

- צרו קובץ חדש בשם "Hello_Python_2"**- כתבו בתוכנית את שורות הקוד הבאות (בשיטה המועדפת עליכם):***#Here is the program*

```

print(" welcome to Python Class!!")
name = input (" what is your name?!!")
print(name, " - is a beautiful name!!")
heigh=float(input("Enter your height\n"))
if ( heigh>1.6 ) :
    print (" You are in a basketball team!!")
else:
    print (" You are in a football team!!")

```

לחצו על Run, הקלידו את הנתונים המבוקשים, מה הפלט שהתקבל?:

לחצו על Run, מה הפלט שהתקבל אם הקלדתם בגובה את הערך 1.8 :

לחצו שוב על Run, מה הפלט אם הקלדתם בגובה את הערך 1.5 :

שנו בהתאם את שורות הקוד המתאימות לשורות הבאות:

```

print(" welcome to Python Class!!")
name = input (" what is your name?!!")
print(name, " - is a beautiful name!!")
heigh=float(input("Enter your height\n"))

if (heigh>1.8) :
    print (" You are in a basketball team!!")
elif(heigh<1.6):
    print (" You are in a football team!!")
else:
    print (" You can be in any team you want!!")

```

לחצו על Run, מה הפלט שהתקבל הפעם אם הקלדתם בגובה את הערך 1.8 :

לחצו על Run, מה הפלט שהתקבל הפעם אם הקלדתם בגובה את הערך 1.6 :

ביטוי לוגי פשוט מורכב משני ביטויים וביניהם אופרטור לוגי לפי המפורט בטבלה

Operator	Description	משמעות משמאל לימין
==	Checks if the value of two operands is equal or not, if yes then condition becomes true.	שווה
!=	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	שונה
>	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	שונה
<	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	גדול
<=	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	קטן
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	גדול שווה
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	קטן שווה

ביטוי לוגי מורכב

There are following logical operators supported by Python language. Assume variable a holds 10 and variable b holds 20, then:

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true, then the condition becomes true.	(a and b) is true.
or	Called Logical OR Operator. If any of the two operands are non zero, then the condition becomes true.	(a or b) is true.
not	Called Logical NOT Operator. Used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false.	not(a and b) is false.

תרגול:

תרגיל 1 - דוגמא לשימוש בביטוי לוגי פשוט בתוכנית:

פעולות	עבור קלט	הפלט
<pre> heigh=float(input("Enter your height\n")) if (heigh>1.6): print(" You are in a basketball team!!") else: print(" You are in a football team!!") </pre>	1.75	
	1.60	
	1.45	
	1.65	

תרגיל 2

דוגמא לתנאי מורכב:

```
math = int(input("Insert grade in math"))
computer = int(input("Insert grade in Computer Science"))
english = int(input("Insert grade in English"))
if (math>85 and computer>85 and english>85) :
    print("You're in a technological scientific reserve
classroom")
elif(math<56 or computer<56 or english<56 ):
    print("You are in a regular classroom")
else:
    print("You're in a mofet classroom")
```

פרטו, מבלי להריץ, מה יהיה הפלט לפי הקלט למשתנים בהתאם לטבלה:

מתמטיקה	מדעי המחשב	אנגלית	פלט
90	95	100	
85	90	85	
56	56	56	
45	90	87	
78	64	72	
45	88	92	

א. הריצו את התוכנית ובידקו את תשובתכם. האם צדקתם? _____

ב. מה משמעות and ? _____

ג. מה משמעות or ? _____

ד. הסבר מה משמעות המילים השונות במבנה התנאי :

if
else
elif

תרגיל 3

מספר יקרא פולינדרום, אם כאשר נקרא אותו מימין לשמאל או משמאל לימין נקבל את אותו מספר, למשל: 121, 545, 898 וכו'. ...

תכננו וכתבו תכנית הקולטת מספר שלם תלת ספרתי, בודקת האם הוא פולינדרום ומדפיסה הודעה מתאימה.

תרגיל 4

התשלום לאכסניה, עבור לינה, נעשה לפי 100 ₪ לחדר. באכסניה יש 4 מיטות בחדר. מדריך קבוצה רוצה לדעת כמה תעלה הלינה לקבוצה. אם מספר האנשים בקבוצה, אינו מתחלק ל-4 בדיוק, נדרש חדר נוסף לשארית.

התכנית תשאל כמה אנשים בקבוצה ותחשב מה עלות הלינה, השלימו את

הנתונים החסרים בתוכנית

```
group_size = int(input('Enter how many people in group ? : '))
rooms = group_size ____ 4
if(group_size % ____ ) > 0:
    rooms = ____ + 1
print ('The cost is:', rooms * 100, 'Shekels')
```

תרגיל 5

כתבו תוכנית המבקשת כקלט את צבע הרמזור להולכי רגל. אם הצבע ירוק

(Green), מדפיסה Pass. אחרת תדפיס Stop.

התכנית צריכה להבין כי הקלט green, GREEN, Green וכל קומבינציה של אותיות קטנות וגדולות הוא ירוק.

תרגיל 6**תנאי מקונן ו-elif**

לעיתים לאחר בדיקת תנאי, נדרש לבדוק תנאי נוסף. במקרה זה בתוך הקוד המותנה, תופיע

הוראת תנאי נוספת. למצב זה קוראים **קינון**.

כל קינון גורם להזחה נוספת.

עבור מקרים בהם נדרשת בדיקה מרובה, כאשר מעוניינים לטפח רק במקרה שהתנאי מתקיים (True) קיים ב-python מבנה של: אם ... אחרת אם ... אחרת אם ... וכולי.

מבנה זה, מונע קינון והזחות רבות :

מבנה אלגוריתמי	מבנה ההוראה בשפת python
אם <תנאי לוגי> (שורות קוד המותנה בקיום התנאי) [אחרת אם (שורות קוד המותנה באי קיום התנאי)] [אחרת אם (שורות קוד המותנה באי קיום התנאי הקודם)] ... וכך הלאה	<pre>if<condition>:[code line] [code lines] elif: code lines] elif: code lines] elif: code lines] . . . and so on</pre>

כתוב תכנית המבקשת ציון בין 0 ל-100, יש לבדוק שהקלט חוקי.

אם הציון 100 תדפיס Excellent

אם הציון שווה או יותר מ-90 תדפיס Very good

אם הציון שווה או יותר מ-80 תדפיס Good

אם הציון שווה או יותר מ-60 תדפיס You should do better

אחרת תדפיס You should relearn

תרגיל 7

תנאי לוגי מורכב - תנאי לוגי מורכב הוא תנאי הכולל בדרך כלל שניים או יותר תנאים

פשוטים, ופעולות של יחס לוגי ביניהם. היחסים הלוגיים הם :

הפעולה	משמעות	משמעות
<ביטוי>and<ביטוי>	פעולת וגם	מחזירה True רק עם שני הביטויים True. אחרת מחזירה False
<ביטוי>or<ביטוי>	פעולת או	מחזירה True אם לפחות אחד מהביטויים True. אחרת מחזירה False
<ביטוי>not	פעולת לא	הופכת את המצב הלוגי של הביטוי: מחזירה True אם הביטוי False. מחזירה False אם הביטוי True.

דוגמאות:

א. התנאי הבדוק האם x נמצא בתחום שבין 5 ל-10 ($5 < x < 10$) יירשם:
<code>10 > x and x > 5</code>
ב. התנאי הבדוק האם האור ברמזור הוא אדום או ירוק יירשם:
<code>color == 'green' or color == 'red'</code>

ג. התנאים: יכולים להירשם:	<code>a!=b, a<>b</code> <code>not (a==b)</code>
ד. התנאים: יכולים להירשם: וכן יכולים להירשם:	<code>a >= b , a <= b</code> <code>not (a < b) ,not (a > b)</code> <code>(a>b) or (a==b),(a>b) or (a==b)</code>

נתבקשנו לכתוב תכנית שתקלוט שעה, ותדפיס האם זו שעת מנוחה (כלומר בין 14 ל-16 או בין 23 ל-6 למחרת) או שעת פעילות. השלימו את היחס הלוגי המתאים:

```
time=int(input('Enter current hour:'))
if((time >= 14) ____ (time <16)) ____ (time >= 23) ____ (time <6):
    print ('It is rest time')
else:
    print ('It is active time')
```

תרגיל 8

שנה מעוברת (29 לפברואר) מתרחשת כל 4 שנים.
אבל אם זו תחילת מאה (למשל 1800, 1900), זו לא תהיה שנה מעוברת.
אבל אם זו תחילה של מאה רביעית בסידרה (למשל 400, 800 ... 2000...) זו כן תהיה שנה מעוברת.
כתבו תכנית המבקשת שנה, ומדפיסה הודעה האם היא מעוברת - leap year
או לא. נסו להשתמש בהוראת תנאי אחת בלבד עם תנאי מורכב.

תרגיל 9

יעל מארגנת כל 5 עוגיות באריזה
כתבו קטע קוד הקולט מספר שלם המציין מספר עוגיות ומצייר על המסך באמצעות הצב את
מספר האריזות. אם נותרו עוגיות ללא אריזה יש לצייר עיגולים כמספר העוגיות שנותרו
אחרת יצוירו ריבועים ככמות האריזות
לדוגמה עבור 17 עוגיות ← 3 אריזות מרובעות ו- 2 עיגולים, עבור 20 עוגיות ← נצייר 4 אריזות
מרובעות

תרגיל 10

א.

הפקודה `alex.pos()` מחזירה את המיקומים `X,Y` של הצב .

`alex.pos()[0]` - יחזיר את ערך ה- `X` של הצב

`alex.pos()[1]` - יחזיר את ערך ה- `Y` של הצב

כתבו מה מבצע קטע הקוד הבא :

```
import turtle
wn = turtle.Screen()
alex = turtle.Turtle()
alex.shape("turtle")
turtle.speed("slow")
while True:
    while alex.pos()[0] != 200:
        alex.forward(5)
    while alex.pos()[0] != -200:
        alex.backward(5)
turtle.mainloop()
```

ב.

שדרגו את קטע הקוד. הציבו 2 צבים משני צדדי הישר שיתקדמו כל עוד אינם פוגשים זה בזה .
כאשר הם נפגשים הם חוזרים לנקודת ההתחלה ושוב מתקדמים. כל צב נע במהירות אחרת.