

#DAY

12

DAY 12 OF 200 DAY'S PYTHON CHALLENGE



STATIC METHOD

A static method in Python is a method that is bound to a class rather than the objects of that class. This means that a static method can be called without instantiating an object of the class.



Day 4.py

```
class Person:  
    @staticmethod  
    def is_adult(age):  
        return age >= 18  
  
print(Person.is_adult(20))
```

EXAMPLE



Day 4.py

```
class pwskills2:  
    def student_details(self, name , mail_id, number):  
        print(name, mail_id, number)  
  
    @staticmethod  
    def mentor_class(list_mentor):  
        print(list_mentor)  
  
    @staticmethod  
    def mentor_mail_id(mail_id_mentor):  
        print(mail_id_mentor)  
  
    @classmethod  
    def class_name(cls):  
        pwskills2.mentor_mail_id(['abhi@gmail.com', 'sudh@gmail.com',  
        'krish@gmail.com'])  
        cls.mentor_class(['abhi', 'sudh', 'krish'])  
  
    def mentor(self, mentor_list):  
        self.mentor_class(['abhi', 'sudh', 'krish'])  
        print(mentor_list)  
  
pwskills2.class_name()
```

MAGIC FUNCTION

Python Magic methods are the methods starting and ending with double underscores ‘__’. They are defined by built-in classes in Python and commonly used for operator overloading.

They are also called Dunder methods, Dunder here means “Double Under (Underscores)”. And it work really magical way.



Day 4.py

```
dir(int)
a = 10
a.__add__(5) # or a + 5 are equal
```

PROPERTY DECORATOR

The property decorator in Python is a built-in function that allows you to define properties for a class. Properties are special attributes that can have getters, setters, and deleters. The @property decorator is used to define the property name in the class, and the getter, setter, and deleter methods are defined as normal methods.



Day 4.py

```
class Person:  
    def __init__(self, name):  
        self._name = name  
  
    @property  
    def name(self):  
        return self._name  
    @name.setter  
    def name(self, value):  
        self._name = value  
    @name.deleter  
    def name(self):  
        del self._name
```

EXAMPLE



Day 4.py

```
class test2:  
    def __init__(self, course_price, course_name):  
        self.course_name = course_name  
        self.__course_price = course_price  
  
    @property  
    def course_price_access(self):  
        return self.__course_price  
  
    @course_price_access.setter  
    def course_price_set(self, price):  
        if price <= 3500:  
            pass  
        else:  
            self.__course_price = price  
  
    @course_price_access.deleter  
    def delete_course_price(self):  
        del self.__course_price  
  
t2 = test2(34545, 'data science')  
t2.course_price_set = 3600  
t2.course_price_access
```

PYTHON

Assignment Questions



1. Who developed Python Programming Language?
2. Which type of Programming does Python support?
3. Is Python case sensitive when dealing with identifiers?
4. What is the correct extension of the Python file?
5. Is Python code compiled or interpreted?
6. Name a few blocks of code used to define in Python language?
7. State a character used to give single-line comments in Python?
8. Mention functions which can help us to find the version of python that we are currently working on?
9. Python supports the creation of anonymous functions at runtime, using a construct called

10. What does pip stand for python?
11. Mention a few built-in functions in python?
12. What is the maximum possible length of an identifier in Python?
13. What are the benefits of using Python?
14. How is memory managed in Python?
15. How to install Python on Windows and set path variables?
16. Is indentation required in python?

Assignment 2:

Marks: 20

- 1. Write a program to accept percentage from the user and display the grade according to the following criteria:**

Marks	Grade
>90	A
>80 and <=90	B
>=60 and <=80	C
below 60	D

- 2. Write a program to accept the cost price of a bike and display the road tax to be paid according to the following criteria:**

Tax	Cost Price(in Rs)
15%	>100000
10%	>50000 and <= 100000
5%	< = 50000

- 3. Accept any city from the user and display monuments of that city.**

City	Monument
Delhi	Red Fort
Agra	Taj Mahal
Jaipur	Jal Mahal

- 4. Check how many times a given number can be divided by 3 before it is less than or equal to 10.**

- 5. Why and When to Use while Loop in Python give a detailed description with example**

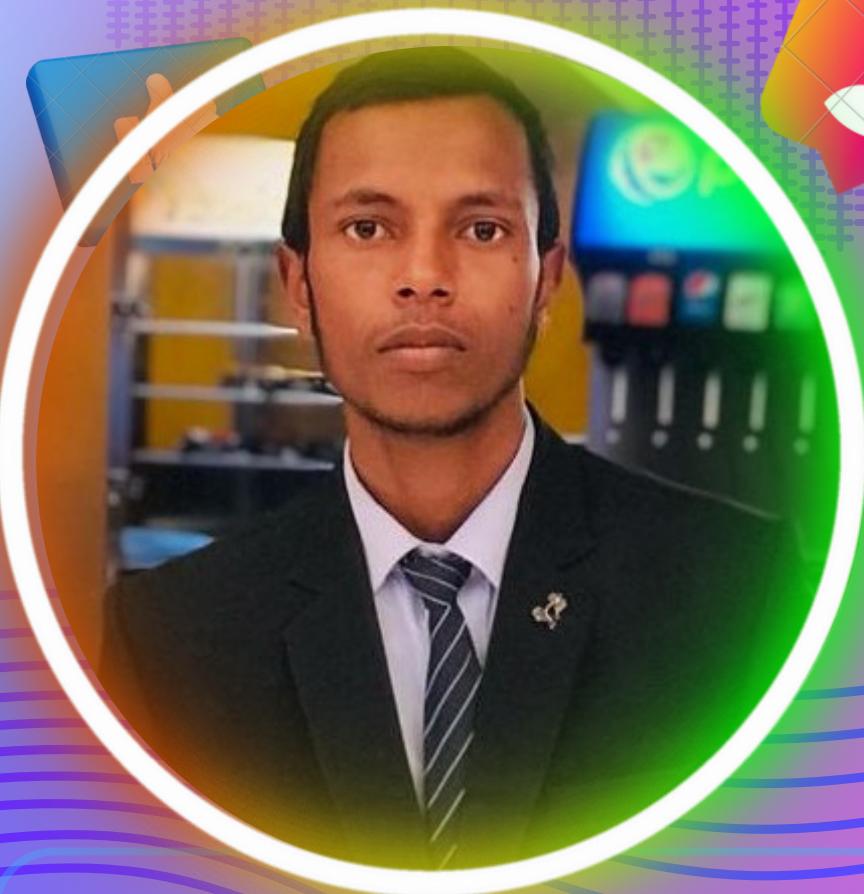
- 6. Use nested while loop to print 3 different pattern.**

- 7. Reverse a while loop to display numbers from 10 to 1.**

- 8. Reverse a while loop to display numbers from 10 to 1**

Note:- Please create a Google Document and write your answers and upload the shareable link of the Google Document with view access during the submission of the assignment.

FOLLOW FOR MORE



FOLLOW ME

