חשוב לקרוא , להבין ולתרגל את כל מרכיבי התוכנית לפי הסדר . יש קשר היררכי בין הסעיפים !

חשוב להדגיש שיש להקפיד ללמד על פי ההנחיות המתפרסמות בכל שנה !

- ללמד ולתרגל את הפעולה(פונקציה) Random בצורה בסיסית (בחירת מספרים בתחומים מסויימים)
- בהגדרת ושימוש בפעולות(פונקציות) – **יש ללמד** ולתרגל זימון לפונקציה ,הדפסת ערכים והחזרת ערך …return.
- ללמד ולתרגל שימוש בסיסי במחרוזות – כפי שיבואר בדוגמאות בהמשך .
- קיימות תוכניות להעשרה – כדי לאתגר את התלמידים ולאפשר להם לשלוט בחומר כראוי .

יעדי הלמידה והתרגול :

- להקנות ללומדים מושגי יסוד ומיומנויות בתרגול של הבסיס לשפת תכנות
- להקנות מושגים בסיסיים בחשיבה לוגית בצורה חוויתית בשילוב פעילויות תכנותיות במחשב.
- להקנות הרגלי עיצוב ותכנון, משמעת עצמית, בדיקה עצמית, עבודה כיחיד ובקבוצה, כישורי הפשטה, התמודדות עם סיבוכיות וחשיבה מודולרית.

פירוט נושאי התרגול לפי תוכנית הלימודים הקיימת והשעות המומלצות

מספר נושא	שם הנושא	שעות התנסות	שעות עיוניות	סה"כ שעות
1	**מבוא ומושגי יסוד במדעי המחשב / תכנות** • הגדרות מושגים : מחשב,תוכנית מחשב, נתונים והוראות • זיכרון המחשב – אוסף של תאים שלכל אחד ישנו שם(כתובת) ותוכן .(תוכן התא וכתובתו בשפה הבינארית) תא בזיכרון יכול להכיל נתון או הוראה . תא המכיל נתון – יכיל תו אחד מהמקלדת (לפי תווי ה ASCI) או ערך מספרי • תוכנית בשפה עילית – מול שפת מכונה . • קומפיילר מול אינטרפרטר • אלגוריתם (pseudo-code) : שפה הדומה במבנה שלה לשפת מחשב אך אינה צמודה לכללי דקדוק נוקשים כמו שפת תכנות פורמלית . אלגוריתם מילולי פשוט (מציאת הכדור הכבד מבין 9 כדורים על ידי 2 שקילות-מקסימום) • חומרה ותוכנה • אמצעי קלט ופלט • יחידות הקלט והפלט, מעבד, זיכרון, יחידת בקרה, יחידת החישובים(אריתמטית-לוגית) *אנאלוגיה של חיבור 2 מספרים :* *למשל : 3+6 - 6,3 הם הנתונים והסימן + היא ההוראה (אוסף נתונים והוראות= מידע או תוכנית מחשב) .*			

	הנתונים וההוראה עוברים מיחידת הקלט(המקלדת) *באמצעים אלקטרוניים לזיכרון .....* *מוקצים 3 תאי זיכרון : שניים לנתונים ואחד להוראה...*			
2	**הכרת השפה והכלים הבסיסיים** **להלן המלצה – אפשר בכל סביבה אחרת , למשל:** **visual studio code** • התקנת – Python רצוי גירסאות 3.X • שימוש בעורך – כדאי להתחיל ב notepad++ • היכרות עם Command line • סביבת העבודה והפיתוח PyCharm • קובץ תוכנית בפייתון מיוצג על ידי שם ואחריו .py : x.py **<u>כללי כתיבה בפייתון :</u>** • כדי לרשום הערה יש להשתמש בסולמית (#) .למשל: *קלוט מספר שלם #* *mis1=int(input("enter first number: "))* • חשוב להקפיד על הזחה כאשר כותבים מספר פקודות שיתבצעו יחד כבלוק (בשפות האחרות) – דוגמאות בהמשך.			
3	**מושגים בסיסיים במדעי המחשב –** • קבוע מספרי : מספר שלם (integer) או בקיצור int , מספר ממשי (float) • פעולות החשבון(אופרטורים) מספרים ממשיים(+ , - , * כפל , חילוק / , חזקה **) • פעולות החשבון מספרים שלמים(+ , - , *, חילוק //, שארית %) • הגדרת ביטוי חשבוני . • מספר חיובי= מספר הגדול מ 0 . שלילי =קטן מ 0 . 0 עצמו ניטרלי. • חשוב! לתרגל תרגילי המרה מביטויים במתמטיקה לייצוג ביטויים בשפות תכנות			
4	**הוראות קלט / פלט**. • **<u>קלט :</u> תמיד יקלט ערך מחרוזתי (טקסטואלי).** כדי לקבל ערך מספרי יש להמיר אותו לערך שלם או ממשי . למשל: mis1=int(input("enter first number: ")) תוצג הודעה למשתמש על גבי המסך – enter first number . הערך שיקלט יומר למספר שלם (int) והוא יוצב לתוך המשתנה mis1.			

- קליטת ערך מחרוזתי למשתנה str -

```
str = input("enter any string: ")
```

- **<u>פלט:</u>**

-הדפסת הודעות על המסך : print ("shalom")

-הדפסת ערכים של משתנים(יודפס הערך המופיע בתא x): print(x)

-הדפסת ערך(תוצאה) של ביטוי חשבוני : print(a+b)

אם למשל יוצב המספר 10 לתא a ולתא b יוצב 20 ....

```
F:\PYTHON>python
>>> a=10
>>> b=20
>>> print(a+b)
30
```

-שילוב הדפסת ערכי משתנים .

<u>למשל</u>- תוכנית הקולטת 2 מספרים שלמים ומדפיסה את סכומם בשלמים בצורת תרגיל ואת ממוצעם בערך **<u>ממשי</u>**.

שורות בהן מופיע הסימן # הינן הערות למתכנת.

```
קלוט מספר שלם ראשון #
mis1=int(input("enter first number: "))
קלוט מספר שלם שני #
mis2=int(input("enter second number: "))
sum=mis1+mis2
מתקבלת מנה בערך ממשי (אם רוצים ערך שלם- יש להשתמש בפעולת החילוק // ) #
avg=sum/2
print (mis1,"+",mis2,"=",sum)
print ("memutza: ",avg)
```

הפלט במסך יראה כך:

```
F:\PYTHON>python ex1.py
enter first number: 20
enter second number: 15
20 + 15 = 35
memutza:  17.5
```

מחרוזות (בסיס)

- אוסף של תווים המופיעים בין ' ' או " ".

5	לכל תו ישנו מיקום. התו הראשון במקום 0. האחרון במקום 1- למשל: המחרוזת 'tikshuv' הוצבה למשתנה המחרוזתי str: str='tikshuv', str , במקום הראשון במחרוזת str[0] יופיע התו s. דוגמא לתוכנית: str = **'megamat Tikshuv'** print('str = ', str) print('str[0] = ', str[0]) print('str[-1] = ', str[-1]) print('str[1:5] = ', str[1:5]) print('str[5:-2] = ', str[5:-2]) פלט במסך: F:\PYTHON>python str3.py str = megamat Tikshuv str[0] = m str[-1] = v str[1:5] = egam str[5:-2] = at Tiksh • הסימן + במחרוזות משמעותו: שירשור או צירוף. למשל: בתוכנית הבאה: להציב את התו a במשתנה x. את התו b במשתנה y ולהציב את השירשור שלהם במשתנה z. x="a" y="b" z=x+y print(z) הפלט במסך יהיה: ab			
6	**הוראת תנאי ,ביטויים ופעולות לוגיות (בוליאניים)** • מבנה הוראת התנאי – (בסוף הוראת התנאי חייבים להופיע:) **if** <הוראת התנאי (ביטוי יחס)>**:** הוראה או סידרת הוראות לביצוע כאשר התנאי מתקיים(בהזחה) **else:** הוראה או סידרת הוראות לביצוע כאשר התנאי **<u>לא מתקיים</u>**(בהזחה) • תווי היחס בתנאי: num1 == num2 (פעם אחת = זו הוראת הצבה) שיוון num1 > num2 גדול מ			

			num1 < num2 קטן מ
			num1 >= num2 גדול או שווה
			num1 <= num2 קטן או שווה
			num1 != num2 שונה
			
			דוגמא לתוכנית הקולטת מספר טבעי(שלם וחיובי) ומדפיסה את ההודעה אם הוא זוגי (odd) או אי-זוגי (Even):
			שימו לב להזחה של ההוראה המתבצעת אם התנאי מתקיים וכן אם אינו מתקיים !
			num=int(input("Enter int number: "))
			if (num%2==0):
			print("Even")
			else:
			print("Odd ")
			דוגמת הרצה במסך:
			F:\PYTHON>python zugi.py
			Enter int number: 15 נקלט המספר 15
			Odd
			
			F:\PYTHON>python zugi.py
			Enter int number: 8 נקלט המספר 8
			Even
			
			• הפקודה elif הינה צירוף של else if.
			
			• תנאים מורכבים : פעולה לוגית or, and , not
			דוגמא לתוכנית פשוטה(לא חכמה) הקולטת 3 מספרים ומדפיסה את ערכו של הגדול ביותר :
			
			num1=int(input("Enter first number: "))
			num2=int(input("Enter second number: "))
			num3=int(input("Enter first number: "))
			if(num1>num2) **and** (num1>num3) :
			print (num1, " is the bigest")
			elif (num2>num1) **and** (num2>num3):
			print (num2, " is the bigest")
			elif (num3>num1) and (num3>num2):
			print (num3, " is the bigest")
			else:
			print("the numbers are equal")
			
			דוגמת הרצה במסך :
			
			F:\PYTHON>python if1.py
			Enter first number: 3	

```
Enter second number: 2
Enter first number: 5
5    is the biggest

F:\PYTHON>python if1.py
Enter first number: 6
Enter second number: 4
Enter first number: 1
6    is the biggest

F:\PYTHON>python if1.py
Enter first number: 2
Enter second number: 8
Enter first number: 2
8    is the biggest

F:\PYTHON>python if1.py
Enter first number: 3
Enter second number: 3
Enter first number: 3
the numbers are equal
```

- תנאי במחרוזות מתייחס לבדיקת היחס לפי הסדר המילוני (בטבלת ה ascii)

למשל – הפקודה הבאה בודקת אם התו a נמצא לפי הסדר המילוני לפני התו c : יודפס על המסך - yes

```
if('a'<'c'):
        print ("yes")
```

```
if (s[0]==s[-1]):
```
אם התו הראשון במחרוזת s שווה לתו האחרון

מספר נושא	שם הנושא	שעות התנסות	שעות עיוניות	סה"כ שעות
7	**לולאות** - חשוב להדגיש מהי בקרת כניסה ויציאה מהלולאה. - ההבדל בין לולאת while ללולאת for. - לתרגל היטב פעולת מניה ופעולת צבירה . חשוב להקפיד לאפס או לאתחל את הערך של משתנים המופיעים בפעולות מניה וצבירה מונה למשל : i=i+1 או כתיבה מקוצרת : i+=1 צובר למשל : sum=sum+num או sum+=num **** נא לקרוא היטב את הדגש הדידקטי שמופיע בלולאת ה FOR** - להקפיד להשתמש בשמות משתנים משמעותיים ! • **לולאת While** **בלולאת while נשתמש אם מספר הפעמים שהלולאה אמורה להתבצע לא ידוע מראש .** מבנה תהליך העבודה בלולאה זו : 1) קביעת ערך התחלתי למשתנה הלולאה 2) **While <condition> :** <condition>הינו תנאי **הוראה 1** בקרת כניסה ללולאה **הוראה 2** אם התנאי מתקיים תתבצע **....** גוף הלולאה. אחרת- הלולאה מסתיימת 3) סידרת ההוראות לביצוע בתוך הלולאה (גוף הלולאה) שיופיעו מוזחים . 4) שינוי ערכו של משתנה הלולאה או קליטת ערך שיבדק בתנאי הלולאה (ב condition)			

דוגמא לתוכנית הקולטת ציון אחד חוקי (מספר בין 0 ל 100 ,כולל) .

התוכנית קולטת מספר שמציין ציון (mark) לפני הלולאה , בודקת אם הוא ציון **<u>לא חוקי</u>**(קטן מ 0 וגדול מ 100) . אם כן מציגה הודעה שאינו חוקי וקולטת מספר אחר במקומו. הלולאה מסתיימת כאשר יקלט מספר חוקי !

<u>שם התוכנית : while-mark.py</u>

```
mark=int(input("enter mark 0-100: "))          קליטת מספר בין 0 ל 100
while (mark<0) or (mark>100):                  בדיקה אם המספר מהווה ציון
    print ("invalid mark")                     גוף הלולאה מכיל 2 הוראות ולכן הן מוזחות.
    mark=int(input("enter mark 0-100: "))      קליטת ציון אחר
print("the mark is valid !")                   הודעה שהציון שנקלט חוקי כי הלולאה הסתיימה
```

בפלט ההרצה על המסך רואים שנקלטו 4 מספרים – עד שנקלט מספר חוקי .

```
F:\PYTHON>python while-mark.py
enter mark 0-100: -40
invalid mark
enter mark 0-100: 150
invalid mark
enter mark 0-100: 300
invalid mark
enter mark 0-100: 70
the mark is valid !
```

<u>המשך לולאות</u>

<u>לולאת while</u>

דוגמא לתוכנית המשתמשת בלולאת while ומשתנה בוליאני :

התוכנית תדפיס True לגביי כל מספר עוקב מ 0 עד 5 (לא כולל)

```
b = True        ← הסבר: מוגדר משתנה בוליאני b שמאותחל ל "אמת"
i = 0           ← המונה i שיוצר את המספרים העוקבים מאותחל ל 0
while b:        ← כל עוד ערך המשתנה b הוא true בצע את גוף הלולאה
    i=i+1       ← הוסף למונה i את המספר 1
    print(b)    ← הדפס את הערך הבוליאני הנוכחי
    if i == 5:  ← אם ערך המונה הוא 5 הצב במשתנה
        b = False ← הבוליאני b את הערך "שקר" (False)
print(b)        הדפס את תוכן המשתנה b בסיום הלולאה
```

פלט ההרצה במסך :

```
F:\PYTHON>python bool1.py
True
True
True
True
True
False
```

• לולאת for

בלולאת for נשתמש אם מספר הפעמים שהלולאה אמורה להתבצע ידוע מראש!!

מבנה פקודת הלולאה :

For <variable> in < range> :

<u>משמעות :</u> עבור משתנה הלולאה (variable) בצע את גוף הלולאה בתחום המוגדר (range) **לא כולל הערך הסופי ב range.** למשתנה הלולאה כברירת מחדל מתווסף הקבוע המספרי 1 .הוראות גוף הלולאה יופיעו לאחר שורת ה for בהזחה.

<u>דוגמה לתוכנית המדפיסה את כל המספרים העוקבים בתחום 0 עד 8</u>
(לא כולל 8) :

```
for i in range(8):
  print (i)
```

פלט על המסך: (פלט אנכי)

```
F:\PYTHON>python for1.py
0
1
2
3
4
5
6
7
```

התוכנית הבאה תבצע אותו הדבר אלא הפלט יופיע בשורה אחת עם התו פסיק(,) שיוצג בין המספרים. האפשרות end=","" גורמת לאחר כל ערך שיודפס להדפיס , .

```
for i in range(8):
   print (i, end =",")
```

הפלט :

```
F:\PYTHON>python for1.py
0,1,2,3,4,5,6,7,
```

לולאת for הפועלת על מחרוזת ....

התוכנית הבאה : המשתנה index מציין את מיקום התו הראשון .
בשורת ה for משתנה הלולאה הוא tav ותחום ערכי המשתנה הם 'acdef'
בתוך גוף לולאת ה for מופיעות 2 הוראות : הדפסה ומניה (index=index+1)
לפי הפלט שמתקבל ניתן להבין מה מבצעת הוראת ההדפסה .

```
index=0
for tav in 'acdef':
   print('at index {} tav is {} '.format(index,tav))
   index+=1
```

***הפקודה format נועדה לעצב את הפלט כפי הנראה בהרצה בהמשך. אין חובה ללמוד אותה (להעשרה בלבד – כדי להציג פלט בצורה ידידותית למשתמש . הסוגריים {} מתייחסות בהתאמה לערכים שיודפסו בתוך הפקודה format .**

פלט על המסך :

```
F:\PYTHON>python for_char1.py
at index 0 tav is a
at index 1 tav is c
at index 2 tav is d
at index 3 tav is e
at index 4 tav is f
```

המשך לולאות

- **לולאות for מקוננות : דוגמא ללוח הכפל 5X5**

```
for i in range(1,6):
    for j in range(1,6):
        print(i*j, end='  \t|  ')
    print()
    print('----------------------------------------')
```

גוף לולאה פנימית

for i….. – מציין את גוף הלולאה החיצונית
\t משמעותו להשתמש ב tab - כ 12 מקומות בשורה .

פלט המסך :

```
F:\PYTHON>python kefel.py
1   | 2  | 3  | 4  | 5  |
----------------------------------------
2   | 4  | 6  | 8  | 10 |
----------------------------------------
3   | 6  | 9  | 12 | 15 |
----------------------------------------
4   | 8  | 12 | 16 | 20 |
----------------------------------------
5   | 10 | 15 | 20 | 25 |
----------------------------------------
```

דגש דידקטי ◄ ◄ ◄

פעולות מניה וצבירה מבחינת המחשב (התוכנית) הינן פעולות הצבה לכל דבר . כאשר המחשב ניתקל בפעולת ההצבה(השמה) שמצויינת ע״י הסימן = , הוא מתחיל מצד ימין של סימן השיוויון ומציב את הערך של הביטוי החשבוני לתוך המשתנה המופיע מצד שמאל !

בפעולת מניה מוסיפים ערך של קבוע מספרי למשתנה . בדר״כ מתווסף למשתנה הערך 1 . תפקיד המונה – למנות כמה פעמים מתבצעת פעולה מסויימת . בתוכנית הבאה – כמה פעמים נקלט מספר שמתחלק ב 5 ללא שארית .

בפעולת צבירה מוסיפים ערך של משתנה אחר לצובר.
למשל : sum=sum+num . למשתנה sum מתווסף ערך של num ובכל פעם ב num יכול להיות מספר אחר(בשונה מפעולת המניה שתמיד מתווסף למונה אותו ערך מספרי)

חשוב להקפיד להציב ערך התחלתי (לאתחל) את המונה והצובר בתחילת התוכנית .
ככלל – אם אותו משתנה יופיע משני צידי סימן השיוויון(בהצבה) צריך תמיד לזכור לאתחל אותו ! ◄ ◄ ◄

- **תוכנית הקולטת 5 מספרים ומדפיסה כמה מבין המספרים שנקלטו (מונה m) מתחלקים ב 5 ללא שארית .**

```
m=0
for i in range (1,6):
    num =int(input("enter interger number:  "))
    if ( num  % 5 == 0 ):
        m=m+1
print(m)
```

דוגמת פלט :

```
F:\PYTHON>python div5.py
enter interger number:  34
enter interger number:  15
enter interger number:  20
enter interger number:  18
enter interger number:  100
3
```

- **תוכנית הקולטת 5 מספרים שלמים , מוצאת ומדפיסה את המספר הגדול ביותר (max) , את מיקומו מבין סדרת המספרים שנקלטים (המיקום של המספר הראשון הוא – 0 (המשתנה max_indx) וכן את המספר הקטן ביותר מבין המספרים שנקלטו (min) ואת מיקומו הסידורי של המספר הקטן ביותר מבין כלל המספרים שנקלטו (min_indx) .**

<u>הלוגיקה בכתיבת התוכנית הינה :</u>

תחילה נקלט המספר הראשון (מספר שלם,יכול להיות שלילי) ומוצב למשתנה max ולמשתנה min כי נכון לעכשיו המספר הראשון הוא גם הקטן ביותר וגם הגדול ביותר ומיקומו ברשימה – 1 .

לאחר מכן הלולאה קולטת עוד 4 מספרים :הראשון נקלט לפני הלולאה, לכן תחום הלולאה מתחיל ב 2 ומסתיים ב6 –לא כולל 6 : ערכי משתנה הלולאה i יהיו 2,3,4,5 .

בגוף הלולאה- עבור כל מספר שנקלט יבדק אם הוא גדול מהמספר הגדול ביותר עד עכשיו- max ואם המספר קטן מהמספר הקטן ביותר עד עכשיו min.אם כן הערך שנקלט יוצב למשתנים בהתאם .

```
max=int(input("enter the first number : "))
min=max
max_indx=1
min_indx=1
for i in range (2,6):
   num=int(input("enter the first number : "))
   if(num>max):
      max=num
      max_indx=i
   if(num<min):
      min=num
      min_indx=i
print ("the bigest numner is: ",max," in place- ",max_indx)
print ("the loewst numner is: ",min," in place- ",min_indx)
```

דוגמת הרצה למספרים שנקלטו (משמאל לימין) :

-10 , 30 , -55 , 68 ,120

המספר 120 הוא הגדול ביותר מבין המספרים שנקלטו והוא במיקום 5

המספר 55 - הוא הקטן ביותר מבין המספרים שנקלטו והוא במיקום 3

```
F:\PYTHON>python max_min_for.py
enter the first number : -10
enter the first number : 30
enter the first number : -55
enter the first number : 68
enter the first number : 120
the bigest numner is:  120  in place-  5
the loewst numner is:  -55  in place-  3

F:\PYTHON>
```

מספר נושא	שם הנושא	שעות התנסות	שעות עיוניות	סה"כ שעות
8	**פונקציות – נקראות גם פעולות .** • פונקציה(נקראת גם – פעולה) היא קטע תוכנית(קטע קוד) המכיל הוראות לביצוע משימה מוגדרת . • כל פונקציה תוגדר על ידי שם יחודי . • קריאה(זימון) לפונקציה יעשה על ידי כתיבת שם הפונקציה וסוגריים עגולים לאחריה . אם קיים פרמטר – הוא יופיע בתוך הסוגריים **קיימות פונקציות מובנות בשפה למשל :** - פונקציה המקבלת כפרמטר מחרוזת ומחזירה את מספר התווים במחרוזת : פונקצית len (length) לדוגמא : - str="tikshuv python" print (len(str)) במחרוזת str קיימים 14 תווים (תו הרווח נחשב כתו לכל דבר) לכן הפלט יהיה : 14 (להלן ההרצה) F:\PYTHON>python s6.py 14 **בתוכנית הבאה :** קולטים מחרוזת למשתנה str, קולטים תו אחד למשתנה char . בודקים בעזרת לולאת for שעוברת על כל התווים במחרוזת str החל מהתו הראשון במחרוזת(במקום 0) ועד לתו האחרון ובודקת ***<u>האם התו במיקום i במחרוזת str שווה לתו הבודד שנקלט ל char</u>*** . אם כן היא מונה (mone) כמה תווים כאלו קיימים. mone=0 str = input("enter any string: ") char = input("enter one character only : ") for i in range(0,len(str)): ***<u>if (str[i]= =char):</u>*** mone=mone+1 print (mone) **דוגמת הרצה :** F:\PYTHON>python str5.py enter any string: i am learning python in tikshuv enter one character only : i 4 **במחרוזת שנקלטה :** i am learning python in tikshuv **האות i מופיעה 4 פעמים במחרוזת str.** • פונקציות שפה נוספות : max() ,min() ,int() , float() ,char()			

המשך פונקציות :

פונקצית המספרים האקראיים :

מספרים אקראיים (Random) נוצרים(מוגרלים) על ידי המחשב כאשר מבקשים לייבא את המודול random באמצעות הפקודה :

import ranom. פונקציית השפה random.randint(1,8) מגרילה מספרים אקראיים **שלמים** בתחום 1 עד 8 – בדוגמא זו- כולל 1 וכולל 8

תוכנית המגרילה 15 מספרים אקראיים בין 1 ל 8 (כולל) ומדפיסה אותם בשורה אחת .

```
import random
for i in range (1,15):
   number = random.randint(1,8)
   print(number, end="," )
```

הרצה: התוכנית הורצה 4 פעמים ובכל פעם הוגרלו מספרים שונים בין 1 ל 8

```
C:\tmp>python random1.py
6,7,4,1,4,4,6,8,1,1,7,1,4,7,

C:\tmp>python random1.py
3,6,7,2,3,5,2,3,6,8,2,8,4,4,

C:\tmp>python random1.py
1,8,8,4,8,1,6,1,7,2,1,5,4,6,

C:\tmp>python random1.py
5,7,5,2,5,2,2,7,3,3,3,1,2,3,
```

פונקציות המוגדרות על ידי המשתמש :

- הפונקציה תוגדר על ידי המילה השמורה def . לאחריה תרשם שם הפונקציה . בסיום הגדרת הפונקציה יש לרשום נקודותיים(:) בדוגמה בתוכנית הבאה שם הפונקציה הינו add. הפונקציה מקבלת שני פרמטרים – x,y בתוך סוגריים(בהתחלה מאתחלים אותם ל 0. גוף הפונקציה מכיל 2 הוראות :

```
   sum=x+y
   return sum
```

מזמנים את הפונקציה בשורת ההדפסה :

```
print ("the sum is: ",add(num1,num2))
```

הערכים של שני המספרים שנקלטו מוצבים לתוך הפרמטרים בפונקציה . המשתנה num1 יוצב לפרמטר x , num2 יוצב ל y.

הפונקציה מחזירה את סכום המספרים – return sum למקום שבו היא זומנה .

```
def add(x=0,y=0):
   sum=x+y
   return sum
num1=int(input("enter first number: "))
num2=int(input("enter second number: "))
print ("the sum is: ",add(num1,num2))
```

הרצה :

```
F:\PYTHON>python fun-add.py
```

```
enter first number: 4
enter second number: 9
the sum is: 13
```

המשך פונקציות :

- הפקודה return מפסיקה את ביצוע התוכנית ומחזירה ערך .אם הפקודה return נמצאת בתוך לולאת for ולכן- ביצוע הלולאה יפסק ויוחזר הערך .

דוגמא :

מספר ראשוני הינו מספר שמתחלק רק בעצמו וב 1 . המספר 2 הינו היחיד שהוא גם ראשוני וגם זוגי .
הפונקציה prime (מספר ראשוני) מקבלת כפרמטר מספר (n) ומחזירה ערך בוליאני – אם הוא ראשוני יוחזר true , אחרת יוחזר false . בצורה הפשוטה ביותר – נבדוק אם המספר מתחלק לכל המספרים בין 2 עד למספר עצמו (לא כולל) . אם הוא מתחלק לאחד מהם תחזיר הפונקציה true וביצוע לולאת ה for יפסק.

- גוף הפונקציה כתוב בהזחה מתחת למילה השמורה def ומסומנת ברקע אפור !

```
def prime(n):
    if (n==1):
        return False
    elif (n==2):
        return True;
    else:
        for x in range(2,n):
            if(n % x==0):
                return False
        return True
print(prime(13))
```

הרצה : עבור הזימון למספר 13 – יוחזר true כי הוא ראשוני.

```
F:\PYTHON>python fun-rishoni.py
True
```

עבור הזימון למספר 84– יוחזר false כי הוא לא- ראשוני.

```
print(prime(84))
F:\PYTHON>python fun-rishoni.py
False
```

9

רשימות (lists)

- סידרה של ערכים המופיעים בתוך סוגריים מרובעות ,מופרדות על ידי פסיק(,) ויש להם שם אחד משותף . הערכים ברשימה יכולים להיות מכל סוג טיפוס(למשל: מספרים,מחרוזות וכדו') . האיבר הראשון ברשימה נמצא במקום 0.
(רשימה כמו מערך בשפות אחרות)

בתוכנית הבאה מספר פעולות חיוניות על גבי רשימה:
(זו רק דוגמא, ניתן ללמד גם פעולות כגון: ...in, not in, count)

מיקום אברי הרשימה ← 0 1 2 3 4 5

```
list=[1,4,7,2,5,10]                    שורה 1 - הגדרת רשימה
print(list[0:5])                       שורה 2 - הדפסת אברי הרשימה –לא כולל מקום 5
list.append(99)                        שורה 3 - הוספת המספר 99 בסוף הרשימה
print(list)                            שורה 4 –הדפסת הרשימה החדשה עם האיבר 99 שהתווסף לסוף
list[1]=33                             שורה 5 – הוספת הערך 33 לאחר המספר במיקום 1 (השני ברשימה)
print(list)                            שורה 6 –הדפסת אברי הרשימה לאחר הוספת 33
list.sort()                            שורה 7 – מיון אברי הרשימה בסדר עולה
print(list)                            שורה 8 - הדפסת הרשימה ממויינת בסדר עולה
print(list.index(7))                   שורה 9 – הדפס את מיקום האיבר ברשימה ששוה ל 7
list.remove(2)                         שורה 10 - להסיר מהרשימה את האיבר שערכו 2
print(list)                            שורה 11 -הדפסת הרשימה ללא המספר 2
list.pop(3)                            שורה 12-להוציא מהרשימה את המספר הנמצא במקום 3 שערכו הוא 10
print(list)                            שורה 13 -הדפסת הרשימה הנוכחית- ללא הערך 10
print ("orech reshima: ",len(list))    שורה 14-הדפסת מספר האברים ברשימה

for i in list:                         שורה 15 - הדפסת אברי הרשימה בשימוש לולאה
    print (i)
```

הרצת התוכנית:

```
F:\PYTHON>python l2.py
[1, 4, 7, 2, 5]                        פלט שורה 2 -
[1, 4, 7, 2, 5, 10, 99]                פלט שורה 4 -
[1, 33, 7, 2, 5, 10, 99]               פלט שורה 6 -
[1, 2, 5, 7, 10, 33, 99]               פלט שורה 8 -
3                                      פלט שורה 9 -
[1, 5, 7, 10, 33, 99]                  פלט שורה 11 -
[1, 5, 7, 33, 99]                      פלט שורה 13 -
orech reshima:  5                      פלט שורה 14 -
1                                      פלט שורה 15 -
5
7
33
99
```

			המשך רשימות (lists)
			• **תוכנית המדפיסה את כל האברים ברשימה ששוים בערכם למיקומם הסידורי ברשימה (האיבר הראשון במקום 0)**
			a=[0,1,3,5,6,3,7,8,2,9]
			print (a)
			for i in range (10):
			if (a.index(a[i])== a[i]):
			print (a[i])
			פלט הרצה :
			F:\PYTHON>python ldemo.py
			[0, 1, 3, 5, 6, 3, 7, 8, 2, 9]
			0
			1
			9	

10			**מילונים (Dictionaries) – ירד השנה**	
11			**רשומות (Tuples) - ירד השנה**	

12			**נספחים**
			רשימות- מיון : תרגילים להעשרה – למתקדמים
			<u>מיון בועות לא יעיל</u> : הפונקציה מקבלת רשימת מספרים טבעיים ומדפיסה רשימה ממויינת בסדר עולה (בלי flag)
			def bubblesort(L):
			for t in range(len(L)-1,0,-1):
			for i in range(t):
			if L[i]>L[i+1]:
			tmp=L[i]
			L[i]=L[i+1]
			L[i+1]=tmp	

			L=[5,7,10,15,16]
			bubblesort(L)
			print(L)
			
			<u>מיון בועות יעיל בעזרת "דגל":</u> הפונקציה מקבלת רשימת מספרים טבעיים ומדפיסה רשימה ממויינת בסדר עולה תוך שימוש ב flag .
			def bubblesort(L):
			  t = len(L)-1
			  flag='false'
			  while((t > 0) and (**flag=='false**')):
			  flag='true'
			  i=0
			  while (i < t):
			  if L[i]>L[i+1]:
			  tmp=L[i]
			  L[i]=L[i+1]
			  L[i+1]=tmp
			  flag='false'
			  i+=1
			  t-=1
			L=[5,7,10,1,6,30,40,50,100]
			bubblesort(L)
			print(L)	