Day 15 Class of DevOps

Agenda – python data types and how to identify data types in python program by using Linux commands

1. What is data types in python

Data types is a storage of value variable in specific operation that can perform on those values is known as data type.

Data types are known as

* Int data type
* Float data type
* String data type
* Boolean data type
* List data type
* Tuple data type
* Dictionary data type
* Unique number data type
* Int data type: Int data type is used to determine the integer values

Example: 20, 30, 45

* Float data type: Float data type is used to determine the decimal values

Example: 2.5, 3.4, 5.5

* String data type: String data type is used to determine the characters is denoted with single quote or double quote

Example: ‘python’ “Hello world”

* Boolean data type: Boolean data type is used to determine either True or false

Example: a=b True/false

* List data type: List data type is used to determine to store different items in single variable

Example: [apple, banana, mango]

[1, 2, 3]

[2.2, 3.2, 4.5]

* Tuple data type: Tuple is a data type is used to determine to store a unchangeable data with open braces

Example: (23, 44, 45)

* Dictionary data type: Dictionary data type is used to determine the values such as list, int, string, tuple in one variable

Example: {“Name”: “John”, “age”: 25}

* Unique number data type: Unique number data type is used to determine the values such as number like integers and floating values

Example: [1, 2, 3]

1. How to know data types in python

* To know about the int data type we need to use print(f"int\_decl\_x is {type(int\_decl\_x)}")

Example:

int\_decl\_x = 20

print(f"int\_decl\_x is {type(int\_decl\_x)}")

Out Put: int\_decl\_x is <class 'int'>

We can know every data type by using type command

int\_decl\_x = 20

float\_decl\_f = 2.5

string\_decl\_s = "sai"

bol\_decl\_is\_boolt = True

bol\_decl\_is\_boolf = False

List\_decl\_nums = [1, 2, 3]

tuple\_decl\_tup = (10, 20)

dict\_decl\_dictionary = {"name": "Sai", "age": 25}

unique\_numbers = {1, 2, 3}

print(f"int\_decl\_x is {type(int\_decl\_x)}")

print(f"float\_decl\_f is {type(float\_decl\_f)}")

print(f"string\_decl\_s is {type(string\_decl\_s)}")

print(f"bol\_decl\_boolt is {type(bol\_decl\_is\_boolt)}")

print(f"bol\_decl\_boolf is {type(bol\_decl\_is\_boolf)}")

print(f"List\_decl\_nums is {type(List\_decl\_nums)}")

print(f"tuple\_decl\_tup is {type(tuple\_decl\_tup)}")

print(f"dict\_decl\_dictionary is {type(dict\_decl\_dictionary)}")

print(f"unique\_numbers is {type(unique\_numbers)}")

Out Put:

int\_decl\_x is <class 'int'>

float\_decl\_f is <class 'float'>

string\_decl\_s is <class 'str'>

bol\_decl\_boolt is <class 'bool'>

bol\_decl\_boolf is <class 'bool'>

List\_decl\_nums is <class 'list'>

tuple\_decl\_tup is <class 'tuple'>

dict\_decl\_dictionary is <class 'dict'>

unique\_numbers is <class 'set'>

1. How to push program to another branch from local repository to remote repository

First of all we need to initialize to local repository by using git command as git init once it got initialize we need to add the files from machine to local repo by using command as git add <file\_name> once it got add we need to commit the file by using command as git commit –m <add file commit> once we commit the file we need to remote the file by using command as git remote add origin <git url> once we remote the file we need to push the file to remote repo for existing repo with new branch by using command as git push –u origin master --force once we hit enter file would be move to remote repository