Python

TomerBu

נושאים להיום:

Lists
Loops
Slicing and indexing
List Comprehension
Functions

נתונים 3 מספרים:

איך ניתן למצוא מי המספר הקטן ביותר?

```
x = int(input("Enter a number"))
y = int(input("Enter a number"))
z = int(input("Enter a number"))


smallest = min(x, y, z)
```

נתונים 3 מספרים:

איך ניתן למצוא מי המספר הקטן ביותר?

```
x = int(input("Enter a number"))
y = int(input("Enter a number"))
z = int(input("Enter a number"))

lst = [x, y, z]

smallest = lst[0]

for num in lst:
    if num < smallest:
        smallest = num

print(smallest)
```

נתונים 3 מספרים:

איך ניתן למצוא מי המספר הקטן ביותר?

```
x = int(input("Enter a number"))
y = int(input("Enter a number"))
z = int(input("Enter a number"))

lst = [x, y, z]

smallest = lst[0]

for num in lst:
    if num < smallest:
        smallest = num

print(smallest)
```

החלפה בין ערכים:

קלאסי:

```
x = 400
y = 200

if x > y:
    #swap the values
    #x = y # x = 200, y = 200
    temp = x # temp = 400
    x = y # x = 200
    y = temp # y = 400
```

temp = 400

x = 200

y = 400

החלפה בין ערכים:

פתרון בפייתון:

```
x = 400
y = 200

if x > y:
    #swap the values
    x, y = y, x
```

נתונים 3 מספרים:

איך ניתן למיין אותם:

בלי להשתמש בפונקציה המובנית **sorted**

1. עם משפטי תנאי
2. עם לולאה
3. מצאו איך להשתמש בפונקציה המובנית **sorted**

נתונים 3 מספרים:

איך ניתן למיין אותם:

```
x = 400
y = 200
z = 100

if x > y:
    #swap the values
    x, y = y, x
if y > z:
    #swap the values
    y, z = z, y
if x > y:
    #swap the values
    x, y = y, x
```

x = 200, y = 400, z = 100

x = 200, y = 100, z = 400

x = 100, y = 200, z = 400

נתונים 3 מספרים:

איך ניתן למיין אותם:

```
lst = [500, 200, 100, 500, 300, 100]

for i in range(len(lst) - 1):
    for j in range(len(lst) - 1):
        if lst[j] > lst[j+1]:
            lst[j], lst[j+1] = lst[j+1], lst[j]


print(lst)
```

נתונים 3 מספרים:

איך ניתן למיין אותם:

```
lst = [500, 200, 100, 500, 300, 100]
```

פונקציה שיוצרת לנו רשימה חדשה וממוינת:

```
print(sorted(lst))  # [100, 100, 200, 300, 500, 500]
```

```
lst = [500, 200, 100, 500, 300, 100]
```

פונקציה שממיינת את הרשימה המקורית:

```
lst.sort()

print(lst)  # [100, 100, 200, 300, 500, 500]
```

עוד על תנאים בפייתון:

משפט תנאי מקוצר בפייתון
(לא קיים האופרטור ?:)

פייתון שומרת על קריאות וקוד מובן:

```
# קלוט ציון – אם הציון גדול מ56
# "תדפיס "עברת את המבחן
# "אחרת תדפיס "נכשלת במבחן


grade = int(input("Enter your grade: "))
print("Success" if grade > 56 else "Failure")
```

עוד פעולות על רשימות:
שימוש בIntelisense

```
# re
lst.
lst.
lst.
     append
     reverse
     pop
     extend
     insert
# up remove
lst[ copy
     count
# re index
prin sort
prin clear
     __add__
lst.sort()
```

```
def sort(
    *,
    key: None = None,
    reverse: bool = False
) -> None: ...

def sort(
    *,
    key: (int) -> Support
    reverse: bool = False
) -> None: ...
```

Sort the list in ascending order and return None.

עבודה עם Slices

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

print(fruits[0])  # apple
print(fruits[-1])  # mango

print(fruits[-2])  # kiwi
```

אפשר לעבוד עם אינדקס שלילי:
האיברים מהסוף להתחלה

גישה מאוד נוחה לאיבר האחרון:
במקום
fruits(len(fruits) - 1)

עבודה עם Slices

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

print(fruits[1:3] )  #1(inclusive) to 3(exclusive)
```

התחלה באינדקס 1 כולל

סיום לא כולל ה3

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]


# start defaults to 0
print(fruits[:3])

# end defaults to len(list)
print(fruits[2:])  # 2 to end
```

ב"מ של start הוא 0

ב"מ של end הוא len(lst)

עבודה עם Slices

```
# all the elements but the last #['apple', 'banana', 'cherry', 'kiwi']
print(fruits[:-1])
```

כל הרשימה חוץ מהאיבר האחרון

מה עושה הקוד הבא?

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

fruits2 = fruits
fruits2.clear()
print(fruits)
```

2 מצביעים לאותו אובייקט
2 מצביעים לאותה כתובת בזכרון

למה ליצור עותק?
כך לא נגרום לתופעות לוואי
מי שמשתמש בפונקציה לא רוצה שתעלם לו הבננה!

כשיוצרים עותק
אפשר לבצע כל פעולה על העותק
מבלי לגרום לתופעות לוואי

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

fruits2 = fruits.copy()
fruits2.remove("banana")
print(fruits)
print(fruits2)
```

שימוש בSlices ליצירת עותק:

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

fruits2 = fruits[:]  # copy the list
```

slice[start:end:step]

הפרמטר השלישי נקרא step

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# # [1, 3, 5, 7, 9]
print(numbers[::2]) # step 2
```

ברירת המחדל של step היא 1

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]


print(numbers[::-1]) # step -1
```

קפיצה שלילית:
ערך ב"מ של start הוא האיבר האחרון
ערך ב"מ של end הוא האיבר הראשון

אפשר לעבוד עם אינדקסים ומחרוזות:

```
phrase = "Monty Python!"
print(phrase[:-1]) #Monty Python

print(phrase[::-1]) #nohtyP ytnoM
```

איך נעתיק רק חלק מהאיברים לרשימה חדשה:

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

# copy all the fruits but the banana
fruits_copy = fruits[:1] + fruits[2:]

# copy all the fruits but the banana
fruits_copy = []
for fruit in fruits:
    if fruit != "banana":
        fruits_copy.append(fruit)
```

נתונה רשימה עם מספרים:
רוצים להעתיק את המספרים שגדולים מ56:

או שרוצים להעתיק רק את המספרים הזוגיים:

```
from random import randint
random_numbers = [randint(1, 100), randint(1, 100), randint(1, 100), randint(1, 100)]

even_numbers = []
for number in random_numbers:
    if number % 2 == 0:
        even_numbers.append(number)
```

List Comprehension

```
from random import randint
random_numbers = [randint(1, 100), randint(1, 100), randint(1, 100), randint(1, 100)]


even_numbers = [num for num in random_numbers if num % 2 == 0]
```

האופרטור in והאופרטור not in

```
#in ניתן לבדוק האם מחרוזת מכילה מחרוזת נתונה באמצעות האופרטור
#not in ניתן לבדוק האם מחרוזת מכילה מחרוזת נתונה באמצעות האופרטור


print("a" in "apple")  # True
print("b" in "apple")  # False

print("banana" not in "apple pie")  # True


# also works with lists:
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
print("apple" in ["apple", "banana", "cherry"])  # True
print("apple" in fruits)  # True
```

מיני תרגיל:

נתונה הרשימה הבאה:

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
```

יש ליצור רשימה חדשה עם כל הפירות שמכילים את האות a
יש להשתמש בlist comprehension

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

fruits_copy = [fruit for fruit in fruits if "a" in fruit]
print(fruits_copy)  # ['apple', 'banana', 'mango']
```

מיני תרגיל:

משחק איש תלוי:

אפשר להגדיר בנק מילים ולהגדיר מילה רנדומלית.

להציג למשתמש _____

```
from random import choice

fruits = ["apple", "banana", "cherry", "kiwi", "mango", "orange", "pear",
"pineapple", "strawberry", "watermelon"]

# random fruit:
random_word = choice(fruits)

underscores = ["_" for letter in random_word]
print(underscores)
```

```
underscores = ["_" for _ in random_word]
print(underscores)
```

כשלא משתמשים במשתנה
לא ניתן לו שם
כך יובהר למתכנתים בצוות שלא עשינו שימוש במשתנה

מיני תרגיל:

האופרטור כפל עובד גם על מחרוזות :)

```
print("🍏"*3) #🍏🍏🍏
```

```
print("🍏"*len(random_word))
```

המרת מחרוזת לרשימה של אותיות:

```
list("abcdefg") # ['a', b', 'c', 'd', 'e', 'f', 'g']
```

```
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

# random fruit:
random_word = choice(fruits)


apples = list("🍏"*len(random_word))
print(apples) #['🍏', '🍏', '🍏', '🍏', '🍏', '🍏', '🍏', '🍏', '🍏']
```

מיני תרגיל:

איש תלוי

1 הגרילו מילה מתוך רשימת המילים הבאות:
["dog", "cat" "elephant", "home"]

2 יש ליצור רשימה עם מקפים במקום האותיות של המילה שהוגרלה
לדוגמא - עבור dog
יש ליצור רשימה שתכיל ["_" ,"_" ,"_"]

3 קלטו מהמשתמש אות:
בעזרת לולאת)for i in range
עברו על הרשימה עם המקפים
אם המשתמש ניחש אות נכונה - להחליף את המקף.

לדוגמא:
אם המשתמש ניחש את האות o במילה dog
הרשימה המעודכנת תהיה _o_

4 לולאה שתריץ את כל זה כל עוד יש קו תחתון

מיני תרגיל:

```
from random import choice

# faker library
word_list = ["dog", "cat", "elephant", "home"]

chosen_word = choice(word_list)                    1

#testing code
print(f'Pssst, the solution is {chosen_word}.')

display = ["_" for _ in chosen_word]               2

#show the display
print(display)

guess = input("Guess a letter: ").lower()


if guess not in chosen_word:                       3
    print("Wrong")
else:
    # update the display
    for i in range(len(chosen_word)):
        if guess == chosen_word[i]:
            display[i] = guess

print(display)
```

מיני תרגיל:

```
from random import choice

# faker library
word_list = ["dog", "cat", "elephant", "home"]

chosen_word = choice(word_list)

#testing code
print(f'Pssst, the solution is {chosen_word}.')

display = ["_" for _ in chosen_word]

#show the display
print(display)

while "_" in display:
    guess = input("Guess a letter: ").lower()

    if guess not in chosen_word:
        print("Wrong")
    else:
        # update the display
        for i in range(len(chosen_word)):
            if guess == chosen_word[i]:
                display[i] = guess

    print(display)
```

1

2

3

4

מיני תרגיל:

1 צרו מודול עם רשימה של מצבים למשחק איש תלוי

https://gist.github.com/chrishorton/8510732aa9a80a03c829b09f12e20d9c

וייבאו את התמונות למשחק

2 הוסיפו משתנה חדש בשם lives שערכו יהיה שווה לכמות התמונות שיש לכם

כשאתם מציגים את הDisplay - כדאי להציג את התמונה המתאימה לפי המשתנה lives

3 בכל פעם שהמשתמש עונה לא נכון להפחית מהערך של המשתנה lives

4 עדכנו את הלולאה שתרוץ כל עוד יש למשתמש חיים וגם יש underscore בdisplay

5 מחוץ ללולאה - בדקו האם למשתמש נותרו חיים - אם כן הוא ניצח. אחרת הפסיד.

פונקציות:

קטע קוד עם שם - שניתן לעשות בו **שימוש חוזר.**
פונקציות יכולות לקבל פרמטרים.
פונקציות יכולות להחזיר ערך.

```
def say_hello():
    print("Hello!")


say_hello()
```

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


say_hello('Jerry')
```

פונקציות:

קטע קוד עם שם - שניתן לעשות בו **שימוש חוזר.**
פונקציות יכולות לקבל פרמטרים.
פונקציות יכולות להחזיר ערך.

```
def add(x, y):
    return x + y

result = add(3, 4) # 7

print(result)
```

בנק של פונקציות:

mio.py U ×

ממשו את הפונקציות הבאות:

mio.py

```
1  # random lower case letter
2  # random upper case letter
3  # random digit
4  # random punctuation
```

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

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)
```

בנק של פונקציות:

לולאת קלט:

```
def get_letter(message = "Enter a single letter: "):
    #loop until we get a single letter
    while True:
        char = input(message).strip()
        if len(char) == 1 and char.isalpha():
            return char
        print("Invalid input, please try again")

letter = get_letter("enter your guess:")
print(letter)
```

כדי לבדוק שהקלט תקין - נריץ את הקוד שלנו בלולאה:
הלולאה תרוץ לפחות פעם אחת.

בגוף הלולאה נבדוק - בעזרת משפט תנאי
אם הקלט תקין (עומד בדרישות שלנו)

אם הקלט תקין - נחזיר ערך - והלולאה תעצר (return עוצר את הפונקציה ואת הלולאה)
אם הקלט לא תקין - נדפיס הודעה "קלט לא תקין" והלולאה תחזור על עצמה עד שהתנאי יתקיים.

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

שיעורי בית:

ממשו את הפונקציה הבאה
לפי ההנחיות שהסברנו בשיעור:

```
def get_bool(message = "yes no please:"):
```

בצעו קלט:
אם הקלט הוא y, yes, true, 1, yep, yap, yeah
הפונקציה תחזיר true

אם הקלט הוא n, no, false, 0, nope, nay, nah
הפונקציה תחזיר false

אחרת - הזנת קלט לא תקין - נסה שוב

רמזים:

המירו את הקלט לlower case
לפני ההשוואה - ובצעו strip

if kelet in ["y", "yes"]

שאלת בונוס:
פונקציה לקלט של מספר שלם (try)
פונקציה לקלט של מספר עשרוני

הפונקציות יכולות לקבל טווח אופציונלי

שיעורי בית: