

## 78. Define PIP.

PIP stands for Python Installer Package. As the name indicates, it is used for installing different python modules. It is a command-line tool providing a seamless interface for installing different python modules. It searches over the internet for the package and installs them into the working directory without the need for any interaction with the user. The syntax for this is:

```
pip install <package_name>
```

## 79. Are there any tools for identifying bugs and performing static analysis in python?

Yes, there are tools like **PyChecker** and **Pylint** which are used as static analysis and linting tools respectively. **PyChecker** helps **find bugs** in python source code files and **raises alerts** for code issues and their complexity. **Pylint** checks for the module's coding standards and supports different plugins to enable custom features to meet this requirement.

## 80. Differentiate between deep and shallow copies.

## 82. Write python function which takes a variable number of arguments.

A function that takes variable arguments is called a function prototype.

**Syntax:**

```
def function_name (*arg_list)
```

For example:

```
def func (*var) :  
    for i in var:  
        print(i)
```

```
func(1)
```

```
func(20, 1, 6)
```

The \* in the function argument represents variable arguments in the function.

### 83. WAP (Write a program) which takes a sequence of numbers and check if all numbers are unique.

You can do this by converting the list to set by using set() method and comparing the length of this set with the length of the original list. If found equal, return True.

```
def check_distinct(data_list):  
    if len(data_list) == len(set(data_list)):  
        return True  
    else:  
        return False;  
  
print(check_distinct([1,6,5,8])) #Prints True  
print(check_distinct([2,2,5,5,7,8])) #Prints False
```

```
# Without using set() method
```

```
lst = [2,5,7,2,8]
```

```
lst2 = []
```

```
for i in lst:
```

```
    if i not in lst2:
```

```
        lst2.append(i)
```

```
if len(lst) == len(lst2):  
    print('Yes, sequence contains Unique values')  
  
else:  
    print('No, sequence contains Non Unique values')
```

**88. Write a Program to match a string that has the letter 'a' followed by 4 to 8 'b's.**

```
import re  
def match_text(txt_data):  
    pattern = 'ab{4,8}'  
    if re.search(pattern, txt_data): #search for  
pattern in txt_data  
        return 'Match found'  
    else:  
        return ('Match not found')  
print(match_text("abc")) #prints Match not found  
print(match_text("aabbbbbbc")) #prints Match found
```

**89. Write a Program to convert date from yyyy-mm-dd format to dd-mm-yyyy format.**

```
from datetime import datetime  
new_date = datetime.strptime("2021-08-01", "%Y-%m-%d").strftime("%d:%m:%Y")  
print(new_date)
```

**90. Write a Program to combine two different dictionaries. While combining, if you find the same keys, you can add the values of these same keys. Output the new dictionary**

```
from collections import Counter
```

```
d1 = {'key1': 50, 'key2': 100, 'key3':200}
```

```
d2 = {'key1': 200, 'key2': 100, 'key4':300}
```

```
new_dict = Counter(d1) + Counter(d2)
```

```
print(dict(new_dict))
```

**Output:** {'key1': 250, 'key2': 200, 'key3': 200, 'key4': 300}