

Python

TomerBu

נושאים להיום:

Functions

Args

Exceptions

Data Structures

משחק Higher Lower ש"ב:

```
from random import randint

rand_number = randint(1, 100)

while True:
    guess = int(input("Enter your guess:"))
    if guess < rand_number:
        print("HIGHER!")
    elif guess > rand_number:
        print("LOWER!")
    else:
        print("CORRECT!")
        break
```

משחק Higher Lower ש"ב:

```
from random import randint

rand_number = randint(1, 100)
print(f"psstt then number is: {rand_number}")

attempts = 0
win = False
while attempts < 5 and not win:
    guess = int(input("Enter your guess:"))
    attempts += 1
    if guess < rand_number:
        print("HIGHER!")
    elif guess > rand_number:
        print("LOWER!")
    else:
        print("CORRECT!")
        win = True

if win:
    print(f"Congratulations! You've guessed the number in {attempts} attempts.")
else:
    print(f"Sorry, you've used all your attempts. The number was {rand_number}.")
```

פונקציות

קטע קוד עם שם

מאפשר שימוש-חוזר-בקוד

Code Reuse!

פונקציות יכולות לקבל פרמטרים

פונקציות יכולות להחזיר ערך

פונקציות

```
def say_hello():  
    print("🌈")  
    print(f"Hello!")  
  
def say_goodbye():  
    print(f"Goodbye!")  
    print("🌈")
```

```
say_hello()  
say_hello()  
say_hello()  
say_hello()
```

קוד באנגלית!

כתבנו 4 פעמים באנגלית say_hello()
זוה בדיוק מה שהקוד עושה

פונקציות

```
def say_hello_to(name):  
    print("🌈")  
    print(f"Hello, {name}!")
```

```
say_hello_to("Alice")  
say_hello_to("Bob")  
say_hello_to("Joe")
```

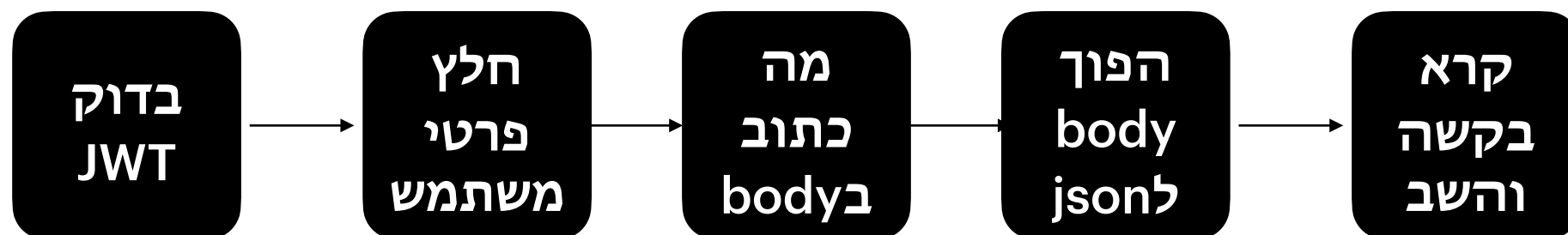
פונקציות עם פרמטרים:

פונקציות

```
def rating(rate):  
    return "★" * rate  
  
print(rating(5))
```

פונקציות שמחזירה ערך:

פונקציות כפס-ייצור
function chain:



בנק של פונקציות לשימוש חוזר
כתיבה באנגלית
כתיבה של תהליך שלם מיחידות מבנה

בנק של פונקציות

בנו את הפונקציות הבאות:

random lower case letter

הפונקציה לא מקבלת פרמטרים - ומחזירה

random upper case letter

הפונקציה לא מקבלת פרמטרים - ומחזירה

random digit

הפונקציה לא מקבלת פרמטרים - ומחזירה

random punctuation

הפונקציה לא מקבלת פרמטרים - ומחזירה

מאגר ידע - שמרו כל מה שלמדתם

לפעמים הבאות - כך לא תשכחו כלום

מי שסיים - אשמח שתכתבו בצ'אט - סיימתי / fi

אשמח גם שתשלחו פתרונות - אוכל לתת לכם משוב

אשמח שתשלחו פה פתרונות - יעזור לעוד סטודנטים שרוצים להשוות

בנק של פונקציות

```
from random import choice
from string import ascii_lowercase, ascii_uppercase, digits, punctuation

def random_lower():
    return choice(ascii_lowercase)

def random_upper():
    return choice(ascii_uppercase)

def random_digit():
    return choice(digits)

def random_punctuation():
    return choice(punctuation)
```

תרגול - כתיבת בנק פונקציות

קלט של מספר שלם
כולל בדיקות תקינות של הקלט
מאפשר השקעה בקוד - שיהיה איכותי
קוד שעובר בדיקות לאורך זמן
מקבל שיפורים

```
int(input(message))
```

קלט של מספר

1

קלט של מספר
כולל בדיקות תקינות

2

תרגול - כתיבת בנק פונקציות

פרמטר עם ערך ברירת מחדל



```
def get_int(message = "Enter a number: "):  
    return int(input(message))
```

1

```
age = get_int()  
age = get_int("whats your age: ")
```

2

תרגול - כתיבת בנק פונקציות

```
def get_int(message = "Enter a number: "):  
    while True:  
        user_input = input(message)  
  
        if user_input.isdigit():  
            return int(user_input)  
        print("try again")
```

תרגול - כתיבת בנק פונקציות

```
def get_int(message = "Enter a number: "):  
    while True:  
        user_input = input(message)  
  
        if user_input.isdigit() and user_input[0] != "0":  
            return int(user_input)  
        else:  
            print("Try again")  
  
get_int("whats your age: ")
```

תרגול - כתיבת בנק פונקציות

```
def get_boolean(message = "boolean True/False Yes/No: "):  
    while True:  
        user_input = input(message)  
  
        if user_input.lower() in ["yes", "yap", "yeah", "y", "true", "yy", "yep", "sure", "1"]:  
            return True  
        elif user_input.lower() in ["no", "nah", "n", "false", "nope", "0"]:  
            return False  
        print("try again")  
  
get_boolean("would you like to continue? yes or no: ")
```

מיני תרגיל:

קלטו מהמשתמש אות אחת בלבד באנגלית
אם המשתמש הזין 2 אותיות ויותר - try again

```
def get_boolean(message = "boolean True/False Yes/No: "):  
    while True:  
        user_input = input(message)  
  
        if user_input.lower() in ["yes", "yap", "yeah", "y", "true", "yy", "yep", "sure", "1"]:  
            return True  
        elif user_input.lower() in ["no", "nah", "n", "false", "nope", "0"]:  
            return False  
        print("try again")  
  
get_boolean("would you like to continue? yes or no: ")
```

כתבו פונקציה שמקבלת מספר פעמים num ומחזירה קו_תחתון כמספר הפעמים שביקשו כלומר num * "_"

```
def get_int(message = "Enter a number: "):  
    while True:  
        user_input = input(message)  
  
        if user_input.isdigit() and user_input[0] != "0":  
            return int(user_input)  
        else:  
            print("Try again")  
  
get_int("whats your age: ")
```


מיני תרגיל:

```
def get_letter(message="Enter a single letter: "):  
    while True:  
        user_input = input(message).strip()  
        if len(user_input) == 1 and user_input.isalpha():  
            return user_input  
  
    print("try again.")
```

כתבו פונקציה שמקבלת מספר פעמים num
ומחזירה קו_תחתון כמספר הפעמים שביקשו
כלומר num * "_"

הערות תיעוד:


```
/**  
  this is a description in javascript  
**/
```

```
def underscores(num):  
    """  
    Returns a list of underscores of length num.  
    If num is 5, it returns ["_", "_", "_", "_", "_"].  
    If num is 0, it returns an empty list.  
    If num is negative, it returns an empty list.  
    """  
  
    # return "_" * num  
    return ["_"] * num  
  
underscores()
```

(function) `def underscores(num: Any) -> Any`

Returns a list of underscores of length num. If num is 5, it returns ["_", "_", "_", "_", "_"]. If num is 0, it returns an empty list. If num is negative, it returns an empty list.

type annotations:



```
def underscores(num:int):  
    """  
    Returns a list of underscores of length num.  
    If num is 5, it returns ["_", "_", "_", "_", "_"].  
    If num is 0, it returns an empty list.  
    If num is negative, it returns an empty list.  
    """  
    # return "_" * num  
    return ["_"] * num  
  
underscores(2)
```



(function) `def underscores(num: int) -> list[str]`

Returns a list of underscores of length num. If num is 5, it returns ["_", "_", "_", "_", "_"]. If num is 0, it returns an empty list. If num is negative, it returns an empty list.

ריבוי ארגומנטים: args

הפונקציה מקבלת את words כאוסף של מילים
(אפשר לרוץ על המילים בלולאה)

```
def cold_print(*words):  
    for word in words:  
        print(f"❄️{word}❄️")
```

בקריאה לפונקציה הארגומנטים מועברים בלי [] או ()

```
cold_print("Elsa", "Anna")
```

ריבוי ארגומנטים: args

פונקציה שמקבלת רשימה:

```
def my_sum(numbers):  
    s = 0  
    for num in numbers:  
        s += num  
  
    return s  
  
my_sum([4, 3, 5, 1])
```

פונקציה שמקבלת args:

```
def my_sum(*numbers):  
    s = 0  
    for num in numbers:  
        s += num  
  
    return s  
  
my_sum(5, 1, 20, 3, 5)
```

מקרים שימושיים:

פונקציות לוג כגון
print, log

פונקציות כתיבה לקובץ

פונקציות מתמטיות כגון sum, min, max ועוד...

sum_hand(...card)

מיני תרגיל:

כתבו פונקציה שמקבלת args של מספרים:
הפונקציה תחזיר את הממוצע שלהם

```
def avg(*numbers):  
    return sum(numbers) / len(numbers)  
  
def asum(*numbers):  
    return sum(numbers)
```

Tuple = Immutable List

רשימה קבועה שאי אפשר לשנות בה איברים

מבנה נתונים - בדומה לרשימה
לאיברים יש אינדקס מתחיל באפס

```
my_list = [1, 2, 3]  
my_tuple = (1, 2, 3)
```

```
my_list = [1, 2, 3]  
my_tuple = (1, 2, 3)
```

סינטקס שונה ליצירה: ←

Tuple = Immutable List

רשימה קבועה שאי אפשר לשנות בה איברים

מבנה נתונים - בדומה לרשימה
לאיברים יש אינדקס מתחיל באפס

```
# Lists and Tuples
my_list = [1, 2, 3]
my_tuple = (1, 2, 3)

# Accessing Elements
print(my_list[0]) # Output: 1
print(my_tuple[0]) # Output: 1

# Modifying Elements
my_list[0] = 10
# my_tuple[0] = 10 # This will raise an error

# Length
print(len(my_list)) # Output: 3
print(len(my_tuple)) # Output: 3

# Iterating
for item in my_list:
    print(item)

for item in my_tuple:
    print(item)
```

שימוש:

*מי שקורא את הקוד מבין שהערך קבוע
*יעילות קבועים יותר מהירים

זאת המהות -

אחרי ההפסקה נראה דוגמאות לשימוש

:Tuples

```
numbers = (10, 20, 30)
```

```
numbers[0] = 200#TypeError: 'tuple' object does not support item assignment
```

```
def get_gas_prices():  
    return 8, 8.1, 9
```

```
print(get_gas_prices()[0])  
print(get_gas_prices()[1])  
print(get_gas_prices()[2])
```

```
def get_gas_prices():  
    return 8, 8.1, 9
```

```
gas = get_gas_prices()[0]  
diesel = get_gas_prices()[1]
```

Tuple de structuring

פירוק של Tuple

```
def get_gas_prices():  
    return 8, 8.1, 9
```

```
gas, diesel, electric = get_gas_prices()
```

```
t = get_gas_prices()  
print(t) # (8, 8.1, 9)
```

Tuple de structuring

פירוק של Tuple

```
def stats(*numbers):  
    """  
    Returns the minimum, maximum, and total of the given numbers.  
    Example: stats(1, 2, 3, 4, 5) returns (1, 5, 15)  
    min, max, total = stats(1, 2, 3, 4, 5)  
  
    print(stats(1, 2, 3, 4, 5)) # (1, 5, 15)  
    print(stats(1, 2, 3, 4, 5)[0]) # 1  
    """  
    return min(numbers), max(numbers), sum(numbers)
```

```
min, max, total = stats(1, 2, 3, 4, 5)
```

Data Structures

מבני נתונים:

מבנה נתונים מסודר לפי אינדקס
ניתן לעריכה/שינוי

list

מאוד יעיל לגשת לאיבר לפי אינדקס

מבנה נתונים מסודר לפי אינדקס
לא ניתן לעריכה/שינוי

tuple

מאוד יעיל לגשת לאיבר לפי אינדקס

מבנה נתונים לא ממוין ניתן לעריכה ומונע כפילויות!

set

מניעת כפילויות, פעולות על קבוצות

מבנה נתונים key-value pairs

אין חשיבות לסדר

מפתחות ייחודיים - ערך חדש יותר ידרוס את הקודם

dict

ייצוג של אובייקטים

מאוד יעיל לגשת לאיבר במילון לפי השם שלו

Data Structures

מבני נתונים:

set

מבנה נתונים לא ממוין ניתן לעריכה ומונע כפילויות!

מניעת כפילויות, פעולות על קבוצות

אין כפילויות בקבוצה/סט

```
winners = set()

winners.add("Alice")
winners.add("Bob")
winners.add("Alice")
winners.add("Alice")
winners.add("Alice")
winners.add("Alice")

print(winners) # {'Alice', 'Bob'}
```

Data Structures

מבני נתונים:

set

יש להגדיל 3 זוכים בהגרלה:

```
students = {"Mendy", "Eden", "Itay", "Yasmin", "Shani", "Shahar", "Eli", "Noam", "Tal", "Hila"}
```

```
from random import choice
```

```
students = {"Mendy", "Eden", "Itay", "Yasmin", "Shani", "Shahar", "Eli", "Noam", "Tal", "Hila"}
```

```
# Convert the set to a list
```

```
students = list(students)
```

```
winners = set()
```

```
while len(winners) < 3:
```

```
    winners.add(
```

```
        choice(students)
```

```
    )
```

```
print(winners)
```

כל עוד מספר האיברים קטן מ-3

נוסיף איבר רנדומלי לקבוצה.

סט - מונע כפילויות - ברגע שהגודל 3 סימן שכל האיברים שונים זה מזה.

Data Structures

מבני נתונים:

יצירת Set

```
# creating sets
```

```
s1 = set() #empty set
```

יצירת סט חדש וריק:

```
# set from items:
```

```
s2 = {1, 2, 3, 6}
```

יצירת סט מאיברים קיימים:

```
my_list = ['a', 'b', 'c']
```

```
# convert list to set
```

```
s3 = set(my_list)
```

המרת רשימה לסט:

```
# create a set from string:
```

```
s4 = set('hello') # {'h', 'e', 'l', 'o'}
```

יצירת סט ממחרוזת

Data Structures

מבני נתונים:

פעולות:

```
s = {1, 2, 3, 4, 5}
```

פעולות - הוספה, הסרה, קריאה

```
s.add(6) # added
```

```
s.add(6) # rejected
```

אין עדכון בקבוצה כי בקבוצה אין מקומות קבועים.

```
s.update({7, 8, 9})
```

```
print(s) # {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
if 7 in s:
```

```
    s.remove(7) # raises KeyError if item does not exist!
```

```
# remove if exists, does not raise error
```

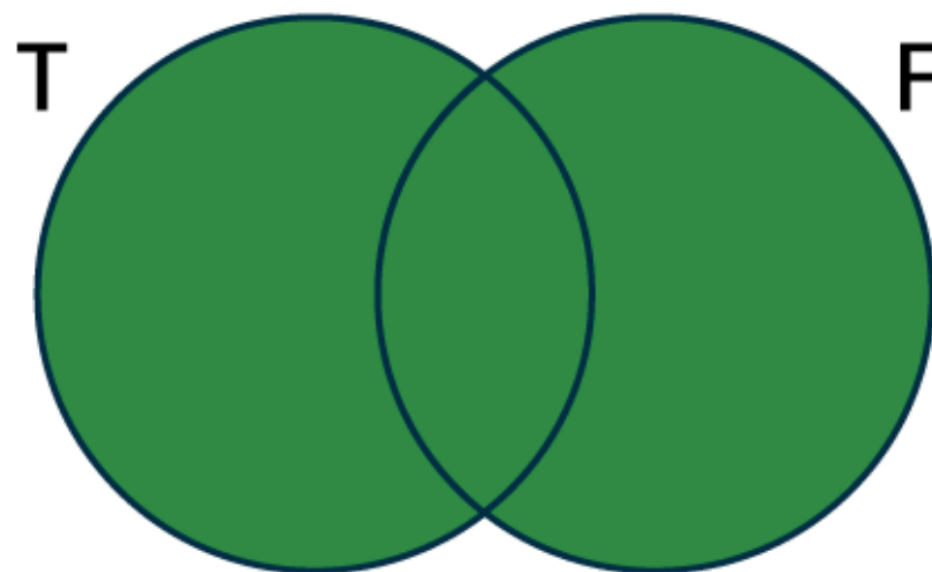
```
s.discard(8)
```


Data Structures

מבני נתונים:

פעולות על קבוצות:
איחוד

```
batman_actors = {'Christian Bale', 'Michael Keaton', 'Ben Affleck', 'Robert Pattinson'}  
american_actors = {'Christian Bale', 'Ben Affleck', 'Tom Hanks'}  
union = batman_actors.union(americans_actors)  
  
print(union)  
print(batman_actors | american_actors)
```



Union = $T \cup F$

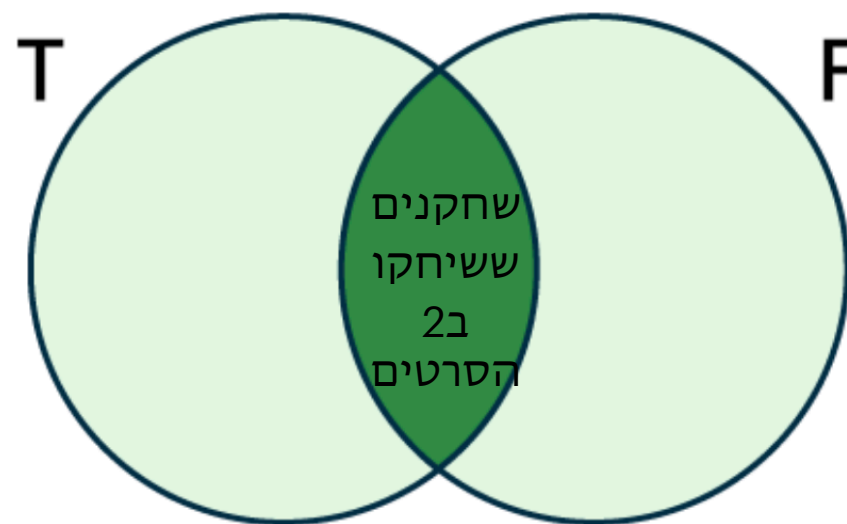
Data Structures

מבני נתונים:

פעולות על קבוצות:
חיתוך

שחקנים בבתמן

שחקנים באמריקן



$$\text{Intersection} = T \cap F$$

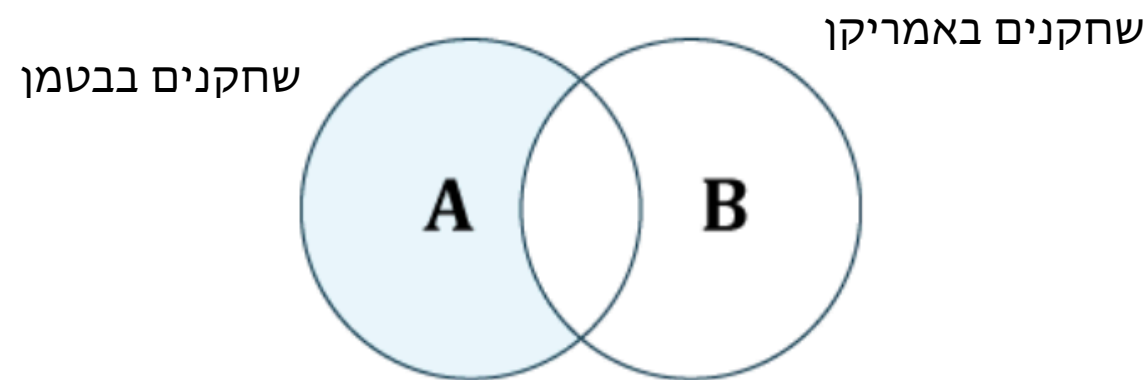
```
batman_actors = {'Christian Bale', 'Michael Keaton', 'Ben Affleck', 'Robert Pattinson'}  
american_actors = {'Christian Bale', 'Ben Affleck', 'Tom Hanks'}  
intersection = batman_actors.intersection(americian_actors)  
print(intersection) # {'Ben Affleck', 'Christian Bale'}
```

Data Structures

מבני נתונים:

פעולות על קבוצות:
הפרש

Difference



Notation :
A - B

שחקנים ששיחקו בבאטמן אבל לא באמריקן

```
batman_actors = {'Christian Bale', 'Michael Keaton', 'Ben Affleck', 'Robert Pattinson'}  
american_actors = {'Christian Bale', 'Ben Affleck', 'Tom Hanks'}  
difference = batman_actors.difference(americn_actors)  
print(difference) # Output: {'Michael Keaton', 'Robert Pattinson'}
```

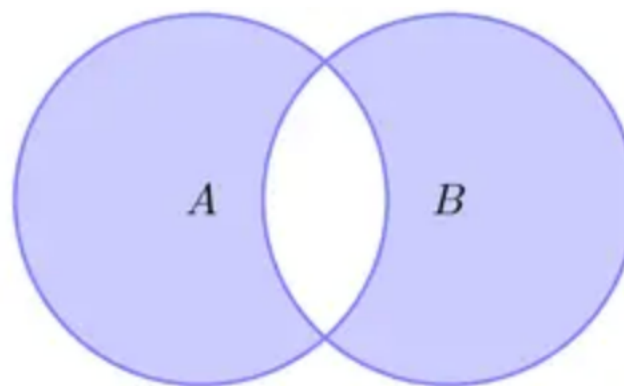
Data Structures

מבני נתונים:

פעולות על קבוצות:
הפרש סימטרי

שחקנים באמריקן

שחקנים בבטמן



שחקנים ששיחקו או בבטמן או באמריקן אך לא בשניהם

```
batman_actors = {'Christian Bale', 'Michael Keaton', 'Ben Affleck', 'Robert Pattinson'}  
american_actors = {'Christian Bale', 'Ben Affleck', 'Tom Hanks'}  
difference = batman_actors.symmetric_difference(americ
```

```
print(difference) # {'Robert Pattinson', 'Michael Keaton', 'Tom Hanks'}
```

Data Structures

מבני נתונים:

```
chat_people = ['Tamir', 'Nina', 'Raz', 'Tamir', 'Doron']

# convert list to set:
chat_set = set(chat_people)

# convert set to list:
chat_list = list(chat_set)

# one liner to remove duplicates
chat_people = list(set(chat_people))
```

Data Structures

מבני נתונים מילון:

ייצוג של אובייקטים

מאוד יעיל לגשת לאיבר במילון לפי השם שלו

מבנה נתונים key-value pairs

אין חשיבות לסדר

מפתחות ייחודיים - ערך חדש יותר ידרוס את הקודם

dict

חזיפוש ברשימה - n איטרציות:

ברשימה עם 4 איברים - 4 איטרציות לחיפוש

```
names = ['Christian Bale', 'Michael Keaton', 'Ben Affleck', 'Robert Pattinson']
```

ברשימה האינדקס הוא 0, 1, 2, (מספרי)

```
my_dict = {  
    'Christian Bale': 40,  
    'Michael Keaton': 31,  
    'Ben Affleck': 59,  
    'Robert Pattinson': 43  
}
```

חזיפוש במילון - 1 איטרציות:

במילון יש איטרציה אחת!

```
print(my_dict['Christian Bale'])
```

במילון האינדקס הוא המילה עצמה - ולכן יעילות מקסימלית של פעולה אחת לגישה לאיבר

שיעורי בית

(1) כתבו פונקציה שמקבלת מספר - אם המספר חיובי הפונקציה תחזיר True

אחרת הפונקציה תחזיר False

(2) כתבו פונקציה שמקבלת רשימה של מספרים - הפונקציה תחזיר את הסכום של כל האיברים ברשימה.

(3) כתבו פונקציה שמקבלת רשימה של מספרים - הפונקציה תחזיר את האיבר הקטן ביותר ברשימה.

(4) כתבו פונקציה שמקבלת רשימה של מספרים - הפונקציה תחזיר את האיבר הגדול ביותר ברשימה.

(5) כתבו פונקציה שמקבלת מילה ומדפיסה כל אות במילה פעמיים:

לדוגמא עבור המילה avi הפונקציה תדפיס aavvii

*5 הוסיפו פרמטר sep לפונקציה כדי שתדפיס את האותיות הכפולות עם מפריד:

לדוגמא: הפונקציה תדפיס aa-vv-ii עבור sep="-"

עוד דוגמא: הפונקציה תדפיס aa#vv#ii עבור sep="#"

(6) כתבו פונקציה שמקבלת מילה ומזיזה את כל האותיות אות אחת קדימה -

לדוגמא עבור הקלט ace הפונקציה תדפיס bdf

לדוגמא עבור הקלט abc הפונקציה תדפיס bcd

(7) כתבו פונקציה שמקבלת מספר ומחזירה True אם המספר הוא פלינדרום:

פלינדרום הוא מספר שאם נהפוך את הספרות שלו נקבל בדיוק את אותו המספר:

עבור 121 הפונקציה תחזיר True

עבור 252 תחזיר True

עבור 522 תחזיר False

שיעורי בית:

(8) כתבו פונקציה שמקבלת מספר שלם ומדפיסה את המספר השלם הבא.
(אם הפונקציה מקבלת את המספר 1 הפונקציה תדפיס 2)

(9) כתבו פונקציה המקבלת 2 רשימות של מספרים ומדפיסה את הרשימה שסכום המספרים בה הגדול מבין ה-2.
(10) כתבו תוכנית שקולטת 5 מספרים (לולאה).

עבור כל מספר שנקלט יש להדפיס את הספרות שמרכיבות את המספר בסדר הפוך.

דוגמא:

123 //input

321 //output

(10) הדפיסו את לוח הכפל בעזרת לולאות for

(11) צרו רשימה של רשימות שמכילה את האיברים של לוח הכפל

(12) הציגו את הslice של לוח הכפל מגודל 3*3

תרגילים:

"אנא הזן מספר בין 1 ל4 בלבד"

כתבו פונקציה שמקבלת קלט של מספר שלם:

הפונקציה תקבל 3 פרמטרים:

(1) הודעה לבקשת קלט

(2) מספר מינימום

(3) מספר מקסימום

ומתמודדת עם קלט לא תקין.

שיעורי בית:

עבור הקבוצות הבאות:

```
country_A = {"Python", "Machine Learning", "High-Speed Rail", "Startups", "Cloud Computing"}  
country_B = {"Python", "Cybersecurity", "High-Speed Rail", "Renewable Energy", "Blockchain"}
```

יש להדפיס:

- (1) את החיתוך בין 2 הקבוצות
- (2) את האיחוד בין 2 הקבוצות
- (3) את ההפרש בין 2 הקבוצות
- (4) את ההפרש הסימטרי בין 2 הקבוצות

שיעורי בית:

עבור הרשימה הבאה:

```
numbers = [1, 2, 3, 2, 4, 1, 5, 6, 6, 7]
```

יש להדפיס:

(1) את כל האיברים הייחודיים

(2) * את כל האיברים שמופיעים פעם אחת בלבד

(3) * עבור כל מספר יש לספור כמה פעמים הוא הופיע

ייתכן שימוש בלולאות

ש"ב: קריאה מומלצת

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set

צרו לעצמכם AI ביתי מותאם - מבית מטא

תודה לרוני על ההמלצה החמה!

<https://www.jjude.com/tech-notes/run-owui-on-mac/>

<https://gptforwork.com/help/ai-models/custom-endpoints/set-up-ollama-on-macos>

<https://ollama.com/search>