# מתודות קסם

זוכרים את הדוגמא מהפרק על הכימוס בפייתון עם המחלקות A ו-B?   
מתי שהפעלנו את הפונקציה dir() על המחלקה B היינו אמורים לראות רק מתודות ושדות של B שהגדרנו, אבל למרבה ההפתעה קיבלנו שם רשימה של תכונות שיש למחלקה שבכלל לא ידענו על קיומן:

print(dir(B))  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
['\_A\_\_private\_function', '\_B\_\_private\_function', '\_\_class\_\_', '\_\_delattr\_\_', '\_\_dict\_\_', '\_\_dir\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_format\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_', '\_\_init\_subclass\_\_', '\_\_le\_\_', '\_\_lt\_\_', '\_\_module\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_', '\_\_repr\_\_', '\_\_setattr\_\_', '\_\_sizeof\_\_', '\_\_str\_\_', '\_\_subclasshook\_\_', '\_\_weakref\_\_', 'public\_function']

המכנה המשותף לכל התכונות האלה הוא שהן מתחילות בשני קווים תחתונים ומסתימות בשני קווים תחתונים.  
אז מה אלו התכונות האלה?   
המתודות האלו נקראות "מתודות קסם" והן אינן נועדו לשימוש ישיר אלא הן מתעוררות בצורה עקיפה.   
למעשה יצא לנו כבר לראות כמה שדות ומתודות כאלו, למשל המתודה \_\_init\_\_() היא סוג של מתודת קסם שמאתחלת אובייקט מטיפוס מחלקה כלשהי, והיא מופעלת אוטומטית כאשר יוצרים אינסטנס חדש למחלקה.  
לרוב משתמשים במתודות קסם כדי לדרוס אופרטורים, למשל כדי לדרוס את האופרטור '+' נדרוס את המתודה \_\_add\_\_(self,other), או עבור אופרטור '==' נדרוס את \_\_eq\_\_(self,other) .  
אך למעשה יש עוד הרבה סוגים של מתודות קסם מלבד אופרטורים.   
ניתן לחלק את מתודות הקסם לכמה סוגים:   
  
1. אופרטורים  
כפי שכבר ציינו קודם למתודות קסם יש את האפשרות לדרוס אופרטורים.   
שלא כמו ב-c++ או c# המתודות שמגדירות את סוג אופרטור אינן כייצוג שלו בפועל:

|  |  |
| --- | --- |
| אופרטור | מתודה |
| + | \_\_add\_\_(self,other) |
| - | \_\_sub\_\_(self,other) |
| \* | \_\_mul\_\_(self, other) |
| // | \_\_floordiv\_\_(self, other) |
| / | \_\_truediv\_\_(self, other) |
| % | \_\_mod\_\_(self, other) |
| \*\* | \_\_pow\_\_(self, other[, modulo]) |
| > | \_\_lt\_\_(self, other) |
| <= | \_\_le\_\_(self, other) |
| == | \_\_eq\_\_(self, other) |
| != | \_\_ne\_\_(self, other) |
| >= | \_\_ge\_\_(self, other) |
| > | \_\_gt\_\_(self, other) |
| << | \_\_lshift\_\_(self, other) |
| >> | \_\_rshift\_\_(self, other) |
| & | \_\_and\_\_(self, other) |
| | | \_\_or\_\_(self, other) |
| ^ | \_\_xor\_\_(self, other) |

אם מוסיפים לחתימת המתודות את האות 'i' לפני השם האופרטור ניתן לדרוס את ה'syntactic sugar' אופרטור עם השמה, למשל עבור '=+' נדרוס את המתודה \_\_iadd\_\_(self,other), או בשביל '-=' את \_\_isub\_\_(self,other) .

2.אתחול והריסה של אובייקט-   
את המתודוה \_\_init\_\_() אנחנו כבר מכירים, אבל בפועל אנחנו לא צריכים את המתודה כדי ליצור אובייקט חדש, אפילו יותר מזה אפשר להוסיף למחלקה בלי תכונות תכונות חדשות:

טקסט