

תרגיל 3: פונקציות

🔧 חלק א': שאלות פשוטות על פונקציות

♦ שאלה 1: ברכה אישית

כתוב פונקציה בשם `greet(name)` שמקבלת שם ומחזירה את ההודעה:
שלום! <name> ,

♦ שאלה 2: חישוב ריבוע

כתוב פונקציה `square(num)` שמחזירה את הריבוע של המספר.

♦ שאלה 3: סכום של שני מספרים

כתוב פונקציה `add(a, b)` שמחזירה את הסכום של שני מספרים.

♦ שאלה 4: כפולה של מספר

כתוב פונקציה `is_multiple_of_3(num)` שמחזירה `True` אם המספר מתחלק ב-3, אחרת `False`.

♦ שאלה 5: אורך של מחרוזת

כתוב פונקציה `string_length(text)` שמחזירה את מספר התווים במחרוזת.

♦ שאלה 6: הדפסת הודעה N פעמים

כתוב פונקציה `repeat_message(msg, n)` שמדפיסה את ההודעה `msg` בדיוק `n` פעמים.

♦ שאלה 7: המרה לצורת שאלה

כתוב פונקציה `to_question(text)` שמוסיפה סימן שאלה לסוף הטקסט, אם הוא לא כבר שם.

חלק ב': שאלות מתקדמות יותר 🧪

שאלה 8: ניתוח מחרוזת – ספירת מילים ותווים ✓

כתוב פונקציה בשם `analyze_text` שמקבלת מחרוזת ומחזירה טופל (tuple) עם שני ערכים:

- מספר המילים
- מספר התווים (ללא רווחים)

שאלה 9: בדיקת פלינדרום ✓

כתוב פונקציה `is_palindrome` שבודקת האם מחרוזת היא פלינדרום (מתעלמים מרווחים ואותיות רישיות).

שאלה 10: תדירות מילים בטקסט ✓

כתוב פונקציה `word_frequency` שמקבלת טקסט ומחזירה מילון של מילים ומספר הפעמים שכל אחת הופיעה.

שאלה 11: מילה הכי ארוכה ✓

כתוב פונקציה `longest_word` שמחזירה את המילה הכי ארוכה בטקסט.

שאלה 12: אורך ממוצע של מילה ✓

כתוב פונקציה `average_word_length` שמחזירה את האורך הממוצע של מילה בטקסט, מעוגל לשתי ספרות אחרי הנקודה.

Category	Function / Method	Description
String Methods	<code>split()</code>	Splits a string into a list of words based on whitespace.
	<code>replace(" ", "")</code>	Removes all spaces from the string.
	<code>lower()</code>	Converts all characters in the string to lowercase.
	<code>endswith("?")</code>	Checks if the string ends with a question mark.
	<code>join(iterable)</code>	Joins elements of an iterable into a single string.
	<code>isalpha()</code>	Checks if a character is an alphabetic letter (ignores punctuation/whitespace).
List Operations	<code>len(list)</code>	Returns the number of items in a list or the number of characters in a string.
	<code>max(list, key=len)</code>	Returns the longest item in the list (e.g., the longest word).
	List comprehension	E.g. <code>[c for c in text if c.isalpha()]</code> to filter only alphabetic characters.
Dictionary Operations	<code>dict.get(key, 0)</code>	Retrieves a value by key, or 0 if the key doesn't exist.
	<code>dict[key] = value</code>	Adds or updates a key-value pair in the dictionary.
Built-in Functions	<code>print()</code>	Prints output to the console.
	<code>round(number, 2)</code>	Rounds a number to 2 decimal places.
	<code>sum(iterable)</code>	Sums all the elements in an iterable (e.g., total word lengths).
	<code>range(n)</code>	Generates a sequence of numbers from 0 to n-1, typically used in loops.

פתרונות

```
# 1 שאלה
def greet(name):
    return f"שלום, {name}!"

# 2 שאלה
def square(num):
    return num ** 2

# 3 שאלה
def add(a, b):
    return a + b

# 4 שאלה
def is_multiple_of_3(num):
    return num % 3 == 0

# 5 שאלה
def string_length(text):
    return len(text)

# 6 שאלה
def repeat_message(msg, n):
    for _ in range(n):
        print(msg)

# 7 שאלה
def to_question(text):
    if not text.endswith("?"):
        return text + "?"
    return text
```

```
# 8 הלל
def analyze_text(text):
    word_count = len(text.split())
    char_count = len(text.replace(" ", ""))
    return word_count, char_count

# 9 הלל
def is_palindrome(text):
    cleaned = ''.join(text.lower().split())
    return cleaned == cleaned[::-1]

# 10 הלל
def word_frequency(text):
    words = text.lower().split()
    freq = {}
    for word in words:
        freq[word] = freq.get(word, 0) + 1
    return freq

# 11 הלל
def longest_word(text):
    words = text.split()
    return max(words, key=len)

# 12 הלל
def average_word_length(text):
    words = text.split()
    avg = sum(len(word) for word in words) / len(words)
    return round(avg, 2)
```
