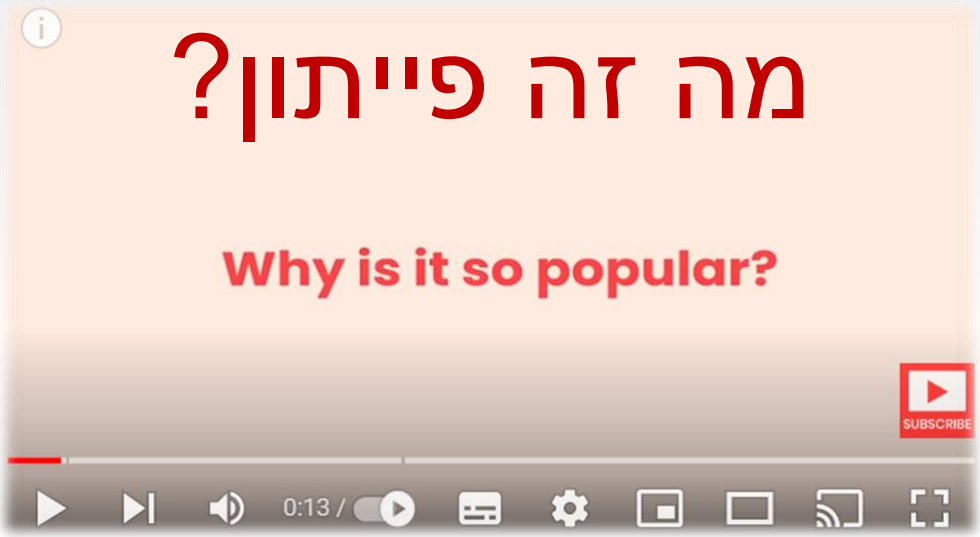
מבוא ל-PYTHON

- הכרות עם שפת התכנות פייתון (PYTHON)
- נכיר את סביבת העבודה של PYTHON
- תחביר השפה:
  - ✓ משתנים
  - ✓ סוגי משתנים
  - ✓ השמה
  - ✓ הדפסה
  - ✓ אופרטורים
  - ✓ תיעוד

נצפה בסרטון



https://www.youtube.com/watch?v=Y8Tko2YC5hA

הכרות עם שפת התכנות פייתון (python)

- פייתון היא שפת תכנות פשוטה המשמשת מתכנתים בהרבה מאוד תחומים. שפת תכנות היא אוסף של חוקים שונים, כגון חוקים תחביריים אשר באמצעותם ניתן להגדיר למחשב באופן מפורט פקודות ופעולות שעליו לבצע.
- פייתון היא דוגמא ל**שפה עילית** כמו שפות אחרות שאולי שמעתם עליהן כגון C, C++, Perl ו- Java. לעומתן ישנן **שפות סף** שנקראות גם "שפת מכונה" או "שפת אסמבלי".
- מחשב הוא מכונה המבצעת פקודות הכתובות בשפה המובנת לו, שפת המכונה. לכן, על מנת להריץ תכנית מסוימת, הכתובה בשפה עילית כמו פייתון שאינה שפת המכונה של המחשב, ראשית יש לתרגמה לשפה זו.

הפילוסופיה מאחורי השפה:

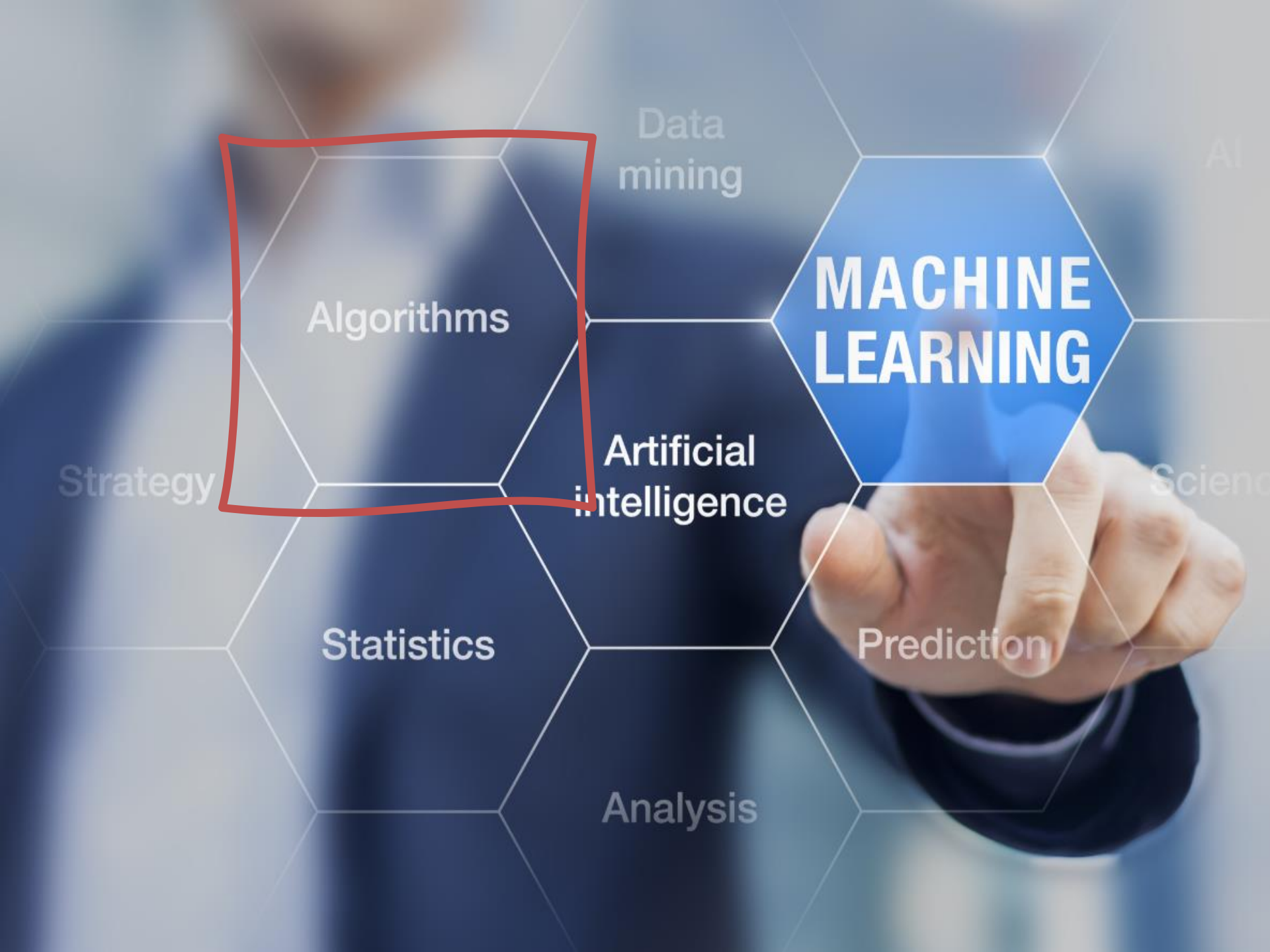
- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

אתרים מומלצים ללמידת Python

הכל נמצא ברשת, צריך רק לדעת איפה לחפש:

אתרים ללמוד עצמי

- The Algorithm Design Manual by Steven S Skiena (highly recommended)
- Python Documentation
- Think Python: How to Think Like a Computer Scientist by Allen B. Downey
- Python Programming WikiBook
- The Python Tutorial official tutorial
- Programiz


Data
mining
Algorithms
MACHINE
LEARNING
Strategy
Artificial
intelligence
Statistics
Prediction
Analysis

משתנים(Variables)

- סוגי משתנים
- הדפסת משתנים
- השמה
- קליטת ערכים מהמשתמש

משתנים (Variables)

- משתנה יכול להיות מסוגים שונים:

- 'str' זה קיצור של string, **מחרוזת של תווים**. בד"כ מחרוזת בתוך מרכאות בודדות או כפולות (' ', " ").
- 'int' זה קיצור של integer, שזהו **מספר שלם**.
- 'float' (floating point) זהו מספר לא שלם, **עשרוני (נקודה צַפָּה).**
- 'bool' זה קיצור של Boolean. **True** ו- **False** הם ערכים בעלי ערך בוליאני (לוגי), אותם נכיר בהמשך

- בהמשך נכיר מערכים של מספרים (רבים מהמשתנים בהם נשתמש יהיו מערכים של מספרים: וקטורים, מטריצות).

- ע"מ ליצור משתנה שנקרא a ושערכו 1, כיתבו פשוט a=1

משתנים (Variables)

שמות המשתנים:

- ✓ case sensitive (a≠A).
- ✓חייבים להתחיל באות או בסימן _. אח"כ כל קומבינציה של אותיות, מספרים או _ היא קבילה.
- ✓שמות משמעותיים
- ✓הערות לתיעוד: תפקיד המשתנה בקוד

משתנים (Variables)

השמה: **<u>פעולת ההשמה</u>** יוצרת משתנים חדשים ומציבה בתוכם ערכים, זאת ע"י שימוש בסימן 'שווה' =

ניתן להציב ערך ולשנות את הערך

X = "Al"

ניתן לקלוט ערך מהשתמש

X = input("מה ראשי התיבות של בינה מלאכותית?")

X = int(input("בן כמה אתה?"))

להפוך את הערך למספר מסוג int

הדפסה(print)

- הדפסת מחרוזת
- הדפסת ערך השמור במשתנה

הדפסה ()print - מחרוזת

כדי להדפיס טקסט על המסך נשתמש בפייתון בפקודת ה- print

print (“Hello, World!")

תוצאת פעולת ה- print היא הצגה על המסך של הטקסט:

Hello, World!

הטקסט המודפס חייב להיות בסוגריים.

בנוסף, הטקסט המודפס נמצא בתוך מרכאות, אך הן אינן מוצגות בתוצאה הסופית.

ניתן להשתמש במרכאות בודדות (' ') או בכפולות (" ").

הדפסה print() – מלל המורכב ממחרוזת ומהערך במשתנה

```
my_name = "merav"
my_age = 4 # not a lie
my_height = 160 # centimeter
my_eyes = "Brown"
print ("Let's talk about %s." % my_name)
print ("She's %d centimeters tall." % my_height)
print ("She's got %s eyes and has %d childrens." % (my_eyes, my_age))
```

```
Let's talk about merav.
She's 160 centimeters tall.
She's got Brown eyes and has age 4 childrens.
```

ניתן להדפיס מספר משתנים בפקודה אחת, כפי שמופיע בדוגמה

Format Character	Variable Type
%d	Integer
%s	String
%f	Float
%r	Integer, String, Float

בתוך גרשיים - יש לכתוב מחרוזת ולשלב את סימן הפורמט % ולאחריו סוג הפורמט בהתאם לסוג המשתנה

בסיום הגרשיים יש להוסיף % ואת המשתנה שיבוא במקום

הדפסה ()print – מלל המורכב ממחרוזת ומהערך במשתנה

```
pi=3.1415926535
print ("The value of Pi is: %.2f"  %  pi) #2 digits after dot
print ("The value of Pi is: %.4f"  %  pi) #4 digits after dot
```

```
The value of Pi is: 3.14
The value of Pi is: 3.1416
```

שימו לב ❤️ כאשר נכתוב

%.xf

משמעות הדבר הדפסת x ספרות לאחר הנקודה העשרונית

אופרטורים

- סוגי אופרטורים

אופרטורים - פעולות חשבון פשוטות

ניתן להשתמש בחיבור, חיסור, כפל וחילוק כפי שרגילים.

```
> -1
-1
> 4 + 2
6
> 4 - 1
3
> 4 * 3
12
> (50 - 5 * 6) / 4
5.0
> 2 ** 10
1024
>
```

חוקי סדר הפעולות: כפל וחילוק קודמים לחיבור וחיסור.

יש להשתמש בסוגריים בשעת הצורך.

פעולת החזקה: n^k

k הוא המעריך ו- n הוא הבסיס.

את הפעולה בפייתון נכתב כ- n ** k

אופרטורים - פעולת החילוק

ישנם כמה אופרטורים שמשתמשים בהם בפעולות חילוק:

1. חילוק רגיל:

סימון החלוקה "/" מספק תשובה שהיא תוצאה של חלוקת שני מספרים, והתוצאה תוצג כמספר עשרוני משתנה מסוג float

אופרטורים - פעולת החילוק

2. חילוק לקבלת המספר השלם ללא השארית:

לצורך כך נשתמש בסימון החלוקה "//".

התשובה תהיה מספר שלם ללא השארית, תוצאה של חלוקת מספרים floor division (המחשב מעגל את תוצר החילוק כלפי מטה.)

לדוגמא: תוצאת החילוק של 3/2 תהיה 1 ולא 1.5.

אופרטורים - פעולת החילוק

3. חילוק לקבלת השארית בלבד:

לצורך כך נשתמש בסימן החלוקה "%".

התשובה היא השארית ללא המספר השלם, תוצאה של חלוקת מספרים modulo

לדוגמא: תוצאת שארית החילוק של 12.4%3 תהיה 0.4.

אופרטורים - פעולת החילוק

```
> 20/3
6.666666666666667
>
>
> 20 / 3
6.666666666666667
>
> 20 // 3
6
>
> 20 % 3
2
>
```

1. חילוק רגיל

2. חילוק לקבלת המספק השלם

2. חילוק לקבלת השארית

כמה שיותר הערות

שורה

הערות בתכנית (שורות המתחילות ב "#")

הערות הן מאוד חשובות בתכניות מחשב. משתמשים בהם על מנת לתאר בשפת דיבור פשוטה (אנגלית) מה מטרת הפעולות בתכנית.

בנוסף על כך ניתן להשתמש בהערות על מנת למנוע באופן זמני מחלקים של התכנית להתבצע. ברוב סביבות העבודה שורה שמסומנת כהערה תופיע בצבע אחר.

!!!!!

פיסקה

!!!!!

תרגול

- צרו את המשתנים הבאים:
  - משתנה בשם FirstVar שערכו 65- כפול 47 חלקי 10000
  - משתנה בשם SecondVar שערכו קוסינוס של הערך המוחלט של משתנה FirstVar.
  - משתנה בשם ThirdVar שערכו המספר השלם הקרוב ביותר של SecondVar בחזקת FirstVar.
- הדפיסו על המסך את התשובות בצורה ברורה, לדוגמא:

The value of FirstVar is: -0.3055