

1. Difference between == and "is"

=> We use the == operator in Python when the values of both the operands are equal. Thus, the condition would become true here.

is operator would evaluate to true whenever the variables present on either side of an operator would point towards the very same object. Otherwise, it would give us a false evaluation.

2. How to delete a key from a dictionary?

=> using the del keyword, using pop()

ex:1

```
dictValue = {"rani": 5, "rahul": 210, "santu": 10}
```

```
del dictValue['rahul']
```

ex:2

```
dictValue = {"rani": 5, "rahul": 210, "santu": 10}
```

```
removed_value = dictValue.pop('santu')
```

3. How to remove item from list

=> The remove() method removes the specified item from list.

The pop() method removes the element specified index from list.

Ex:1

```
listItem = [5, 50, 500]
```

```
listItem.remove(50)
```

Ex:2

```
listItem = [5, 50, 500]
```

```
listItem.pop(1)
```

4. What is a decorator in python?

=> A decorator is a function that takes another function as an argument and extend its functionality and return modified function with extended functionality.

The main objective of decorator function is we can extend the functionality of existing function without modifying that function.

ex:

```
def smart_decor(func):
```

```
    def inner_func(x,y):
```

```
        if x > y:
```

```
            f=func(x,y)
```

```
            return f
```

```
        else:
```

```
            return y-x
```

```
    return inner_func
```

```
@smart_decor
```

```
def sub(x,y):
```

```
    return x-y
```

```
print(sub(20,10))
```

5. What is a lambda function in python

=> A lambda function is a small anonymous function. A lambda function can take any number of arguments, but can only have one expression. In this function we can't use the def keyword.

Syntax:

lambda arguments : expression

ex:

```
x = lambda a : a + 50
print(x(50))
```

6. What is a list comprehension in python

=> List comprehension offers a shorter syntax when you want to create a new list based on the values of an existing list.

ex:

```
newlist = [x+1 for x in range(20)]
print(newlist)
```

7. What is the difference between a tuple and a list in Python?

1. lists are mutable(we can modify), tuples are immutable(we can't modify)
2. List represent square brackets "[]", tuple represent parenthesis "()"
3. List consume more memory, Tuple consume less memory
4. List ex.- list1=[10,20,30,40,50]
- tuple ex.- tuple1=(10,20,30,40,50)

8. What does "self" refer to in a class?

1. self is reference variable always pointing to current object.
2. The first argument to the Constructor and instance method is always self.
3. We are not required to pass value to the self variable, Internally PVM is responsible to provide a value.
4. The main Purpose of self variable within the class is to declare instance variables and to access values of instance variable.
5. self is not a keyword and an instance of 'self', we can use any name like "delf" or "kelf" etc, But recommended to use "self".

9. What are negative indexes and why are they used?

=> Negative indexing is used in Python to manipulate sequence objects such as lists, arrays, strings, etc. Negative indexing retrieves elements from the end by providing negative numbers.

ex:-

```
numbers = [1, 2, 3, 4, 5]
last = numbers[-1]
```

```
print(last) #5
```

10. What is the difference between a module and a package in python?

Module:

The module is a simple Python file that contains collections of functions and global variables and with having a .py extension file. It is an executable

A module is a single file (or files) that are imported under one import and used. E.g.

```
import my_module
```

Package:

The package is a simple directory having collections of modules. This directory contains Python modules and also having. __init__.py file by which the interpreter interprets it as a Package.

A package is a collection of modules in directories that give a package hierarchy.

```
from my_package.abc import a
```

11. When to use the assert keyword in python? What does it do?

=>The assert keyword is used when debugging code.

The assert keyword lets you test if a condition in your code returns True, if not, the program will raise an AssertionError.

ex.-

```
x = "puspendu"  
assert x == "Rani", "x should be 'puspendu'"
```



Puspendu Sahu

Kolkata, West Bengal- 721451

7872045257 | puspendusahoo786@gmail.com

<https://skylight4u.pythonanywhere.com/>

<https://www.linkedin.com/in/puspendu-sahu/>

Objective

Full Stack Django developer highly skilled and innovative software development with 2 years of hands-on experience designing, developing, and maintained applications. Focused and efficient learner with practical expertise in web application, a full stack developer position with Intellect.

Experience

- Aakins Global Pvt. Ltd., Bangalore** Aug 2021 - June 2023
Software engineer
I am responsible for implementing the MVT architecture in developing the web applications with the help of Django framework, and responsible for debugging and troubleshooting the programming related issues.

Education

- New Horizon College of Engineering Bangalore** 2021
Masters in Computer Application(MCA)
8.47 CGPA
- Prabhat Kumar College, Contai, West Bengal** 2018
Bachelor of Computer Application (BCA)
66.60%
- Keshary High School, West Bengal** 2015
Higher Secondary
75.4%
- Maljamuna K.P.S. Vidyapith** 2013
Secondary
58.28%

Skills

- Python/Python 3
- Django Framework
- Django Rest API(DRF)
- Django channel(WebSocket)
- PostgreSQL/SQL/SQLite
- HTML/CSS/JavaScript
- Bootstrap
- JQuery/JQuery plugins
- React Js/React Redux
- Docker
- Git/GitHub
- Full-Stack Development

Projects

- ANNA PRITHIBI SEEDS - Agriculture Domain**
Team Size: 1
Role: Freelance Web Developer
Technology: Python 3, Django ,CSS, HTML, Bootstrap, Javascript, JQuery
Responsibilities: My responsible for developing the frontend and backend modules using Python on Django

Web Framework to create a user-friendly website interface, and I contribute to analysis and design of new requirements and fault/Bug fixes.

- **Tomes - Electrical & Construction Domain**

Team Size: 3

Role: Software Engineer

Technology: Python 3, Django, DRF ,CSS, HTML, Bootstrap, Javascript, JQuery

Responsibilities: My responsible for developing the frontend and backend modules using Python on Django Web Framework to create a user-friendly website interface, and I contribute to analysis and design of new requirements and fault/Bug fixes.

- **Web Scripting (Self Project)**

In this project I implement fetching data from website and store data into the database and showing Front End.

Technology: Python (Programming Language) · Django · MySQL · HTML5 · Cascading Style Sheets (CSS) · JavaScript · Bootstrap · jQuery · AJAX.

Achievements & Awards

- Achievements & Awards NPTEL Online Certification for the course of “Introduction to R Software” of duration 8 weeks (Aug-Oct 2019) with 81% marks (Elite Silver Certificate)

Interests

- Learning New Technology
- Exploring Nature
- Cooking

Languages

- Bengali
- English
- Hindi

DECLARATION

- I hereby declare that above mentioned details are correct to the best of my knowledge and belief.