OOP?

→ We write a code around the object (real life entity)

→ Methods and Properties (variables) related to the real life entity.

For example

Vehicle
Properties - Name, Speed, Colour, type
Methods speedup gearup

Person
Properties → name, gender
Methods → eat(), walk(), dance()

```
@staticmethod?
It does not take either cls or obj(self) reference.
## When do we create them:-
-> Just do something with the parameters given like
    changing the format of given date(like utility
    or helper functionalities)
-> It improves code readability, signifying that the
    method
    does not depend on state of the object itself.
-> It reduces memory usage as compared to classmethod.
-> It allows for method overriding
```

Vrite code in Python 3.6 (drag lower right corner to resize code editor)

```
1  class Student:
2      class_var = 5
3      def __init__(self,name,roll_no,marks):
4          self.name = name
5          self.roll_no = roll_no
6          self.marks = marks  ## it will be a dictionary
7
8      @staticmethod
9      def percentage_formula(marks_obtained_list,total):
10         return (sum(marks_obtained_list)/total)*100
11
12     def calculate_percentage(self):
13         total = len(self.marks)*100
14         self.printPercentage(Student.percentage_formula
15
16     def printPercentage(self,percentage):
17         print('The percentage is ',percentage)
18
19 def happy():
20     print("I am happy to know that")
21
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (2 steps)