

## קונבנציות כתיבה ב-Python: תקציר

### כללי

מסמך זה הינו תקציר לקונבנציות עיקריות לכתיבה בשפת Python. המסמך מבוסס על PEP8<sup>1</sup> והוא חלקי בלבד.

### שמות משתנים ופונקציות

#### כללי

שמות יהיו לעולם משמעותיים ואינדיקטיביים.

#### קונבנציות שמות

באותיות לטיניות קטנות, כאשר מילים מופרדות בקו-תחתון. אין להשתמש בקידומות כלשהן לשמות משתנים כדי להעיד על סוג המשתנה.

**נכון:** `name_of_student = 'Moshe'`  
**לא נכון:** `str_name_of_student = 'Moshe'`  
`nameOfSudent = 'Moshe'`  
`n = 'Moshe'`

(1) שמות משתנים

יהיו באותיות קטנות, כאשר מילים מופרדות בקו-תחתון.

**נכון:** `get_student_grade(student_name)`  
**לא נכון:** `getStudentGrade(studentName)`

(2) שמות פונקציות

משתנים שאסור לשנות את ערכם יכתבו באותיות גדולות, עם קו-תחתון להפרדה בין מילים.

**נכון:** `BEST_PROGRAM = 'Gvhaim'`  
**לא נכון:** `best_program = 'gvahim'`

(3) קבועים

#### תיעוד

(4) Documentation Strings : יש ליצור מחרוזות תיעוד לכל קובץ (מודול), כך שיכיל את שם הכותב ותיאור קצר של הפונקציונליות שהמודול מספק.

(ב) חובה ליצור מחרוזות תיעוד לכל פונקציה ומתודה.

(ג) הערה: מחרוזות התיעוד לא מחליפות את השמות האינדיקטיביים.

(5) תיעוד בעזרת הערות :

(א) יש לתעד את הקוד כך שיהיה מובן, בדגש על חלקים מורכבים.

(ב) אין צורך לתעד כל שורה ושורה, ואין צורך לתעד דברים מובנים מאליהם.

<sup>1</sup> <https://www.python.org/dev/peps/pep-0008>

6) אין לכתוב בקוד "מספרי קסם". במקום זאת, יש ליצור קבוע מתאים כאשר משתמשים במספר כלשהו בעל משמעות.

```
print sys.argv[2]
USER_NAME = 2
print sys.argv[USER_NAME]
```

7) יש להשתמש בקבועים True ו-False המובנים בשפה כדי לייצג "אמת" או "שקר" (ולא, למשל, 0 או 1).

8) לכל משתנה בקוד יהיה ייעוד אחד בלבד. אין ליצור משתנים עם מספר שימושים שונים.

9) יש "להקיף" אופרטורים ברווח אחד מכל צד (חוץ ממינוס המציין שלילה):

```
1 + 2 * 3 == -3 + 10
1+2*3==-3+10
```

```
birthday_string = "happy birthday!"
birthday_string="sad birthday :("
```

10) רווח יבוא אחרי פסיק, לא לפניו.

```
example_function(1, 2, 3)
example_function(1 , 2 , 3)
```

11) אין לכתוב שורות ארוכות מידי. כמות תווים המקסימלית בשורה הינה 79.

**כללי אצבע חשובים**

12) בכדי לבדוק האם אובייקט הוא None יש להשתמש באופרטור is:

```
if object is None:
    ...
if object == None:
```

13) אין להגדיר mutable objects (כמו list או dictionary) בתור פרמטרים דיפולטיים לפונקציה.

```
def function(default=None):
    if default is None:
        default={}
```

```
def function(default={}):
```

14) אין להשתמש ב-keywords של פייתון בתור שמות משתנים.

```
list = [1, 2]
numbers_list = [1, 2]
```