הסבר על הסימון <tuple_name>* שמאפשר להגדיר שפונקציה תקבל מספר לא מוגבל של פרמטרים:

דוגמא פשוטה: פונקציה שמקבלת מספר כלשהו של מספרים ומחזירה את המכפלה שלהם (אם לא קיבלה אף מספר תחזיר 1).

```
def mult(*params):
        result = 1
        print(params)
        for elem in params:
                reuslt *= elem
        return result
```

כל הקריאות הבאות לפונקציה יהיו חוקיות:

```
mult()

mult(3,4)

mult(5,3,1,2)
```

למעשה params הינו tuple שיכיל 0 או יותר ערכים שהועברו כפרמטרים ל mult.

הפלט שיתקבל מהפעלה של mult(3,4) הינו:

```
>>> x = mult(3,4)

(3,4)

>>> x

12
```

עבור tuple בשם t למשל, ניתן להשתמש בסימון t* כדי להעביר את תוכן ה tuple כאוסף של פרמטרים לתוך פונקציה. לדוגמא:

```
>>> t = (3,4,5)

>>>x = mult(*t)

(3,4,5)

>>>x

60
```

נחזור לדוגמא שכוללת ביטויי lambda:

נגדיר את הפונקציה הבאה שמקבלת לפחות שני פרמטרים lst, func ויתכן שפרמטרים נוספים שיכנסו לתוך tuple בשם params. הפונקציה מפעילה את func על הרשימה ומעבירה את הפרמטרים הנוספים שניתנו לה, אם היו כאלה, בזה אחר זה.

```
def print_consecutive_sublist(lst, func, *params):
    return func(lst, *params)
```

נגדיר פונקציה נוספת בשם func באמצעות ביטוי lambda. הפונקציה מקבלת 3 פרמטרים (רשימה ושני אינדקסים) ומדפיסה את תת הרשימה בין האינדקסים שקיבלה או את המחרוזת bad range אם האינדקסים לא ניתנו בסדר הנכון.

```
func = lambda lst, i, j : print(lst[i:j] if i <j else "bad range")
```

נגדיר את הרשימה להיות:

```
lst = [i for i in range (100)]
```

כעת נקרא ל print_consecutive_sublist עם lst, הפונקציה func, ושני פרמטרים נוספים: 50, 55.

שני הפרמטרים האחרונים יכנסו לתוך tuple בשם params.

הפלט שיתקבל הינו:

```
>>> print_consecutive_sublist([i for i in range (100)], func, 50, 55)
[50, 51, 52, 53, 54]
```

הפלט שיתקבל מההפעלה הבאה הינו:

```
>>> print_consecutive_sublist([i for i in range (100)], func, 55, 45)
bad range
```

כעת נגדיר את הפונקציה print_sublist שמקבלת לפחות שני פרמטרים lst, func ויתכן שפרמטרים נוספים שיכנסו לתוך tuple בשם params. הפונקציה מפעילה את func על הרשימה lst ועל הtuple params.

```
def print_sublist(lst, func, *params):
   return func(lst, params)
```

נגדיר את func להיות:

```
func = lambda lst, tup : print([x for x in lst if x in tup])
```

הפלט מההרצות הבאות יהיה:

```
>>> print_sublist([i for i in range (100)], func, 50, 55)
[50, 55]
>>> print_sublist([i for i in range (100)], func, 50, 55, 20)
[20, 50, 55]
```