**סדנת AI - דף הסבר למשתתפת**

אז ברוכות הבאות לאתגר ה-AI של ה-BOOTCAMP.

כפי שנמסר לכן בתדריך שעברתן, המטרה שלכן באתגר זה היא לממש אלגוריתמים אוטומטיים ומתוחכמים שיסייעו בסופו של דבר להגדלת הספקי האימוץ האפשריים.

נזכיר את סיפור הרקע שוב – למי שפספסה:

# רקע למשימה

יש לנו רשימת מאמצים.

הרשימה איננה סטטית – כל שנייה מתווסף מאמץ חדש.

לכל מאמץ יש רשימה של חיות שמציינת איזה חיה הוא מעוניין לאמץ (כך שמאמץ עם רשימה המכילה את החיה חתול בלבד מסמל מאמץ שמעוניין לאמץ חתול בלבד, ואילו מאמץ עם הרשימה: חתול, כלב, תוכי מעוניין לאמץ אחת מחיות אלו ולכן אם יקבל אחת מהן יהיה מרוצה).

בנוסף, יש לנו מרחב גדול של משבצות אשר בהן מופיעות חיות (בכל משבצת מופיעה חיה בודדה, יש משבצות ללא חיות).

המטרה: לשלוח מאמצים למשבצות אשר מכילות חיות שהם מעוניינים בהם.

למה זה בעיה ואי אפשר פשוט ללכת בגישת הניסוי והטעייה (שלח מאמץ למשבצת – אין שם חיה? שילך למשבצת אחרת)? התשובה:

1. לוקח זמן לשלוח מאמץ למשבצת – חבל לבזבז אותו לחינם
2. מאמצים מאוד לא מרוצים ששולחים אותם לחינם. הם עלולים לתבוע את החברה שלנו או להפסיק לעבוד איתה. ולכן על כל שליחה מיותרת של מאמץ, החברה שלנו תשלם "קנס" כספי. על כן, המוטיבציה שלנו הוא לשלוח מאמצים למשבצות טובות בלבד.

אם כן, איך נדע לאן לשלוח מאמץ כך שהאימוץ יהיה בהכרח מוצלח?

לשם כך יש ברשותנו רחפנים שאנחנו יכולים להטיס למשבצות שנבחר.

לכל רחפן כזה יש סנסורים משוכללים שיכולים לאסוף: "גרף תנועה של חיה" באותה המשבצת.

אם נצייד את הרחפן באלגוריתם למידת מכונה משוכלל (המשימה של צוות 2 בכל מוצר), הוא יוכל לשערך בצורה די סבירה – על פי גרף התנועה שאסף, איזו חיה נמצאת במשבצת.

צוות 3, יממש אלגוריתם שייקח את המידע שהרחפנים אספו ועל פיו יחליט לאן לשלוח מאמצים (בהנחה שאלגוריתם של צוות 2 מושלם – אין פה ממש אתגר, פשוט לשלוח מאמץ למשבצת בה נאסף גרף תנועה של חיה שהוא מעוניין בה. אולם – במציאות אלגוריתם למדה שכזה עשוי להיות לא מדויק, ואלגוריתם חכם של צוות 3 יכול לפצות על כך).

ומה צוות 1 עושה בכל העניין הזה? ובכן – מטרתו של צוות 1 הוא לכתוב אלגוריתם שמתפעל את הרחפנים – קרי: מחליט לאיזה משבצות לשלוח אותם לצורך איסוף המידע הסנסורי של גרפי התנועה (אם אתן מרימות כרגע גבה על סמך איזה מידע הוא אמור לעשות זאת ולמה יש פה בכלל אלגוריתמיקה מתוחכמת ולא מנגנון אקראי פשוט – אל דאגה: הכל יוסבר בהמשך).

עבודה טובה של כל אחד מהצוותים תתרום לביצועים טובים יותר של תהליך האימוץ הכולל – כל רכיב חכם משפיע על הביצועים הסופיים. כמובן, שכמו בחיים – אלגוריתם מאוד מתוחכם של צוות מסוים יוכל לפצות על מימוש פחות טוב של צוות אחר ובכך להרים את איכות המוצר.

# אז איך מתחילים?

שאלה מצוינת. טוב ששאלתן.

אז כאמור, העבודה היא בצוותים.

על כל צוות להוריד את הקוד שבתא המתאים לו בטבלה הבאה:

|  |  |
| --- | --- |
| צוות | קישור לקוד |
| צוות 1 | [קוד קבוצה 1](https://gitlab.com/nir199820/cat_dog_hackaton_drone_handler) |
| צוות 2 | [קוד קבוצה 2](https://gitlab.com/nir199820/cat_dog_hackaton_animal_classifier) |
| צוות 3 | [קוד קבוצה 3](https://gitlab.com/nir199820/cat_dog_hackaton_animal_adopter_matcher) |

אז הורדתן את הקוד הרלוונטי לקבוצה שלכן. מה הלאה?

אז במהלך האתגר אתן תעבדו על מחברת JUPYTER מפורטת שהכנו עבורכן.

איפה היא נמצאת? אז בתוך התיקייה שהורדתן יש תיקייה בשם team1\_solution (במידה ואתן צוות מספר 1, אם אתן צוות מספר 2 – שמה יהיה: team2\_solution, ואם אתן צוות מספר 3 שמה יהיה – team3\_solution). היכנסו לתיקייה זו.

בתוך תיקייה זו יש קובץ עם סיומת ipynb – זאת מחברת ה-JUPYTER הרלוונטית לאתגר שלכן. פיתחו אותה באמצעות ה-anaconda navigator והתחילו לעבוד עליה על פי ההוראות הרשומות בתוכה.

שכחתן איך פותחים מחברת JUPYTER? אין בעיה – ישנן הוראות מפורטות בנספח המופיע בסוף מסמך זה.

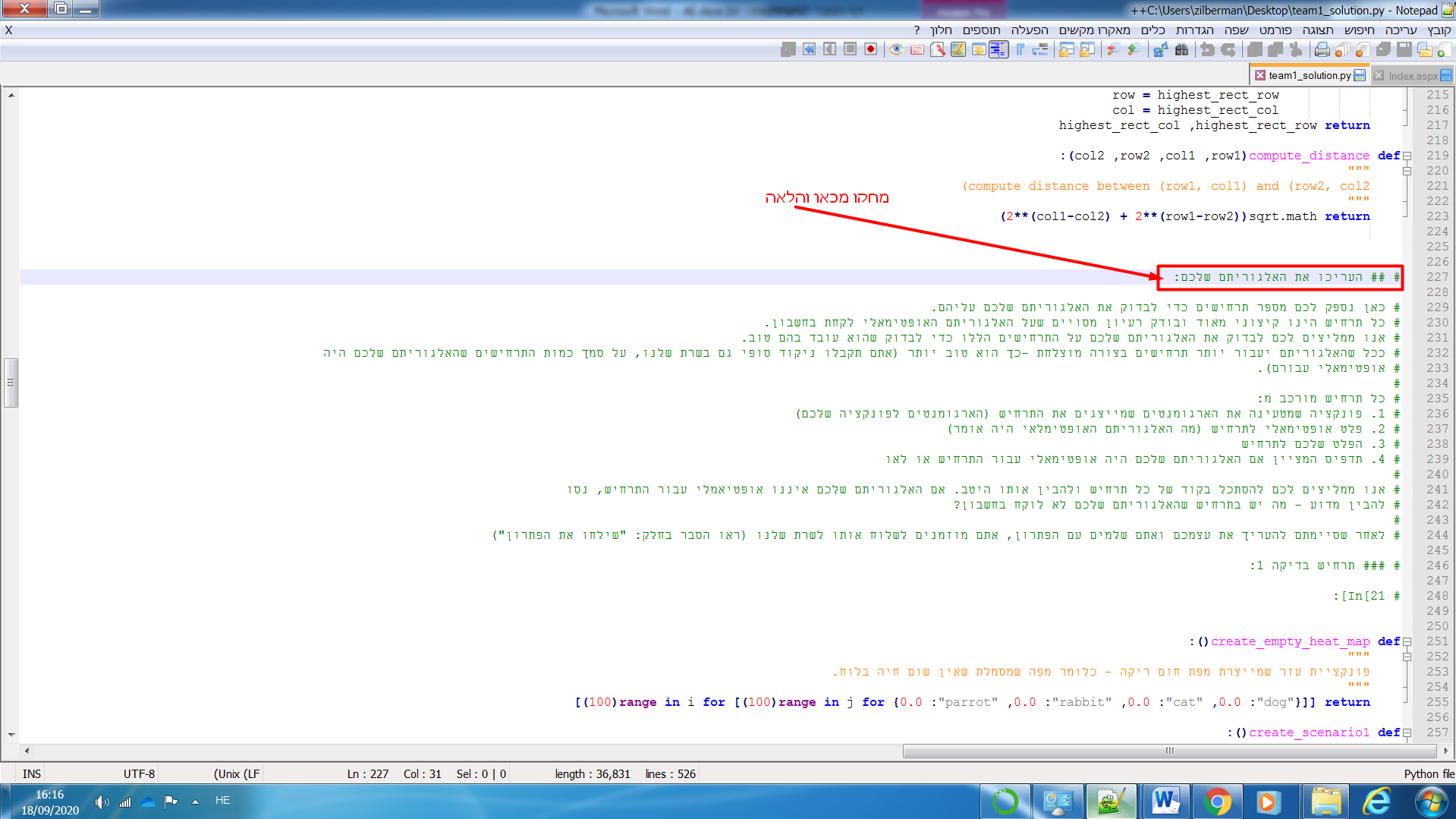
המחברת מכילה הוראות מפורטות. קראו אותם בקפידה וממשו את הפונקציה המרכזית שניתנה לכם במחברת. המחברת מכילה כבר מימוש בסיסי מאוד לפונקציה. אתן רשאיות להרחיב אותו או לחלופין להחליפו במימוש אחר לחלוטין – רק הקפידו לא לשנות את חתימת הפונקציה. ניתן לממש פונקציות עזר כרצונכן. בהצלחה!

# מימשתן את הפונקציה במחברת, מה הלאה? או במילים אחרות: איך שולחים את הפתרון ומה אמור לקרות לאחר השליחה?

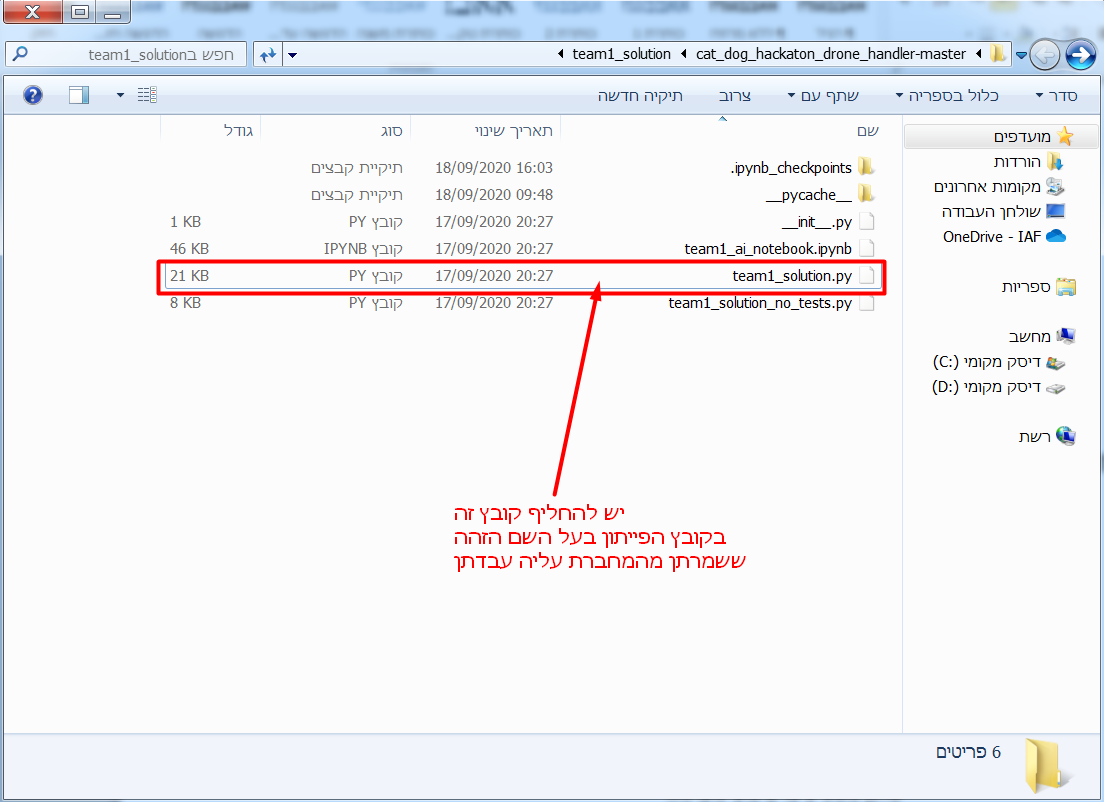
דבר ראשון ברכותינו, זה לא מובן מאליו.

לאחר שסיימתן לכתוב את הפתרון במחברת ששלחנו לכן, עליכן:

1. לשמור את המחברת כקובץ פייתון (בסוף מסמך זה, הוספנו הסבר קצר כיצד לעשות זאת בצורה פשוטה – נספח 2)
2. למחוק מקובץ הפייתון את החלקים המיותרים: המחברת שסיפקנו לכן מכילה גם פונקציות בדיקה שעוזרות לכן להבין עד כמה הקוד שלכן טוב. יש למחוק חלקים אלו מקובץ הפייתון ששמרתן. איך מזהים מה צריך למחוק? חפשו את החלק בקוד שמכיל את ההערה: "העריכו את הקוד שלכן" ומחקו את כל הקוד ממנו והלאה (ניתן לפתוח את קובץ ה-PYTHON באמצעות כל IDE, כולל SPYDER שהותקן יחד עם ANACONDA ואפילו באמצעות NOTEPAD++):



1. מצויין, כמעט סיימנו. כעת יש לכם קובץ פייתון שמור בשם teamX\_solution.py (כאשר X זה מספר הצוות שלכן). כעת עליכן לשים אותו במקום המתאים בתיקיית הקוד שהורדתן מה-gitLab: באותה התיקייה בה נמצאת מחברת ה-jupyter בה עבדתן (teamX\_solution - כאשר X הוא מספר הצוות שלכן) יש קובץ בשם teamX\_solution.py – החליפו אותו בקובץ בעל השם הזהה ששמרתם מתוך מחברת ה-jupyter עליה עבדתן (ומחקתם ממנו את הקוד הלא רלוונטי הקשור להערכה).



1. מה שנותר כעת הוא להריץ את שרת הפייתון המקומי ואז סיימתן (שרת הפייתון המקומי ישלח לשרת המרכזי שלנו את ההחלטות שהאלגוריתמים שלכן עשו. החלטות אלו ישוקפו במסך האפליקציה של הפרוייקט שלכן ובדף התוצאות).

איך מריצים את שרת הפייתון המקומי – קראו כל כך בנספח 3: "כיצד להריץ את שרת הפייתון המקומי".

1. אוקיי, אז עבדתם על מחברת ה-JUPYTER, ברגע שהייתן מרוצות אז שמרתן ממנה קובץ פייתון ושמתם אותו במקום המתאים בתיקיית הפרוייקט ואפילו הרצתם את שרת הפייתון המקומי. אבל זה לא כל כך מסעיר, נכון? איך "רואים" את התוצר של מה שעשיתן. ובכן – 3 דרכים:
   1. תחילה ניתן לראות את ה-SCORE שהפתרון שלכן קיבל (ציון בין 0 ל-100). כדי לראות את ה-SCORE ניתן להיכנס לקישור הבא וללחוץ על כפתור ה"PLAY": <http://129.213.158.42:9000/graphiql?query=query%20%7B%0A%20%20calculateScore%7B%0A%20%20%20%20score%0A%20%20%20%20challenge%0A%20%20%7D%0A%7D>

את ה-PORT שבקישור החליפו ב-PORT המתאים למספר הפרוייקט/מוצר שלכן (מוצר/פרוייקט מספר 1 = פורט 9001, מוצר/פרוייקט מספר 2 = פורט 9002 וכך הלאה).

חוץ מלראות את ה-SCORE של הפרוייקט/מוצר אליו אתן שייכות (ובתוך כך גם את הציון של הצוות הספציפי שלכן), ניתן גם לראות את מצבכן יחסית לקבוצות ולצוותים האחרים. לצורך כך ניתן להיכנס לקישור הבא:

<http://129.213.158.42:3003/>

* 1. בנוסף, תוכלו לראות את הבקשות שהשרת שלכן שולח לאפליקציה באמצעות הממשק הבא:

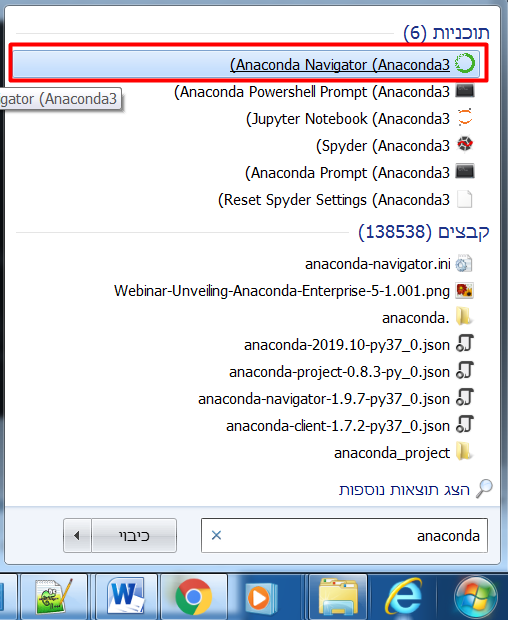
<http://129.213.158.42:9000/graphiql?query=query%20%7B%0A%20%20openEvents(isOpen%3Atrue)%7B%0A%20%20%20%20id%0A%20%20%20%20gridCell%7B%0A%20%20%20%20%20%20x%0A%20%20%20%20%20%20y%0A%20%20%20%20%20%20id%0A%20%20%20%20%7D%0A%20%20%7D%0A%7D>

החליפו את הפורט בכתובת זו בפורט המתאים לקבוצה/פרוייקט/מוצר שלכן (מוצר 1 – פורט 9001, מוצר 2 – פורט 9002 וכך הלאה)

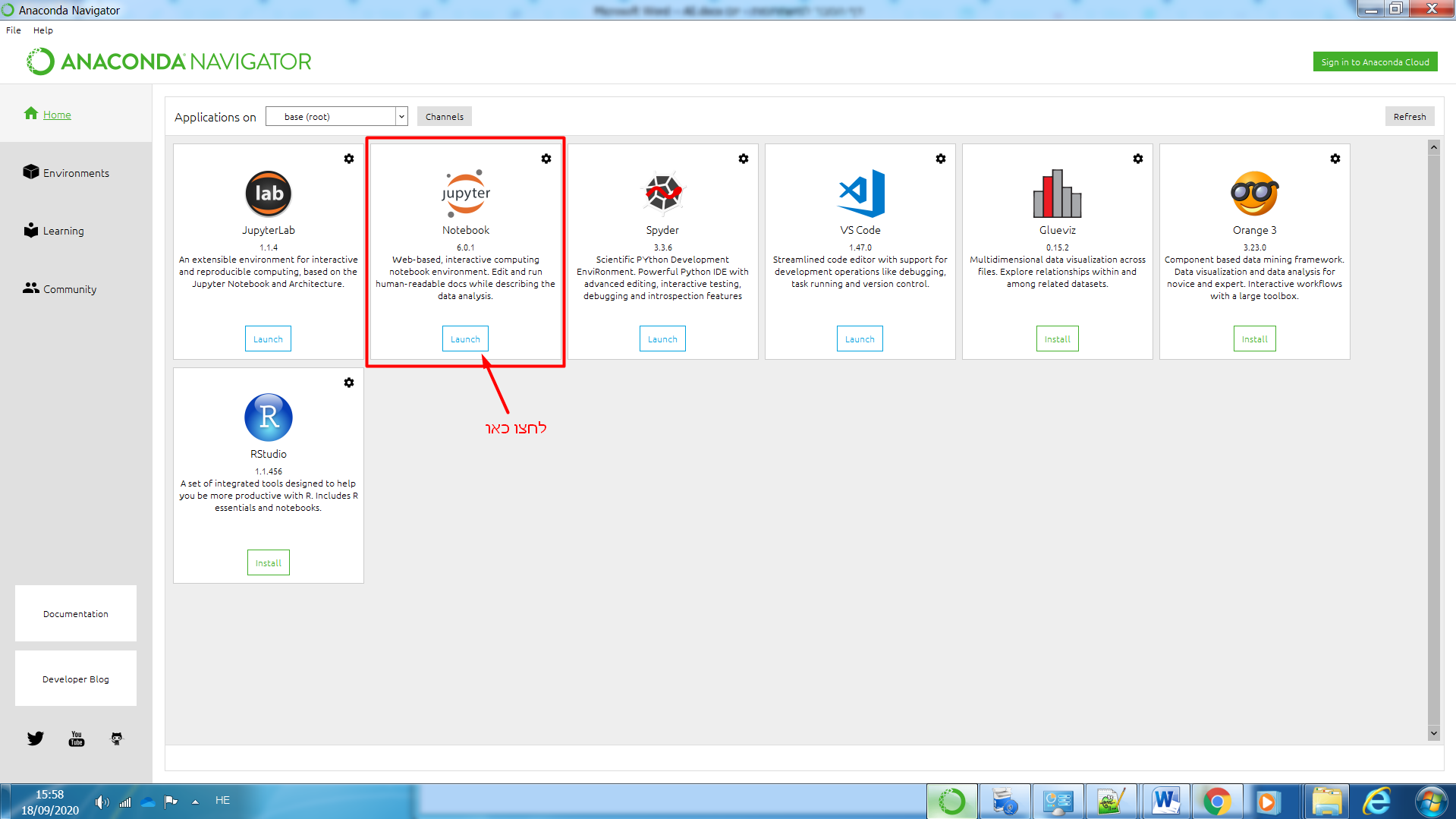
* 1. מלבד הציון הספציפי שקיבלתן על התוצר שכתבתן, ניתן גם לראות ב"עיניים" כיצד התוכנה מתנהגת עם האלגוריתם החדש שמימשתן עבורה, לצורך כך פשוט היכנסו למסך ה-UI המתאים של התוכנה שמימשתן אתמול, במסגרת אתגר ה-UI. אתן מוזמנות לראות כיצד התוכנה מתנהגת? האם האלגוריתם שלכן גורם לה לעבוד כמצופה? תמיד נחמד לראות איך האלגוריתמים שלנו משתקפים בפועל, ולכן זה החלק החוויתי שממליצים לא לוותר עליו.

# נספח 1 – איך פותחים מחברת JUPYTER

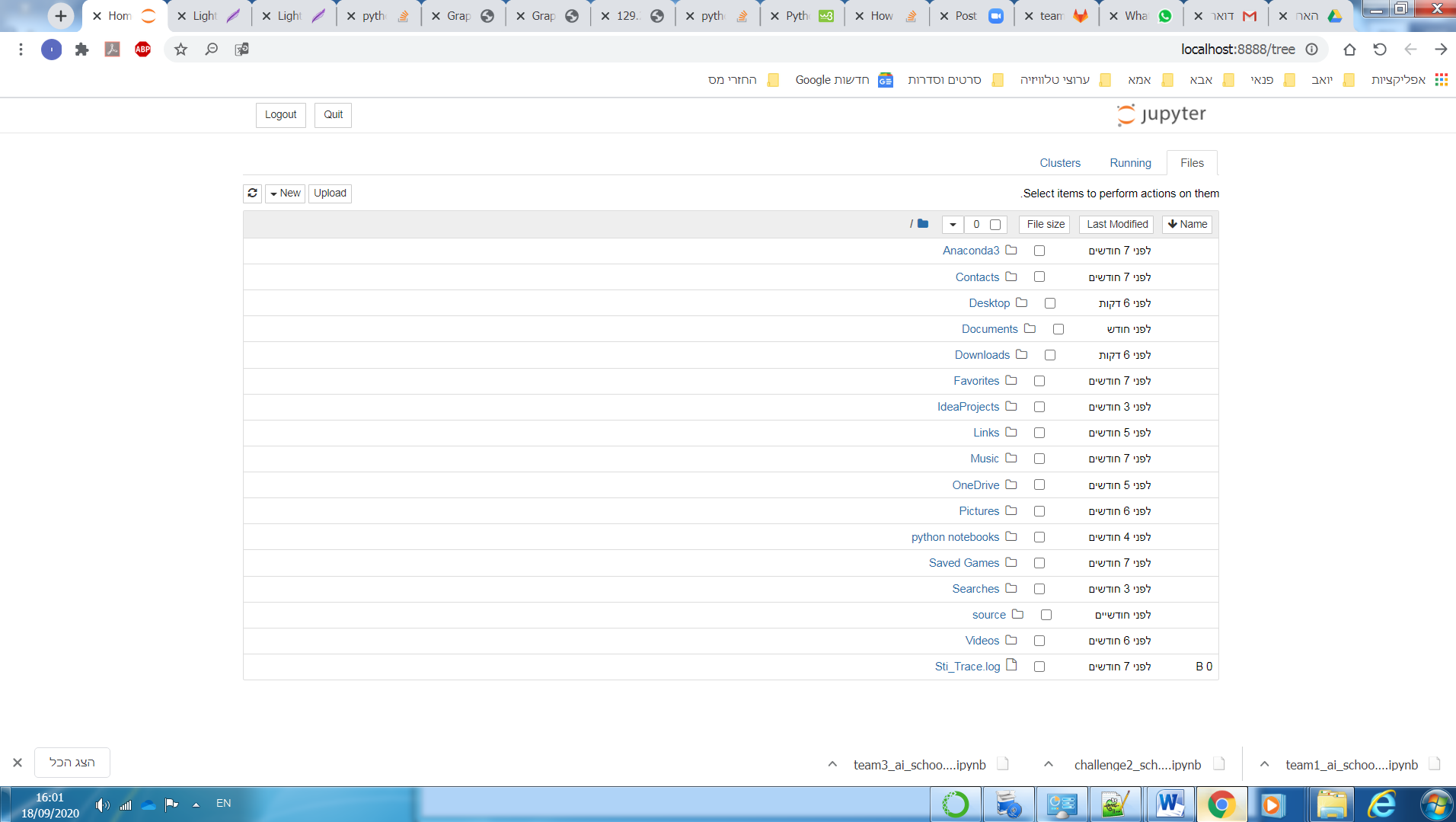
פתחו את ה-ANACONDA NAVIGATOR. תוכנה זו הותקנה יחד עם חבילה anaconda שהתבקשתן להתקין:



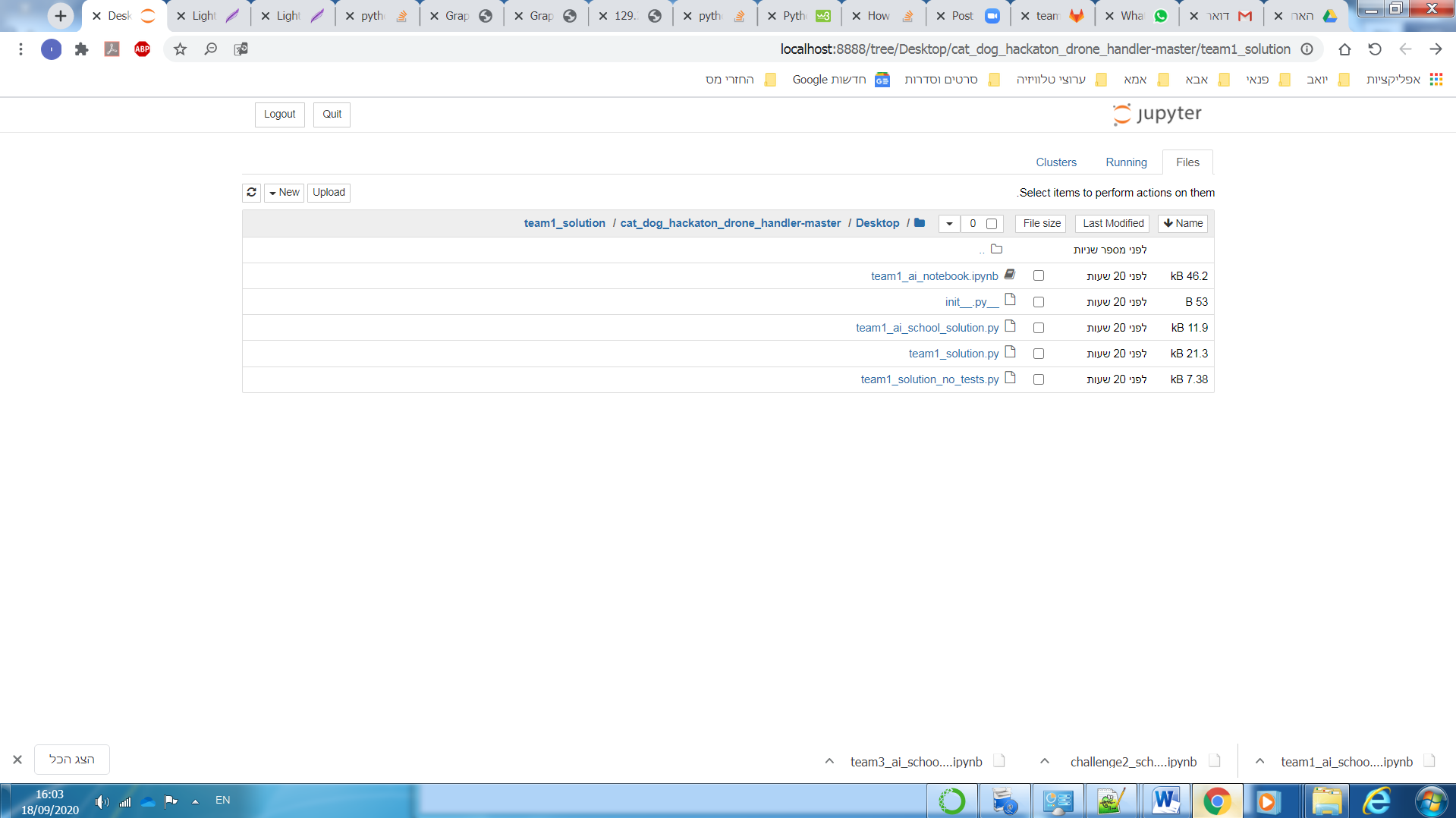
לתוכנה לוקח מעט זמן להיטען, אך ברגע שהיא תיפתח, היכנסו ל-jupyter notebook:



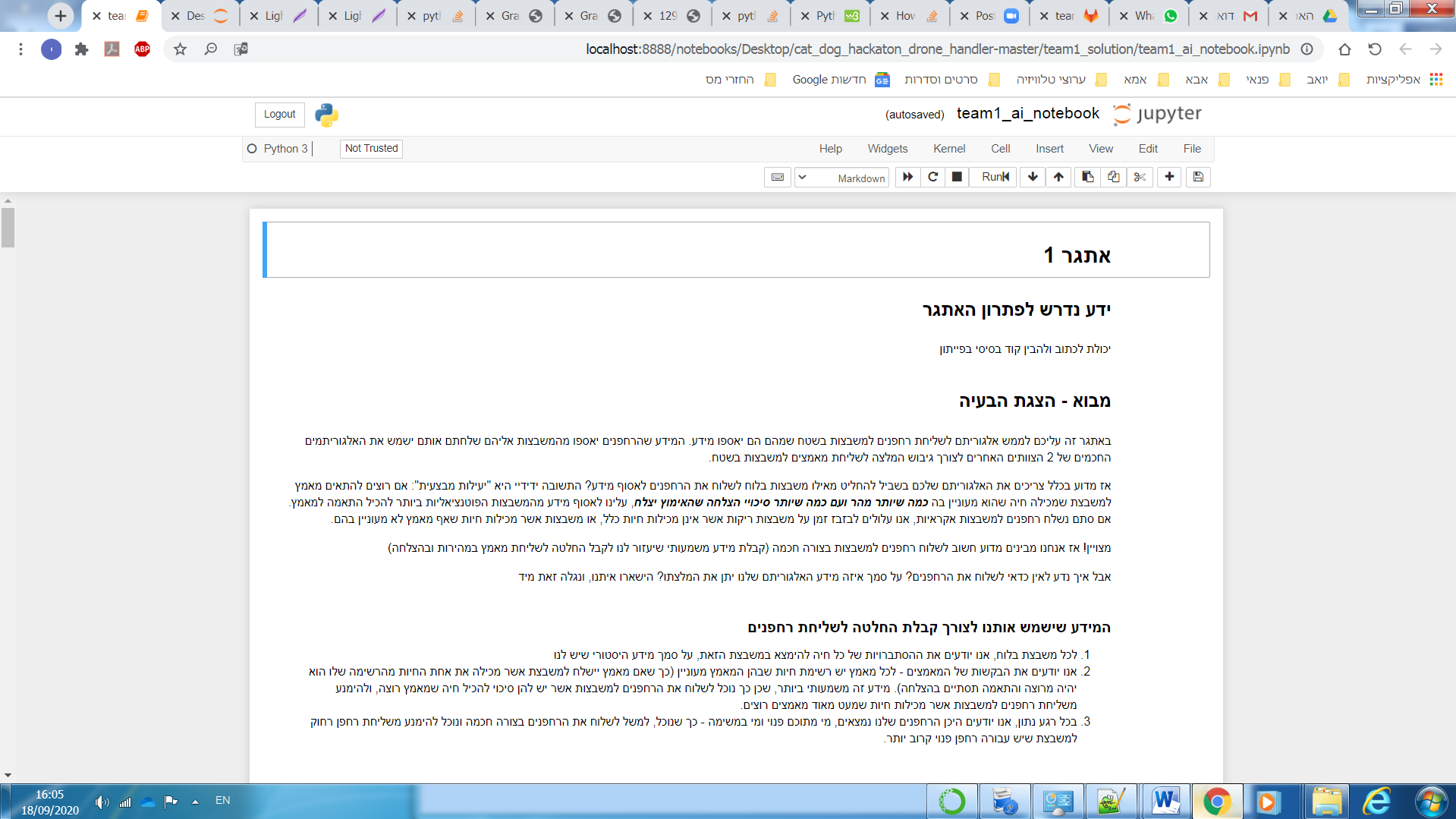
ייפתח בדפדפן המסך הבא:



המסך למעשה מכיל את היררכיית התיקיות במחשב שלכן – פשוט נווטו בתוכו למקום בו המחברת שהורדתן נמצאת:

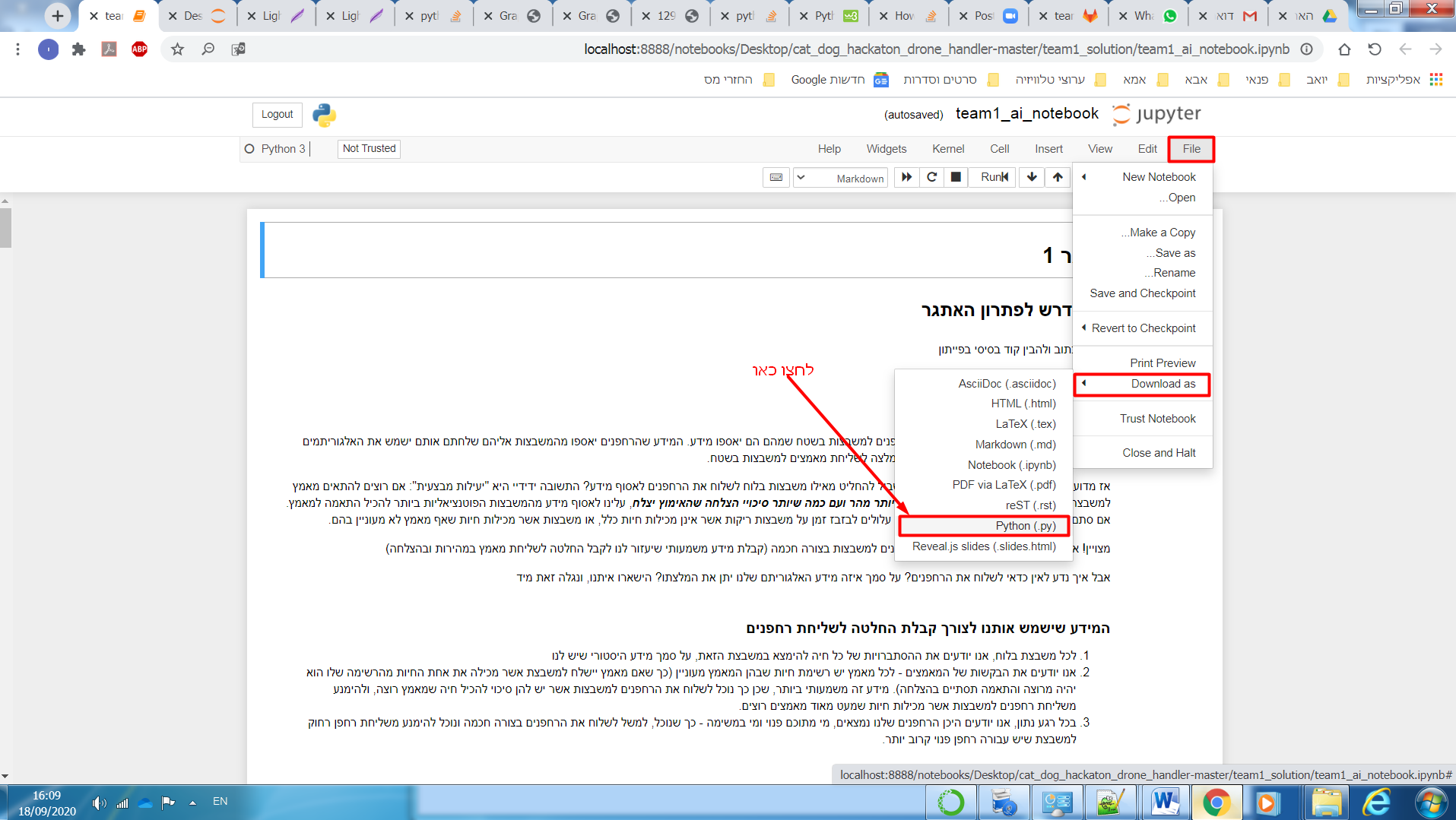


ופשוט היכנסו אליה על ידי לחיצה על שמה – ייפתח מסך שנראה פחות או יותר כך:

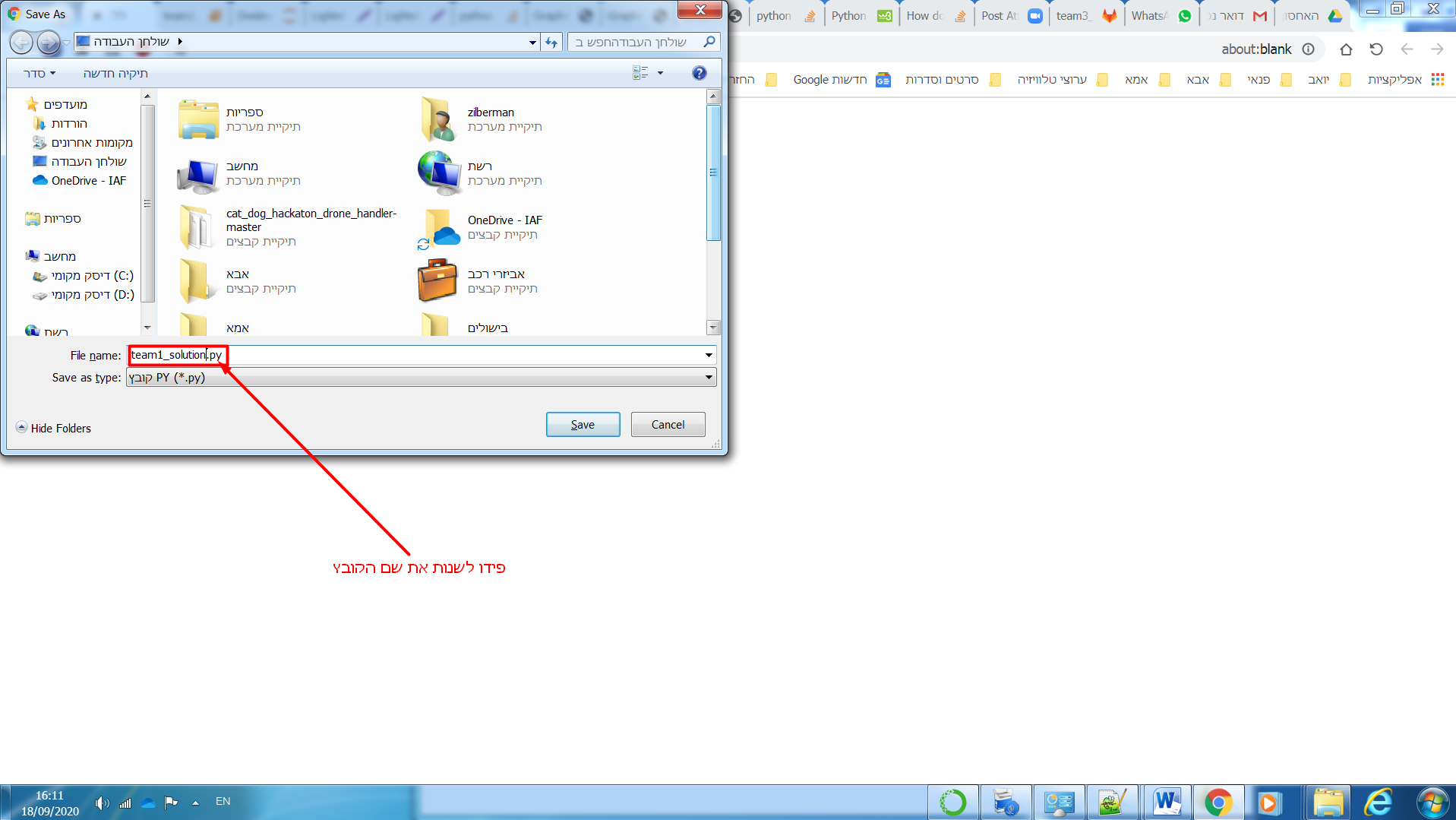


זהו, זאת המחברת. פשוט קראו את תוכנה וממשו את הקוד הרלוונטי בתוכה. בהצלחה!

# נספח 2 – כיצד לשמור מחברת JUPYTER כקובץ פייתון



המסך הבא שתגיעו אליו ייראה כך:



הקפידו לשנות את שם הקובץ ל-team1\_solution.py (במידה ואתן קבוצה מספר 1, אם אתן קבוצה מספר 2 – שנו את שם הקובץ ל-, team2\_solution.py ואם אתן קבוצה מספר 3 – שנו את שם הקובץ ל- team3\_solution.py).

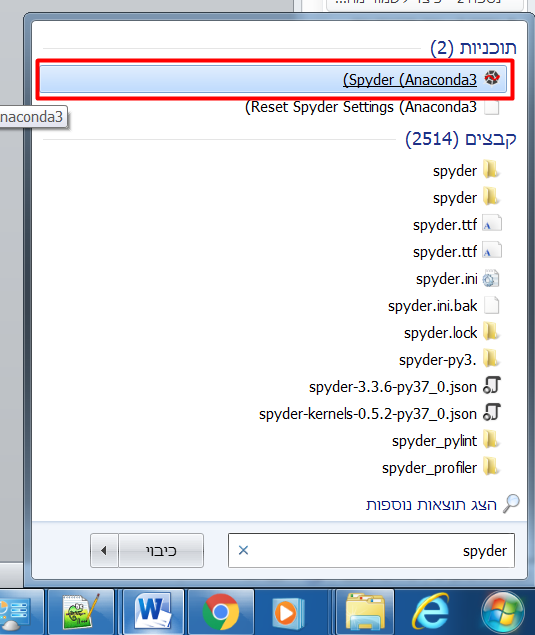
# נספח 3 – איך מריצים את שרת הפייתון המקומי

מדריך זה יתייחס לכיצד מרימים את השרת מקומית באמצעות SYPDER- IDE שהותקן לכן יחד עם ANACODA.

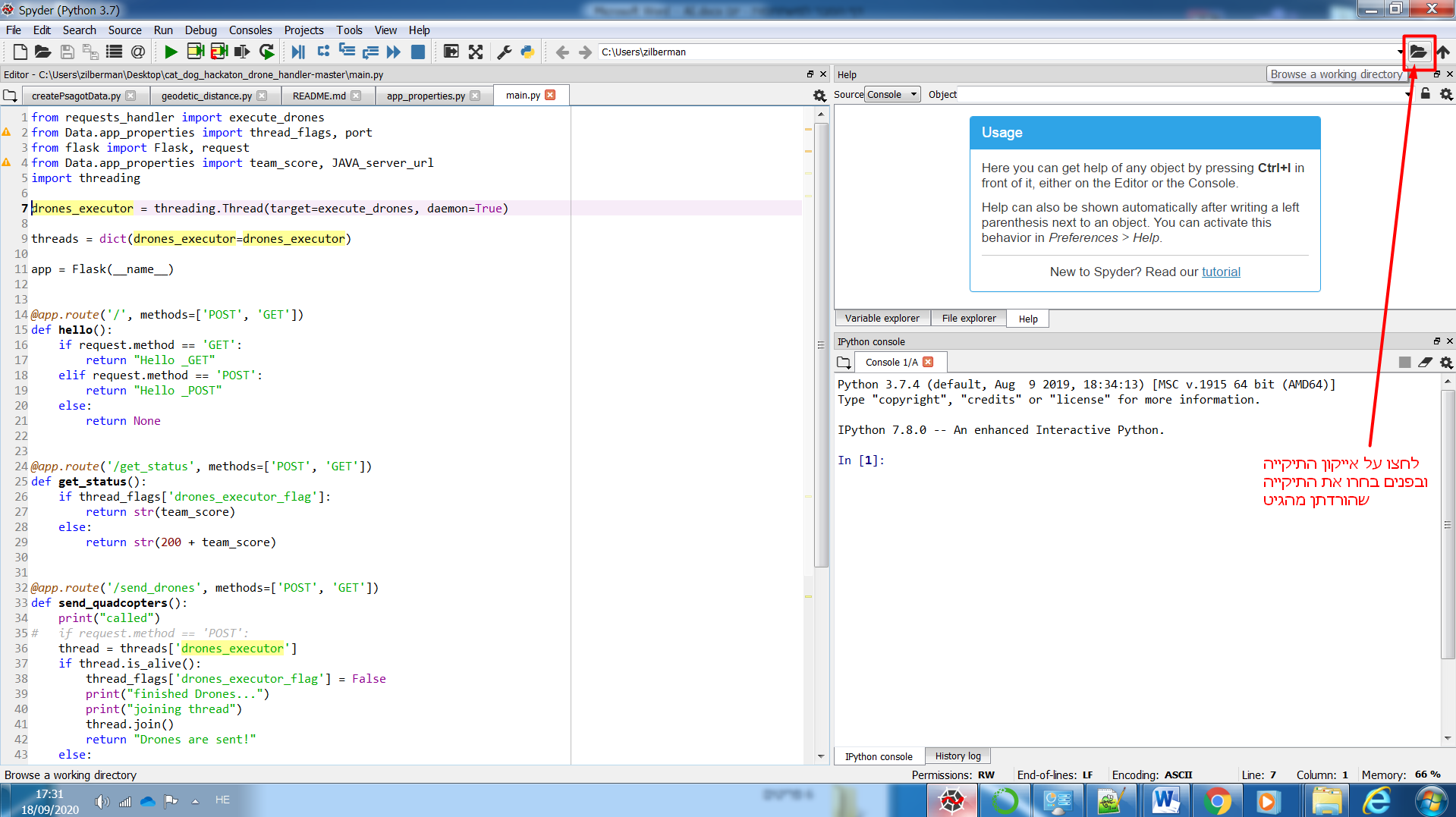
לפני שמריצים את השרת בפעם הראשונה, צריך להתקין אותו. לאחר מכן, הרצתו כל פעם מחדש היא פשוטה מאוד. נפריד ל-2 המקרים:

## הרצה ראשונה של השרת – איך מתקינים את השרת?

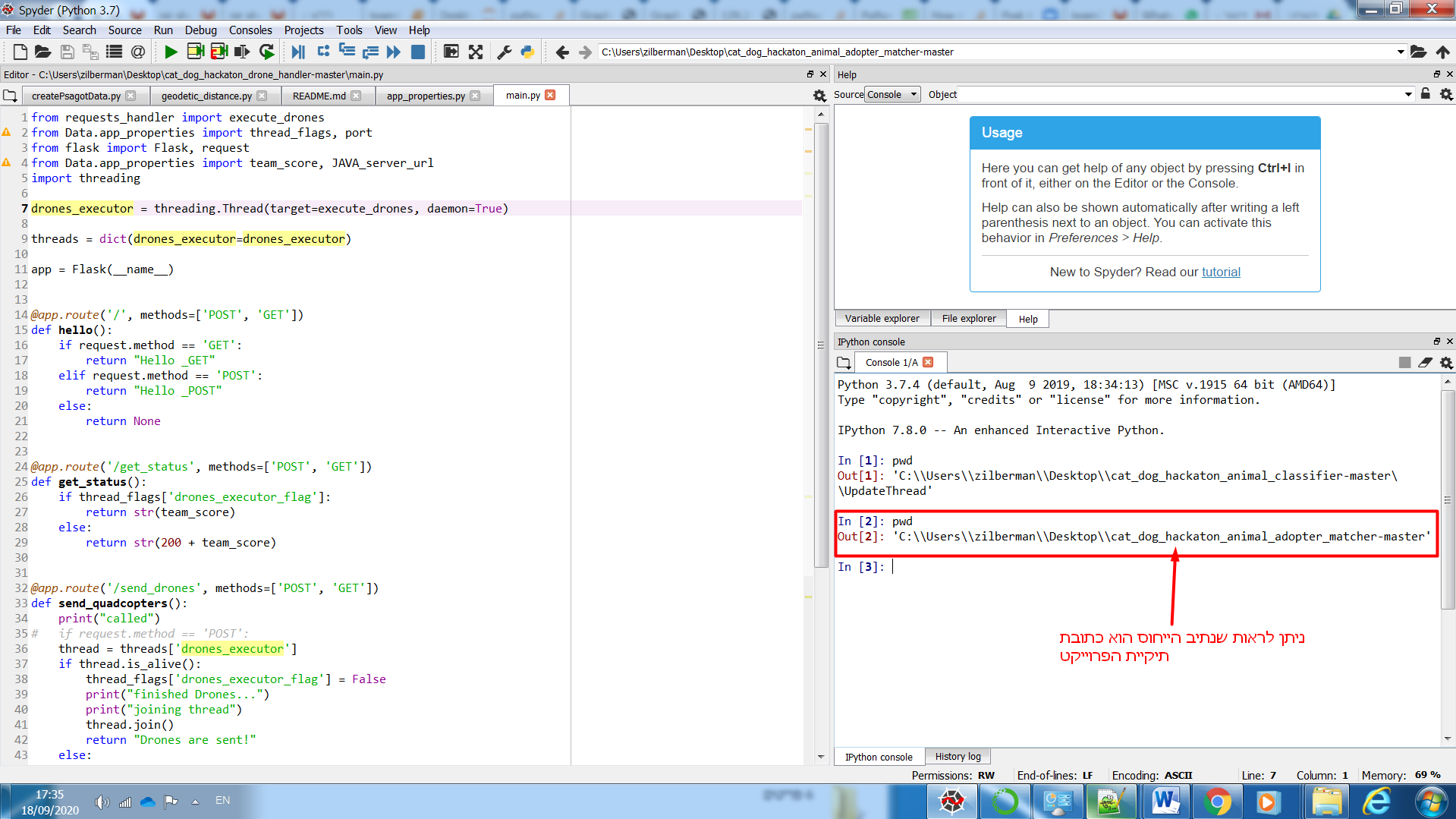
ראשית היכנסו ל-SPYDER:



כעת, לאחר שה-SPYDER נפתח, שנו את תיקיית הייחוס להיות התיקייה של הפרוייקט שהורדתן:

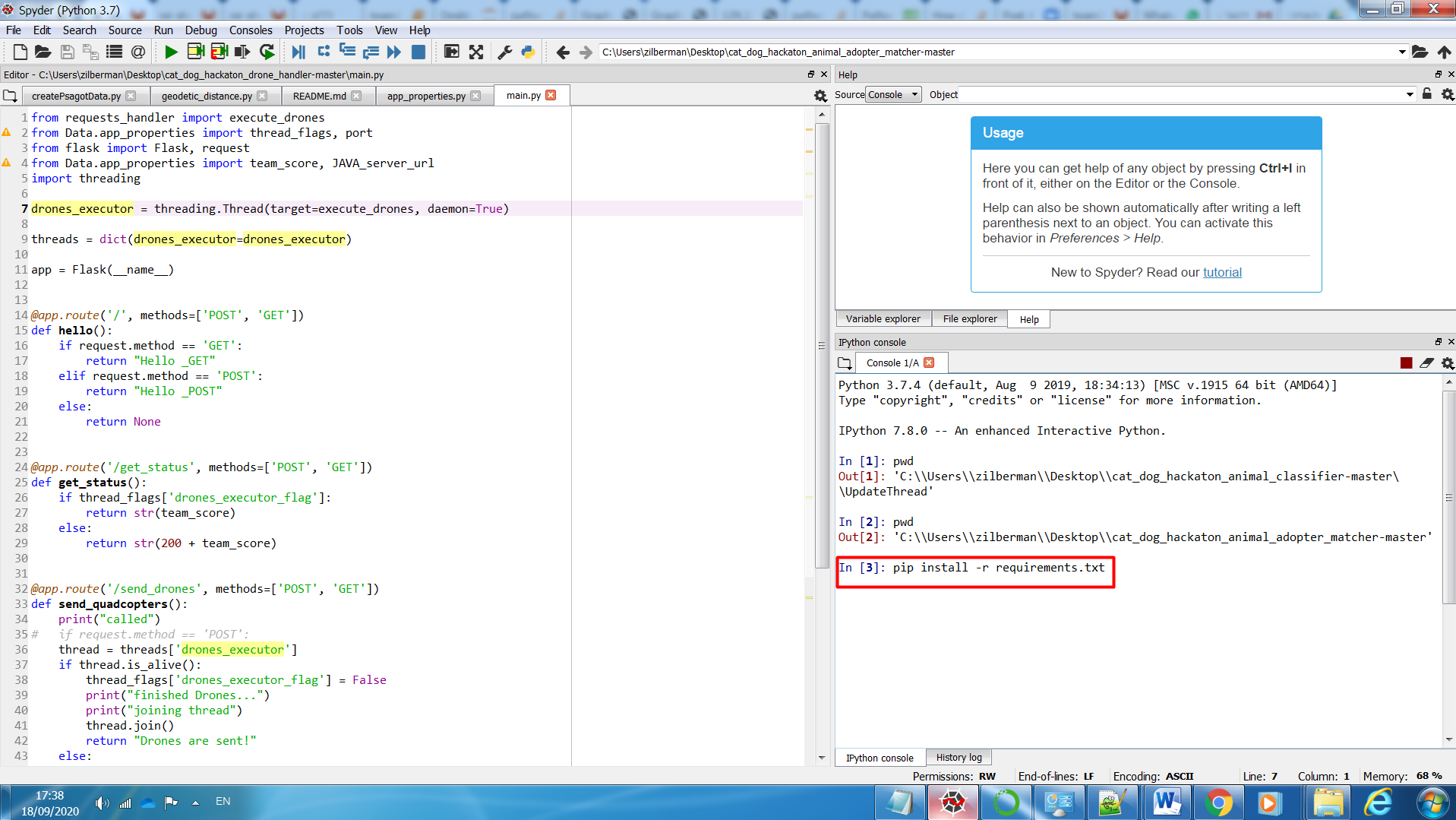


כדי לוודא שתיקיית הייחוס עודכנה להיות תיקיית הפרוייקט – ניתן לרשום pwd ב-ipython console ולוודא שהכתובת המוחזרת היא אכן הכתובת של תיקיית הפרוייקט שהורדתן מהגיט:

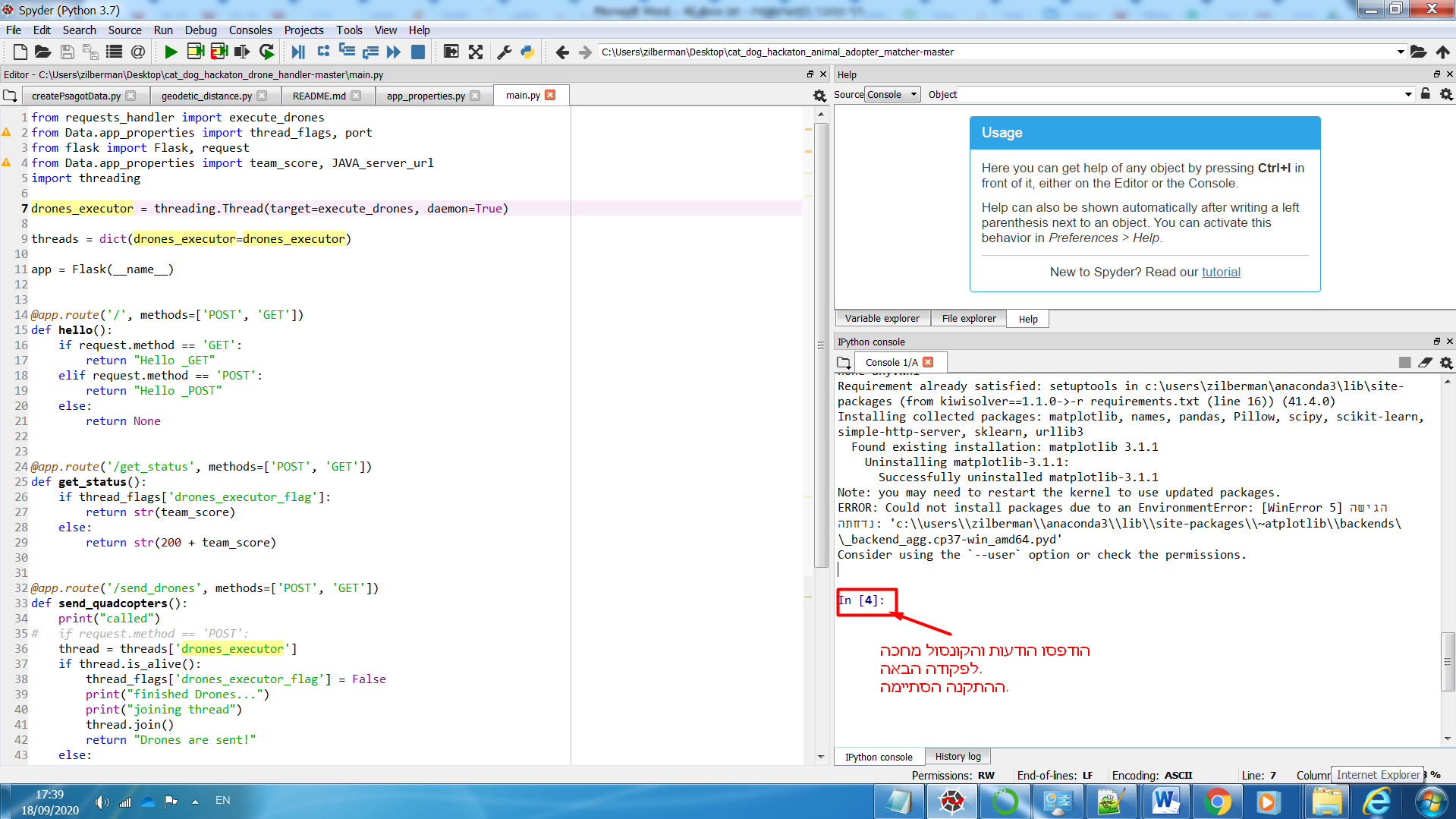


כעת הקלידו ב-ipython console את הפקודה:

**pip install -r requirements.txt**



המתינו מספר דקות (2-3 דקות, אולי פחות). יתחילו להיות מודפסות הודעות ה-ipython console. אתן תדעו שההתקנה הסתיימה כאשר ה-ipython console יחכה לפקודה הבאה מכן:



מצויין. אז התקנו את השרת.

עוד דבר קטנטן ואנחנו מוכנים להרצת השרת:

1. היכנסו לתיקייה DATA שנמצאת בתוך תיקיית הפרוייקט שלכן
2. פיתחו את הקובץ app\_properties.py (ניתן לעשות זאת באמצעות spyder שכבר פתוח אצלכן במחשב על ידי לחיצה על file ואז על open ואז לבחור את הקובץ מהמיקום המתאים שלו)
3. הקובץ שפתחתן מכיל קבוע בשם JAVA\_SERVER\_PORT. כרגע הוא מכיל את הערך 9000. שנו אותו לקבוע המתאים לפרויקט/קבוצה/מוצר בו אתן נמצאות, על פי הטבלה הבאה:

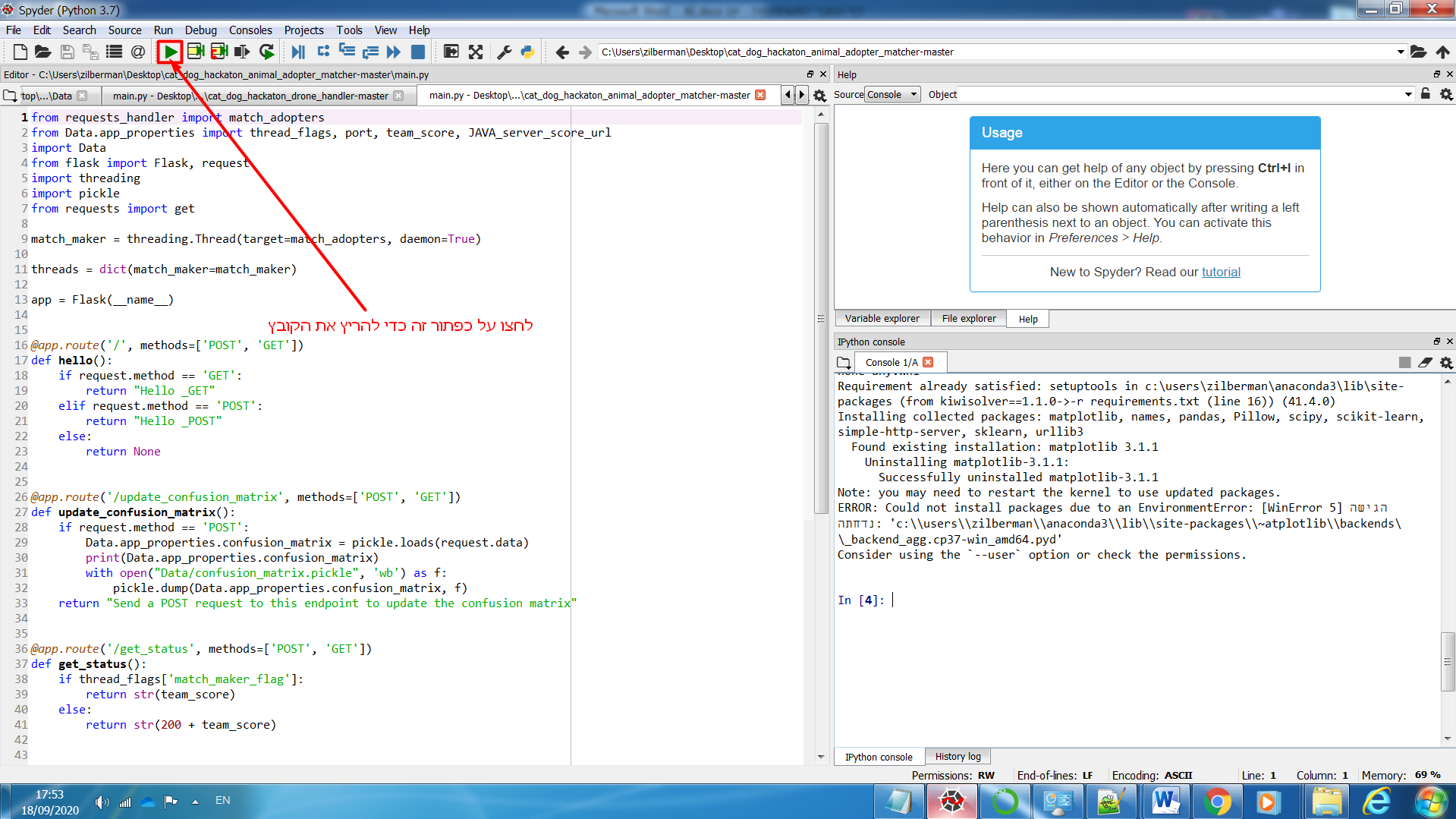
|  |  |
| --- | --- |
| קבוצה/מוצר | JAVA\_SERVER\_PORT |
| 1 | 9001 |
| 2 | 9002 |
| 3 | 9003 |
| 4 | 9004 |
| 5 | 9005 |
| 6 | 9006 |
| 7 | 9007 |
| 8 | 9008 |
| 9 | 9009 |

1. שמרו.

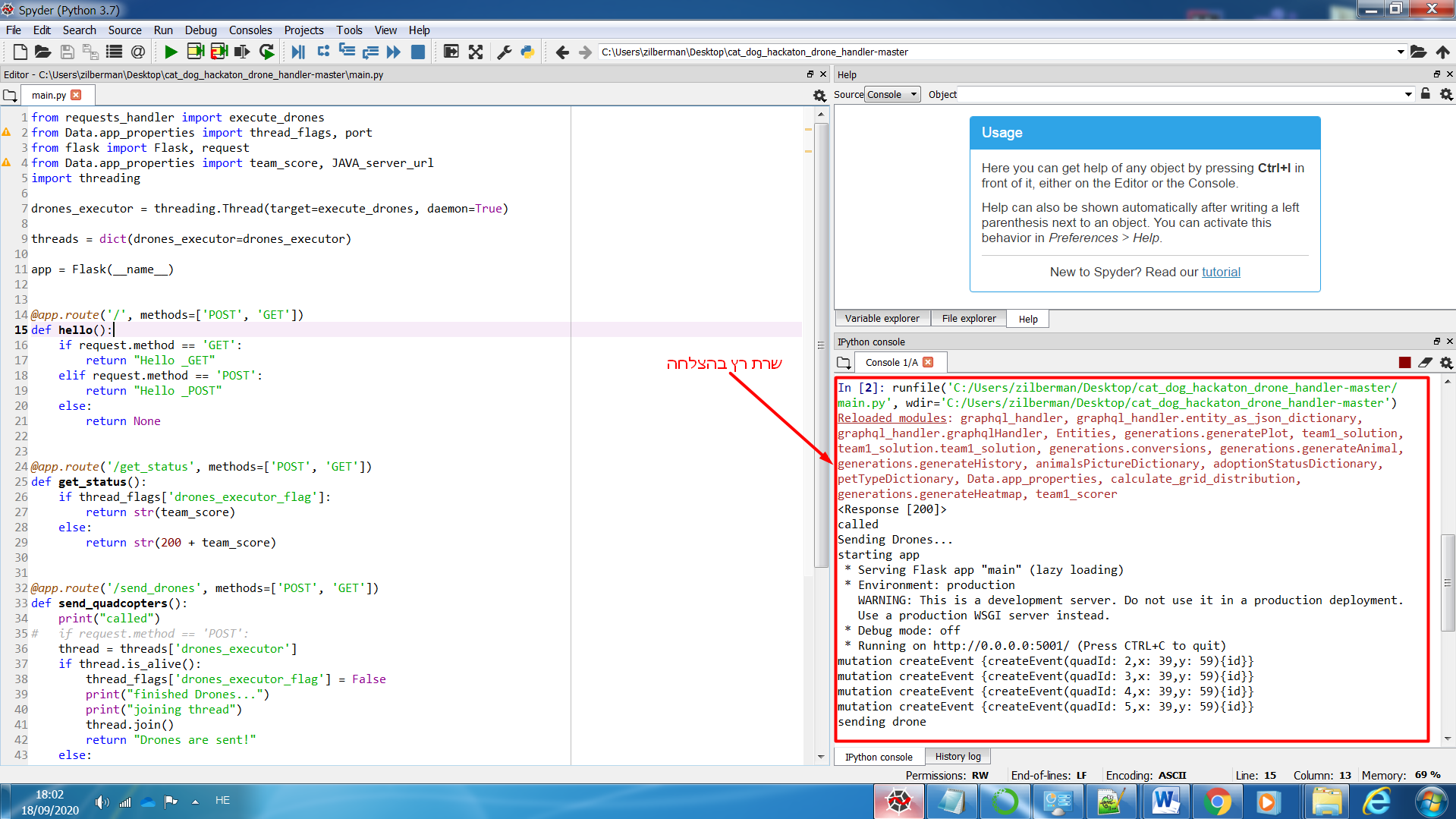
זהו, סיימנו. זאת הייתה פעולה חד פעמית להתקנת השרת. מעתה ואילך בכל פעם שתרצו להריץ את השרת תצטרכו רק לעקוב אחרי הסעיף הקצרצר הבא:

## הרצה של השרת לאחר התקנתו

1. פיתחו את קובץ ה-main של הפרוייקט באמצעות spyder והריצו אותו (זה הכל!)



1. לאחר ההרצה, אתן תראו שה-console מתחיל להדפיס הודעות, זה סימן שהכל עובד. חוץ מזה, תוכלו לראות את מסך ה-scores מתעדכן עם התוצאות של האלגוריתם החדש שלכן וכמובן לראות את האפליקציה שלכן מתנהגת בצורה חדשה על פי הלוגיקה שמימשתן, במסך ה-ui המתאים.



**הערה**: בכל פעם שתעדכנו פתרון שלכן (עדכון במחברת –> שמירה כקובץ פייתון -> החלפתו בתיקיית הפרוייקט), עליכן להפסיק את פעילות השרת (אפשרי על ידי לחצן הריבוע האדום מעל ה-ipython console ב-spyder) ולהפעילו מחדש (השרת משתמש בקובץ python ששמתן לו. לכן אתן צריכות להפסיק את פעילותו ולהריצו מחדש, כדי שירוץ עם הגרסה החדשה של הפתרון שלכן).