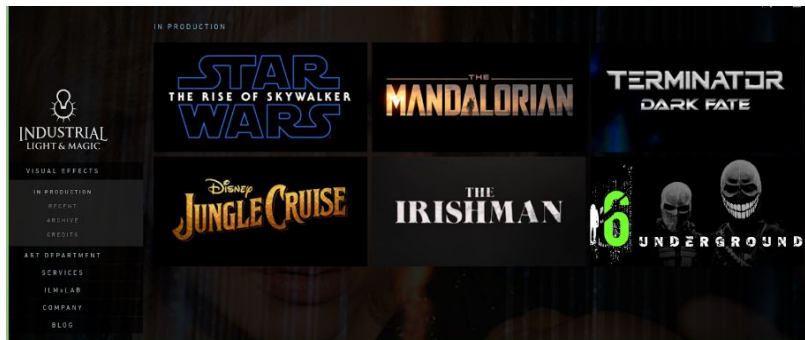




מסקרצ' לפייתון - שיעור בחירה לכיתות ד-ו

פייתון היא אחת משפות המחשב האהובות בעולם. בגלל שהיא יחסית פשוטה, משתמשים בה גם כדי ללמד מחשבים בכל הרמות, וגם כדי לייצר אפליקציות ברמות המקצועיות הגבוהות ביותר.

החברות הבאות, לדוגמא, עושות שימוש רב בפייתון:



בפייתון משתמשים באותיות אנגליות על מנת להסביר למחשב מה עליו לעשות, אבל אין צורך לדעת אנגלית ברמה גבוהה. הרבה מהמילים ניתן להמציא, ומילים אחרות הן פשוטות מאד, כגון: if, sleep, in, for, all.

הביטו בתמונה הבאה. בצד ימין ישנה תוכנית בסקראץ' ובצד שמאל אותה תכנית, כתובה בפייתון. אין צורך לקרוא או לנסות להבין את תוכנית הפייתון בשלב זה.

```

from tkinter import *
from time import sleep

tk = Tk()


MASACH = Canvas(tk, width=400, height=400)
MASACH.pack()

TMUNA1 = PhotoImage(file='figures/costume1.png')
TMUNA2 = PhotoImage(file='figures/costume2.png')

for i in range(20):
    MASACH.create_image(0, 0, anchor=NW, image=TMUNA1, tag='a')
    MASACH.update_idletasks()
    sleep(0.3)
    MASACH.delete('a')
    MASACH.create_image(0, 0, anchor=NW, image=TMUNA2, tag='b')
    MASACH.update_idletasks()
    sleep(0.3)
    MASACH.delete('b')

mainloop()

```





האם ניתן להבין את התוכנית שכתובה בסקראץ'? זו תוכנית שתחזור על פעולה פעמים. או במילים אחרות - לולאה. בכל חזרה, התוכנית תציג את התמונה costume1 למשך 0.3 שניות, ואז את תמונה costume2 למשך 0.3 שניות.



כך נקבל סרטון הנפשה בו נראה שהחתול של סקראץ' הולך.

עכשיו נבחן את התוכנית בפייתון, שבצד שמאל:

```
from tkinter import *
from time import sleep

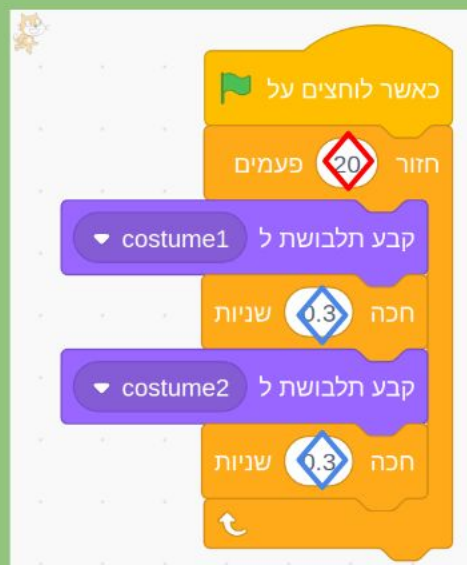
tk = Tk()

MASACH = Canvas(tk, width=400, height=400)
MASACH.pack()

TMUNA1 = PhotoImage(file='figures/costume1.png')
TMUNA2 = PhotoImage(file='figures/costume2.png')

for i in range(20):
    MASACH.create_image(0, 0, anchor=NW, image=TMUNA1, tag='a')
    MASACH.update_idletasks()
    sleep(0.3)
    MASACH.delete('a')
    MASACH.create_image(0, 0, anchor=NW, image=TMUNA2, tag='b')
    MASACH.update_idletasks()
    sleep(0.3)
    MASACH.delete('b')

mainloop()
```



בשלב ראשון אנחנו מגדירים את המסך: **MASACH**. אנחנו בחרנו מילה, שאמנם כתובה באותיות באנגלית, אבל היא פשוטה לנו להבנה. כמובן שהמשך השורה צריך להיות מדויק על מנת שפייתון יבין את המשמעות של **MASACH**, אבל נגיע לדברים כאלו בהמשך.

אחר כך אנחנו מגדירים את התמונות: **TMUNA1** ו**TMUNA2**. וכעת הגיע הזמן להגדיר את הלולאה שתחליף את התמונות, כמו בתוכנית הסקראץ':

```
for i in range(20):
```

כך אנחנו אומרים למחשב לעשות משהו **עבור (for)** כל מספר מתוך 20 מספרים. הדבר שעל המחשב לבצע מופיע מיד אחריו בבלוק של טקסט שמוסט מעט פנימה.

האם ניתן לראות הקשר בין הלולאה של תוכנית הפיתוח ללולאה שבתוכנית
הסקראץ'?

הכנה

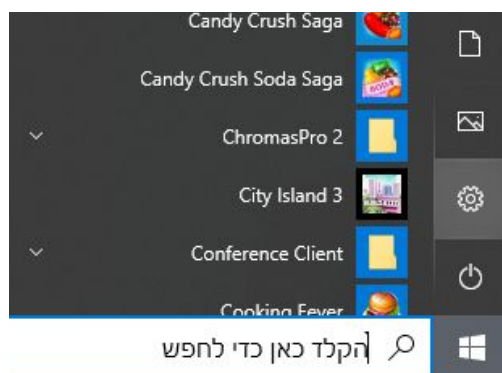
התקנת סביבת העבודה על המחשב

על מנת שנוכל לכתוב תוכניות בפייתון, עלינו להתקין על המחשב מפענח, שיתרגם את התוכניות שלנו לשפת מכונה - שפתו של המחשב, וגם סביבת עבודה, על מנת שנוכל להקליד תוכניות. סביבת העבודה שנעבוד בה נקראת jupyter notebook.

שני המרכיבים האלו כלולים בחבילת תוכנה הקרויה "אנאקונדה" anaconda.

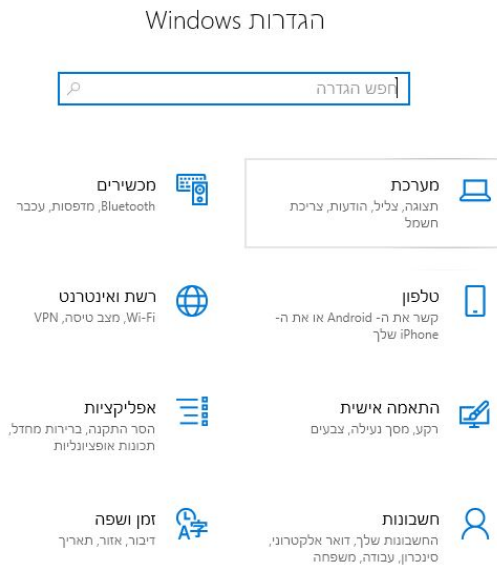
ראשית, על איזה מחשב אנחנו עובדים - 32 או 64 סיביות?
ניתן לברר זאת בקלות:

1. פתחו את תפריט ההגדרות, על ידי לחיצה על הלוגו של חלונות ואח"כ על גלגל השיניים:

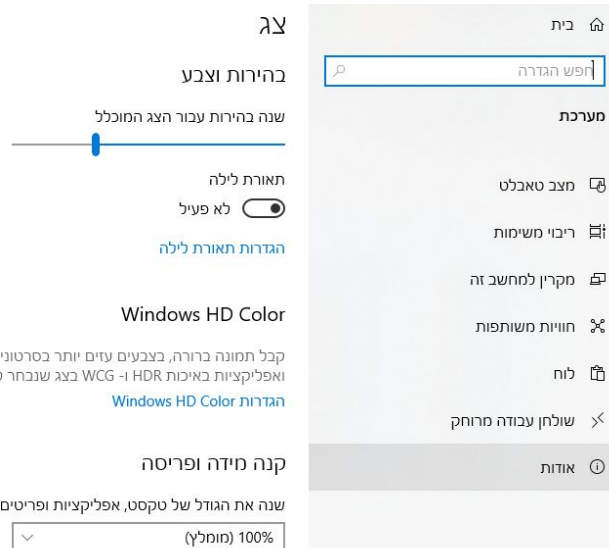


יפתח חלון ההגדרות.

2. לחצו על "מערכת":



3. כעת יש לגלול את התפריט עד למטה ולבחור "אודות":

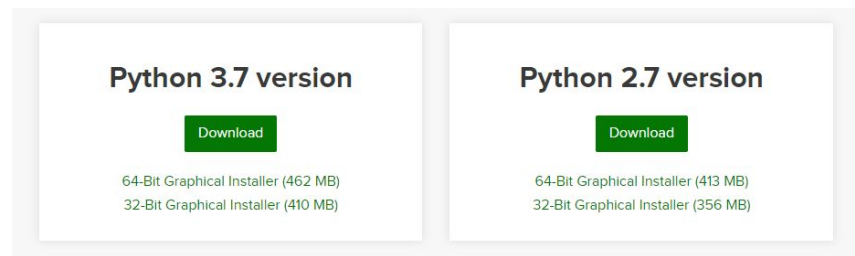


4. בתחתית החלון תוכלו לראות את "סוג המערכת": 32 או 64 סיביות - bit:



כעת נוריד ונתקין את אנקונדה. התהליך יכול להמשיך חצי שעה.

גילשו לכתובת הבאה: <https://www.anaconda.com/distribution>
גללו כלפי מטה ואתרו את החלק הזה של העמוד:



בקורס הזה נעבוד עם גרסת פייתון 3.7. בתחתית הקוביה של פייתון 3.7 ישנם שני קישורים, האחד עבור 64 סיביות והשני עבור 32. הקליקו על הקישור שמתאים למחשב שלכם:

Download

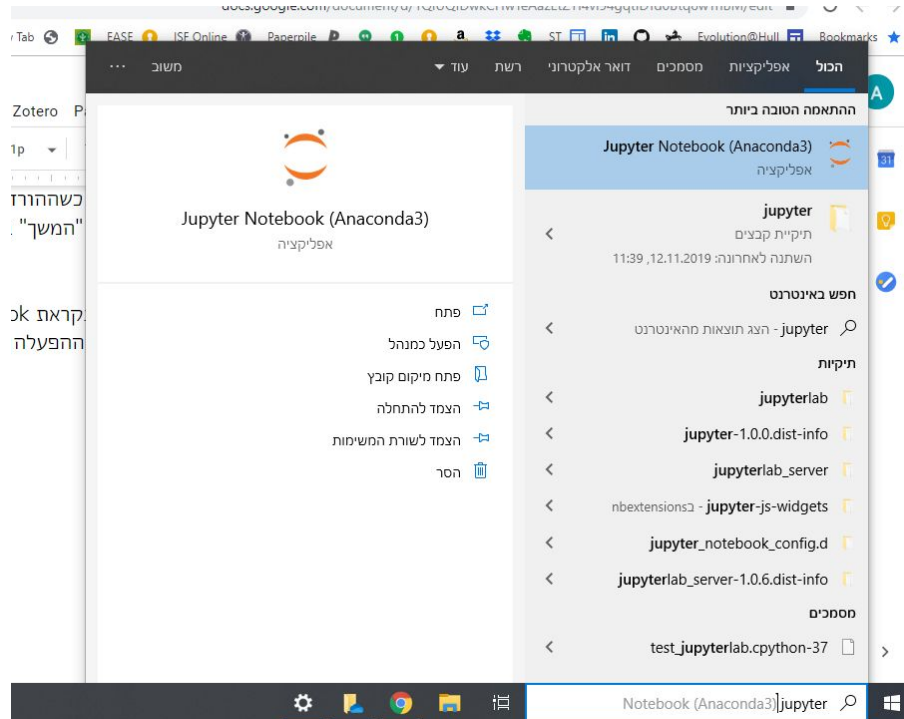
64-Bit Graphical Installer (462 MB)

32-Bit Graphical Installer (410 MB)

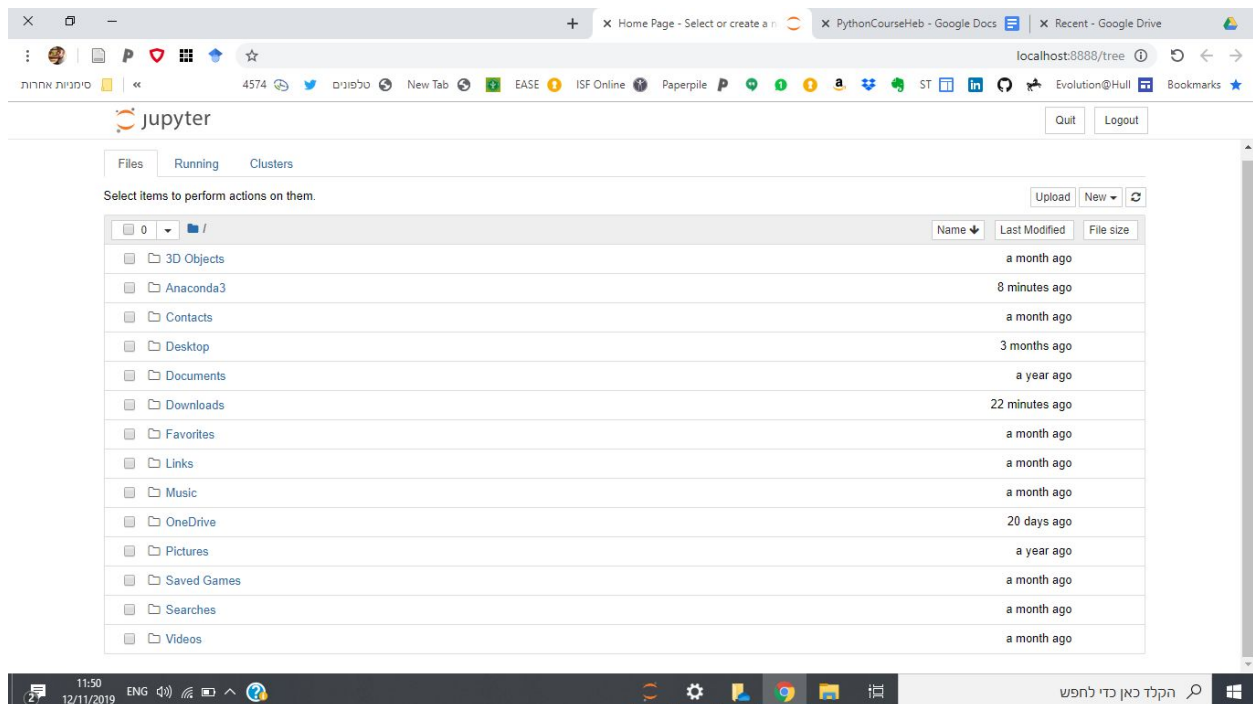
כעת יש להמתין שהמתקין האוטומטי ירד אל המחשב. כשההורדה מסתיימת, יש ללחוץ על הקובץ, ולעקוב אחר ההוראות. ניתן ללחוץ "המשך" בכל שלב, אין צורך לשנות דבר.

כאשר התקנה מסתיימת, נפתח את סביבת העבודה שנקראת jupyter notebook:

חפשו את המילה jupyter בשורת החיפוש של מערכת ההפעלה. המערכת תזהה את התוכנה, וניתן יהיה ללחוץ על מנת להפעיל אותה:



התוכנה נפתחת בדפדפן האינטרנט:



מסך הבית שלה, הוא תכולת התיקיות של המשתמש, כולל התיקיה Documents. זוהי אותה ספריית Documents של המשתמש שממוקמת ב C:\Users\username\Documents, כאשר username הוא השם שלכם.

(הערה: בחלק מהמחשבים החדשים, ספריית המסמכים הרגילה נמצאת בתיקית OneDrive ולא בתיקייה של המשתמש. עם זאת, התיקייה אליה יש גישה מהתוכנה היא זו שצינתי למעלה).

jupyter notebook עובדים עם קבצי מחברת. את קובץ המחברת עבור השיעור הראשון ניתן להוריד בקישור הזה:

https://www.dropbox.com/s/lpepa4qd5ccehy5/StrIntVarFunc_part1.ipynb?dl=0

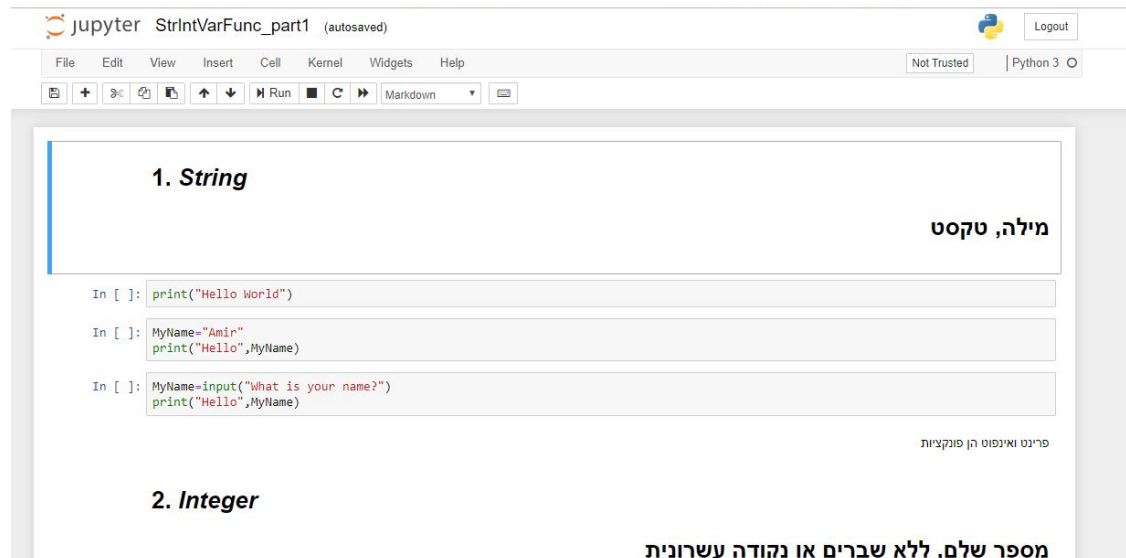
את הקובץ StrIntVarFunc_part1.ipynb, יש להעביר אל תיקיית המסמכים Documents.

פתיחת המחברת בסביבת העבודה

כעת חיזרו אל העמוד של סביבת העבודה שלכם בדפדפן, וליחצו על הקישור לתיקיית "Documents". תופיע רשימת הקבצים בתיקייה, כולל הקובץ החדש - StrIntVarFunc_part1.ipynb, עליו יש ללחוץ כדי לפתוח את המחברת:



לחצו על שם הקובץ (לא על הריבוע שלידו), והמחברת תפתח בלשונית חדשה בדפדפן:



האזורים האפורים בכל מחברת הם אזורים בהם אנו כותבים קוד - הוראות בפייתון. כדי להריץ את הקוד שכתבנו, לוחצים על הכפתור Run בראש העמוד.

כדי לנסות, לחצו על המלבן האפור העליון, בו כתוב

```
print("Hello World")
```

ואז לחצו על הכפתור Run.

מכיון שמשמעות המילה `print` היא הדפס, המחשב הדפיס את המילים `Hello World` שמופיעות בסוגריים, מתחת לריבוע האפור.

עד כאן ההתקנה של סביבת העבודה, וכעת נציג כמה מושגים.

שיעור מספר 1:

חפצים, פונקציות ומשתנים

חפצים

פייתון, כמו שפות רבות אחרות, היא שפה "מוכוונת חפצים". אנחנו מתייחסים אל התוכניות שלנו כאילו היו חפץ.

כדוגמא, נאפיין כעת את החפץ הבא:



החפץ הזה הוא עט עם צבעים מתחלפים, שימו לב שכעת לחוץ הכפתור של הצבע האדום.

לחפצים יש תכונות ומאפיינים. מאפיינים של עט יכולים להיות הצבע שלה ועובי הקו שהיא עושה.

לחפצים יש גם שיטות שימוש. בעט שלנו לדוגמא, שיטת השימוש היא כתיבה. וכשאנחנו כותבים תוכנית, אנחנו אוהבים לפרק זאת להרבה שיטות שימוש קטנות ומוגדרות היטב. את שיטת השימוש "כתיבה" ניתן לפרק למספר שיטות כמו "הנחה על הדף", "פנייה לכיוון מסויים", "התקדמות על גבי הנייר למרחק של 2 ס"מ", "הרמה מהדף" ועוד.

ישנן גם שיטות שימוש שמטרתן לשנות את המאפיינים. לעט שלנו, לדוגמא, יש כפתורים שמאפשרים לנו לשנות את הצבע של העט.

בפייתון אמנם ישנם "חפצים" מאד מגוונים ומעניינים, אבל בחלק זה נכיר את הבסיסיים ביותר, כאלו שנמצאים בכל תוכנית ותוכנית.

החפץ string - "שרשרת אותיות" או במילים אחרות "מילים", או "טקסט"

מילים ומשפטים בפייתון הופכים גם הם לחפץ.

בסעיף הראשון במחברת אפשר לראות איך חפץ string נראה בתוכנית פייתון:

```
In [2]: print("Hello World")
Hello World
```

כדי שהמחשב ידע שאנחנו כותבים מילה או טקסט, המילים צריכות להופיע במרכאות. בשלב זה נקפיד על מרכאות כפולות:

"Hello World"

אם נכתוב את אותן המילים ללא המרכאות, Hello World, המחשב לא יבין אותנו. מאחר ש string הוא חפץ, יש לו מאפיינים ושיטות עבודה.

לדוגמא, מכיוון שבאנגלית יש אותיות קטנות ואותיות גדולות (Upper-case) לחפצים מסוג "מילה", string, יש שיטה שנקראת upper שהופכת את האותיות לאותיות גדולות. על מנת להפעיל שיטות של חפץ, נכתוב אחרי החפץ נקודה "." ואת שם השיטה, בסיום סוגריים:

"Hello World".upper()

נסו להוסיף את השיטה upper למילה שהדפסת קודם לכן והריצו שוב את המלבן על מנת לראות את התוצאה:

```
print("Hello World".upper())
```

פונקציות

במשבצת האפורה הראשונה במחברת, ביצענו פעולת הדפסה. שימו לב שהקריאה לפעולת ההדפסה, לא תלוייה בקיומו של חפץ. כלומר, בשונה מ-upper, פשוט כתבנו `print()`, ללא נקודה או שם של חפץ שקדמו לכך.

פעולה שאינה תלוייה בקיומו של חפץ נקראת **פונקציה**. במקרה של `print()` הכנסנו לתוך הסוגריים את החפץ שברצוננו להדפיס. אולם יכולנו לקרוא לפונקציה גם בלי לספק לה שום חפץ. נסו זאת במחברת:

`print()`

אמנם לא הודפס דבר, אבל הפונקציה פעלה.

משתנים

כל שפות התכנות, וגם פייתון, מאפשרות להשתמש במשתנים.

כאשר בונים מכונה, או כותבים תוכנת מחשב, אנחנו מעוניינים שיהיו לה שימושים רחבים ככל שניתן. לכן מצד אחד, התוכנית או המכונה מוגדרות היטב כדי שתהיינה מדויקות, אך מצד שני, אנחנו מאפשרים למשתמש לשנות דברים.

הביטו בתמונה הבאה:



זוהי תמונה של מכונה לצביעת בובות פלסטיק.

זוהי אינה מכונה לצביעת בובות פלסטיק של מינימאוס בצבעי שחור וורוד, או לצביעת בובת פלסטיק של ילד בצבעי אדום וכחול, אלא מכונה שמאפשרת לשנות כמה דברים בין הפעלה להפעלה, על מנת שניתן יהיה לצבוע בובות בצורות שונות ובצבעים שונים.

לדוגמא, צורת הבובה היא "משתנה". המשתמש יכול להחליף את צורת הבובה, והמכונה תשנה את פעולתה במקצת, כדי להתאים לתוכן החדש של ה-"משתנה".

בפייתון, כשרוצים ליצור משתנה ולהזין אותו בתוכן, עושים זאת באופן הבא:

```
MyName="Amir"
```

עלינו לתת שם למשתנה:

```
MyName
```

ולהזין אותו בתוכן על ידי שימוש בסימן "=".

בדוגמא הזו, התוכן שהזנו לתוך המשתנה, הוא חפץ מסוג string, או מילה: "Amir"

כעת אנחנו יכולים להשתמש במשתנה בתוך ה"מכונה" או התוכנית שלנו:

```
print("Hello " + MyName)
```

שימו לב ששם המשתנה הוא ללא מרכאות.