

הסבר של:

```
if __name__ == '__main__':  
    main()
```

ישנן שתי דרכים להריץ תוכנית פייתון (סקריפט), אחת על ידי הרצה ישירה והשניה על ידי "יבוא" או: import של תוכנית.

לפני שפייתון מריץ את התוכנית (סקריפט), הוא קובע מספר משתני מערכת מיוחדים.

אחד מהם נקרא: `__name__`

בואו וננסה להדפיס משתנה זה ונראה מה הוא מכיל. נריץ זאת כהרצה ישירה. נקרא לתוכנית: `name_main`

```
print __name__
```

מה הודפס?

כעת ניצור תוכנית שניה שבה נבצע את ה "יבוא" של הראשונה. זיכרו שעל ידי ה- import פייתון לא רק מביא את הקוד אלא גם מבצע אותו.

כדי שהתוצאה תהיה יותר ברורה נדפיס בתוכנית הראשונה:

```
print "First program name is: ", __name__
```

נריץ את התוכנית השניה (שמיבאת את הראשונה וכמובן מבצעת אותה).

מה קיבלנו?

נריץ שוב את הראשונה. מה קיבלנו?

נוסיף את השורה:

```
print "Second program name is: ", __name__
```

מה המסקנה? כאשר מריצים תוכנית בצורה ישירה, משתנה המערכת `__name__` מקבל את הערך `__main__`. לעומת זאת כאשר מריצים את התוכנית על ידי import אותו משתנה מקבל את הערך של שם הקובץ שבו נימצאת התוכנית.

כדי להדגים זאת בצורה עוד יותר בולטת ניכתוב בתוכנית הראשונה:

```
if __name__ == '__main__':  
    print 'Run directly'  
else:  
    print 'Run from import'
```

וכעת נריץ את הראשונה ואת השניה ונראה את ההבדלים.

מדוע שנשתמש ב- import ?

כדי להשתמש שוב בקוד שכתבנו ללא צורך לקודד שוב או כדי להשתמש בקוד שמישהו אחר כתב על מנת לא "להמציא" את הגלגל מחדש

למעשה השורות שאיתן התחלנו (למעלה) אומרות: האם קובץ זה מורץ ישירות על ידי פייתון (אז בצע את main()), או שהוא מיובא ואז אל תעשה כלום. על ידי הכנסת שתי שורות אלה לכל תוכנית פייתון, ניתן לייבא אותן ללא חשש שהן תתבצענה ללא שביקשנו.

התוכנית (מודול) שאני מייבאים, יכול בדרך כלל פונקציות שימושיות עבורנו ועל ידי היבוא הן תהפוכנה זמינות עבור תוכניתנו הנוכחית.