

Python

TomerBu

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.

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,
프로그זורליות, פונקציונליות

20 בפברואר 1991

תאריך
השקה

פייטון (באנגלית: Python) היא שפת תכנות עילית דינמית למטרות כללות 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 להגדלת בלוקים של קוד (ללא שימוש בסוגרים או במילים שמורות לצורך כך, כמו ברוב השפות הנפוצות).

פיתון היא שפה מרובה-פרדיוגמות, המאפשרת **תכנים מונחה-עצמים**, **תכנים פרוצדורלי**, ובמידה מסוימת גם **תכנים פונקציוני**. 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].

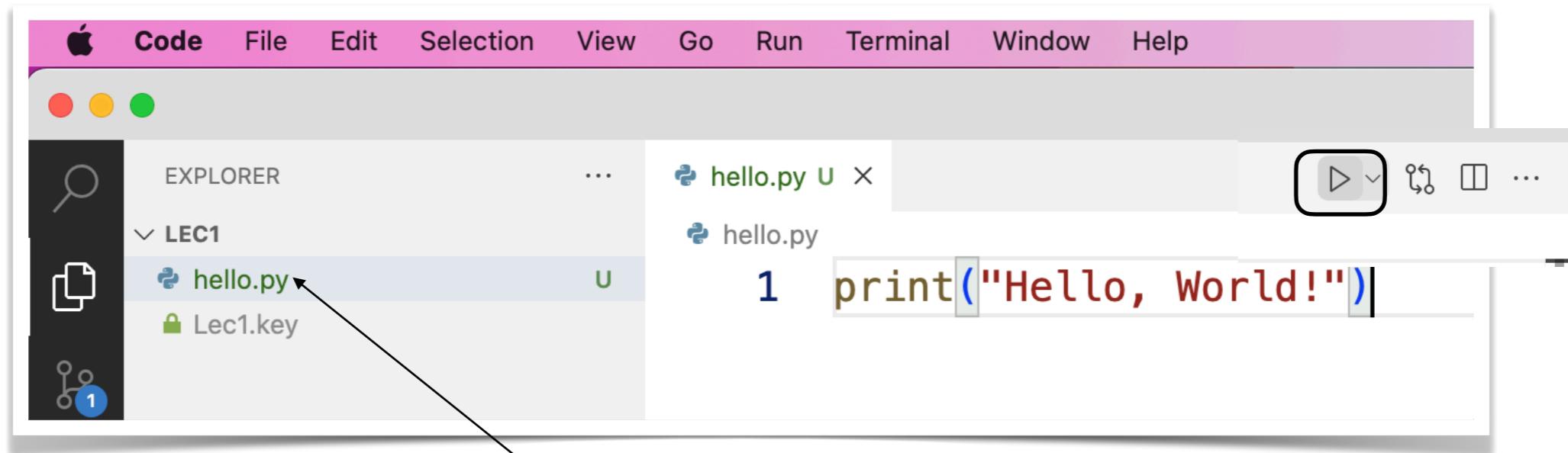
עיצוב ופיתוח [עריכת קוד מקו | עריכה]

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

תוספים ל-VSCode

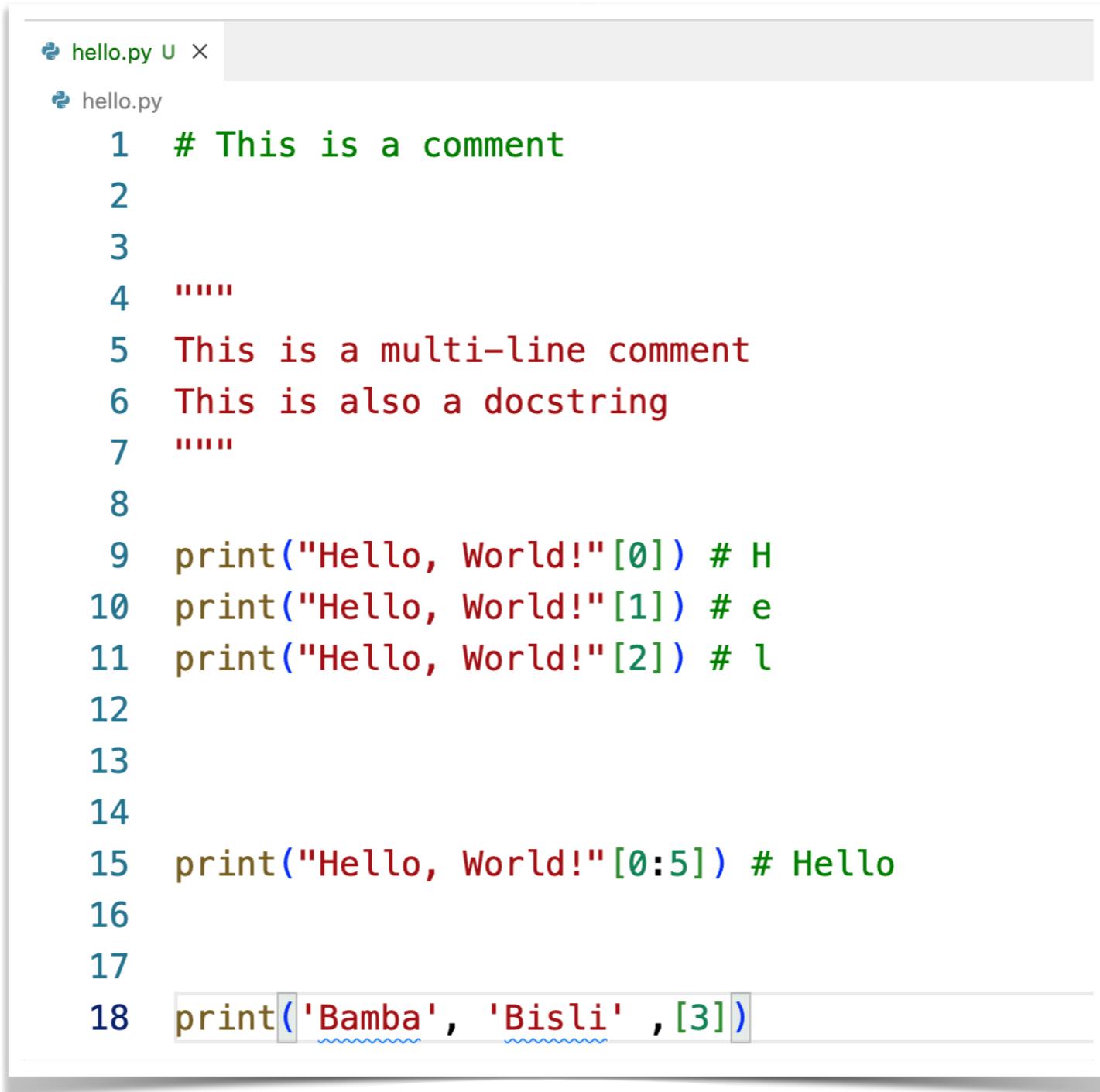


Hello World



קובץ עם סיוםת .py.

Hello World



The image shows a screenshot of a code editor window titled "hello.py". The code is a Python script with the following content:

```
hello.py x
hello.py

1 # This is a comment
2
3
4 """
5 This is a multi-line comment
6 This is also a docstring
7 """
8
9 print("Hello, World!"[0]) # H
10 print("Hello, World!"[1]) # e
11 print("Hello, World!"[2]) # l
12
13
14
15 print("Hello, World!"[0:5]) # Hello
16
17
18 print('Bamba', 'Bisli' , [3])
```

The code includes single-line comments (line 1), multi-line comments (lines 4-7), docstrings (line 6), and string slicing (line 15). The code editor interface shows line numbers on the left and syntax highlighting for keywords and strings.

Hello World

עד אפשרויות להדפסה למסך:

```
print("a", "b", "c", sep="😊") # a😊b😊c  
print("a", "b", "c", sep="😊", end="🐶\n") # a😊b😊c🐶
```

משתנה בפייתו:

```
⚡ 2_input.py U ×  
⚡ 2_input.py > ...  
1 user_name = 'Jerry'
```

קלט בפייתון:

```
user_name = input('Enter your name: ')
print('Hello, ' + user_name + '!')
```

נקlid urk shishmer b'mashtnah user name

ain hakod haze uved?

ma sder haratzah?

2_input.py U ×

2_input.py

```
1 print("Welcome " + input("What is your name? ") + "!")
```

קלט בפייתו:

```
# input of an str:  
user_name = input("What is your name? ")  
  
# input of an int:  
age = int(input("What is your age? "))  
print(age)      # 25  
  
# print the type of a variable:  
print(type(age))          # <class 'int'>  
print(type(user_name))    # <class 'str'>
```

מינוי תרגיל:

יש לקלוט סכום של חשבונו במסעדתך.
יש לקלוט אחוז טיפ שרצוים להשאר.
יש לקלוט את כמות הסועדים.
הדףו כמה כל אחד צריך לשלם.
הדףו כמה הסכום כולל הטיפ.
הדףו את סכום הטיפ שיש להשאר.

אשמה לפתרונות חלקיים ביצ'אט תוך כדי שאתם מתקדמים

מי "שנתקע" אשמה אם תשתפו.

מי שישים - סמננו ביצ'אט.

מוחמנים לשאול מה שרצוים.

פתרונות לתרגיל:

.....

- יש לקלוט סכום של שעבון במסעדה
- יש לקלוט אחוז טיפ שרצוים להשאיר
- 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 שיש להשאיר

.....

```
# inputs:  
bill = float(input("Enter the bill amount: "))  
tip_percent = int(input("Enter the tip percentage: "))  
diners = int(input("Enter the number of diners: "))  
  
# calculations:  
tip_amount = bill * tip_percent / 100  
total = bill + tip_amount  
total_per_person = total / diners  
  
# outputs:  
print(f"Each person should pay: {total_per_person:.2f} $")  
print(f"Total: {total} $")  
print(f"Tip: {tip_amount} $")
```

פתרונות לתרגילים:

Python Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	חזקה	$x ** y$
//	חלוקת שלמים	$x // y$

אי אפשר לחבר מחרוזת עם מספר:

```
age = 29  
name = "dave"
```

```
display = "your name is " + name + "your age is " + age
```

לא תקין
↑
לא ניתן לחבר מספר למחרוזת

```
age = 29  
name = "dave"
```

```
display = "your name is " + name + "your age is " + str(age)  
print(display)
```

עכשו זה תקין
↑

טיפוסים בסיסיים:

```
age = 29
name = "dave"

print('Hello, my name is' + name + "and I am" + str(age) + 'years old.')

# formatted strings (prettier/more readable)
print(f'Hello, my name is {name} and I am {age} years old.')

# some basic types in python:
# str
# int
# float
# bool (True/False)
is_happy = True # or False
```

בקרט זרימה:
Flow Control:
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 equal to b")  
else:  
    print("a is less than b")
```

אין סוגרים עגולים מסביב לתנאי
אין סוגרים מסולסלים - משתמשים בהזחה

מינימליסטי BMI - חישוב ו意义

יש לקלוט משקל וגובה
חשבו BMI לפי הנוסחה
בעזרת משפטי תנאי יש לתת הערכת משקל

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

תרגיל למי שסימן -

קלט של שנה
ונסו לקבוע אם מדובר בשנה מעוברת.

ויקיפדיה - התנאים לשנה מעוברת
Leap year

ברגע שיחסנו BMI אפשר להשתמש במשפטי תנאי כדי להדפיס הערכה:

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

תרגיל למי שסימן:

קלט של חודש ו שנה:
קבעו כמה ימים יש בחודש

מינימליסטי BMI - חישוב BMI

פתרון יעיל:

```
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("Obese")
```

פתרון פחות יעיל - אבל את הסינטקס חשוב להכיר

```
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")
```

יצירה של מודול:

קובץ `my_module.py`.

```
my_module.py U X 8_moduls.py U  
my_module.py > ...  
1 dog = """  
2 (\_,-----' () '---o  
3 .( _ ..... ____ ..... /~"  
4 ..(_)_ ) .. (_)_ )  
5 """  
6  
7 cat = """  
8 | \_ / |  
9 ( . o . o . )  
10 .> .^ .<  
11 """  
12
```

יבוא של מידע ממודול:

my_module.py U

8_moduls.py U X

8_moduls.py

```
1 from my_module import cat, dog, frog
2
3 print(cat)
4 print(dog)
5 print(frog)
```

יבוא המודול:

8_moduls.py

8_moduls.py

```
1 import my_module
```

```
2
```

```
3 print(my_module.cat)
```

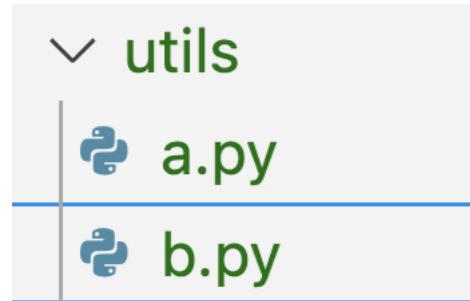


יבוא המודול עם שם חליפי: :alias

```
import my_module as animals
```

```
print(animals.cat)
```

תיקיה למודול:



החל מגרסת 3.3 יוצרים תיקיה והיא מהוות מודול:

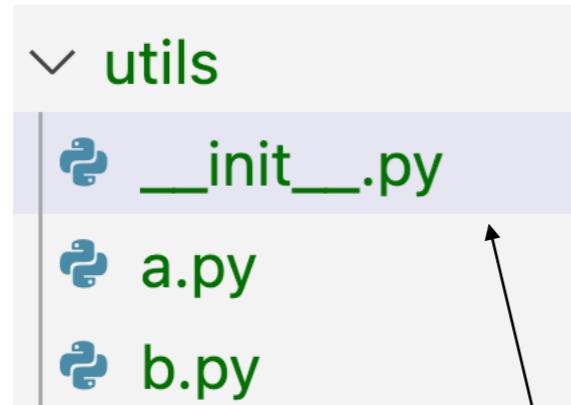
אפשר ליבא מהתיקון עם `import` כרגע:

A screenshot of a code editor showing a file named '8_moduls.py'. The code in the file is:

```
1 from utils import a, b
2
3 print(a.c)
```

The code editor interface includes tabs for '8_moduls.py' and 'U' (untracked), and a status bar at the bottom.

תיקיה למודול:



לתמיכה בגרסאות פיתון ישנות יותר:
נוסיף קובץ בשם uc.__init__

וכך נוכל לעבוד עם מודולים גם בגרסאות ישנות יותר של פיתון

סביבת הפיתוח יוצרת תיקייה __pycache__
מייעלת את תהליך היבוא
যוצרת אינדקס לתיקיה

בקובץ זהה אפשר להוסיף קוד
שירוץ עם טעינת המודול

יבוא מהספריה הסטנדרטית (המבנה)

```
#1) import the entire module
```

```
import random
```

```
print(random.randint(1, 10))
```

```
# 2) import the entire module with an alias
```

```
import random as r
```

```
print(r.randint(1, 10))
```

```
# 3) import a function from a module
```

```
from random import randint
```

```
print(randint(1, 10))
```

כך נגריל מספר בין 1 ל 3

```
# יש להגריל מספר בין 1 ל 3
from random import randint

num = randint(1, 3) #random from 1 to 3 inclusive

print(num)
```

מיני תרגיל:

צרו משחק של
Rock, Paper, Scissors

מיני תרג'il:

צרו משחק של
Rock, Paper, Scissors

א) צרו מודול עם ציורים בASCII
ב) בתוכנית נקלט מהמשתמש קלט:

1 = rock
2 = paper
3 = scissors

ג) נגריל למחשב
מספר בין 1 ל 3
(שימוש במודול random)

ד) לפי כללי המשחק נקבע מי מנצח
(שימוש במשפטי תנאי)

ה) נסו להדפיס את המהלך של המשחק
עם שימוש בASCII מהמודול שיצרתם בשלב א.

מי שישים :

צרו משחק של
Rock, Paper, Scissors, Lizard, Spock

עוד רעיונות לפתרון לתרגילים:

- 1) עבודה עם רשימות
- 2) עבודה עם מיליון
- 3) לולאות

רשימות:

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

```
# Accessing elements in a list
print(my_list[0]) # a
print(my_list[1]) # b
```

```
# len() function
print(len(my_list)) # 5
```

איך נגריל אותה אחת מהרשימה?

נגריל מספר בין 0 ל 4 (האינדקסים של הרשימה)
ונציג את האיבר ברשימה לפי האינדקס

```
print(my_list[randint(0,4)])
```

הגרלה של איבר רנדומלי:

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

פעולות על רשימות:

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

# insert:
my_list.append('f') # add an element to the end of the list
my_list.insert(6, 'g') # insert an element at a specific index

# update:
my_list[0] = 'A' # update the element at index 0

# delete:
# delete by index:
del my_list[6] # delete the element at index 6

# delete by index:
my_list.pop(0)

# delete the last item:
my_list.pop()

# delete by value:
my_list.remove('c')

# print the list:
print(my_list)
print(my_list[0])
```

לולאות `for in`

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

# forEach loop
for f in fruit:
    print(f)

# loop from 0 to 3 exclusive # 0, 1, 2
for i in range(3):
    print(i)

# loop from 2 to 5 exclusive # 2, 3, 4
for i in range(2, 5):
    print(i)

# loop from 1 to 10 exclusive with step 2 # 1, 3, 5, 7, 9
for i in range(1, 10, 2):
    print(i)      ↑      ↑      ↑
                  start    end    step

# run 3 times:
for _ in range(3):
    print("Hello")
```

מיני תרגיל:

יש להציג רשימה עם האפשרויות של המשחק Rock Paper Scissors
ולהציג איבר רנדומלי מהרשימה

תרגיל סיכום:

צרו מחולל סיסמאות:

```
lower_case_letters = 'abcdefghijklmnopqrstuvwxyz'  
upper_case_letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  
digits = '1234567'  
special_chars = '@#$%^&'  
  
password_letters = ""  
  
if input('use lower') == 'y':  
    password_letters += lower_case_letters
```

קלטים:

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

נרצה בוללהה לפי אורך הסיסמה:

for i in range(16):

נשתמש ברנדום כדי למצוא את רנדומליות

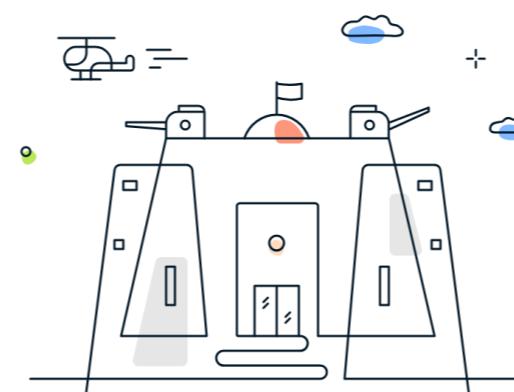
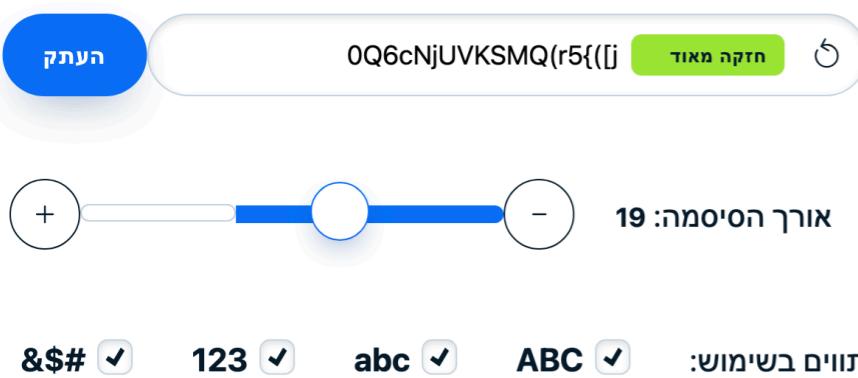
הנחיות:

למי שהתרגיל מסובך:

קלט רק של כמה התווים.
להגריל סיסמה שמורכבת רק מאותיות.

מחלול סיסמאות אקראיות

יציאת סיסמאות חזקות ומאובטחות כדי לשמר על החשבונות המקצועיים.



פתרונות לתרגיל מחולל סיסמות:

```
from random import choice
from string import ascii_lowercase, ascii_uppercase, digits
punctuation = '#$%+*'

# Get user input
length = int(input('Enter the length of the password: '))

include_lower = input('Include lowercase letters? (y/n): ').lower() == 'y'
include_upper = input('Include uppercase letters? (y/n): ').lower() == 'y'
include_digits = input('Include digits? (y/n): ').lower() == 'y'
include_punctuation = input('Include punctuation? (y/n): ').lower() == 'y'

# all the letters for the password
letters = ''
password = ''

# add user choice to the letters list
if include_lower:
    letters += ascii_lowercase #letters = 'abcdefghijklmnopqrstuvwxyz'
if include_upper:
    letters += ascii_uppercase #letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
if include_digits:
    letters += digits #letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
if include_punctuation:
    letters += punctuation #letters = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789#%+*'

for _ in range(length):
    password += choice(letters)

print(password)
```

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 במחוזת **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אות 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.

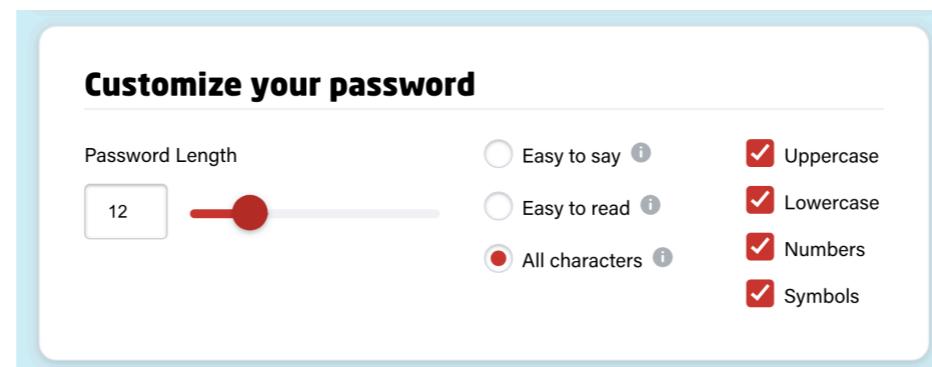
האם לחייב Lowercase

האם לחייב Uppercase

האם לחייב ספרות

האם לחייב תווים מיוחדים

התוכנה תפלוט סיסמא רנדומלית



תרגיל בonus:

צופו קיסר:

נגיד רשימה עם כל האותיות:
נבקש מהמשתמש קלט שהוא רוצה להצפין.
נבחר הزاה

for letter in user_input:
 מחשב את האות החליפית

עם המתודה `:find`

```
index = letters.find('a') # 0
```

```
index += shift
```