### Is python numpy better than lists?

**Ans:** We use python numpy array instead of a list because of the below three reasons:

1. Less Memory
2. Fast
3. Convenient

### How do you calculate percentiles with Python/ NumPy?

**Ans:** We can calculate percentiles with the following code

```
import numpy as np
a = np.array([1,2,3,4,5])
p = np.percentile(a, 50) #Returns 50th percentile, e.g. median
print(p)
```

### Q91. Which of the following statements create a dictionary? (Multiple Correct Answers Possible)

a) d = {}
b) d = {“john”:40, “peter”:45}
c) d = {40:”john”, 45:”peter”}
d) d = (40:”john”, 45:”50”)

**Answer:** b, c & d.

Dictionaries are created by specifying keys and values.

### Q92. Which one of these is floor division?

a) /
b) //
c) %
d) None of the mentioned

**Answer:** b) //

When both of the operands are integer then python chops out the fraction part and gives you the round off value, to get the accurate answer use floor division. For ex,

5/2 = 2.5 but both of the operands are integer so answer of this expression in python is 2. To get the 2.5 as the answer, use floor division using //. So, 5//2 = 2.5

Q95. Which of the following is an invalid statement?

a)abc=1,000,000
b) a b c = 1000 2000 3000
c)a,b,c=1000,2000,3000
d) a_b_c = 1,000,000

Answer: b) a b c = 1000 2000 3000

Spaces are not allowed in variable names.

**Q96. What is the output of the following?**

```
try:
if '1' != 1:
raise "someError"
else:
print("someError has not occured")
except "someError":
print ("someError has occured")
```

a) someError has occured
b) someError has not occured
c) invalid code
d) none of the above

**Answer:** c) invalid code

A new exception class must inherit from a BaseException. There is no such inheritance here.

**Q97. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1] ?**

a) Error
b) None
c) 25

d) 2

**Answer:** c) 25

**Q98. To open a file c:scores.txt for writing, we use**

***a) outfile = open("c:scores.txt", "r")***
***b) outfile = open("c:scores.txt", "w")***
***c) outfile = open(file = "c:scores.txt", "r")***
***d) outfile = open(file = "c:scores.txt", "o")***
**Answer:** b) The location contains double slashes ( ) and w is used to indicate that file is being written to.

**Q99. What is the output of the following?**

```
f = None

for i in range(5):
with open("data.txt", "w") as f:
    if i > 2:
        break

print f.closed
```

a) True
b) False
c) None
d) Error

**Answer:** a) True

The WITH statement when used with open file guarantees that the file object is closed when the with block exits.

**Q100. When will the else part of try-except-else be executed?**

a) always
b) when an exception occurs
c) when no exception occurs
d) when an exception occurs into except block

**Answer:** c) when no exception occurs

The else part is executed when no exception occurs.

## Q5.What is pep 8?

***Ans:*** PEP stands for **Python Enhancement Proposal.** It is a set of rules that specify how to format Python code for maximum readability.

## Q10.What are local variables and global variables in Python?

**Global Variables:**

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

**Local Variables:**

Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

## 14. Is indentation required in python?

***Ans:*** Indentation is necessary for Python. It specifies a block of code. All code within loops, classes, functions, etc is specified within an indented block. It is usually done using four space characters. If your code is not indented necessarily, it will not execute accurately and will throw errors as well.

## Q17.What is __init__?

***Ans:*** __init__ is a method or constructor in Python. This method is automatically called to allocate memory when a new object/ instance of a class is created. All classes have the __init__ method.

Here is an example of how to use it.

```
class Employee:
    def __init__(self, name, age,salary):
        self.name = name
        self.age = age
        self.salary = 20000
E1 = Employee("XYZ", 23, 20000)
# E1 is the instance of class Employee.
# __init__ allocates memory for E1.
print(E1.name)
print(E1.age)
print(E1.salary)
```