ספר קורס

CYBER SCHOOL

לולאות for ו- while

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

CYBER SCHOOL

מידע על ספר לימוד זה

ספר לימוד זה נועד לסייע לכם ללמוד ולסקור את החומר הנלמד במהלך הקורס מבוא לפייתון. הוא מכיל מידע, הרלוונטי לשיעורים הנלמדים בכיתה, וכן הפניות לפעילויות מעבדה שכדאי לתרגל כדי להגביר את רמת הידע שלכם.

התוכן בספר לימוד זה כולל יסודות תכנות וכיצד לעבוד עם שפת התכנות פייתון.

מקרא

בלוקים של טקסט צבעוני עשויים להופיע לאורך ספר הלימוד כדי להפנות את הקוראים למקור ספציפי או להעשיר אותם במידע נוסף.

הבלוקים של הטקסט כוללים:

משימת מעבדה
זהו זמן טוב לבחון את קבצי המעבדה הרלוונטיים (על פי קוד המעבדה) ולתרגל את מה שנלמד עד כה.

טוב לדעת
מידע נוסף או תובנות לגבי הנושא. מידע זה נועד לצורך העשרה בלבד ואינו חלק מהחומרים עליהם תיבחנו.

טיפ
מידע שימושי שיש בו כדי לסייע לתלמידים ללמוד את החומר או לעבוד עם כלים מסוימים.

מידע נוסף
קישורים או הפניות לחומר חיצוני שניתן להשתמש בהם כדי להרחיב את הידיעות שלכם לגבי הנושא.

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

לולאות For & While

שיעור זה הינו מבוא לתכנות לולאות for ו-while בשפת פייתון. בנוסף יושם דגש על שיטות ניהול על-מנת להבטיח כתיבת קוד מסודר ויעיל.

מה הן לולאות?

לולאות הן בלוקים של קוד שפועלות כל עוד תנאי מסוים מתקיים. הן כלי חיוני בכתיבה ותכנות שעוזר למתכנתים להימנע מלכתוב את אותה שורת קוד שוב ושוב.

לדוגמה, הדפסת המילה "Hello" 10פעמים יכולה להתבצע בשתי שורות קוד, במקום עשר פקודות הדפסה הכתובות אחת מתחת לשנייה.

Python מאפשרת לולאות *for* ו-*while* דומות, אך משמשות במקרים שונים:

- לולאות For יכולות לחזור על הצהרות X פעמים.
- לולאות While חוזרות על הצהרות ברציפות, עד שמשהו בבלוק הקוד עוצר אותן.

לולאת For

הלולאה *for* ב-Python חוזרת על רצפים או אובייקטים שחוזרים על עצמם, כגון רשימות, tuples ומחרוזות. הם משמשים בתוכניות לביצוע פעולה חוזרת בצורה יעילה ללא צורך בכתיבת שורות קוד רבות.

שימו לב שלולאות *for* יבוצעו ללא בדיקה מוקדמת לאימות התנאי, בניגוד ללולאות *while* שעבורן התוכנית תבדוק תחילה אם התנאי חוקי.

בדוגמה למטה, המילה "Test" מוגדרת כמחרוזת במשתנה בשם "Word". לאחר מכן נגדיר tuple עם ארבעה מספרים שלמים במשתנה "num". לאחר מכן, אנו יוצרים לולאת for, שבתוכה אנו מגדירים משתנה נוסף "i" המייצג כל תו במשתנה word. הצעד האחרון הוא ההדפסה של תוצרי התהליך.

הלולאה חוזרת על עצמה כמספר התווים במשתנה. אותו דבר קורה עם המשתנה num, לפיו לולאת for חוזרת על עצמה ככמות המספרים השלמים במשתנה num.

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

```
word = "test"
num = (1,2,3,4)

for i in word:
    print(i)

print()

for n in num:
      print(n)
```

```
==========================================================
t
e
s
t

1
2
3
```

טווח בהשוואה לרשימות בלולאות

לולאות For יכולות לשמש בכדי להדפיס רשימה שלמה לפי ברירת מחדל, או חלק מרשימה, על ידי בחירת הפרמטרים שיכללו בלולאה, כמוצג בדוגמה להלן.

פונקציית הטווח יכולה לשמש להדפסת סט ערכים ספציפי בתוך רשימה.

בקוד המופיע להלן, אנו מגדירים את המשתנה fruit עם רשימה של חמישה fruits. לאחר מכן אנו יוצרים שלוש לולאות. הראשונה חוזרת כמספר הפעמים שיש פריטים ברשימה, ומדפיסה כל פריט עם כל איטרציה של הלולאה.

הלולאה השנייה חוזרת כמספר הפעמים המוגדרים בפונקציית הטווח, שבמקרה שלנו זה חמש פעמים, 0-5, לא כולל 5.

הלולאה השלישית משתמש בפונקציה בתוך פונקציה, כדי לחזור חמש פעמים. פונקציית Len() סופרת כמה פריטים נמצאים במשתנה fruit, ומשתמשת במספר הזה כחלק מהטווח.

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

```
fruit = ["Watermelon", "Banana", "Apple", "Pineapple", "Strawberry"]
for i in fruit:
   print(i)

for i in range(0,5):
   print(i)

for i in range(0,len(fruit)):
   print(i)

==================================================
Watermelon Banana Apple Pineapple Strawberry

0 1 2 3 4

0 1 2 3 4
```

הערה: פונקציית הטווח יכולה להשתמש בשיטת *traversal* כדי ליישם טווח מספרים שלמים (כגון 0,10).

חישוב גודל לולאה

הפונקציה ***len()*** סופרת את מספר הפריטים של אובייקט, כגון מחרוזות ורשימות.

הפונקציה ***range()*** מקבלת עד שלושה מספרים שלמים ומחזירה את הטווח של האובייקט. המספר השלם הראשון הוא נקודת ההתחלה, השני הוא נקודת הסיום והשלישי הוא גודל הצעד.

בקוד שלהלן, הפונקציה len() משמשת לספירת מספר הפריטים במשתנה השמות והדפסת המספר. כך, אם אורך הרשימה משתנה, שורת ההדפסה עדיין תוכל להדפיס את המספר הנכון של פריטים ברשימה.

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

```
namelist = ["David", "John", "Cooper", "Nick"]
print(namelist, "The length is {}".format(len(namelist)))

colorlist = ["Blue", "Red", "Yellow"]

for i in range(8):
    print(i)
```

```
==========================================================
['David', 'John', 'Cooper', 'Nick'] The length is 4
0
1
2
3
4
5
6
7
```

משימת מעבדה - לולאת טווח

תרגול יצירת טווח עם קלטים שונים בלולאה.

משימת מעבדה - Loops in Nested Lists

תרגול של הפעלת Nested Loops.

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

CYBER SCHOOL

לולאת While

תנאי While מפעיל בלוק (Block) של הצהרות עד שנמצא תנאי False.

בדוגמה למטה, נוצרת לולאת While כדי לבדוק בכל איטרציה של הלולאה אם משתנה המונה קטן או שווה ל-6. אם הוא לא, הלולאה תמשיך, ואם הוא כן, הלולאה תסתיים. כל איטרציה של הלולאה תוסיף 1 למשתנה Counter.

```
counter = 0

while counter <= 6:
   counter += 1
   print(counter)
```

מצב תלוי בדרך כלל במשהו שקורה בתוך בלוק הקוד. לכן, בשלב מסוים, התנאי כבר לא יהיה נכון והלולאה תסתיים.

אם התנאי הוא False, הלולאה לא תבוצע, והתוכנית תמשיך לבלוק הקוד הבא.

טיפ

כאשר משתמשים בלולאת While, חשוב לבדוק שהתנאי בסופו של דבר יהיה False, או שהלולאה לעולם לא תסתיים!

פקודות Break

ניתן להשתמש בפקודות Break רק בלולאות. בפייתון, **break** תיצא מהלולאה אם התנאי הוא False או True, בהתאם לתוכן התוכנית.

שימו לב שניתן להשתמש בפקודות Break גם בלולאות For וגם ב-While.

המטרה היא לצאת מלולאה כאשר תנאי מסוים מתקיים.

בדוגמה למטה, בכל איטרציה של לולאת While היא בודקת אם משתנה המונה קטן מ-50. אם כן, היא נכנסת ללולאה ומוסיפה 1. בלולאה יש פקודת ***if*** שבודקת אם המונה שווה ל-7. אם כן, היא מדפיסה הודעה ושוברת את מעגל הלולאה.

```
counter = 0

while counter < 50:
   counter += 1
   if counter == 7:
      print("The counter reached the max number: {}".format(counter))
      break
```


כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

```
  else:
    print("The current number is: {}".format(counter))
```

```
The current number is: 1
The current number is: 2
The current number is: 3
The current number is: 4
The current number is: 5
The current number is: 6
The counter reached the max number: 7
```

פקודת Continue

ניתן להשתמש בפקודת **Continue** גם בלולאת for וגם בלולאת while כדי לדלג על איטרציה. היא מתנהגת כאילו הקוד הגיע לסוף הבלוק.

פקודות Continue משמשות בכדי לדלג על יתר הקוד בתוך לולאה, באיטרציה הנוכחית בלבד. בניגוד ל-**Break,** הלולאה לא מסיימת אלא ממשיכה אל האיטרציה הבאה.

בדוגמה למטה, לולאת while בודקת אם משתנה המונה קטן מ-10. אם כן, היא נכנסת ללולאה ומוסיפה 1 למונה. בלולאת while, יש פקודת ***if*** שבודקת אם משתנה המונה שווה ל-7. אם כן, היא תדפיס "Skipping one increment" ותמשיך את הלולאה.

```
counter = 0

while counter < 10:
  counter += 1

  if counter == 7:
    print("Skipping one increment")
    continue

  print("Current number: {}".format(counter))
```

```
Current number: 1
Current number: 2
Current number: 3
Current number: 4
Current number: 5
Current number: 6
Skipping one increment
Current number: 8
```


כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

```
Current number: 9
Current number: 10
```

פקודת Pass

Pass היא placeholder עבור פקודה נדרשת שתוזן מאוחר יותר. השמטת פקודה בנקודות מסוימות בתוכנית עלולה לגרום לשגיאה, לכן נעשה שימוש במשפט **pass,** כדי לאפשר לקוד להמשיך לרוץ בלעדיו לעת עתה.

בדוגמה מטה, לולאת While בודקת אם משתנה המונה קטן מ-50. אם כן, היא נכנסת ללולאה ומוסיפה 1 למונה. בלולאת while יש פקודת ***if*** שבודקת אם משתנה המונה שווה ל-7. אם כן, היא תדלג על האיטרציה ותמשיך את הלולאה.

```
counter = 0

while counter < 50:
   counter += 1
   if counter == 7:
      pass

   else:
      print("The current number is: {}".format(counter))
```

```
The current number is: 1
The current number is: 2
The current number is: 3
The current number is: 4
The current number is: 5
The current number is: 6
The current number is: 8
The current number is: 9
```

Pass בהשוואה ל Continue

Pass פירושה "אין קוד לביצוע כאן" והיא תאפשר לתוכנית להמשיך עם הקטע הבא בלולאה, או לעבור לקטע הבא של הקוד.

Continue מאלצת את הלולאה להתחיל באיטרציה הבאה, כלומר היא תקפוץ חזרה לתחילת הלולאה.

כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383

בדוגמה למטה, הלולאה for חוזרת על עצמה כמספר הפעמים שיש תווים במשתנה המילה. אם המשתנה ***x*** שווה ל-***s***, היא תעבור את האיטרציה. בדוגמה למטה, כאשר הלולאה מגיעה לתו ***s***, היא מדפיסה "Continue execute" וממשיכה עם הלולאה.

```
word = "test"
for x in word:
    if x == "s":
        print("Pass Execute")
        pass
    print(x)

print()

for x in word:
    if x == "s":
        print("Continue execute")
        continue
    print(x)
```

```
t
e
Pass Execute
s
t

t
e
Continue execute
t
```


כל הזכויות שמורות © סייבר סקול בע"מ, אילנות 7, כרמיאל | 077-7781383