### **Or = ||**

And = &&

for(let i=0;i<5;i++){

console.log(i)

}

if(x > 5){

}

else if(x<5){

}

else{

}

### **דף עבודה – תרגול JavaScript**

1. **הדפס את שמך ללוג**
2. **הצג את שמך באמצעות alert**
3. **קלוט באמצעות prompt את שמך הפרטי ושמור במשתנה. הצג את המשתנה על ידי alert**
4. **קלוט באמצעות prompt את שמך הפרטי ושמור במש+תנה ולקלוט את שם משפחתך ושמור במשתנה . הצג את שמך המלא באמצעות alert**
5. **כתוב תוכנית המדפיסה את המספרים מ-1 עד 10 באמצעות לולאת for.  
   כתוב תוכנית הקולטת שני מספרים ומדפיסה את כל המספרים בין המספר הראשון לשני (כולל).**
6. **כתוב תוכנית הקולטת מספר ומדפיסה האם הוא זוגי או אי זוגי.**
7. **כתוב תכנית הקולטת שני מספרים גדול וקטן ומדפיסה את המספרים מהגדול לקטן**
8. **כתוב תוכנית המדפיסה את כל המספרים המתחלקים ב 3 מ 10 עד 50**
9. **כתוב תוכנית שיוצרת מערך של 5 שמות ומדפיסה כל שם בלולאה**
10. **כתוב תוכנית המגדירה מערך של מספרים ומדפיסה את הסכום של כל המספרים**
11. **כתוב תוכנית המגדירה מערך של מספרים ומדפיסה את ממוצע של כל המספרים**
12. **כתוב תוכנית הקולטת מחרוזת מהמשתמש ומדפיסה את האורך שלה.**

**שיעורי בית**

1. **כתוב תוכנית שקולטת מספר ומדפיסה את הריבוע שלו.**
2. **כתוב תוכנית שקולטת שם ומדפיסה את אותו שם 5 פעמים בלולאת for.**
3. **כתוב תוכנית שקולטת גיל ומדפיסה אם מותר לו להיכנס לקולנוע (18 ומעלה).**
4. **כתוב תוכנית שמדפיסה את האותיות A עד Z (קוד ASCII) באמצעות לולאה.**
5. **כתוב תוכנית הקולטת 5 מספרים מהמשתמש ושומרת אותם במערך. לאחר מכן הדפס את המערך.**
6. **כתוב תוכנית שמדפיסה את המספרים מ־1 עד 30, אבל אם המספר מתחלק ב־7 – הדפס במקום זה "בום".**
7. **כתוב תוכנית שמייצרת מערך של 10 מספרים רנדומליים בין 1 ל-100 ומדפיסה את כולם.**
8. **כתוב תוכנית שקולטת מחרוזת ומדפיסה כל תו (אות) בשורה נפרדת.**
9. **כתוב תוכנית שקולטת מספר ובודקת אם הוא מתחלק גם ב־2 וגם ב־3.**
10. **כתוב תוכנית שמדפיסה את כל המספרים מ־10 עד 1 בלולאת while.**
11. **כתוב תוכנית שמבקשת מהמשתמש שם, וממשיכה לבקש שוב עד שייכנס "admin".**
12. **כתוב תוכנית שמבקשת מהמשתמש מספר עד שיכניס מספר זוגי.**
13. **כתוב תוכנית שמדפיסה את סכום כל המספרים מ־1 עד 100.**
14. **כתוב תוכנית שמדפיסה את כל המספרים בין 1 ל־50 שמכילים את הספרה 3 (למשל: 3, 13, 23, 30, 31...).**
15. **כתוב תוכנית שקולטת מספר ומדפיסה את כמות הספרות שבו.**
16. **כתוב תוכנית שמדפיסה את כל המספרים מ־1 עד 100 ומוסיפה כוכבית (\*) ליד כל מספר אי זוגי.**
17. **כתוב תוכנית שמדפיסה את המספרים בין 20 ל־40, אבל מדלגת על כל מספר שמתחלק ב־4.**
18. **כתוב תוכנית שמבצעת ספירה עולה מ־0 עד מספר שהמשתמש הכניס (עם prompt).**
19. **כתוב תוכנית שמקבלת שם ומדפיסה אותו הפוך (ללא פונקציות מובנות).**
20. **כתוב תוכנית שמבצעת ספירה לאחור ממספר שהמשתמש הכניס עד 0.**

**שיעורי בית**

1. **כתוב פונקציה המקבלת כפרמטר מספר ומדפיסה את הריבוע שלו.**
   1. **כתוב תכנית מזמנת את הפונקציה עם המספר 19 ומדפיסה את התוצאה**
   2. **כתוב תכנית הקולטת על ידי prompt מהמשתמש מספר ומדפיסה את המספר**
   3. **כתוב תכנית הקולטת על ידי input type מספר ומדפיסה את המספר**

1. **כתוב פונקציה המקבלת שם ומדפיסה את אותו שם 5 פעמים בלולאת for.**
   1. **כתוב תכנית מזמנת את הפונקציה עם השם david**
   2. **כתוב תוכנית הקולטת שם ידי prompt מהמשתמש ומזמנת את הפוקנציה**

1. **כתוב פונקציה המקבלת כפרמטר גיל ומדפיסה אם מותר לו להיכנס לקולנוע (18 ומעלה).**
   1. **כתוב תכנית מזמנת את הפונקציה עם הגיל 10. מה הודפס ללוג**
   2. **כתוב תוכנית הקולטת שם ידי prompt את גיל המשתמש ומזמנת את הפונקציה**

1. **כתוב פונקציה המקבלת כפרמטר מערך של מספרים מספרים ומחזירה את הממוצע .**
   1. **כתוב תכנית המצהירה על מערך numbers = [1,3,9,1 התכנית תפעיל את הפונקציה והדפס את ממוצע מערך**
   2. **כתוב תכנית הקולטת על ידי prompt חמישה מספרים בלולאה. התכנית תוסיף את המספרים למערך בשם numbers . התוכנית תפעיל את הפונקציה עם המערך והדפס את הממוצע שלהם**
2. **כתוב פונקציה המדפיסה את המספרים מ־1 עד 30, אבל אם המספר מתחלק ב־7 – הדפס במקום זה "בום".**
   1. **הפעל את הפונקציה**
3. **כתוב פונקציה שמייצרת מערך של 10 מספרים רנדומליים בין 1 ל-100 ומדפיסה את כולם.**
4. **כתוב פונקציה המקבלת מחרוזת ומדפיסה כל תו (אות) בשורה נפרדת. printLetters**
   1. **קלוט את שמך מהמשתמש על ידי prompt והפעל את הפונקציה על שמך**
5. **כתוב פונקציה המקבלת כפרמטר מספר ומחזירה אמת אם הוא מתחלק גם ב־2 וגם ב־3. אחרת מחזירה שקר isDevide2And3** 
   1. **הפעל את הפונקציה עם המספר 3**
   2. **הפעל את הפונקציה עם המספרים 1-100**
6. **כתוב פונקציה המדפיסה את כל המספרים מ־10 עד 1 בלולאת while.**
   1. **הפעל את הפונקציה**
7. **כתוב פונקציה שמבקשת מהמשתמש שם על ידי prompt , וממשיכה לבקש שוב עד שייכנס "admin".**
   1. **הפעל את הפונקציה**
8. **כתוב פונקציה שמבקשת מהמשתמש מספר עד שיכניס מספר זוגי. הפונקציה תדפיס את המספר הזוגי בלבד** 
   1. **הפעל את הפונקציה**

1. **כתוב פונקציה שמדפיסה את סכום כל המספרים מ־1 עד 100.**
   1. **הפעל את הפונקציה**
2. **כתוב פונקציה שמדפיסה את כל המספרים בין 1 ל־50 שמכילים את הספרה 3 (למשל: 3, 13, 23, 30, 31...).**
   1. **הפעל את הפונקציה**
3. **כתוב פונקציה המקבלת כפרמטר מספר ומדפיסה את כמות הספרות שבו.**
4. **כתוב פונקציה שמדפיסה את המספרים בין 20 ל־40, אבל מדלגת על כל מספר שמתחלק ב־4.**
   1. **הפעל את הפונקציה**
5. **כתוב פונקתיה שמקבלת שם ומדפיסה אותו הפוך (ללא פונקציות מובנות).**
   1. **קלוט מהמשתמש שם והפעל את הפונקציה**

basic js

condition

קילטו = prompt

הציגו = alert או console.log לבחירתכם

1. קילטו שני מספרים והציגו את הקטן מביניהם. אם שניהם שווים יש להציג אחד מהם בלבד.
2. קילטו ציון והציגו "Pass" אם הציון גדול או שווה ל-60, אחרת הציגו "Fail".
3. קילטו גיל, הציגו הודעה המציינת האם ניתן להוציא רישיון נהיגה או לא. אם לא – יש להציג את מספר השנים שיש להמתין.
4. קילטו מספר, הציגו "חיובי", "שלילי" או "אפס" כתלות במספר.
5. קילטו מספר, הציגו:
6. "המספר הינו מספר שלילי"
7. "המספר הינו אפס"
8. "המספר הינו בין 1 ל-100 כולל"
9. "המספר הינו בין 101 ל-1000 כולל"
10. "המספר גדול מ-1000"
11. קילטו ציון והציגו:
12. "נכשל" אם הציון בין 0 ל-59 כולל.
13. "מספיק" אם הציון בין 60 ל-69 כולל.
14. "כמעט טוב" אם הציון בין 70 ל-79 כולל.
15. "טוב" אם הציון בין 80 ל-89 כולל.
16. "טוב מאוד" אם הציון בין 90 ל-99 כולל.
17. "מעולה" אם הציון 100.

קילטו שלושה מספרים, הציגו את המקסימלי מביניהם. אם יש יותר מאחד כזה – הציגו אותו פעם אחת.

קילטו = prompt

הציגו = alert או console.log לבחירתכם

js- Conditions, loop

* **עבור כל טווח מספרים שעליכם להציג – יש להציג את המספרים כולל קצוות. לדוגמה, אם התרגיל מציין להציג מ-1 ועד 1000, הכוונה היא כולל 1 וכולל 1000.**

1. הציגו את כל המספרים מ-1 ועד 1000.
2. הציגו את כל הזוגיים מ-1 ועד 1000.
3. הציגו את כל האי-זוגיים מ-1000 עד 2000.
4. הציגו את כל המספרים מ-1000 עד 1 (סדר יורד).
5. הציגו את כל הזוגיים מ-1000 ועד 1 (סדר יורד).
6. הציגו את כל האי-זוגיים מ-1000 ועד 1 (סדר יורד).
7. קילטו מהמשתמש מספר n, הציגו את כל המספרים המתחלקים ב-3 ללא שארית, מ-1 ועד n.
8. קילטו מהמשתמש מספר n, הציגו את כל המספרים מ-1 ועד n ואז שוב את כל המספרים מ-n ועד 1.
9. קילטו מהמשתמש שני מספרים, הציגו את כל המספרים ביניהם בסדר עולה. אפשר להניח שהראשון קטן מהשני.
10. קילטו מהמשתמש שני מספרים, הציגו את כל המספרים ביניהם בסדר עולה. אין להניח שהראשון קטן מהשני.
11. קילטו מהמשתמש שני מספרים, הציגו את כל המספרים ביניהם בסדר יורד. אפשר להניח שהראשון קטן מהשני.
12. קילטו מהמשתמש שני מספרים, הציגו את כל המספרים ביניהם בסדר יורד. אין להניח שהראשון קטן מהשני.
13. קילטו מהמשתמש שלושה מספרים: first, last ו-num. הציגו את כל המספרים בין first ל-last המתחלקים ב-num ללא שארית, בסדר עולה. אפשר להניח ש-first קטן מ-last.
14. קילטו מהמשתמש שלושה מספרים: first, last ו-num. הציגו את כל המספרים בין first ל-last המתחלקים ב-num ללא שארית, בסדר עולה. אין להניח ש-first קטן מ-last.
15. קילטו מהמשתמש מספרים שוב ושוב עד שמכניס 0.

עבור כל מספר שנקלט (כולל ה-0 שמסיים את הקלט) – הציגו האם הוא מתחלק ב-7 ללא שארית או לא.

1. קילטו מהמשתמש מספרים שוב ושוב עד שמכניס 0.

עבור כל מספר שנקלט (לא כולל ה-0 שמסיים את הקלט) – הציגו האם הוא מתחלק ב-7 ללא שארית או לא.

1. קילטו מהמשתמש מספרים שוב ושוב עד שמכניס מספר שלילי.

עבור כל מספר שנקלט (כולל השלילי שמסיים את הקלט) – הציגו את החזקה השלישית שלו (המספר בחזקת 3).

1. קילטו מהמשתמש מספרים שוב ושוב עד שמכניס מספר שלילי.

עבור כל מספר שנקלט (לא כולל השלילי שמסיים את הקלט) – הציגו את החזקה השלישית שלו (המספר בחזקת 3).

1. קילטו מהמשתמש 100 מספרים, והציגו את סכומם ואת הממוצע.
2. קילטו מהמשתמש 100 מספרים, הציגו את המספר הגדול ביותר שנקלט.
3. קילטו מהמשתמש מספרים שוב ושוב, עד שמכניס מספר שלילי. הציגו כמה מספרים זוגיים נקלטו.
4. אתגר: קילטו מהמשתמש מספר, הציגו את סכום הספרות שלו.

לדוגמה, אם המשתמש הכניס: 8415 יש להציג 18 כי זה סכום הספרות.

1. אתגר: **קילטו מהמשתמש מספרים שוב ושוב עד שמכניס 0 או מספר שלילי.**

עבור כל מספר שנקלט – **הציגו את כל המספרים מ-1 ועד אליו**.

1. אתגר: קילטו מהמשתמש מספר n, הציגו ריבוע מלא בכוכביות בגודל n על n.

לדוגמה, אם המשתמש הכניס 4, יש להציג:

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

בצעו זאת בשתי שיטות: א. ע"י אפשרות הכפלת מחרוזות. ב. ללא הכפלת מחרוזות

1. אתגר: קילטו מהמשתמש רוחב וגובה של מלבן.

הציגו מלבן מלא כוכביות ברוחב ובגובה שנקלטו.

לדוגמה, אם המשתמש הכניס רוחב 8 וגובה 5, יש להציג את המלבן הבא:

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

בצעו זאת בשתי שיטות: א. ע"י אפשרות הכפלת מחרוזות. ב. ללא הכפלת מחרוזות

js- For,while Break, Continue, Nested Loops

1. קילטו מהמשתמש שני מספרים, הציגו את הקטן מבניהם. עם ternary
2. קילטו מהמשתמש מספר, הציגו הודעה המציינת האם המספר מסתיים ב-5 או לא.ternary
3. קילטו מהמשתמש מספרים שלמים, עד שמכניס 0. עבור כל מספר שנקלט, הציגו האם המספר חיובי או שלילי.

בהכנסת 0, אין להציג דבר, אלא לסיים את הלולאה.

1. קילטו מהמשתמש מספרים שלמים, עד שמכניס מספר שלילי. עבור כל מספר שנקלט, הציגו את החזקה השלישית שלו (המספר כפול עצמו 3 פעמים). בהכנסת מספר שלילי – אין להציג דבר אלא יש לסיים את הלולאה.
2. קילטו מהמשתמש מספרים שלמים, עד שמכניס מספר המתחלק ב-7 ללא שארית. עבור כל מספר שנקלט, הציגו האם הוא חיובי, שלילי או 0. בהכנסת מספר המתחלק ב-7 ללא שארית, אין להציג דבר אלא יש לסיים את הלולאה.
3. קילטו מהמשתמש מספרים שלמים, עד שמכניס 0. עבור כל מספר שנקלט (כולל ה-0 המסיים את הקלט), הציגו האם המספר חיובי, שלילי או אפס.
4. קילטו מהמשתמש מספרים שלמים, עד שמכניס מספר שלילי. עבור כל מספר שנקלט (כולל המספר השלילי המסיים את הקלט), הציגו את החזקה השלישית שלו (המספר כפול עצמו 3 פעמים).
5. קילטו מהמשתמש מספרים שלמים, עד שמכניס מספר המתחלק ב-7 ללא שארית. עבור כל מספר שנקלט (כולל המספר המתחלק ב-7 ללא שארית המסיים את הקלט), הציגו האם הוא חיובי, שלילי או 0.
6. הציגו את כל המספרים בין 1 ל-300 כולל קצוות.
7. הציגו את כל המספרים בין (300-) ל-(300+) כולל קצוות.
8. קילטו מהמשתמש מספר שלם חיובי, הציגו את כל המספרים מ-1 ועד אליו כולל קצוות.
9. קילטו מהמשתמש מספר שלם חיובי, הציגו את כל המספרים ממנו ועד 1, כולל קצוות, בסדר יורד.
10. קילטו מהמשתמש מספר שלם חיובי, הציגו את כל המספרים מ-1 ועד אליו כולל קצוות, ואז את כל המספרים ממנו ועד 1 כולל קצוות, בסדר יורד.
11. קילטו מהמשתמש שני מספרים שלמים (יש להניח שהראשון קטן יותר), הציגו את כל המספרים ביניהם, כולל קצוות, מהקטן לגדול.
12. קילטו מהמשתמש שני מספרים שלמים (אין להניח שהראשון קטן יותר), הציגו את כל המספרים ביניהם, כולל קצוות, מהקטן לגדול.
13. קילטו מהמשתמש שני מספרים שלמים (אין להניח שהראשון קטן יותר), הציגו את כל המספרים ביניהם, המסתיימים ב-7, או המתחלקים ב-7, מהקטן לגדול כולל קצוות.

**עבור התרגילים המציגים כוכביות על המסך, נסו לפתור כל תרגיל בשתי שיטות:**

1. **שיטה ראשונה כאילו אנו בשפת תכנות דמיונית שלא מכילה את האופרטור \* הפועל על מחרוזות.**
2. **שיטה שנייה כאילו אנו בשפת Python, שכן מכילה את האופרטור \* הפועל על מחרוזות.**
3. קילטו מהמשתמש מספר n, הציגו ריבוע מלא בכוכביות בגודל n על n.

לדוגמה, אם המשתמש הכניס 4, יש להציג:

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

1. קילטו מהמשתמש רוחב וגובה של מלבן.

הציגו מלבן מלא בכוכביות ברוחב ובגובה שנקלטו.

לדוגמה, אם המשתמש הכניס רוחב 8 וגובה 5, יש להציג את המלבן הבא:

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

19. קילטו מהמשתמש מספר n, הציגו משולש כתלות ב-n לפי הפורמט הבא (דוגמה ש-n שווה 5):

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

1. אני רוצה להדפיס n שורות . לולאה זה רצה כל עוד n גדול מ - 0
   1. כל שורה אני רוצה להדפיס k כוכביות כאשר k =n
2. אני יוריד את n ב 1

N = 5

While n > 0 : # create n lines

K = n

While k > 0: # print k stars

print(“\*” , end=” ”)

k-=1

n-=1

20. קילטו מהמשתמש מספר n, הציגו משולש כתלות ב-n לפי הפורמט הבא (דוגמה ש-n שווה 5):

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

21. קילטו מהמשתמש מספר n, הציגו ריבוע כתלות ב-n לפי הפורמט הבא (דוגמה ש-n שווה 5):

5 4 3 2 1

5 4 3 2 1

5 4 3 2 1

5 4 3 2 1

5 4 3 2 1

22.קילטו מהמשתמש מספרים שוב ושוב עד שמכניס 0 או מספר שלילי. עבור כל מספר שנקלט (ללא ה-0 או המספר השלילי שמסיימים את הקלט) – הציגו את כל המספרים ממנו ועד 1 בסדר יורד.

23. מספר ראשוני הינו מספר המתחלק ללא שארית בעצמו וב-1 בלבד ואין שום מספר אחר המחלק אותו ללא שארית. לדוגמה, המספר 7 הינו מספר ראשוני כי הוא מתחלק ללא שארית רק בעצמו וב-1 ללא שארית. לדוגמה המספר 23 הינו מספר ראשוני כי הוא מתחלק רק בעצמו וב-1 ללא שארית. לדוגמה המספר 15 אינו מספר ראשוני כי הוא מתחלק גם ב-3 וגם ב-5 ללא שארית.

קילטו מהמשתמש מספר n, הציגו הודעה המציינת האם המספר הינו ראשוני או אינו מספר ראשוני.

24. קילטו מהמשתמש 10 מספרים. עבור כל מספר שנקלט, הציגו ריבוע כוכביות בגודל המספר. לדוגמה, משתמש מכניס מספר ראשון 4, יש להציג מיד ריבוע כוכביות בגודל 4X4. המספר הבא בתור שהוא מכניס הינו 7, יש להציג מיד ריבוע כוכביות בגודל 7X7 וכו'.

1. קלוט עשרה מספרים
   1. הצג ריבוע של מספרים

**for i in range(10): # אני רוצה 10 פעמים**

**n=int(input("Enter the number of rows: "))**

**for i in range(n):**

**for j in range(n):**

**print("\*",end="")**

**print()**

25. קילטו מהמשתמש 10 מספרים. עבור כל מספר שנקלט, הציגו ריבוע של מספרים עולים בגודל המספר שנקלט. לדוגמה, משתמש מכניס מספר ראשון 4, יש להציג מיד ריבוע מספרים עולים מ-1 ועד 4 (כלומר 4 שורות שבכל שורה יש את המספרים 1 עד 4). המספר הבא בתור שהוא מכניס הינו 7, יש להציג מיד ריבוע מספרים עולים מ-1 ועד 7 (כלומר 7 שורות שבכל שורה יש את המספרים 1 עד 7) וכו'.

26. קילטו מהמשתמש מספרים חיוביים שוב ושוב עד שמכניס 0 או מספר שלילי. עבור כל מספר שנקלט (ללא ה-0 או השלילי), הציגו ריבוע מספרים עולים מ-1 ועד המספר שנקלט. לדוגמה, משתמש מכניס מספר ראשון 4, יש להציג מיד ריבוע מספרים עולים מ-1 ועד 4 (כלומר 4 שורות שבכל שורה יש את המספרים 1 עד 4). המספר הבא בתור שהוא מכניס הינו 7, יש להציג מיד ריבוע מספרים עולים מ-1 ועד 7 (כלומר 7 שורות שבכל שורה יש את המספרים 1 עד 7) וכו'. אם המשתמש מכניס 0 או שלילי – על הלולאה להסתיים.

27. קילטו מהמשתמש מספרים חיוביים שוב ושוב עד שמכניס 0 או מספר שלילי. עבור כל מספר שנקלט (ללא ה-0 או השלילי), הציגו ריבוע מספרים יורדים מהמספר שנקלט ועד 1. לדוגמה, משתמש מכניס מספר ראשון 4, יש להציג מיד ריבוע מספרים יורדים מ-4 ועד 1 (כלומר 4 שורות שבכל שורה יש את המספרים 4 עד 1 בסדר יורד). המספר הבא בתור שהוא מכניס הינו 7, יש להציג מיד ריבוע מספרים יורדים מ-7 ועד 1 (כלומר 7 שורות שבכל שורה יש את המספרים 7 עד 1 בסדר יורד) וכו'. אם המשתמש מכניס 0 או שלילי – על הלולאה להסתיים.

28. קילטו מהמשתמש 10 זוגות של מספרים. עבור כל זוג הציגו את המספר המקסימלי מביניהם. אם שניהם שווים יש להציג אחד מהם. לדוגמה משתמש מכניס זוג ראשון 4 ו-7, יש להציג שהמקסימלי הוא 7. משתמש מכניס כזוג שני 5 ו-9, יש להציג שהמקסימלי הוא 9, משתמש מכניס כזוג שלילי 6 ו-6, יש להציג שהמקסימלי הוא 6 וכו'.

29. קילטו מהמשתמש זוגות זוגות של מספרים שוב ושוב עד שמכניס זוג מספרים זהה. עבור כל זוג שנקלט (כולל הזוג שמסיים את הקלט), הציגו את המספר המקסימלי מביניהם. אם שניהם שווים יש להציג אחד מהם. לדוגמה משתמש מכניס זוג ראשון 4 ו-7, יש להציג שהמקסימלי הוא 7. משתמש מכניס כזוג שני 5 ו-9, יש להציג שהמקסימלי הוא 9, משתמש מכניס כזוג שלילי 6 ו-6, יש להציג שהמקסימלי הוא 6 ולסיים את הלולאה כי זה זוג מספרים זהים.

30. קילטו מהמשתמש מספר count. לאחר מכן קילטו count שלשות של מספרים. עבור כל שלשה הציגו את המספר המקסימלי מביניהם. לדוגמה, אם המשתמש הכניס count=5, יש לקלוט לאחר מכן 5 שלשות של מספרים. עבור כל שלשה יש להציג את המקסימלי שנקלט ממנה. אז אם שלשה ראשונה הינה 3, 8, 2 – יש להציג שהמקסימלי זה 8. אם שלשה שנייה הינה 9, 3, 9 – יש להציג שהמקסימלי הוא 9. אם שלשה שלישית היא 4, 4, 4 – יש להציג שהמקסימלי הוא 4 וכו'.

**Data structure - list, dictionary**

1. צרו List ריק. הכניסו לתוכו 100 מספרים רנדומליים – כל מספר בין 0 ל-200 (כולל קצוות). הציגו את ה-List.

הציגו את אורך ה-List.

קילטו מספר מהמשתמש, הציגו הודעה המציינת האם המספר קיים ב-List או לא קיים ב-List.

קילטו מספר נוסף מהמשתמש, הציגו הודעה המציינת כמה פעמים המספר קיים ב-List (אם לא קיים זה אומר שהוא קיים 0 פעמים).

קילטו מספר נוסף מהמשתמש, בצעו מחיקה של כל המופעים של המספר הזה מה-List כך שהוא לא יהיה קיים שם כלל. הציגו שוב את ה-List בכדי לוודא שהמספר אכן לא קיים.

קילטו מספר נוסף מהמשתמש, צרו List נפרד והכניסו אליו את כל המספרים הגדולים מהמספר הזה. הציגו את ה-List הנפרד הזה בכדי לוודא שאכן קיימים בו מספרים הגדולים מהמספר שנקלט.

1. צרו Tuple המכיל 10 שמות של מוצרים (Hard-Coded).

הציגו את ה-Tuple הזה.

הציגו את הפריט הראשון שבו.

ע"י לולאה, רוצו על ה-Tuple, וכל פריט בו הציגו בשורה נפרדת אך 3 פעמים ע"י אופרטור ה-\* (Repetition Operator).

חשבו והציגו כמה סה"כ תווים יש בכל השמות יחד.

1. צרו פונקציה הקולטת באמצעות prompt מהמשתמש את ששת משכורות לרשימה .

הציגו את ממוצע המשכורות.

הציגו את המשכורת הגבוהה ביותר.

הציגו את המשכורת הנמוכה ביותר.

Function inputSalary(){

}

זמן את הפונקציה

function inputSalary(){

let salaries = []

for(let i=0;i<6;i++){

let salary = +prompt("enter salary")

salaries.push(salary)

}//end of for

let sum = 0

let max = salaries[0]

let min = salaries[0]

for(let i=0;i<salaries.length;i++){

sum+= salaries[i]

if(salaries[i] > max){

max = salaries[i]

}

if(salaries[i] < min){

min = salaries[i]

}

}// end of for

let avearge = sum/salaries.length

alert("average is :" + avearge + "min is : " + min + " and max is : " + max )

}

inputSalary()

1. צרו Set ריק. הוסיפו אליו 2000 מספרים רנדומליים, כל מספר בין 1 עד 1000 (כולל קצוות).

הציגו את ה-Set.

הציגו את ממוצע המספרים.

קילטו מהמשתמש מספר, הציגו הודעה המציינת האם המספר הזה נמצא ב-Set או לא.

ע"י לולאה, רוצו על ה-Set והציגו כל מספר ואת החזקה הריבועית שלו:

Number: \_\_\_, Power: \_\_\_.

Number: \_\_\_, Power: \_\_\_.

function test(){

let arr = new Set()

for(let i=0;i<5;i++){

let number = +prompt("enter a num")

arr.add(number)

}

arr.add("hello")

console.log(arr)

if(arr.has(3)){

alert("3 exist")

}

if(arr.has("hello")){

alert("hello exist")

}

if(arr.has(100)){

arr.delete(100) // delete value 100

alert("100 removed")

}

}

1. צרו Set ריק. הוסיפו אליו **10000** מספרים רנדומליים, כל מספר בין 1 ל-100 (כולל קצוות).

חשבו את סכום כל המספרים שנמצאים ב-Set. האם הסכום שווה ל-5050? אם כן – איך זה יכול להיות אם הכנסתם מספרים רנדומליים?

Dictionary

let student = {

name:"Avi",

age:17,

grade:92

}

console.log(student.name)// Avi

console.log(student["name"]) // Avi

student.city = "Dimini" // add key city

//change key age

student.age = 18

//change key age

student.age++

//add key password

student.password = "123456"

console.log(student)

//delete key example

delete student.age

console.log(student) // key age removed

if("name" in student){

alert("key name exist")

}

else{

alert("key name not exist")

}

for(let key in student){

console.log(key) // all exist keys - name, password, grade , city

console.log(student[key]) // all exist values

}

console.log(student.age) // undefined

1. צרו Dictionary המכיל מוצרי אלקטרוניקה. לכל מוצר יש קוד (מספר בן 3 ספרות) ושם (מחרוזת).

ה-Key זהו קוד המוצר, וה-Value זהו שם המוצר.

על ה-Dictionary להכיל 3 מוצרים שונים Hard-Coded.

קילטו מהמשתמש מוצר נוסף (קוד + שם). הוסיפו אותו ל-Dictionary.

קילטו מהמשתמש קוד מוצר, הציגו את המוצר השייך לקוד הזה (כמובן שאסור לתוכנית לקרוס).

ע"י לולאה, הציגו את כל המוצרים, כל מוצר בשורה נפרדת בפורמט הבא:

Product ID: \_\_\_, Product Name: \_\_\_.

Product ID: \_\_\_, Product Name: \_\_\_.

…

קילטו מהמשתמש קוד מוצר, בצעו מחיקה של המוצר הזה מהאוסף. מה קורה אם הקוד לא קיים?

הציגו את המוצרים שוב.

let products = {

123:"tv",

562:"radio",

453:"tshirt"

}

let code = prompt("enter a code")

let product = prompt("enter product name")

products[code] = product

let c = +prompt("enter product name")

alert(products[c])

console.log(products)

for(let key in products){

console.log("id is :" + key + " and prodcut is " + products[key])

}

code = prompt("find a code")

if (code in products)

{

alert(products[code] + " exist")

delete products[code]

}

else{

alert("code not exist")

}

console.log(products)

**תרגיל 7**

### **🔹 תרגיל a**

צור אובייקט בשם car עם המאפיינים:  
 brand, model, year — עם ערכים שאתה בוחר.  
 הדפס את כל אחד מהם בעזרת console.log.

### **🔹 תרגיל b**

הוסף לאובייקט car מאפיין חדש בשם color עם ערך כלשהו.

### **🔹 תרגיל c**

שנה את הערך של year ל־2025.

### **🔹 תרגיל d**

מחק את המאפיין model מתוך האובייקט.

### **🔹 תרגיל e**

כתוב פונקציה שמקבלת אובייקט student עם מאפיינים name, grade, ומדפיסה:

שם התלמיד: X, ציון: Y

### **🔹 תרגיל f**

צור אובייקט בשם book עם מפתחות: title, author, pages.  
 כתוב לולאת for...in שמדפיסה את כל המפתחות והערכים שלו.

### **🔹 תרגיל g**

כתוב פונקציה שמקבלת אובייקט ומחזירה את מספר המפתחות בו.

### **🔹 תרגיל h**

צור מערך של אובייקטים בשם products.  
 כל אובייקט מייצג מוצר עם name, price, inStock.  
 הדפס את שמות כל המוצרים שנמצאים במלאי (inStock === true).

//brand, model, year

let car = {

"model":"mazda2",

"year":1987,

"brand":"mazda"

}

console.log("=====================car============")

console.log(car)

console.log("=====================add color============")

car.color = "blue"

console.log(car)

console.log("=====================update year============")

car.year = 2025

console.log(car)

console.log("=====================model deleted")

let model = "model"

// delete car["model"]

// delete car[model]

delete car.model

console.log(car)

let student = {

grade:78,

name:"oren"

}

console.log(student)

//title, author, pages

let book = {

pages:123,

author:"sergei",

title:"bla bla bla"

}

for(let key in book){

console.log(key)

console.log(book[key])

}

function numOfKeys(dict){

let keys = Object.keys(dict)

console.log(keys) // [ ]

console.log(book)

return keys.length

}

alert(numOfKeys(book))

//name, price, inStock

let products = [

{

name:"ball",

price:45,

inStock:20

},

{

name:"tshirt",

price:30,

inStock:100

},

{

name:"iphone",

price:4750,

inStock:2

}

]

for(let i=0;i<products.length;i++){

console.log(products[i]["name"])

console.log(products[i]["price"])

console.log(products[i]["inStock"])

}

for(let i=0;i<products.length;i++){

console.log(products[i].name)

console.log(products[i].price)

console.log(products[i].inStock)

}

for(let i=0;i<products.length;i++){

let product = products[i]

console.log(product.price)

console.log(products.inStock)

console.log(products["inStock"])

}

list and dictionary

1. צרו שלושה אובייקטים Hard-Coded המתארים מכוניות.

לכל מכונית יש יצרן, מודל, שנת ייצור וצבע.

הכניסו את שלושת האובייקטים לתוך List.

רוצו על ה-List בלולאה והציגו את כל האובייקטים, כאשר כל אובייקט יש להציג ע"י הצגה נפרדת של כל מאפיין.

רוצו שוב על כל ה-List והציגו כל אובייקט ע"י לולאה מקוננת המציגה את הפרטים של אותו אובייקט.

1. צרו List המכיל חמישה אובייקטים (Hard-Coded) המתארים בגדים.

לכל בגד יש סוג, מידה, מחיר וצבע.

ע"י לולאה מקוננת הציגו את כל המאפיינים של כל האובייקטים.

הציגו את ממוצע המחירים.

הציגו את המידה הגדולה ביותר.

1. צרו מערך עבור ערכי Geolocation (מיקום גאוגרפי).

לכל אובייקט במערך יש latitude (קו רוחב) ו-longitude (קו אורך).

קילטו מהמשתמש 3 אובייקטים לתוך המערך הזה.

הציגו את כל ה-Geolocations שנקלטו, כל אחד בפורמט הבא:

(latitude, longitude)

(latitude, longitude)

(latitude, longitude)

…

לדוגמה:

(30, 32)

(33, 36)

(40, 55)

מטריצות

1. צרו את המטריצה הבאה:

const matrix = [[12, 23, 34, 45], [56, 67, 78, 89], [10, 20, 30, 40]];

הציגו:

1. את המטריצה כולה.
2. את סכום כל הערכים.
3. את ממוצע כל הערכים.
4. את המספר המקסימלי שבמטריצה.
5. את המספר המינימלי שבמטריצה.
6. את המטריצה בדו-ממד, אך רק את המספרים הזוגיים, כאשר במקום כל מספר אי-זוגי יש להציג "X".
7. את המטריצה בדו-ממד, אך רק את המספרים המתחלקים ב-7 או מסתיימים ב-7, כאשר במקום כל מספר אחר יש להציג כוכבית "\*".
8. צרו מטריצה בשם multiplication\_board המיועדת להכיל את לוח הכפל.

ע"י לולאה מקוננת הכניסו אליה את ערכי לוח הכפל:

הציגו את המטריצה.

חשבו והציגו את סכום כל המספרים.

1. תרגיל אתגר: צרו מטריצה Hard-Coded בשם colored\_items בעלת שלושה List-ים.

על ה-List הראשון במטריצה להכיל 3 פריטים בעלי צבע אדום.

על ה-List השני במטריצה להכיל 4 פריטים בעלי צבע צהוב.

על ה-List השלישי במטריצה להכיל 2 פריטים בעלי צבע ירוק.

הציגו את כל המטריצה.

ע"י לולאות מתאימות מצאו והציגו את הפריט הקצר ביותר (מבחינת תווים). אם יש מספר פריטים כאלו – הציגו את הראשון.

ע"י לולאות מתאימות מצאו והציגו את הפריט הקצר הארוך (מבחינת תווים). אם יש מספר פריטים כאלו – הציגו את כולם.

1. אתגר: צרו מטריצה ריקה עבור ציוני תלמידים.

קילטו מהמשתמש לתוך המטריצה ציונים של 3 תלמידים. לכל תלמיד 4 ציונים (סה"כ 12 ציונים)

הציגו את המטריצה בדו-ממד.

הציגו את ממוצע הציונים הגבוה ביותר של תלמיד (כלומר הממוצע הגבוה ביותר של שורת ציונים).

הציגו את שורת הציונים עם הממוצע הנמוך ביותר (אם יש מספר כאלו – הציגו את השורה הראשונה מבניהם).

# **פונקציות**

1. צרו פונקציה המציגה את ההודעה "Make Things Go Right" על הדף ואז יורדת שורה.

קיראו לה 3 פעמים.

1. צרו פונקציה המציגה את שמכם המלא על המסך.

קיראו לה 100 פעמים.

1. צרו פונקציה המקבלת מחרוזת אחת (כפרמטר לסוגריים).

על הפונקציה להציג את המחרוזת 10 פעמים.

קיראו לפונקציה פעם ראשונה ע"י שליחת מחרוזת Hard-Coded.

קיראו לפונקציה פעם שנייה ע"י שליחת מחרוזת מהמשתמש.

1. צרו פונקציה המקבלת (לסוגריים) שני פרמטרים. הראשון הינו הודעה והשני הינו מספר הפעמים שיש להציג את ההודעה. על הפונקציה להציג את ההודעה כמספר הפעמים שקיבלה. לדוגמה אם קיבלה את המחרוזת Hi ואת המספר 3, היא תציג 3 פעמים את המחרוזת Hi.

קיראו לה פעם ראשונה ע"י שליחת ערכי Hard-Coded.

קיראו לה פעם שנייה ע"י שליחת ערכים מהמשתמש (המשתמש בוחר את המחרוזת ואת מספר הפעמים שיש להציגה).

1. צרו פונקציה המקבלת לסוגריים 3 מספרים. על הפונקציה לחשב ולהציג את הממוצע שלהם.

קיראו לה פעם ראשונה ע"י שליחת ערכי Hard-Coded.

קיראו לה פעם שנייה ע"י שליחת ערכים מהמשתמש.

קיראו לה פעם שלישית ע"י שליחת ערכים רנדומליים שהמחשב מגריל.

1. צרו פונקציה המקבלת לסוגריים 3 מספרים. על הפונקציה לחשב ולהציג את המספר המקסימלי מביניהם.

קיראו לה פעם ראשונה ע"י שליחת ערכי Hard-Coded.

קיראו לה פעם שנייה ע"י שליחת ערכים מהמשתמש.

קיראו לה פעם שלישית ע"י שליחת ערכים רנדומליים.

1. צרו פונקציה המקבלת לסוגריים מספר המתאר איזה סמיילי להציג על המסך.

אם קיבלה 1, תציג :-)

אם קיבלה 2, תציג :-(

אם קיבלה 3, תציג :-/

אם קיבלה 4, תציג ;-)

אם קיבלה 5, תציג ;-(

קיראו לה פעם ראשונה ע"י שליחת מספר Hard-Coded.

קיראו לה פעם שנייה ע"י שליחת מספר מהמשתמש.

קיראו לה פעם שלישית ע"י שליחת מספר רנדומלי בין 1 ל-5

צרו פונקציה נוספת שמציגה את כל חמשת הסמיילים ע"י שימוש בפונקציה הנ"ל.

צרו פונקציה נוספת המציגה 100 סמיילים רנדומליים ע"י שימוש בפונקציה הנ"ל.

1. צרו פונקציה המציגה את השעה הנוכחית (רמז: Google).

הפונקציה לא אמורה לקבל שום ארגומנטים לסוגריים.

קיראו לה פעם אחת.

1. אתגר: צרו פונקציה המקבלת List.

על הפונקציה להציג את ה-List בשורה אחת, כאשר בין כל מספר ומספר מופיע התו "|" אך הוא לא מופיע אחרי המספר האחרון שמוצג.

קיראו לה פעם ראשונה ע"י שליחת List Hard-Coded.

קיראו לה פעם שנייה ע"י שליחת List שנוצר רנדומלית.

1. צרו פונקציה המקבלת List.

על הפונקציה להציג את ממוצע ערכי ה-List.

קיראו לה פעם ראשונה ע"י שליחת Hard-Coded List.

קיראו לה פעם שנייה ע"י שליחת List שנוצר רנדומלית.

1. צרו פונקציה המקבלת לסוגריים שני ארגומנטים: start ו-end.

על הפונקציה להציג על הדף מספר רנדומלי בטווח שנשלח אליה (בין start ל-end כולל קצוות).

קיראו לפונקציה פעם ראשונה ע"י שליחת שני מספרים Hard-Coded.

קיראו לפונקציה פעם שנייה ע"י שליחת שני מספרים מהמשתמש.

קיראו לפונקציה 100 פעמים לצורך הצגת 100 מספרים רנדומליים, כל מספר בין (10-) ל-(10+).

1. קילטו מהמשתמש מספר n, הציגו ריבוע מלא כוכביות בגודל n. לדוגמה אם נקלט 4, יש להציג:

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

יש לחלק לשתי פונקציות. האחת מקבלת (לסוגריים) מספר length ומציגה שורה של length כוכביות. השנייה מקבלת מספר size ומציגה ריבוע כוכביות בגודל size על size.

יש לבדוק את הפונקציה השנייה ע"י שליחת מספר Hard-Coded וע"י שליחת מספר מהמשתמש.

1. קילטו מהמשתמש רוחב וגובה של מלבן, הציגו מלבן מלא כוכביות בממדים שנקלטו. לדוגמה אם נקלט 6 ו-4, יש להציג:

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

יש לחלק לשתי פונקציות – אחת מקבלת מספר ומציגה שורת כוכביות באורך המספר, השנייה מקבלת אורך ורוחב של מלבן ומציגה את כל המלבן.

קיראו לפונקציה המציגה את המלבן פעם ראשונה ע"י שליחת מספרים Hard-Coded, פעם שנייה ע"י שליחת מספרים רנדומליים, קיראו לה פעם שלישית ע"י שליחת מספרים מהמשתמש.

1. צרו פונקציה המקבלת לסוגריים ארגומנט count ומציגה count מלבנים ע"י שימוש בפונקציה מהסעיף הקודם.

קילטו מספר n מהמשתמש, הציגו n מלבנים רנדומליים על הדף.

1. אתגר: קילטו מהמשתמש רוחב וגובה של מלבן, הציגו מלבן ריק בממדים שנקלטו. לדוגמה אם נקלט 6 ו-4, יש להציג:

\* \* \* \* \* \*

\* \*

\* \*

\* \* \* \* \* \*

יש לחלק לפונקציות מתאימות.

קיראו פעם ראשונה ע"י שליחת מספרים Hard-Coded.

קיראו פעם שנייה ע"י שליחת מספרים רנדומליים.

קיראו פעם שלישית ע"י שליחת מספרים מהמשתמש.

1. צרו פונקציה המקבלת מספר n ומציגה בשורה אחת את כל המספרים מ-n ועד 1 בסדר יורד.

אם לדוגמה קיבלה 5, תציג:

5 4 3 2 1

צרו פונקציה נוספת המקבלת מספר size ומציגה משולש מספרים יורד כתלות ב-size. לדוגמה, אם קיבלה 5, תציג:

5 4 3 2 1

4 3 2 1

3 2 1

2 1

1

על הפונקציה השנייה להשתמש בפונקציה הקודמת.

1. קילטו מהמשתמש מספר n, הציגו משולש כוכביות הפוך כתלות ב-n. אם נקלט 4, יש להציג:

\* \* \* \*

\* \* \*

\* \*

\*

יש לחלק לפונקציות מתאימות

1. קילטו מהמשתמש מספר n, הציגו את הצורה הבאה כתלות ב-n, לדוגמה עבור 4 יש להציג:

1 2 3 4 \* \* \* \*

1 2 3 \* \* \*

1 2 \* \*

1 \*

חלקו את התוכנית לפונקציות מתאימות.

1. קילטו מהמשתמש מספר n, הציגו את שני המשולשים הבאים כתלות ב-n, לדוגמה עבור 4 יש להציג:

\* \* \* \*

\* \* \*

\* \*

\*

\*

\* \*

\* \* \*

\* \* \* \*

יש לחלק לפונקציות מתאימות.

1. אתגר: קילטו מהמשתמש מספר n, הציגו את הצורה הבאה כתלות ב-n, אם נקלט 4, יש להציג:

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

שימו לב לסימטריה.

יש לחלק לפונקציות מתאימות.

# **Functions – Returning a Value**

1. צרו פונקציה המקבלת לסוגריים ארגומנט אחד שהינו List מספרים. על הפונקציה לחשב ולהחזיר את ממוצע ערכי ה-List ללא פונקציות מערכת לחישוב ממוצע. קיראו לה 3 פעמים, בכל פעם שילחו לה List אחר. בכל פעם קבלו את התוצאה והציגו אותה.
2. צרו פונקציה המקבלת לסוגריים ארגומנט אחד שהינו List מספרים. על הפונקציה לחשב ולהחזיר את הערך המינימלי שב-List ללא פונקציות מערכת לחישוב מקסימום. קיראו לה 3 פעמים, בכל פעם שילחו לה List אחר. בכל פעם קבלו את התוצאה והציגו אותה.
3. צרו פונקציה המקבלת לסוגריים ארגומנט אחד שהינו List מחרוזות. על הפונקציה לחשב ולהחזיר את **אורך** המחרוזת הארוכה ביותר שב-List (כלומר מחזירה מספר). קיראו לה 3 פעמים, בכל פעם שילחו לה List אחר. בכל פעם קבלו את התוצאה והציגו אותה.
4. צרו פונקציה המקבלת לסוגריים ארגומנט אחד שהינו List מחרוזות. על הפונקציה לחשב ולהחזיר את המחרוזת הארוכה ביותר שב-List (כלומר מחזירה מחרוזת). קיראו לה 3 פעמים, בכל פעם שילחו לה List אחר. בכל פעם קבלו את התוצאה והציגו אותה.
5. צרו פונקציה המקבלת לסוגריים ארגומנט אחד שהינו List מספרים. על הפונקציה לחשב ולהחזיר כמה מספרים גדולים או שווים לממוצע ערכי הList. קיראו לה 3 פעמים, בכל פעם שילחו List אחר. בכל פעם קבלו את התוצאה והציגו אותה.
6. צרו פונקציה המקבלת List, מחשבת את האינדקס של המספר המינימלי שבו ומחזירה את האינדקס. אם יש מספר איברים מינימליים זהים – על הפונקציה להחזיר את האינדקס של הראשון שנמצא. קיראו לה 3 פעמים, בכל פעם שילחו List אחר, בכל פעם קבלו את התוצאה והציגו אותה.
7. צרו פונקציה המקבלת מטריצת מספרים. על הפונקציה לחשב את המספר המקסימלי של המטריצה ולהחזיר אותו. בדקו אותה.
8. צרו פונקציה המקבלת מטריצת מספרים. על הפונקציה לחשב ולהחזיר את ממוצע ערכי המטריצה. בידקו אותה.
9. צרו פונקציה המקבלת מטריצה המכילה את לוח הכפל. על הפונקציה לבדוק אם אכן המטריצה מכילה לוח כפל חוקי. אם כן – תחזיר true, אם לא – תחזיר false. בידקו אותה ע"י שליחת לוח כפל שגוי וע"י שליחת לוח כפל חוקי והציגו הודעה מתאימה.
10. **צרו פונקציה המקבלת מספר. על הפונקציה להחזיר true אם המספר שקיבלה הינו מספר ראשוני (מתחלק בעצמו וב-1 בלבד) או false אם המספר שקיבלה אינו ראשוני. בידקו אותה.**
11. **צרו פונקציה המקבלת List מספרים. על הפונקציה להחזיר true אם כל המספרים ב-List הינם ראשוניים. אם לא – תחזיר false. על הפונקציה להשתמש בפונקציה שכתבתם בסעיף הקודם. בידקו אותה.**
12. **את שתי הפונקציות הקודמות הכניסו ל-Module בשם primes ובידקו את הפונקציות מסקריפט הראשי (app.py).**
13. **צרו פונקציה המקבלת לסוגריים שני ארגומנטים: מינימום ומקסימום. על הפונקציה להחזיר מספר רנדומלי בטווח שהתקבל, כולל קצוות. לדוגמה, אם קיבלה 10 ו-20, תגריל מספר רנדומלי בין 10 ל-20 (כולל קצוות) ותחזיר אותו. קיראו לה 3 פעמים, בכל פעם הציגו את המספר המוחזר.**
14. אתגר: צרו פונקציה המקבלת לסוגריים ארגומנט אחד שהינו גודל List. על הפונקציה לייצר List המלא במספרים רנדומליים (כל מספר בין 1 ל-100 כולל קצוות) ולהחזיר את הList. קיראו לה 3 פעמים, בכל פעם שילחו מספר אחר ובכל פעם הציגו את הList המוחזר. יש להשתמש לצורך כך בפונקציות נוספות שכתבתם.
15. אתגר: צרו פונקציה המקבלת לסוגריים מספר מינימום ומספר מקסימום. על הפונקציה להחזיר List המכילה את כל המספרים הראשוניים הקיימים בטווח שקיבלה. קיראו לה עם טווח מספרים הנקלט מהמשתמש והציגו את הרשימה המוחזרת.

# Python – General Exercise

* כל פונקציה יש לבנות במודול. ( כל הפונקציות לבנות בקובץ בשם list\_module )
* כל פונקציה יש לבדוק(לזמן) מתוך app.py ולוודא שהיא עובדת כמו שצריך.

**.**

1. **צרו פונקציה המקבלת שני מספרים ומחזירה List המכיל את כל המספרים ביניהם**
2. **צרו פונקציה המקבלת List ומחזירה את סכום הרשימה**
3. **צרו פונקציה המקבלת List ומחזירה את ממוצע הרשימה**
4. **צרו פונקציה המקבלת List ומחזירה את מספר הזוגיים**
5. **צרו פונקציה המקבלת שני מספרים ומחזירה List המכילה את כל המספרים הראשוניים ביניהם.**
6. **"מספר מושלם" הינו מספר השווה לסכום מחלקיו. לדוגמה, המספר 6 הינו מספר מושלם משום שהמספרים שמחלקים את 6 ללא שארית הינם 1, 2 ו-3. בגלל ש- 1+2+3 שווה ל-6, אזי 6 נחשב למספר מושלם.**

**צרו פונקציה המקבלת מספר ומחזירה True אם המספר הינו מספר מושלם, או False אם המספר אינו מספר מושלם.**

1. צרו פונקציה המקבלת מספר ומחזירה את המספר המושלם הגדול ממנו.

השתמשו לצורך כך בפונקציה מהסעיף הקודם.

מהו המספר המושלם אחרי 6?

1. הציגו את כל המספרים המושלמים מ-1 ועד 10000.
2. צרו פונקציה המקבלת שני List-ים. על הפונקציה להחזיר True אם כל המספרים של ה-List השני מוכלים כולם ב-List הראשון. אם לא – יש להחזיר False.

l1 = [1, 2, 8] l2 = [100,200,1,2,400, 8] == > True

1. צרו פונקציה המקבלת שני מספרים. על הפונקציה להחזיר True אם המספרים הפוכים אחד לשני. מספר הפוך הינו מספר המכיל סדר ספרות הפוך. לדוגמה המספר 123 נחשב הפוך למספר 321. אם המספרים אינם הפוכים, על הפונקציה להחזיר False.
2. צרו פונקציה המקבלת שתי מחרוזות. על הפונקציה להחזיר True אם המחרוזת השנייה מוכלת במחרוזת הראשונה. לדוגמה, המחרוזת "python programming" מוכלת במחרוזת "amazing python programming ever!".
3. צרו פונקציה המקבלת מטריצה של מספרים שלמים. על הפונקציה להחזיר Tuple המכיל את המספר המינימלי, את הממוצע ואת המספר המקסימלי של ערכי המטריצה.
4. צרו פונקציה המקבלת List של מספרים שלמים. על הפונקציה להחזיר את סכום כלל הספרות שיש ב-List.

לדוגמה, אם קיבלה את ה-List:

[34, 56, 100, 2222, -567, 7]

על הפונקציה להחזיר 15 כי יש סה"כ 15 ספרות ב-List.

1. אתגר: צרו פונקציה המקבלת מספר שלם. על הפונקציה להחזיר מספר הבנוי מכל הספרות שלא קיימות במספר שהתקבל, בסדר יורד. לדוגמה, אם נשלח לפונקציה את המספר 54348272, על הפונקציה להחזיר את המספר 9610 מפני שהוא בנוי מכל הספרות שלא נמצאות במספר שהתקבל, בסדר יורד.

**Python – Exception Handling**

# 

1. במודול בשם list\_cals צרו פונקציה בשם get\_average המקבלת List ומחזירה את ממוצע הערכים שלו. ( raise נטפל ב error)

**אם נשלח List ריק – העלו חריגה מתאימה.**

**אם אחד הערכים ב list הוא לא מספר העלו חריגה ValueError .**

ב-app.py (נעשה try ו except) קילטו מהמשתמש מספר size המתאר גודל של List, צרו List ריק, קילטו size מספרים והכניסו כל אחד ל-List הריק שהגדרתם. שילחו את ה-List לפונקציה, קבלו את הממוצע והציגו אותו על הדף.

אסור לתוכנית לקרוס בשום מקרה. בכל מקרה של חריגה – יש להציג הודעת שגיאה מתאימה.

1. צרו תיקייה בתוך הפרויקט בשם logic. צרו בתיקייה מודול בשם range\_operations. צרו במודולפונקציה בשם generate\_list. על הפונקציה לקבל שלושה פרמטרים:
2. size: גודל ה-list שיש לייצר.
3. min: המספר המינימלי של טווח הערכים הרנדומליים שיש לייצר.
4. max: המספר המקסימלי של טווח הערכים הרנדומליים שיש לייצר.

על הפונקציה לייצר ולהחזיר List בעלת size מספרים רנדומליים, כאשר כל מספר בין min ל-max.

אם הפונקציה קיבלה פרמטר (מבין שלושת הפרמטרים) שאינו מספר שלם (לדוגמה מחרוזת, עשרוני וכו') – יש להעלות חריגה מתאימה.

אם הפונקציה קיבלה פרמטר (מבין שלושת הפרמטרים) שהינו מספר שלילי – יש להעלות חריגה מתאימה.

אם הפונקציה קיבלה min הגדול מ-max – יש להעלות חריגה מתאימה.

ב-app.py – קילטו מהמשתמש את שלושת הערכים (size, min, max), שלחו לפונקציה, קבלו את ה-List שפונקציה החזירה והציגו על המסך.

אסור לתוכנית לקרוס. יש לטפל בכל חריגה שעלולה להתרחש ע"י הצגת הודעת שגיאה מתאימה.

**תרגול try ו except**

1. במודול נפרד צרו פונקציה המקבלת מטריצה של ציונים של כיתה שלמה.

כל שורה במטריצה הינה רשימת ציונים של תלמיד אחד.

על הפונקציה לייצר List אחת שמכילה את כל הציונים של כל התלמידים כולם ולהחזיר את ה-List.

אם נשלחה מטריצה ריקה – יש להעלות חריגה מתאימה.

אם אחד הציונים לא חוקי – יש להעלות חריגה נוספת מתאימה.

ב-app.py צרו מטריצה Hard-Coded של ציוני תלמידים.

בצעו שליחה לפונקציה, קבלו את ה-List המוחזרת והציגו על המסך.

על התוכנית לעבוד ללא קריסות. אם יש קריסה יש להציג את ההודעה המתאימה.

1. במודול נפרד צרו פונקציה המקבלת מספר שלם n וסיפרה digit.

על הפונקציה לחשב ולהחזיר את מספר המופעים של digit בתוך n.

אם לא נשלח מספר שלם ל-n או digit (לדוגמה שלחו מחרוזת, מספר עשרוני וכו') יש להעלות חריגה מתאימה.

ב-app.py קלטו מספר n ו-digit מהמשתמש.

הציגו כמה פעמים מופיעה הספרה digit בתוך n.

לדוגמה, אם נקלט המספר 162636465 והספרה 6, יש להציג את ההודעה "הספרה 6 מופיעה 4 פעמים במספר".

**תרגילי input type כקלט**

1. כתוב פונקציה הקולטת מהמשתמש מספר מ input type שה id שלו number הפונקציה תציג אלרט האם המספר זוגי
2. **כתוב פונקציה הקולטת מהמשתמש את שמו מ input type שה id שלו fname. הפונקציה תציג את השם על פסקה ש id שלה firstname בצבע ירוק אם האות הראשונה היא a , אחרת בצבע אדום.** 
   1. **הצב input type**
   2. **הצב כפתור שמפעיל את הפונקציה**
   3. **הצב פסקה עם id בשם myname שתציג את השם**

**פונקציות חץ**

1. **Function Declaration**
2. **Function Expression**
3. **Arrow Function  
   כל שאלה מבקשת לכתוב פונקציה שמבצעת פעולה מסוימת. על התלמיד לכתוב אותה בכל אחת משלוש הדרכים.**

### **שאלות לתרגול כתיבת פונקציות:**

1. **פונקציה שמקבלת שני מספרים ומחזירה את המכפלה שלהם.**
2. **פונקציה שמקבלת מספר ומחזירה true אם הוא זוגי, אחרת false.**
3. **פונקציה שמקבלת מחרוזת ומחזירה את אורכה.**
4. **פונקציה שמקבלת מחרוזת ומחזירה אותה באותיות גדולות.**
5. **פונקציה שמקבלת שני מספרים ומחזירה את המספר הגדול מביניהם.**
6. **פונקציה שמקבלת מערך ומחזירה את האיבר הראשון בו.**
7. **פונקציה שמקבלת מערך ומחזירה את סכום כל האיברים בו.**
8. **פונקציה שמקבלת שם פרטי ומחזירה את המשפט: "שלום, [שם]"**
9. **פונקציה שמקבלת גיל ומחזירה האם האדם בגיר (18 ומעלה).**
10. **פונקציה שמקבלת שני מספרים ומחזירה את הממוצע שלהם.**

**Callback**

**JavaScript - Callbacks**

1. צרו פונקציה א' המציגה את שמכם על המסך.

צרו פונקציה ב' המקבלת callback לסוגריים וקוראת לה בצורה הבאה: callback()

קיראו לפונקציה ב' ושילחו לה לסוגריים את פונקציה א'.

Function printName(){

console.log(“ofen”)

}

Function f(callback) {

callback();

}

f(printName)

1. להלן פונקציה:

Function cool(callback) {

callback();

}

Function random (){

let number = Math.floor(Math.random() \*100)

console.log(number)

}

cool(random)

קיראו לה ושילחו לה פונקציה שמציגה מספר רנדומלי כלשהו על המסך.

1. להלן פונקציה:

function nice(callback) {

callback(42);

}

קיראו לה ושילחו לה פונקציה כשלהי(בנו לבד פונקציה בשם printNum המקבלת כפרמטר מספר ומדפיסה אותן) והעבירו ל nice את הפונקציה printNum

1. להלן פונקציה:

**נתון**

function amazing(callback) {

const num = callback(42, 128, 37, 81, 66);

console.log("Num: " + num);

}

Function f(a1, a2, a3, a4, a5){

let list = [a1, a2, a3, a4, a5)

let index = Math.floor(Math.random() \* list.length) // 0 - 4

return list[index];

}

amazing(f)

כתבו פונקציה t המקבלת חמישה ארגומנטים (במקרה הנ"ל הם יהיו המספרים שבסוגריים), ומחזירה אחד מהם בצורה רנדומלית.והפעילו את amazing עם הפונקציה t

1. להלן פונקציה:

function cool(paintCallback) {

paintCallback();

}

כתבו פונקציה randomColor המצהירה על מערך עם 4 צבעים. הפונקציה תגריל מספר בין 0 לגודל המערך. ותדפיס אותו. העבירו את הפונקציה randomColor ל cool.

1. להלן פונקציה:

function amazing(paintCallback) {

const paintedColor = paintCallback("Red", "Green", "Blue");

document.write("Painted Color: " + paintedColor);

}

קיראו לה ושילחו לה פונקציה שמקבלת שלושה צבעים (במקרה הנ"ל הם יהיו אדום, ירוק וכחול), וצובעת את המסך בצבע אחד משלושת אלו (בצורה רנדומלית). לאחר מכן היא מחזירה את הצבע שבעזרתו צבעה את המסך.

1. הציגו בכל שנייה מספר רנדומלי על המסך. על המספר להיות בין 1 ל-100 כולל קצוות.
2. שנו בכל שנייה את צבע הרקע של הדף לצבע רנדומלי.

שינוי צבע הדף הינו ע"י הפקודה: document.body.bgColor="some-color"

(אפשר להגריל צבע מבין סדרת צבעים קבועים ואפשר להגריל צבע רנדומלי לגמרי – מה שנוח לכם).

1. הציגו בכל 3 שניות על הדף מערך חדש של 100 מספרים רנדומליים.

## **10. לבדוק אם מילה היא ארוכה או קצרה**

### **מטרה:**

לכתוב פונקציה אסינכרונית(הפונקציה תפעל אחרי 5 שניות נשתנש ב setTimeout) המקבלת מחרוזת ומחזירה אם היא **"קצרה"** (מתחת ל־5 תווים) או **"ארוכה"** (5 תווים או יותר).

### **📋 דרישות:**

* הפונקציה תקבל:  
  + מחרוזת
  + successCallback - נבצע alert עם המילה הצלחה והמחרוזת
  + errorCallback - נבצע alert עם המילה כישלון אורך המחרוזת
* אם המחרוזת ריקה – שלח שגיאה.
* אחרת – אם היא ארוכה מ־5 תווים → החזר "המחרוזת ארוכה", אחרת "המחרוזת קצרה"

**שיעורי בית – Promise**

* עבור כל תרגיל מהתרגילים הבאים יש לבנות פונקציה המחזירה Promise.

יש להשתמש בה לאחר מכן בשתי צורות:

1. ע"י הפונקציות then ו-catch של אובייקט ה-Promise
2. ע"י async-await.
3. צרו פונקציה בשם generate7BoomAfterDelayAsync המקבלת לסוגריים שני ארגומנטים: מספר מינימלי ומספר מקסימלי ומחזירה Promise. על הפונקציה להמתין שנייה ואז להגריל מספר בטווח שקיבלה. אם המספר שהגרילה מתחלק ב-7 או מסתיים ב-7, עליה לדווח אותו ע"י resolve כהצלחה. אם המספר שהגרילה לא מתחלק ב-7 ולא מסתיים ב-7, עליה לדווח ע"י reject הודעת שגיאה.

צרו שתי תיבות טקסט ולחצן. על המשתמש להכניס בתיבות הטקסט מספר מינימלי ומספר מקסימלי. בלחיצה על הלחצן יש לקרוא לפונקציה הנ"ל, להציג את מספר ה-"7 בום" שהצליח או את הודעת הכישלון.

1. צרו פונקציה בשם generatePrimeNumberAfterDelayAsync המקבלת לסוגריים שני ארגומנטים: מספר מינימלי ומספר מקסימלי ומחזירה Promise. על הפונקציה להמתין שנייה ואז להגריל מספר בטווח שקיבלה. אם המספר שהגרילה הינו מספר ראשוני (מתחלק רק בעצמו וב-1 בלבד), עליה לדווח אותו ע"י resolve כהצלחה. אם המספר שהגרילה אינו מספר ראשוני (יש מספר כלשהו שמחלק אותו ללא שארית), עליה לדווח ע"י reject הודעת שגיאה.

צרו שתי תיבות טקסט ולחצן. על המשתמש להכניס בתיבות הטקסט מספר מינימלי ומספר מקסימלי. בלחיצה על הלחצן יש לקרוא לפונקציה הנ"ל, להציג את המספר הראשוני שהוגרל אם יש הצלחה או להציג את הודעת הכישלון.

1. צרו פונקציה בשם generateCuteAnimalAfterDelayAsync שלא מקבלת דבר לסוגריים ומחזירה Promise. על הפונקציה להמתין שנייה ואז להגריל חייה חמודה מסדרה קבועה של החיות הבאות: "חתלתול", "כלבלב", "ארנבון", "תוכון", "עקרב", "עכביש", "ג'וק". אם הגרילה חיה חמודה מהחיות הנ"ל (ואין צורך לפרט את רמת החמידות של החיות הנ"ל), עליה לדווח אותה ע"י resolve כהצלחה. אם הגרילה חיה לא חמודה, עליה לדווח ע"י reject הודעת שגיאה הכוללת בתוכה את שם החיה הלא חמודה שהוגרלה. בלחיצה על לחצן יש לקרוא לפונקציה הנ"ל, להציג את החיה שהוגרלה או את הודעת הכישלון.
2. צרו פונקציה בשם generateWorkingDayAfterDelayAsync שלא מקבלת דבר לסוגריים ומחזירה Promise. על הפונקציה להמתין שנייה ואז להגריל יום משבעת ימי השבוע ("ראשון", "שני"... "שבת"). אם הגרילה יום עבודה ("ראשון" עד "חמישי"), עליה לדווח את היום ע"י resolve כהצלחה. אם הגרילה "שישי" או "שבת", עליה לדווח ע"י reject הודעת שגיאה הכוללת בתוכה את שם היום שאינו יום עבודה שהוגרל.

בלחיצה על לחצן יש לקרוא לפונקציה הנ"ל, להציג את יום העבודה שהוגרל או את הודעת הכישלון.

1. צרו פונקציה בשם getArrayFromServerAsync המקבלת size לסוגריים ומחזירה Promise. על הפונקציה להמתין שנייה ואז להגריל מספר n. אם n זוגי עליה לייצר מערך בגודל size המכיל מספרים רנדומליים ולדווח אותו כהצלחה ע"י resolve. אם n אי-זוגי עליה לדווח ע"י reject הודעת שגיאה שלא ניתן להביא את המערך מהשרת.

בלחיצה על לחצן יש לקרוא לפונקציה הנ"ל, להציג את המערך או את הודעת הכישלון.

1. צרו פונקציה בשם getPizzaFromServerAsync שלא מקבלת דבר לסוגריים ומחזירה Promise. על הפונקציה להמתין שנייה ואז להגריל מספר n. אם n זוגי עליה לייצר אובייקט פיצה המכיל קוטר (רנדומלי בין 10 ל-50), מחיר (רנדומלי בין 20 ל-80), ומספר תוספות (רנדומלי בין 0 ל-4) ולדווח אותו כהצלחה ע"י resolve. אם n אי-זוגי עליה לדווח ע"י reject הודעת שגיאה שלא ניתן להביא אובייקט פיצה מהשרת.

בלחיצה על לחצן יש לקרוא לפונקציה הנ"ל, להציג את אובייקט הפיצה או את הודעת הכישלון.

1. אתגר: צרו פונקציה בשם generateStrongPasswordAsync שלא מקבלת דבר לסוגריים ומחזירה Promise. על הפונקציה להמתין שנייה ואז לייצר סיסמה רנדומלית בגודל 6 תווים הבנויה מאותיות קטנות, אותיות גדולות וספרות. אם הסיסמה מכילה גם אות קטנה, גם אות גדולה וגם סיפרה, עליה לדווח אותה כהצלחה ע"י resolve. אם בסיסמה אין אות קטנה או אין אות גדולה או אין סיפרה עליה לדווח ע"י reject הודעת כישלון המכילה את הסיסמה הלא חזקה שהוגרלה.

בלחיצה על לחצן יש לקרוא לפונקציה הנ"ל, להציג את הסיסמה שהוגרלה או את הודעת הכישלון.

בהצלחה 😊

### **🧠 פונקציות נפוצות לעבודה עם מערכים:**

#### **🟢 פונקציות של חיפוש וסינון:**

| **שם הפונקציה** | **הסבר** |
| --- | --- |
| **filter()** | **מסנן את המערך לפי תנאי, מחזיר מערך חדש.** |
| **find()** | **מחזיר את האיבר הראשון שמתאים לתנאי.** |
| **findIndex()** | **מחזיר את האינדקס של האיבר הראשון שמתאים לתנאי.** |
| **includes()** | **בודק אם ערך קיים במערך. מחזיר true/false.** |
| **indexOf()** | **מחזיר את האינדקס של איבר מסוים (או -1 אם לא נמצא).** |
| **lastIndexOf()** |  |

#### **🔵 פונקציות של לולאות ועיבוד:**

| **שם הפונקציה** | **הסבר** |
| --- | --- |
| **forEach()** | **מריץ פונקציה על כל איבר, לא מחזיר תוצאה.** |
| **map()** | **יוצר מערך חדש לפי שינוי של כל איבר במקור.** |
| **reduce()** | **מצמצם את כל הערכים לתוצאה אחת (למשל סכום).** |
| **some()** | **מחזיר true אם לפחות איבר אחד עומד בתנאי.** |
| **every()** | **מחזיר true רק אם כל האיברים עומדים בתנאי.** |

#### **🔴 פונקציות של הוספה/הסרה:**

| **שם הפונקציה** | **הסבר** |
| --- | --- |
| **push()** | **מוסיף איבר לסוף המערך.** |
| **pop()** | **מסיר את האיבר האחרון.** |
| **shift()** | **מסיר את האיבר הראשון.** |
| **unshift()** | **מוסיף איבר לתחילת המערך.** |
| **splice()** |  |

#### **🟣 פונקציות של מיון ותחזוקה:**

| **שם הפונקציה** | **הסבר** |
| --- | --- |
| **sort()** | **ממיין את המערך (לפי אלפבית/מספרים).** |
| **reverse()** | **הופך את סדר האיברים.** |
| **slice()** | **יוצר תת-מערך מהמערך המקורי.** |
| **concat()** | **מחבר שני מערכים יחד.** |
| **join()** | **מחבר את כל האיברים למחרוזת עם מפריד.** |
| **flat()** | **משטח מערכים מקוננים לרמה אחת.** |
| **fill()** | **דף עבודה** **1.** **צור מערך של מספרים.  מצא את כל המספרים הזוגיים בעזרת filter.** **2.** **צור מערך של מספרים.  הכפל כל מספר פי 3 בעזרת map.** **3.** **צור מערך של שמות.  הדפס כל שם למסך בעזרת forEach.** **4.** **צור מערך של מספרים.  מצא את המספר הראשון במערך שמתחלק ב־5 בעזרת find.** **5.** **צור מערך של מספרים.  בדוק אם יש לפחות מספר אחד שלילי במערך בעזרת some.** **6.** **צור מערך של מספרים.  בדוק אם כל המספרים במערך חיוביים בעזרת every.** **7.** **צור מערך של פירות: ["apple", "banana", "orange"]  מצא את האינדקס של "banana" בעזרת indexOf.** |