

# Python

TomerBu

# **נושאים להיום:**

Lists

Loops

Slicing And Indexing

List Comprehension

Functions

# הגרלה של איבר רנדומלי מרשימה:

practice.py

```
from random import choice  
  
my_list = ['a', 'b', 'c', 'd']  
  
random_letter = choice(my_list)  
  
print(random_letter)
```

practice.py

**python3 practice.py**

# פעולות על רשימות:

practice.py

```
my_list = [1, 2, 3, 4]
```

הוספת איברים:

```
# add items to the end of the list:  
my_list += [5]  
my_list.append(6)
```

לסוף הרשימה:

```
# insert items at a specific position:  
my_list.insert(2, 2.5)
```

למקום ספציפי ברשימה:

```
print(my_list) # [1, 2, 2.5, 3, 4, 5, 6]
```

practice.py

```
python3 practice.py
```

# פעולות על רשימות:

practice.py

```
my_list = ['apple', 'banana', 'fig']
# CRUD = create, Read, update, delete
```

#Read

```
print(my_list)
print(my_list[0]) # apple
```

קריאה מרשימה:

```
# minus 1 from the end
print(my_list[-1]) # fig
```

```
print(my_list[1:3]) # ['banana', 'fig']
```

דבר על slices בהמשך

practice.py

```
python3 practice.py
```

# פעולות על רשימות:

practice.py

```
my_list = ['A', 'B', 'C']
```

קריאה מרשימה:

```
my_list.pop() # removes the last item
```

מחיקת האיבר האחרון:

```
print(my_list) # ['A', 'B']
```

דבר על slices בהמשך

practice.py

```
python3 practice.py
```

# פעולות על רשימות:

practice.py

```
my_list = ['A', 'B', 'C']

next = my_list.pop() # use and remove the last item

print(f"Doctor sees {next}") # C

print(my_list) # ['A', 'B']
```

שמירת האיבר האחרון שהוסר במשתנה:

practice.py

**python3 practice.py**

# פעולות על רשימות:

practice.py

```
my_list = ['A', 'B', 'C']

next = my_list.pop(1)      הסרת איבר באינדקס מסויים:

print(my_list)  # ['A', 'C']
print(next)     # 'B'

#del my_list[0]  # delete the first item
```

practice.py

```
python3 practice.py
```

# פעולות על רשימות:

practice.py

```
# crUd:  
  
# Update:  
  
my_list = ['👑', '💍']  
  
my_list[0] = '👑👑' # update the first item
```

הסרת איבר באינדקס מסוים:

practice.py

**python3 practice.py**

# לולאות

```
fruit = [🍏, 🍌, 🍒, 🍉, 🍇, 🍓, 🍍, 🍎, 🍊, 🍋, 🍑,
```

```
# for Each loop:  
for f in fruit:  
    print(f)
```

קוד שרך 3 פעמים:

```
# loop 3 times:  
# i = 0, 1, 2  
for i in range(3):  
    print(f'{i} *')
```

: אינטראקציית  
0, 1, 2

# לולאות

```
# print all the even numbers from 0 to 10
for i in range(11):
    if i % 2 == 0:
        print(i)
```

```
for i in range(0, 11, 2):
    print(i)      ↑      ↑       ←
                  start   stop    step
```

```
# print the odd numbers:
for i in range(1, 11, 2):
    print(i)
```

מוסכמת שמות - אם לא משתמשים באינדקס - קו תחתון

```
# code that runs 3 times:
for _ in range(3):
    print('hello')
```

# לולאות

```
# random password generator:

letter_count = int(input("How many letters: "))

use_lower = input('Use lower case letters y/n: ')
use_upper = input('Use upper case letters y/n: ')
use_digits = input('Use digits y/n: ')
use_punctuation = input('Use punctuation y/n: ')

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

ascii_bank = ''
if use_lower == 'y':
    ascii_bank += ascii_lowercase

if use_upper == 'y':
    ascii_bank += ascii_uppercase

if use_digits == 'y':
    ascii_bank += digits

if use_punctuation == 'y':
    ascii_bank += punctuation

print(ascii_bank) #abcdefghijklmnopqrstuvwxyz0123456789!"#$%&'()*+,-./:;=>?@[\\]^_`{|}~
```

קלטים:

- 1) אורך הסיסמה המבוקש
- 2) האם להשתמש באותיות גדולות
- 3) האם להשתמש באותיות קטנות
- 4) האם להשתמש בספרות
- 5) האם להשתמש בתווים מיוחדים

# לולאות

```
'no' in ['y', 'yes', 'da', 'si', 'true', 'fi', 'כן', 'oui']
```

קליטים:

- 1 אורך הסיסמה המבוקש
- 2 האם לשתמש באותיות גדולות
- 3 האם לשתמש באותיות קטנות
- 4 האם לשתמש בספרות
- 5 האם לשתמש בתווים מיוחדים

# מחולל סיסמות:

```
# random password generator:

letter_count = int(input("How many letters: "))

use_lower = input('Use lower case letters y/n: ')
use_upper = input('Use upper case letters y/n: ')
use_digits = input('Use digits y/n: ')
use_punctuation = input('Use punctuation y/n: ')

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

ascii_bank = ''
if use_lower == 'y':
    ascii_bank += ascii_lowercase ←———— מומלץ להוסיף כבר כאן אות קטנה לסיסמה

if use_upper == 'y':
    ascii_bank += ascii_uppercase

if use_digits == 'y':
    ascii_bank += digits

if use_punctuation == 'y':
    ascii_bank += punctuation

if not ascii_bank:
    print('You must select at least one character type.')
    exit()

password = ''

for _ in range(letter_count):
    password += choice(ascii_bank)

print(password)
```

# נתונים 3 מספרים: איך נמצא את הקטן מביניהם?

```
from random import randint

x = randint(1, 100)
y = randint(1, 100)
z = randint(1, 100)

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

תרגיל:

עליכם לדאוג שא יכיל את המספר הקטן ביותר  
עליכם לדאוג שז יכיל את המספר האמצעי  
עליכם לדאוג שז יכיל את המספר הגדול ביותר

רמז לפתרון אחד: משפטי תנאי  
רמז לפתרון שני: מethodות של רשימה

## **נתונים 3 מספרים: איך נמצא את הקטן מביניהם?**

```
from random import randint

x = randint(1, 100)
y = randint(1, 100)
z = randint(1, 100)

lst = [x, y, z]

lst.sort()

x = lst[0]
y = lst[1]
z = lst[2]
```

# נתונים 3 מספרים: איך נמצא את הקטן מביניהם?

```
x = randint(0, 400) # 400  
y = randint(0, 400) # 200  
z = randint(0, 400) # 100
```

```
if x > y:  
    x, y = y, x
```

```
if y > z:  
    y, z = z, y
```

```
if x > y:  
    x, y = y, x
```

x = 400  
y = 200  
z = 100

x = 200  
y = 400  
z = 100

x = 200  
y = 100  
z = 400

x = 100  
y = 200  
z = 400

כרגע נחלף בין 2 איברים

# עוד על משפטים תנאיים

```
# קלוט ציון – אם הציון גדול מ 56  
# תדפיס "עברת את המבחן"  
# אחרת תדפיס "נכשלת ב מבחן"
```

```
grade = int(input("Enter your grade: "))  
  
if grade > 56:  
    print("עברת את המבחן")  
else:  
    print("לא עברת את המבחן")  
  
# short if/else syntax:  
  
print("לא עברת את המבחן" if grade > 56 else "עברת את המבחן")
```

# Slicing And Indexing

קוד בג'אווה סקירה פט:  
"hello".slice(0, 2)

```
fruit = ['apple', 'mango', 'banana', 'kiwi']
```

```
# ['mango', 'banana']
print(fruit[1:3]) #1(inclusive) to 3(exclusive)
```

start

התחלת באינדקס כולל

stop

סיום לא כולל

```
fruit = ['apple', 'mango', 'banana', 'kiwi', 'orange', 'grape']
```

```
# start defaults to 0
# first 2 elements
print(fruit[:2])
```

start

ברירת מחדל 0

```
# stop defaults to the len(lst)
# items from index 2 to the end
print(fruit[2:])
```

stop

ברירת מחדל len(lst)

# Slicing And Indexing

קוד בג'אווה סקירה פט:  
"hello".slice(0, 2)

```
fruit = ['apple', 'mango', 'banana', 'kiwi']
```

```
# ['mango', 'banana']
print(fruit[1:3]) #1(inclusive) to 3(exclusive)
```

start

התחלת באינדקס כולל

stop

סיום לא כולל

```
fruit = ['apple', 'mango', 'banana', 'kiwi', 'orange', 'grape']
```

```
# start defaults to 0
# first 2 elements
print(fruit[:2])
```

start

ברירת מחדל 0

```
# stop defaults to the len(lst)
# items from index 2 to the end
print(fruit[2:])
```

stop

ברירת מחדל len(lst)

# Slicing And Indexing

```
# all the items but the last one:  
print(fruit[:-1])
```

כל האיברים חוץ מה אחרונים:

```
numbers = [1, 2, 3, 50, 300, 5]
```

העתקה של רשימות:

```
#shallow copy  
numbers2 = numbers[:]
```

```
#shallow copy  
numbers2 = numbers.copy()
```

# Slicing And Indexing

```
numbers = [1, 2, 3, 50, 300, 5]
```

```
# start, stop, step  
# [1, 3, 300]  
print(numbers[::-2])
```

הקפיצות באינדקס

step

כל האיברים באינדקסים הזוגיים:

[1, 3, 300]

## סדר הפוך:

```
phrase = "Monty python!"  
letters = ['a', 'b', 'c']
```

```
# ['c', 'b', 'a']  
print(letters[::-1])
```

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

```
print("תומך הלאן לגן[::-1])
```

# List Comprehension

## בנייה של רשימה בתחריב ייעודי

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

רוצים רשימה שכל איבר בה הוא ריבוע:

לולאה שרצה 10 פעמים יש לי ? מאפס עד תשע  
נעלה את ? בריבוע ונכנס לרשימה

```
# list comprehension
"""
list of squares:
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
"""

squares = []
for i in range(10):
    squares.append(x**2)

print(squares)
```

<https://docs.python.org/3/tutorial/datastructures.html>

# List Comprehension

## בנייה של רשימה בתחריב ייעודי

```
# numbers = [i for i in range(10)]  
numbers = []  
  
for i in range(10):  
    numbers.append(i)  
  
print(numbers)
```

```
print([i for i in range(10)])
```

```
print([i**2 for i in range(10)])
```

# List Comprehension

## בנייה של רשימה בתחריב ייעודי

```
from random import randint  
  
random_numbers = [randint(0, 100), randint(0, 100), randint(0, 100), randint(0, 100)]
```

רשימה עם 4 מספרים כלשהם:

העתקה של רשימה בעזרת list comprehension

```
copy = [num for num in random_numbers]
```

filtracija של רשימה בעזרת list comprehension

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

# האופרטור `in` ו-`not in`

```
print("a" in "banana")
```

אם האות a נמצאת ברשימה

מילה - רשימה של תווים

אם המספר נמצא ברשימה:

```
if 3 in [1, 2, 3]:  
    print("Found 3!")
```

```
if '7' in '1012389123':  
    print("BOOM")  
else:  
    print(num)
```

# האופרטור in ו-not in

```
# char in string:  
print("a" in "apple") # True  
print("b" in "apple") # False  
  
# 'substring' in 'string'  
print("banana" not in "apple pie") # True  
  
# element in Iterable  
print (1 in [2, 3, 4, 1])
```

דוגמאות לשימוש:

בדיקות של ייחודיות - אם האות שאני מנסה להכניס כבר קיימת - נגריל אותה חדשה.

'123901' in '7'

חיפוש:

## אחרי הפסיקת - איש תלוי ומבוא לפונקציות

```
my_str = 'a'*10 # 'aaaaaaaaaa'
```

```
l = len("hello") #l = 5
```

```
l = len([2, 3, 4]) #l = 3
```

```
lst = list("hello") # ['h', 'e', 'l', 'l', 'o']
```

```
print("🍏" * 10) # 🍏🍏🍏🍏🍏🍏🍏🍏🍏🍏
```

```
print("_" * len("hello")) # _ _ _ _ _
```

```
lst = list("hello") # ['h', 'e', 'l', 'l', 'o']
```

```
lst = list("_" * 10)
```

```
print(lst) # ['_', '_', '_', '_', '_', '_', '_', '_', '_', '_']
```

# משחק איש תלוי:

cow

יש להגריל מילה אחת רנדומלית: יש להשתמש בפונקציה choice

1

[\_,o,\_]

יש ליצור רשימה עם מקפים במקום האותיות במילה הסודית

2

יש להדפיס למשתמש קווים תחתיוים כמנין האותיות \_ \_ \_

3

בלולאת while כל עוד לא ניחסו את כל המילה (כלומר יש עדין קו-תחtooן ברשימה שיצרנו בשלב 2) קלטו מהמשתמש ניחוש - אם האות קיימת במילה - מלאו את התו ברשימה שיצרתם בסעיף 2 אחרית הדפסו - טעית. ובכל מקרה - הלוואה תמשיך לסייע הבא.

4

# שיעור בית:

בלולה על כל המספרים בין 1 ל-100 - יש להדפיס:

## fizz buzz

אם המספר מתחלק ב-3 fizz

אם המספר מתחלק ב-5 buzz

אם המספר מתחלק ב-15 fizzbuzz



# תְּרֵגִיל Higher Lower

ASCII Art

## תיאור המשחק:

המחשב בוחר מספר אקראי בין 1 ל-100, והמשתמש צריך לנחש אותו.

לאחר כל ניחוש, המחשב מציג ASCII Art שמראה אם המספר הנכון **גובה יותר או נמוך יותר** מהניחסו.

## דוחי המשחק:

**טוויך:** המספר הסודי נמצא בין 1 ל-100 12 34

**رمזים:** אחרי כל ניחוש - "LOWER" או "HIGHER" של ASCII Art - ...

**ניצחון:** כשמנצחים נכון - ASCII Art של "!CORRECT" !

**סטטיסטיקה:** המשחק סופר כמה ניחושים לך

דוגמאות ASCII Art:

אם הניחס גןדי:



אם הניחס גבוה מדי:



אם ניחשת נכון:



# תְּרֵגִיל Higher Lower

דרישות התכונות

## דרישות התכונות:

### 1: ASCII Art

צרו פונקציות שמדפיסות ASCII Art במקומ טקסט רגיל:

```
def print_higher(): # ASCII Art של "HIGHER"
def print_lower(): # ASCII Art של "LOWER"
def print_correct(): # ASCII Art של "CORRECT!"
```

### 2: לוגיקת המשחק:

- יצירת מספר אקראי בין 1-100
- לולאת המשחק עד ניהוש נכון
- השווות ניהושים והציג ASCII Art מתאים
- ספרת מספר הנניהושים

### 3: תכונות נוספות:

- בדיקה קלט תקין ( רק מספרים בין 00-100 )
- הודעות ברכה ופרידה יפות
- אפשרות למשחק נוסף בסיסום

### 4: עיצוב ASCII Art

השתמשו אתרים כמו [patorjk.com/software/taag](http://patorjk.com/software/taag) לייצרת ASCII Art מקצועי!