Super 50 Questions of Python part 2 by Shobhit

Q1. What is Namespaces in Python ?

- Built-in Namespace - Built-in Namespace include python built in objects. These are available everywhere until python interpreter in running.
- Global Namespace - refers variable and objects declared at the level of main program.Those global variables can be obtained by all functions that might be in your script. We can make a local variable a global variable by using global keyword.
- Local Namespace - local variables or objects which are defined within a function block and can only be accessed inside that function.
- Enclosing Namspace -

Q2. What is LEGB rule ?

This rule sets the priority of variables or objects of different scope as follow
Local Scope > Enclosing Scope > Global Scope > Built-in Scope

Q3. What is Decorator function ?

Decorator function is a function which is used to execute something before or after the intended function call.
This decorator function takes a function as argument and returns a function.
Abhishek sir example - hiding error form arrogant boss.
.

```
def dec1(func1):
    def nowexec():
        print("Executing now")
        func1()
        print("Executed")
    return nowexec

def who_is_harry():
    print("Harry is a good boy")

who_is_harry = dec1(who_is_harry)
who_is_harry()
```

There are various things needed to keep in mind

- Overwriting the target function by passing target function as parameter to decorator function.

- Creating one more function inside the decorator function and returning it (but never call it)
- The process of overwriting can be shorten by writing @_decorator_function_name just above the target function (don't write () after decorator function name.

Q4. What is Anonymous Functions in Python ?

Anonymous functions are also called **lambda functions** in Python because instead of declaring them with the standard **def** keyword, lambda keyword is used.

Anonymous functions are used when there is a requirement of a **nameless function** for a **short period of time**. These functions are **created at runtime**. Depending upon the program you need, lambda function **works in conjunction** with **filter(), map()** and **reduce():**

```
m = lambda n:n*2
print(m(3))
```

```
6
```

Lambda function with if else condition

```
mx = lambda m,n:m if m>n else n
print(mx(5,6))
```

```
6
```

Q5. What is the use of filter(), map() and reduce() function ?

filter() This function creates a sequence from a sequence, based on the condition (function/logic) provided. The function should return true or false value to check whether the particular element from sequence needed to keep or remove.

map() This function creates a sequence from a sequence , based on the condition (function/logic) provided. The function should return a calculated value according to the argument provided to it, raken form the sequence.

reduce() - This function commulates the element taken from the sequence one by one and perform the desired operation on the element. The lambda function takes two arguments one represents commoulation while other is an element taken from the sequence.reduce function comes in the **functools** package.

```
nums = [3,2,6,8,4,6,2,9]

evens = list(filter(lambda n : n%2==0,nums))

doubles = list(map(lambda n : n*2,evens))

print(doubles)

sum = reduce(lambda a,b : a+b,doubles)

print(sum)
```

Q6. What is iterator ?

Q7. What is generator ?

Generator is just like a function (declared with def keyword) but instead of return single value it returns multiple values via “yield” keyword. During iterating on a generator when flow of execution (inside generator) encounters yields keyword, it remembers the particular position from generator (at function from where it encountered the yield) and on next iteration, continue execution from last position and flow goes until the last “yield” is not encountered.

```
def gen():
    i=10
    yield i
    i+=1
    yield i
    i+=1
    yield i
for m in gen():
    print(m)

# Output
# 10
# 11
# 12
# 13
```

Q8. What is interpreted language

Q9. What is __init__() in python ?

Q10. What is slicing in python ?

Q11. What is the difference between Array and List

Q12. Explain terinary operator or conditional operator in python ?

Q13. What are the key features of python ?

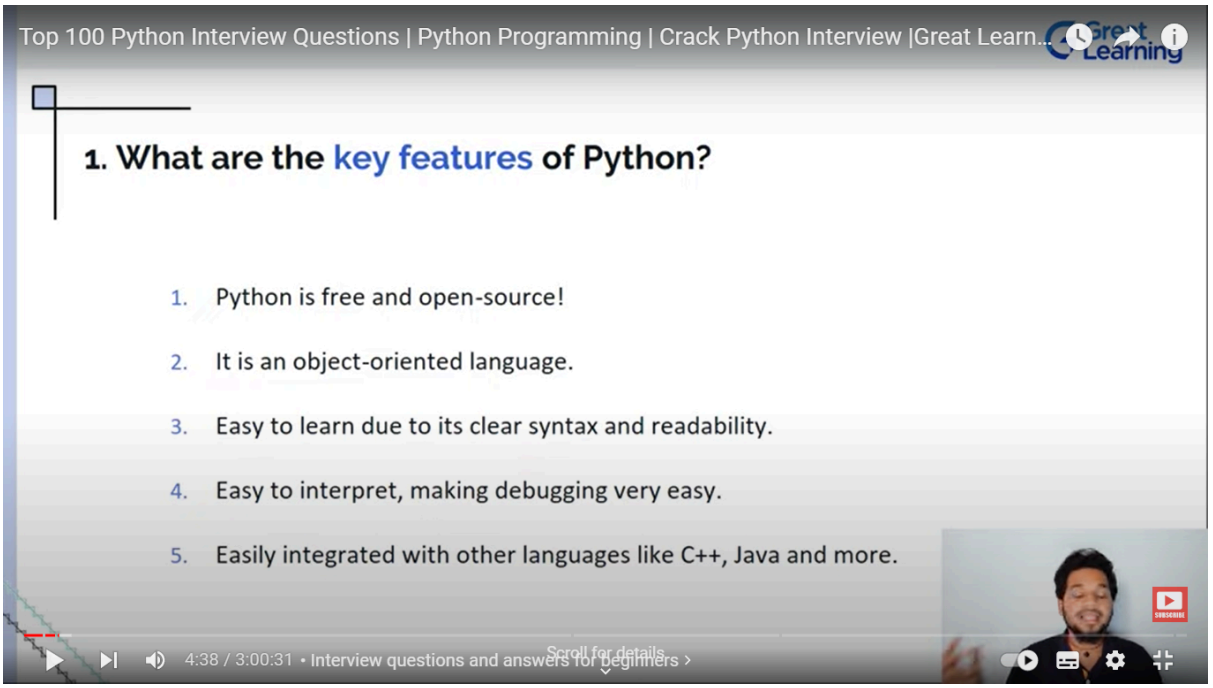
Q14. How you can concanate two tuples ?

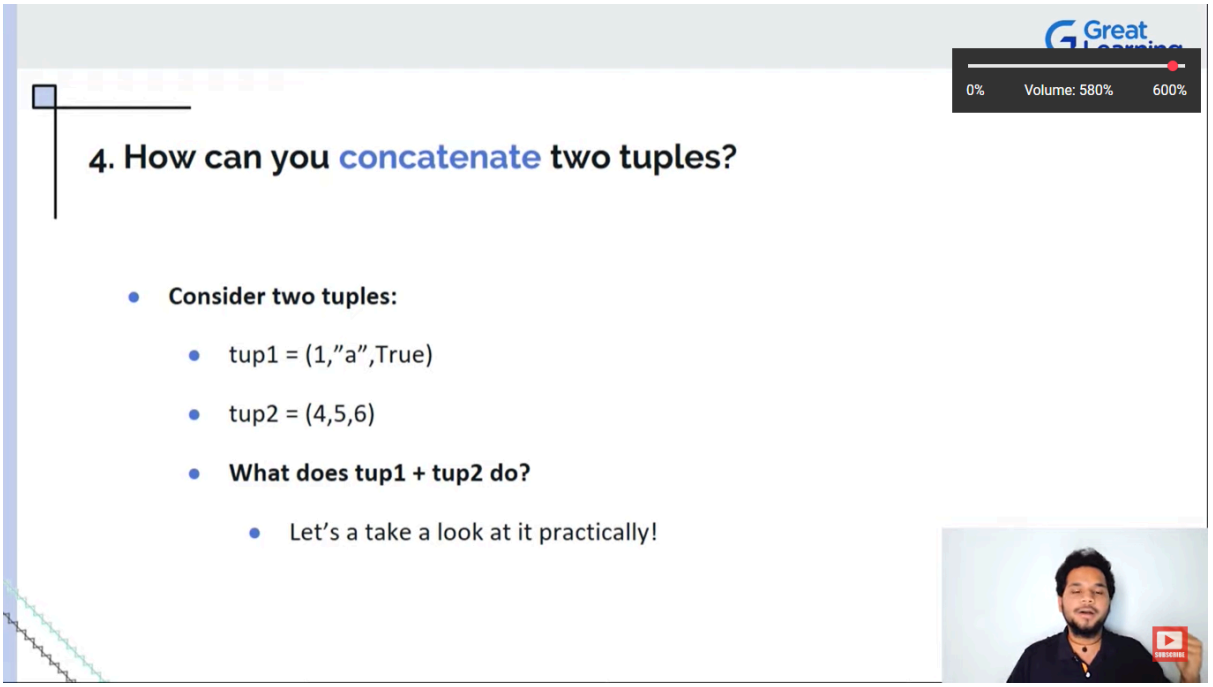
Q13.

Q13.

Q13.

Q13.


Top 100 Python Interview Questions | Python Programming | Crack Python Interview |Great Learn...
1. What are the key features of Python?
1. Python is free and open-source!
2. It is an object-oriented language.
3. Easy to learn due to its clear syntax and readability.
4. Easy to interpret, making debugging very easy.
5. Easily integrated with other languages like C++, Java and more.
4:38 / 3:00:31 • Interview questions and answers for beginners ›
Scroll for details


0%
Volume: 580%
600%
4. How can you concatenate two tuples?
Consider two tuples:
tup1 = (1,"a",True)
tup2 = (4,5,6)
What does tup1 + tup2 do?
Let's a take a look at it practically!

Great Learning

0% Volume: 580% 600%

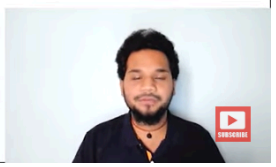
6. How can you initialize a 5x5 NumPy array with only zeroes?

- NumPy has a brilliant function – **zeros()** that does exactly this!
- n1 = np.zeros((5,5))
- Let's check out the output practically!

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

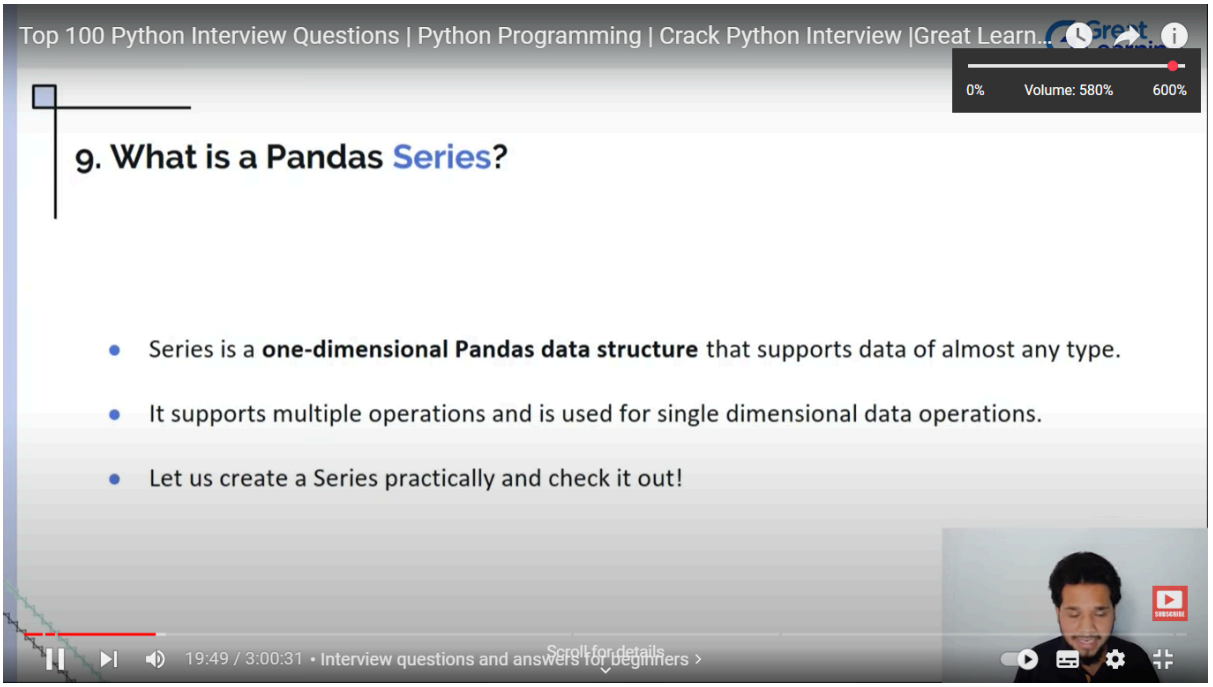
7. What is Pandas?

- Pandas is an **open-source library** that has a very rich set of data structures for data based operations.
- Pandas can deal with a **large variety of files** and is one of the most important tools to have a grip on.
- It provides two fantastic data structures – **Series and DataFrames**!



8. What are DataFrames?

- DataFrames are **mutable datatypes** present in the Pandas library.
- Provides support for heterogeneous data
- Here is how a **CSV file** can be read using Pandas:
  - df=pd.read_csv("mydata.csv")
  - Let's check it out practically!


Top 100 Python Interview Questions | Python Programming | Crack Python Interview |Great Learn...
0%
Volume: 580%
600%
9. What is a Pandas Series?
Series is a one-dimensional Pandas data structure that supports data of almost any type.
It supports multiple operations and is used for single dimensional data operations.
Let us create a Series practically and check it out!
19:49 / 3:00:31 • Interview questions and answers for beginners ›
Scroll for details