

פרק 4 אופרטורים

עמוד	הנושא
2	4. אופרטורים : פעולות אריתמטיות, לוגיות, השמה והשוואה
3	4.1 אופרטורים אריתמטיים (חשבוניים)
3	4.2 אופרטורים לוגיים
4	4.3 אופרטורים להשוואה
4	4.4 פעולות לבדיקת זהות
5	4.5 פעולות על ביטים
6	4.6 פעולות השמה
7	4.7 פעולות חברות - Membership Operators
7	4.8 המתודות floor() ו Ceil() ו round()
8	4.9 תרגילים

בפרק 4 בתוכנית הלימודים של הנדסאי אלקטרוניקה של משרד החינוך בנושא שפת פייתון רשום :

פרק 4 : אופרטורים

יעדים

היכרות עם מערך האופרטורים בשפת Python.

תכנים

1. אופרטורים בסיסים
 2. ההבדל בין אופרטור אונארי לבינארי
 3. אופרטור קידום/הפחתה עצמיים
 4. הכפלה/חלוקה ושארית החלוקה
 5. אופרטורי השוואה
 6. אופרטורי השמה
 7. ערך רצפה, ערך גג ועיגול:
 - floor
 - ceil
 - round
 8. האופרטור "is"
 9. אופרטורים על סיביות (Bits Wise)
- סיכום שעות ההוראה : 8 שעות עיוניות ו- 4 שעות התנסותיות. סה"כ 12 שעות.

4. אופרטורים : פעולות אריתמטיות, לוגיות, השמה והשוואה

האופרטורים – פעולות - משמשים לביצוע פעולות בין משתנים וערכים.

האופרטורים בפיתון מחולקים לקבוצות הבאות :

- פעולות אריתמטיות (חשבוניות) .
- פעולות לוגיות .
- פעולות השוואה.
- פעולות זהות.
- פעולות השמה.
- פעולות על חברויות.

4.1 אופרטורים אריתמטיים (חשבוניים)

בטבלה הבאה נניח ש $x=9$ ו $y=4$

האופרטור	השם	דוגמה	הסבר
+	חיבור	$x+y = 9 + 4 = 13$	חיבור $x+y$
-	חיסור	$x-y = 9 - 4 = 5$	חיסור $x-y$
*	כפל	$x*y = 9 * 4 = 36$	כפל $x*y$
/	חלוקה	$x/y = 9 / 4=2.25$	חלוקה x/y
%	שארית (מודולו)	$x\%y = 9 \% 4 = 1$	שארית לאחר חלוקה של x/y . תוצאת החלוקה השלמה היא 2 ונשאר שארית 1.
**	חזקה או שורש	$x**y = 9^4 = 6561$	חזקה של x^y
//	חילוק בתצוגת מנה (שלמה - Floored division)	$x//y = 9//4=2$ $-9//4 = -3$	המספר השלם הקרוב לרצפה של החלוקה x/y . התוצאה "האמיתית" היא 2.25 והמספר השלם הקרוב לרצפה הוא 2. בחלוקה של מספרים שליליים כמו $-9/2$ התוצאה "האמיתית" היא -2.25 והמספר השלם הקרוב לרצפה היא -3 כי זה המספר שהוא כמו רצפה ל -2.25 .

טבלה 1 : פעולות אריתמטיות

הערה לנושא החזקה : ניתן להשתמש באופרטור חזקה ** גם למציאת שורש אם נשתמש בחזקה כשבר.

דוגמה : מצא את השורש של המספר 45.

```
>>> 45**0.5
```

```
6.708203932499369
```

דוגמה נוספת : נמצא את השורש הרביעי של 6561 שהוא תוצאה של 9^4 .

```
>>> 6561**0.25
```

```
9.0
```

הערה לנושא האחרון בטבלה שהוא חילוק בתצוגת מנה שלמה // : הרצפה של כל מספר חיובי היא הערך השלם

של המספר השלם שמתחתיו שמשמש לו כרצפה, כלומר ללא החלק של השבר. לדוגמה : הריצפה של 1.001 היא 1 וגם של 1.999 היא 1. עבור מספרים שליליים הריצפה היא המספר השלילי השלם שמתחתיו. לדוגמה : -1.001 היא -2 וגם של -1.999 היא -2 .

4.2 אופרטורים לוגיים

אופרטורים לוגיים משמשים להצהרות עם תנאי . אם התנאי מתקיים מקבלים True אם התנאי לא מתקיים מוחזר False .
בטבלה הבאה נניח ש $x=9$ ו $y=4$

האופרטור	השם	דוגמה	הסבר
and	וגם and . טבלת אמת F and F=F F and T=F T and F=F T and T=T	$x < 10$ and $y > 3$ נקבל True	כאן מתקיימים 2 התנאים.
or	או or F or F=F F or T=T T or F=T T or T=T	$x < 10$ or $y > 10$ נקבל True	כאן התנאי הראשון הוא True והשני False והתוצאה True
not	היפוך not	$\text{not}(x < 10 \text{ and } y < 10)$ נקבל False	תוצאת הסוגריים היא True ואחרי היפוך היא False

טבלה 2 : אופרטורים לוגיים

4.3 אופרטורים להשוואה

אופרטורים להשוואה משמשים להשוואה בין ערכים. התוצאה המוחזרת היא True או False .
בטבלה הבאה נניח ש $x=9$ ו $y=4$

האופרטור	השם	דוגמה	הסבר
==	האם שווה ?	<code>print(x==y)</code> בודקים האם x שווה ל y . נקבל False .	היות ו x לא שווה ל y נקבל False
!=	האם לא שווה ?	<code>if (x!=y) :</code> <code>print("x != y")</code> בודקים האם x לא שווה ל y . נקבל True	היות ו x לא שווה ל y נקבל True

		True ואת ההדפסה x!=y	
>	גדול מ ?	if (x>y) : print("x bigger than y")	אם x גדול מ y מוחזר True ולכן נקבל את ההדפסה x bigger . then y
<	קטן מ ?	print(x<y) נקבל False	מודפס False כי x לא קטן מ y .
>=	גדול או שווה ?	print(x>=y) נקבל True	
<=	קטן או שווה ?	print(x<=y) נקבל False	

טבלה 3 : אופרטורים להשוואה

4.4 פעולות לבדיקת זהות

משתמשים בפעולות לבדיקת זהות כדי לבדוק האם האובייקט הוא אותו אובייקט ונמצא באותה כתובת בזיכרון.

הפעולה לא בודקת האם הערך שיש בשני האובייקטים הוא שווה !

יש 2 אופרטורים : **is** ו **is not**

האופרטור is בודק האם 2 משתנים הם אותו אובייקט . אם כן מוחזר True . אם הם לא אותו אובייקט מוחזר False .
הפעולה is not עובדת הפוך. מוחזר True אם הם לא אותו אובייקט ו False אם הם אותו אובייקט.

דוגמה:

```
x = ["dollar", "euro"]
y = ["dollar", "euro"]
z = x
```

```
print(x is z)
```

True # בגלל שהם אותו אובייקט והם באותן כתובות בזיכרון
: נקבל

```
print(x is y)
```

False # אפילו שערכם שווה הם לא אותו אובייקט ולא נמצאים באותו כתובות
בזיכרון

```
print(x == y)
```

True # כי הערכים שלהם : נקבל
שוים

4.5 פעולות על ביטים

פעולות על ביטים משמשות להשוואה בינארית של מספרים. בטבלה הבאה נניח ש $x=9$ ו $y=4$

האופרטור	השם	דוגמה	הסבר
&	AND	$x \& y \Rightarrow 0$	$x=1001 \ y=0100 \Rightarrow 0$
	OR	$x y \Rightarrow 13$	$x=1001 \ y=0100 \Rightarrow 13$
^	XOR	$x \wedge y \Rightarrow 13$	$x=1001 \ y=0100 \Rightarrow 13$
~	NOT	$\text{not}(x \wedge y) \Rightarrow \text{False}$	$\text{not}(x \wedge y \Rightarrow 13) = \text{False}$
<<	הזזה שמאלית עם מילוי אפסים	$x \ll 2 \Rightarrow 36$	$9 \ll 1 \Rightarrow 18 \ \ll 1 \Rightarrow 36$
>>	הזזה ימנית עם שמירה על הסימן	$x \gg 2 \Rightarrow 2$	$9 \gg 1 \Rightarrow 4 \ \gg 1 \Rightarrow 2$

טבלה 4 : פעולות על ביטים

4.6 פעולות השמה

הפעולה	דוגמה	הסבר (הערה – במקום שרשום b הכוונה ערך בינארי)
=	$x=9$	
+=	$x+=4 \ x=x+4 = 13$	
-=	$x-=4 \ x=x-4 = 9-4 = 5$	
=	$x=4 \ x=x*4=9*4 = 36$	
/=	$x/=4 \ x=x/4=9/4 = 2.25$	
%=	$x\%=4 \ x=x\%4 = 9\%4=1$	
//=	$x//4 \ x=x//4 = 9//4 = 2$	
=	$x=4 \ x=x**4 = 9^4= 6561$	
&=	$x\&y \ x=x\&y = 9\&4=0$	$9 \& 4= b1001 \& b0100 = b0000 = 0$
=	$x =y \ x==x =y = 9 =4=13$	$9 4= b1001 b0100 = b1101 = 13$
^=	$x\^=3 \ x=x\^3=9\^3=10$	$9 \wedge 3= b1001 \wedge b0011 = b10010 = 10$

>>=	x>>=3 x>>3=9>>3=1	הזזה ימנית מחלקת את הערך ב 2 תוך שמירת הסימן ומזניחה שארית.
<<=	x<<=3 x=x<<3=9<<3=72	הזזה שמאלית מכפילה את המספר ב 2.

אופרטורים של השמה משמש להעביר ערכים למשתנים. בטבלה הבאה נניח ש $x=9$ ו $y=4$
טבלה 5 : פעולות השמה

4.7 פעולות חברות - Membership Operators

משתמשים בפעולות חברות כדי לבדוק האם רצף מסוים נמצא באובייקט.

יש שני אופרטורים. א. **in** . ב. **not in**

לדוגמה:

```
colors = ["Red", "Green", "Blue"]
print("Green" in colors)
```

התשובה שנקבל היא True .

4.8 המתודות floor() ו Ceil() ו round()

שלוש המתודות נמצאות במודול math של פייתון ולפני השימוש בהן יש לרשום **import math** .

4.8.a המתודה floor() מקבלת מספר כלשהו ומחזירה את המספר השלם הקרוב כלפי מטה למספר ששלחנו אליה.

דוגמה :

```
print(math.floor(0.123))      # 0
print(math.floor(1.3456))    # 1
print(math.floor(15.021))    # 15
print(math.floor(-5.3))      # -6
print(math.floor(-20.654))   # -21
print(math.floor(10.0))      # 10
print(math.floor(1.9999))    # 1
```

4.8.b המתודה ceil() מקבלת מספר כלשהו ומחזירה את המספר השלם הקרוב כלפי מעלה למספר ששלחנו אליה.

```
print(math.ceil(0.123))      # 1
print(math.ceil(1.3456))    # 2
print(math.ceil(15.021))    # 16
```

```
print(math.ceil(-5.3))      # -5
print(math.ceil(-20.654)) # -20
print(math.ceil(10.0))     # 10
print(math.ceil(1.9999))   # 2
```

4.8 המתודה round() מקבלת 2 ערכים. הראשון הוא מספר והשני הוא כמה ספרות אחרי הנקודה רוצים. היא מחזירה את המספר הממשי המעוגל (נקודה צפה – floating point) למספר ששלחנו אליה עם מספר הספרות הרצוי אחרי הנקודה. ברירת המחזר למספר הספרות הוא 0 (אם לא שלחנו אליה את מספר הספרות אחרי הנקודה שרוצים) ואז מוחזר המספר השלם.

דוגמאות :

```
print(round(5.76543, 2))    # 5.77
print(round(3.12345,3))    # 3.123
print(round(-12.654321,4)) # -12.6543
print(round(-123,456))     # -123
```

4.9 תרגילים

לכל התרגילים נניח ש $x=9$ ו $y=4$ והוגדר `import math`.

1. יש לרשום פקודה שתדפיס את מכפלת הערכים x ו y .
2. יש לרשום פקודה שתדפיס את החלוקה בין x ל y .
3. יש לרשום 2 שורות שייבדקו האם x לא שווה ל y ותוציא הדפסה מתאימה.
4. מה נקבל מקטע התוכנית הבאה ?

```
if(x==y or x>y):
    print("YES")
```

5. מה נקבל מקטע התוכנית הבאה ?

```
if(x==y and x>y):
    print("YES")
```

6. מה נקבל מקטע התוכנית הבאה ?

```
if(x%y == 1):
    print("yes")
```

7. מה נקבל בקטע התוכנית הבא ?

```
x = 9
```



```
y=11  
x&=y  
print(x)
```

8. מה נקבל מקטע התוכנית הבא ?

```
x = 9  
y=4  
y|=x  
print(y)
```

9. מה נקבל עבור $x^{\wedge}y$?

10. מה נקבל עבור $x \gg= 2$?

11. מה נקבל עבור $x \ll=y$?

12. מה נקבל עבור $z \gg 2$ אם ידוע ש $z=-40$?

13. מה נקבל עבור $w \gg 3$ אם ידוע ש $w=-40$?

14. מדוע מקבלים False בתוכנית הבאה ?

```
colors = ["Red", "Green", "Blue"]  
print("Green " in colors)
```

15. יש להדפיס את תוצאות התרגילים הבאים:

```
print(math.floor(0.123456))  
print(math.floor(-15.0213))  
print(math.floor(-25.3))  
print(math.floor(20.654))  
  
print(math.ceil(-15.32))  
print(math.ceil(-120.654))  
print(math.ceil(40.0))  
print(math.ceil(-1.9999))  
print(round(15.1276543, 2))  
print(round(-13.12345,3))  
print(round(-12.65432101,6))  
print(round(-123,456))
```