

Python

TomerBu

מטרות:

צד שרת בשפת פיתון
עם ספריות פופולריות

תיק עבודות - עוד Backend

עוד שפה ולראות איך זה לעבוד בשפה אחרת

בפיתון נוח מאד לכתוב סקורייפים

פיתון פופולרית למידת מכונה

data science

web scraping

אלקטרוניקה

בדיקות תוכנה

מבנה של המודולים הבאים:

3 שיעורי פיתון - שפת תכנות

7 שיעורי צד שרת

Django + DRF

סדר היום:

הפסקה ראשונה:

10:30-11:00

הפסקה שנייה:

13:30 - 12:30

הפסקה שלישיית:

15:00 - 14:40

**נהייה בקשר בקבוצת
קבוצת וואטסאפ
עליה חומרים לגיטהאב**

נושאים להיום:

**סביבת פיתוח והתקנות
משתנים
קלט/פלט
משפטי תנאי
לולאות
ייבוא של מודולים**

התקנות:

Python

<https://www.python.org/downloads/>

The screenshot shows the Python.org website's download section. At the top, there's a dark header with tabs for "Python", "PSF", "Docs", "PyPI", "Jobs", and "Community". Below the header is the Python logo. To the right of the logo are buttons for "Donate", "Search", "GO", and "Socialize". A blue navigation bar below the logo has links for "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The main content area features a large yellow banner with the text "Download the latest version for macOS" and a "Download Python 3.13.1" button. Below this, text says "Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)". It also mentions "Want to help test development versions of Python 3.14? [Pre-releases](#), [Docker images](#)". To the right of the text is a cartoon illustration of two boxes descending from the sky on parachutes.

התקנות:

Python

4 . 2 . 1

MAJOR *Minor* patch

הוספת פיצ'רים

תיקוני באגים

הוספת עדכוןים משמעותיים

יכול לגרור קוד שלא עובד

אחרי השדרוג לספריה

מה חדש?

קוד ישן

ימשיך לתפקיד ברגיל

כשנשדרוג לספריה

מבוא



ערך זה עוסק בשפת תכנות. אם התכוונתם לנחש, ראו פיתון.

פייטון Python



```
setOfNumbers = []
print("How many random numbers do you want to generate?")
max = int(input())

for i in range (max):
    setOfNumbers.append(random.randrange(1,101,1))
setOfNumbers.sort()
print(setOfNumbers)

print("Which number do you want to find in the set of random numbers")
searchNumber = int(input())

firstPos = 0
lastPos = max-1
found = False

while (not found and firstPos <= lastPos):
    midPos = int((firstPos + lastPos)/2)

    if (searchNumber == setOfNumbers[midPos]):
        found = True
    else :
        if (searchNumber < setOfNumbers[midPos]):
            lastPos = midPos - 1
        else :
            firstPos = midPos + 1

if (found) :
    print("Your item is in the list")
else :
    print("Your item is not in the list")
```

**פרדיוגמות מונחית-עצמים,
프로그זרולית, פונקציונלית**

20 בפברואר 1991

תאריך
השקה

פייטון (באנגלית: Python) היא שפת תכנות **עלילית דינמית** למטרות כלליות מהנפוצות ביותר, ומודרגת באופן עיקרי כאחת משפות התכנות הפופולריות ביותר^{[2][3]}. פיתון תוכנה תוך שימוש דגש על קראות הקוד, וכוללת מבנים המיעדים לאפשר ביטוי של תוכניות מורכבות בדרך קצרה וברורה.

אחד המאפיינים הבולטים בתחום השפה הוא השימוש בהזחה להגדרת בלוקים של קוד (ללא שימוש בסוגרים או במילים שמורות לצורך כך, כמו ברוב השפות הנפוצות).

פיתון היא שפה מרובה-פרדיוגמות, המאפשרת **תוכנות מונחה-עצמים**, **תוכנות פרוצדורלי**, ובמידה מסוימת גם **תוכנות פונקציונלי**. לשפה **ספרייה** סטנדרטית גדולה וענפה, והיא תומכת באופן מובנה בהרחביה שלא אל שפות אחרות (בתלות במפרש: C, C++, Java, C# ו-C#).

לשפה שתי גרסאות ראשיות, פיתון 2 ופיתון 3. פיתון 2 איננה בפיתוח יותר, והתמכה בה הסתיימה בשנת 2020^[4].

עיצוב ופיתוח [עריכת קוד מקור | עריכה]

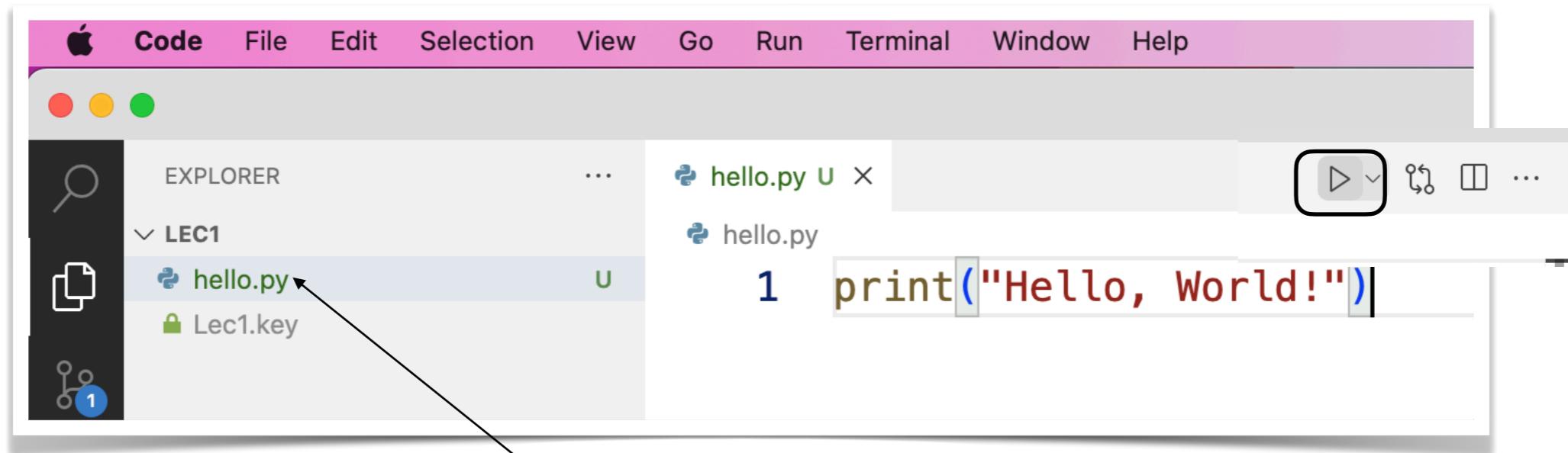
השפה נוצרה על ידי חידוongan רוסום **בתחילת שנות התשעים**, במקור מתוך פיתוח של **שפת התכנות**

תוספים לVSCode

The screenshot shows the Microsoft Store interface for extensions available for Visual Studio Code. The extensions listed are:

- autopep8**: Formatting support for Python files using the autopep8 formatter. Developed by Microsoft.
- Python Debugger**: Python Debugger extension using debugpy. Developed by Microsoft.
- Python**: Python language support with extension access points for IntelliSense (Pylance), Debugging (P...). Developed by Microsoft. This extension is highlighted with a rounded rectangle.
- Pylance**: A performant, feature-rich language server for Python in VS Code. Developed by Microsoft.
- autoDocstring - Python Docstring Generator**: Generates python docstrings automatically. Developed by Nils Werner. An "Install" button is visible next to the developer name.
- Python Environments**: Provides a unified python environment experience. Developed by Microsoft.

Hello World



קובץ עם סיוםת .py.

Hello World

```
print('hello')
print("Hello")
print(f"Hello {1+1}")
```

```
# single line comment
```

```
"""
multi-line
comment
"""
```

```
"This is a string"
```

קלט מהמשתמש

```
name = input("Please enter your name: ")  
age = input("Please enter your age: ")  
  
# This program asks for the user's name and age, then prints a greeting.  
#print("Hello,",name, "you are", age, "years old.")  
print(f'Hello, {name}. You are {age} years old.')
```

סוגי משתנים:

data_types.py

```
String          문자열
Integer         정수
Float           실수
Boolean         논리값 True False
```

```
age = 30 #int
price = 19.99 #float
name = "John" #string
is_active = True #boolean
is_active = False #boolean
```

אופרטורים בפייתו:

operators.py

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

אופרטורים בפייתו:

operators.py

```
x = 1
y = 2

# operators:
s = x + y # 3
sub = x - y # -1
mult = x * y # 2
div = x / y # 0.5
```

```
# more operators:
division = 50 / 6 # 8.3333
floor_division = 50 // 6 # 8
mod = 50 % 6 # 2
power = 2 ** 8 # 2*2*2*2*2*2*2*2 (8 times)
```

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

קלט של מספר:

```
name = input("Enter your name: ") #name:str
age = int(input("Enter your age: ")) #age:int
price = float(input("Enter your price: "))# price:float

print(type(age)) #<class 'int'>
print(type(price)) #<class 'float'>

print(type(name)) #<class 'str'>
```

.....

- יש לקלוט סכום של צבון במסעדה
 - יש לקלוט אחוז טיפ שרוצים להשייר
 - יש לקלוט את כמות הסועדים
 - הディיסו כמה כל אחד צריך לשלם
 - הディיסו כמה הסכום כולל הטיפ
 - הディיסו את סכום הטיפ שייש להשייר
-

```
# inputs:  
bill = float(input("Enter the bill amount: "))  
tip_percentage = int(input("Enter the tip percentage: "))  
num_diners = int(input("Enter the number of diners: "))  
  
# calculations:  
tip_amount = bill * tip_percentage // 100  
total_amount = bill + tip_amount  
total_per_diner = total_amount / num_diners  
  
# outputs:  
print(f"Total per diner: {total_per_diner:.2f}")  
print(f"Total amount including tip: {total_amount:.2f}")  
print(f"Total tip amount: {tip_amount}")
```

בקרת זרימה: לולאות ומשפטים תנאי:

```
a = int(input("Enter a number: "))
b = int(input("Enter a number: "))

if a > b:
    print("a is greater than b")
elif a < b:
    print("a is less than b")
else:
    print("a is equal to b")
```

נקודותים אחרי התנאי

הזהה במקום סגור מסולסל

indentation

מיני תרגיל - חישוב BMI

$$\text{BMI} = \frac{\text{Weight in kilogram}}{(\text{Height in meter})^2}$$

ברגע שיחסנו BMI אפשר להשתמש במשפטי תנאי כדי להדפיס הערה:

BMI	Weight class
below 18.5	underweight
18.5 - 24.9	normal
25.0 - 29.9	overweight
30.0 and up	Very overweight

$$\text{bmi} = 80 / (1.8^{**2}) = 24.7$$

קלטו מהמשתמש משקל בקילוגרמים

אשמה לפתרונות חלקים בצ'אט תוך כדי שאתם מתקדמים

קלטו מהמשתמש גובה במטרים

מי "שנטקע" אשמה אם תשתפו.

חשבו BMI

מי שישים - סמננו בצ'אט.

בעזרת משפטי תנאי - הדפיסו הערה רפואית

מוחמנים לשאול מה שרצו.

**פתרונות פחות יעיל:
יותר קוד ופחות ביצועים**

```
if bmi < 18.5:  
    print("Underweight")  
elif 18.5 <= bmi < 25:  
    print("Normal weight")  
elif bmi >= 25 and bmi < 30:  
    print("Overweight")  
else:  
    print("Obese")
```

פתרון יעיל:
פחות קוד ויתר ביצועים:

```
weight = int(input("Enter your weight in kg: "))
height = int(input("Enter your height in cm: ")) / 100

bmi = weight / height ** 2

if bmi < 18.5:
    print("Underweight")
elif bmi < 25:
    print("Normal weight")
elif bmi < 30:
    print("Overweight")
else:
    print("Very Overweight")
```

מודולציה = חלוקה של הקוד לקבצים: `import`

כל קובץ בפייתון הוא מודול: אפשר ליבא ממנו:

```
# animals.py
dog = r"""
  / \_@_\_
  (   0
  / \_/
  / \_/
"""

squirrel = r"""
  /\_,-~/
  @ @ | /,-:
  \ _T/-.-\(
  / , - \
  \ , / - \
  ((/_\(_,-'
"""

lazy_cat = r"""
  \,-'---,,-
  /,-'---,,-
  |,4- ) )-,,-; \(
  \---((/_\(_,-'
"""

..."""
```

המשתנים שרצוים ליבא
שם הקובץ

```
from animals import dog, lazy_cat, squirrel
print(lazy_cat)
print(dog)
print(squirrel)
```

מודולציה = חלוקה של הקוד לקבצים: import

ייבוא של כל המודול:

```
# animals.py
dog = r"""
  / \_@\
  (   0
  / \_/
  /   U
"""

squirrel = r"""
  |\_,_~_
  (@ @) /,-
  \_T/-._(
  / \_/
  | \_, / \
  ((/_\_) , -
"""

lazy_cat = r"""
  \,-,---,,-
  / .-`-,-,-
  |,4- ) )-, -.-;(
  \_--( _/--`-,-\_
"""

....
```

שם הקובץ

```
import animals
```

```
print(animals.squirrel)
print(animals.lazy_cat)
print(animals.dog)
```

שם תחליף ל קישור

```
import animals as a
```

```
print(a.squirrel)
print(a.lazy_cat)
print(a.dog)
```

תיקיה למודול:

The image shows a file tree on the left and three code editors on the right.

File Tree:

- utils
 - a.py
 - b.py

Code Editors:

- a.py:** Shows the assignment `num1 = 10`. The variable `num1` is highlighted in blue.
- b.py:** Shows the assignment `num2 = 20`. The variable `num2` is highlighted in blue.
- 10_use_modules.py:** Shows the code:

```
1 from utils import a, b
2
3 print(a.num1)
4 print(b.num2)
```

The variable `a` is highlighted in purple, and `b` is highlighted in blue.

תיקיה למודול:

```
▽ utils
  ▽ a.py
  ▽ b.py
```

```
from utils.a import num1
print(num1)
```

יבוא מודול המספריה המובנית של פיתון Python Standard Library

The screenshot shows a browser window displaying the Python documentation for the `random` module. The URL is `docs.python.org/3/library/random.html`. The page title is "random — Generate pseudo-random numbers". On the left, there's a "Table of Contents" sidebar with links to various sections like Bookkeeping functions, Functions for bytes, etc. Below it are "Previous topic" (`fractions`) and "Next topic" (`statistics`). At the bottom of the sidebar are "Report a bug" and "Show source" links. The main content area starts with a "Source code" link to `Lib/random.py`. It describes the module's purpose of implementing pseudo-random number generators for various distributions. It details integer selection from a range, sequence selection, random permutations, and sampling without replacement. It also covers distributions on the real line (uniform, normal, lognormal, negative exponential, gamma, beta) and angles (von Mises). The `random()` function is highlighted as generating a random float uniformly in the half-open range $0.0 \leq X < 1.0$. The Mersenne Twister is mentioned as the core generator, known for its speed and thread-safety. A warning box states that the pseudo-random generators should not be used for security purposes, suggesting the `secrets` module instead. A "See also" box cites a paper by Matsumoto and Nishimura. A note at the bottom cautions about thread contention when using the global generator or multiple instances.

דוקומנטציה רשמית של פיתון:

cano כתוב הכל

מcano אתרים כמו w3schools
MSCMIM ומקזרים

בעידן הAI אפשר לתרגם עם LLM

נעתייך את הטקסט נדביך בצאת גי פי טי ונקבל תרגום

יבוא מודול המספריה המובנית של פיתון Python Standard Library

10_use_modules.py ×

10_use_modules.py > ...

```
1 # יבוא את המודול:  
2 import random  
3  
4 r = random.randint(1, 6)  
5  
6 print(r)  
7  
8
```

(variable) def randint(
 a: int,
 b: int
) -> int

Return random integer in range [a, b], including both end points.

מיני תרגיל:

```
# Rock Paper Scissors ASCII Art
```

```
# Rock
print("""  
---' _____)  
      (_____  
      (_____  
      (_____  
---. __(_ )  
""")
```

```
# Paper
print("""  
---' _____)  
      _____)  
      _____)  
      _____)  
---. _____)  
""")
```

```
# Scissors
print("""  
---' _____)  
      _____)  
      _____)  
      (_____  
---. __(_ )  
""")
```

יש להגריל מספר בין 1 ל 3 כולל

אם המספר יצא אחד - יש להדפיס ציור של rock

אם המספר יצא 2 - יש להדפיס ציור של paper

אם המספר יצא 3 - יש להדפיס ציור של scissors

```
import random  
  
x = random.randint(1, 3)  
  
if x == 1:  
    print(r"""  
---' _____)  
      (_____  
      (_____  
      (_____  
---. __(_ )  
""")  
  
elif x == 2:  
    # Paper  
    print(r"""  
---' _____)  
      _____)  
      _____)  
      _____)  
---. _____)  
""")  
  
elif x == 3:  
    # Scissors  
    print(r"""  
---' _____)  
      _____)  
      _____)  
      (_____  
---. __(_ )  
""")
```

מיני תרגיל:

ascii.py

```
rock = r"""
---' \_____
      (_____)
      (_____)
      (_____)
---.|__(__)
_____""
```

```
paper = r"""
---' \_____
      _____)
      _____)
      _____)
---.|______)
_____""
```

```
scissors = r"""
---' \_____
      _____)
      _____)
      (_____)
---.|__(__)
_____""
```

12_rock_paper.scissors.py

```
from ascii import rock, paper, scissors
import random

options = [rock, paper, scissors]
x = random.randint(0, 2)
print(options[x])
```

מיני תרגיל:

ascii.py

```
rock = r"""
---' \_____
      (_____)
      (_____)
      (_____)
---.__(__)
_____""
```

```
paper = r"""
---'    ____)
      (____)
      (____)
      (____)
---.______)
_____""
```

```
scissors = r"""
---'   ____)
      (____)
      (____)
      (____)
---.__(__)
_____"""
```

הפונקציה choice מגרילה איבר רנדומלי ברשימה

12_rock_paper.scissors.py

```
from ascii import rock, paper, scissors
import random
```

רשימה

```
options = [rock, paper, scissors]
```

פונקציה מובנית בפייתון - שmagicaliber איבר רנדומלי ברשימה #
print(random.choice(options))

מיני תרג'il:

```
from random import choice

letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

random_letter = choice(letters)

print(random_letter)
```

משחקי מזל - להגריל קלף או שניים

מחולל סיסמות

שיק רנדומלי

בחירה זוכים בהגרלה

רשימות בפייתון:

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

# insert at the end:(push)
my_list.append('e')
my_list += 'fg'
my_list += ['h', 'i']

my_list.insert(9, 'j')

print(my_list) #['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

הוספת איברים לרשימה:
append
insert
+=

לולאות

```
fruit = ["apple", "pear", "orange", "lemon", "banana", "watermelon", "grapes", "strawberry", "mango", "cherry", "peach", "pineapple"]

# loop 3 times
for i in range(3):
    print(i) # 0, 1, 2

# for f in fruit list:
for f in fruit:
    print(f)
```

תרגיל - מחולל סיסמה רנדומלית

קלט - כמה אותיות לסיסמה

המשתמש מבקש 7 אותיות

נגריל 7 אותיות ונדפיס

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',  
'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
```

```
fruit = ["🍏", "🍐", "🍊", "🍋", "🍌", "🍉", "🍇", "🍓", "🥝", "🍒", "🍑", "🍍"]
```

שיעור בית:

לולאות **for**

השתמשו בלולאת **for** כדי להדפיס את כל המספרים מ-1 עד 20.

השתמשו בלולאת **for** כדי להדפיס בנפרד את כל האותיות בשמם.

צרו תוכנית המדפיסה את כל המספרים הזוגיים בין 1 ל-30

צרו תוכנית המדפיסה את כל המספרים מ-1 עד 100. בכל פעם שהתוכנית הגיע למספר שמכיל את הספרה 7, תחליף אותו במחרוזת **Boom**.

כתבו תוכנית אשר תבקש מהמשתמש להקליד הודעה ולאחר מכן מכн את מספר הפעמים שברצוננו להדפיס אותה. התוכנית תדפיס את הודעה כמספר הפעמים המבוקש.

כתבו תוכנית המקבלת טקסט מהמשתמש ומחזירה את מספר האותיות הקטנות ומספר האותיות的大写字母 שהtekst מכיל. לדוגמה, הטקסט: "Hello There" מורכב מ-2 אותיות גדולות ו-8 קטנות.

כתבו תוכנית המקבלת מספר שלם ומדפיסה בחזרה את כל הספרות אשר לא נמצאות במספר זהה.

שיעור בית:

הגרילו בין האפשרויות "אבן" "נייר" או "מספריים":
رمز יש להגריל מספר שלם והשתמשו במשפטי תנאי:

העשרה חפשו בגוגל:
ascii art rock paper scissors

```
import random

num = random.randint(1, 3)

if num == 1:
    print('Rock')
elif num == 2:
    print('Paper')
elif num == 3:
    print("Scissors")
```

הגרילו עבורי המחשב
מספר בין אחד ל-3

שאלו את המשתמש מה הוא בוחר:

- 1 אבן
- 2 נייר
- 3 מספריים

הציגו את ה-ascii של המתמודדים וממי ניצח

שיעור בית:

1) קלטו מספר דוא ספרתי והדפיסו את סכום הספרות

"45"

2) קלטו מספר תלת ספרתי והדפיסו את המספר בסדר הפוך

רמז: לכל TWO במחוזות יש אינדקס:

"Hello"[0]

שיעור בית:

1) קלטו מספר:

2) אם המספר גדול מ-10

הדפיסו גדול מ10

3) אם המספר קטן מאפס - יש להדפיס שלילי.

4) * אם המספר זוגי - יש להדפיס זוגי.

קלטו מספר א

קלטו מספר ב

יש להדפיס מי הגדל וממי הקטן.

* נסו להחליף ביניהם כך שהמספר הקטן ישמר בא

והמספר הגדל ישמר בע

*

שיעור בית:

צרו תוכנת קונסול להגרלה של סיסמה רנדומלית:

קלטו מהמשתמש כמהות תווים רצויים.

האם לחייב Lowercase

האם לחייב Uppercase

האם לחייב ספרות

האם לחייב תווים מיוחדים

התוכנה תפלוט סיסמא רנדומלית

