבל :

פייתון:

def test(x):

    y=0

    for i in range(x):

        y+=i

    return y

cython:

cpdef int test(int x):

cdef int y=0

cdef int i = 0

for i in range(x):

y+=i

return y

והשתמש במודול timeit שמאפשר להריץ פונקציה מתוך מודול כמה פעמים, ולמדוד כמה זמן לקח להריץ את התוכנית:

import timeit

cy = timeit.timeit('cy\_example.test(500)',

                    setup='import cy\_example',

                    number = 10000 )

py = timeit.timeit('py\_example.test(500)',

                    setup='import py\_example',

                    number = 10000 )

print(f'cy = {cy}')

print(f'py = {py}')

print(f'Cython is {py/cy}x faster')  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
cy = 0.003027751000000002

py = 0.279153536

Cython is 92.19831353370861x faster

יצא ש-cython מהירה פי בערך 92 יותר מתוכנית פייתון רגילה.  
 יש עוד הרבה מה לראות בנוגע ל-cython מומלץ להסתכל ב[מדריך הרשמי שלהם](https://cython.readthedocs.io/en/latest/index.html).