



# DevNet Experts.

Topic- Python

 +91 9892028199

 [devnetexperts@gmail.com](mailto:devnetexperts@gmail.com)



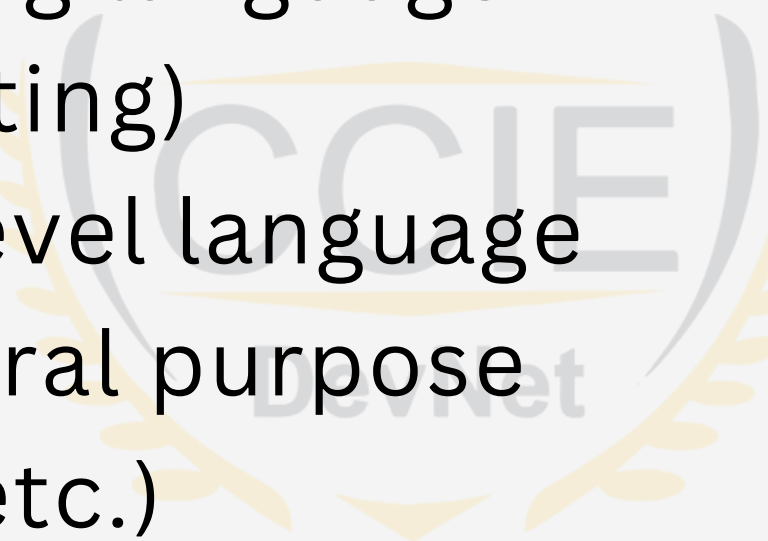
# OVERVIEW

- Python Programming Language
- Variables
- Data Types
- Functions
- Modules and Packages
- Netmiko



# Python Programming Language.

- Python was created by Guido van Rossum and first released in 1991.
- Python started as a scripting language (similar to Linux shell scripting)
- It has evolved into a high level language which can be used for general purpose programming (like Java, C etc.)
- It is very popular as it has low barrier of entry.
- It is very easy to learn due to its simple english like syntax.

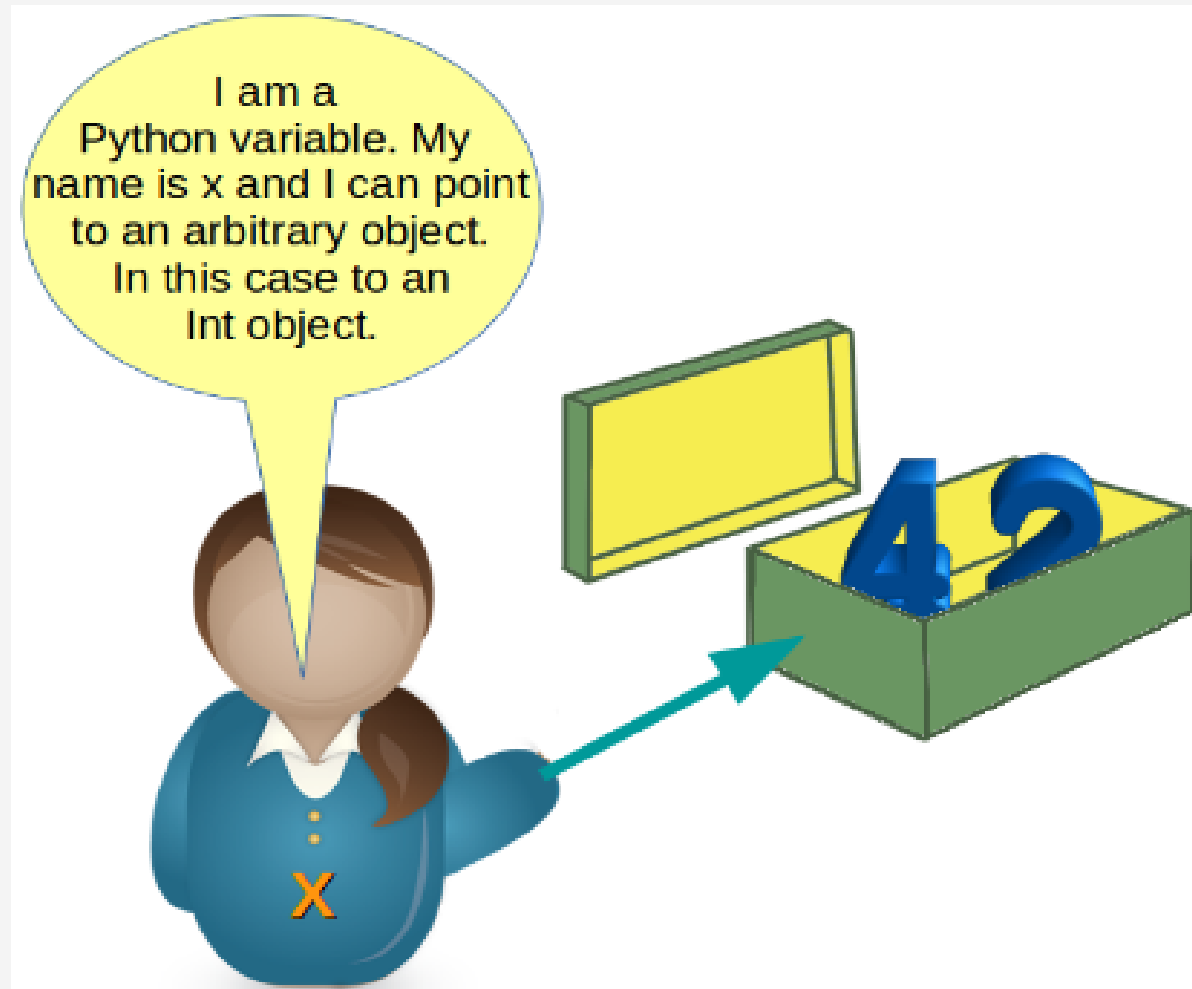


# Python Programming Language (cont.)

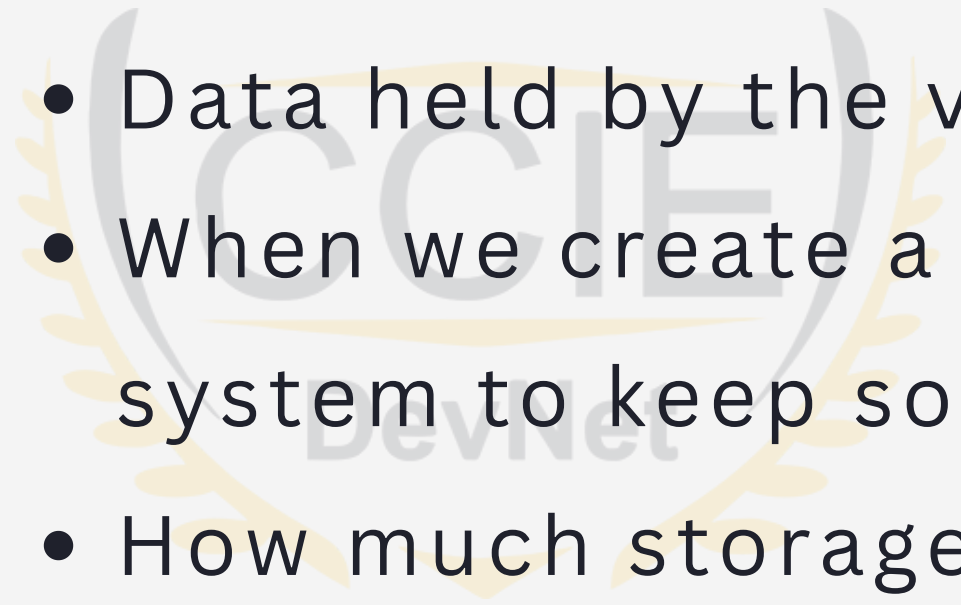
- It has wide range of applications ranging from Web Development, Scientific and Mathematical Computing to Desktop Graphical User Interfaces.
- Python interpreters are available for many operating systems including network operating systems.
- Cisco puts Python on many of its devices and releases many tools using Python.
- This makes Python a great choice for network engineers looking to add programming to their skillset.



# Variables.



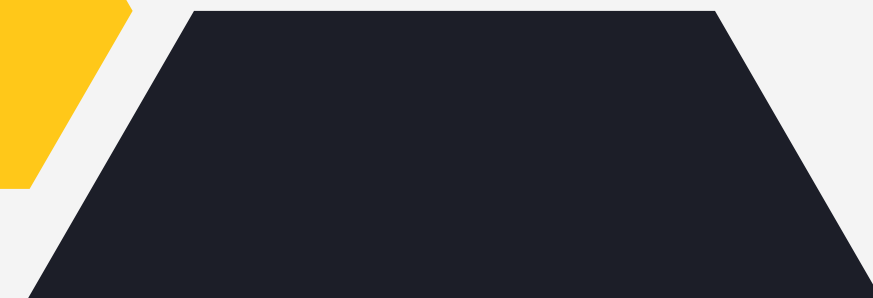
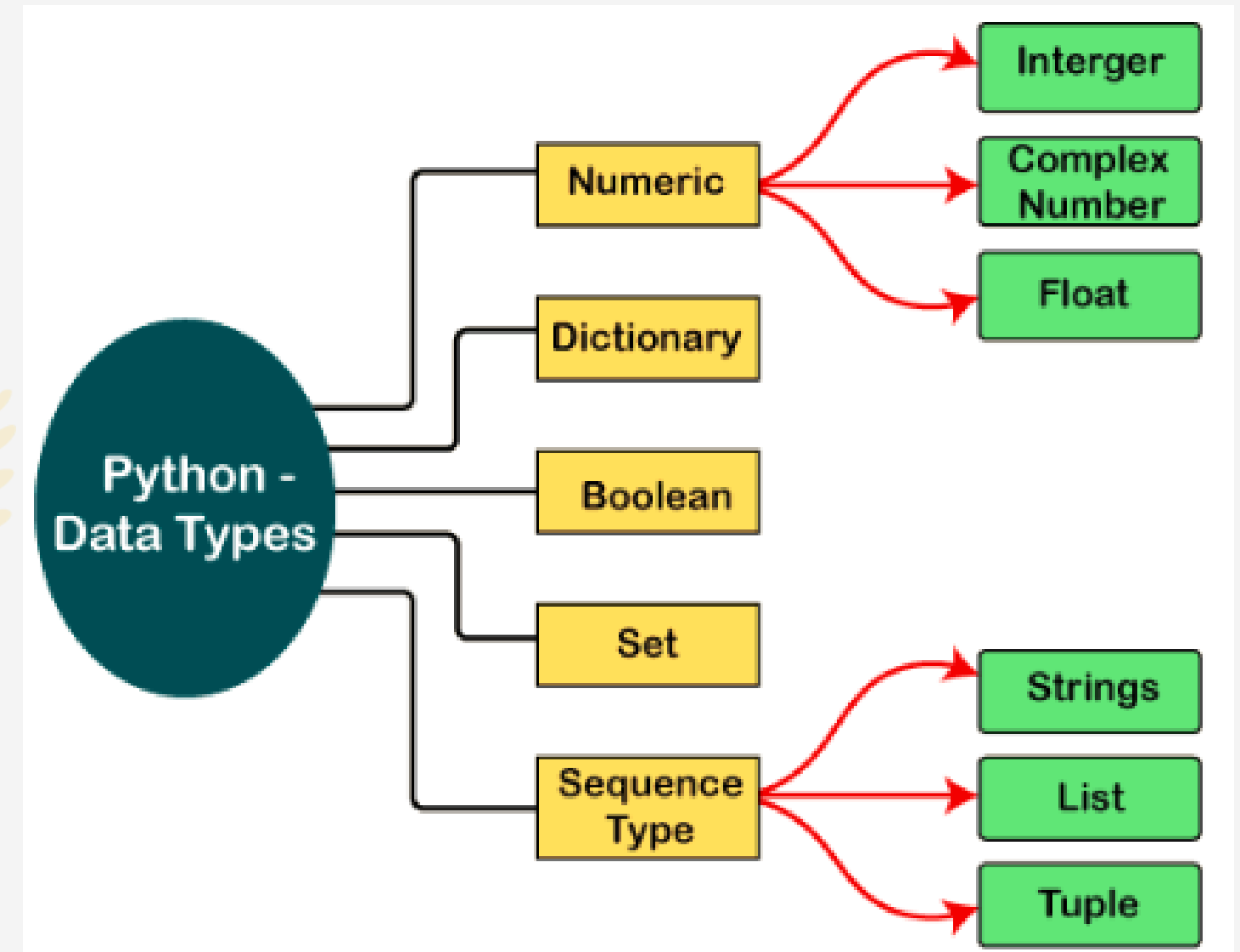
- Variables are named locations used to store data in memory.
- Think of it like a container that holds data.
- Data held by the variable can change later.
- When we create a variable, we are telling the system to keep some storage aside.
- How much storage to keep aside depends on the type of data that is going to be stored in that variable.



# Data Types

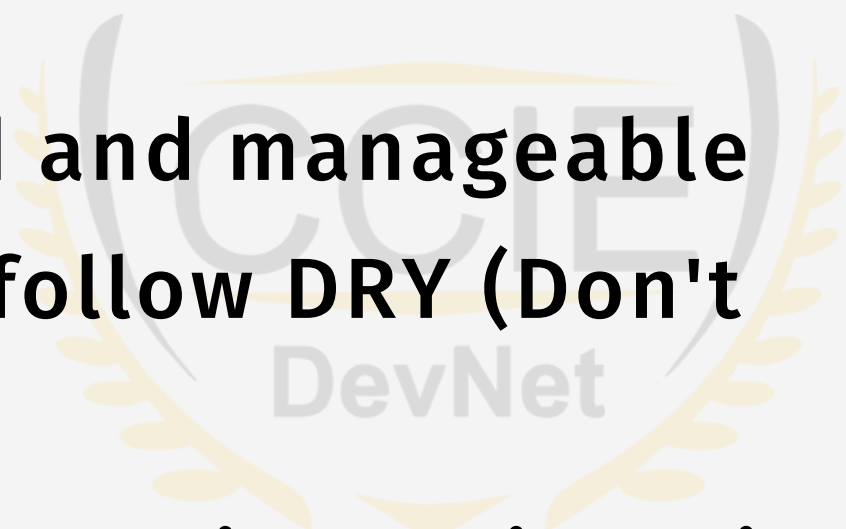
- Different types of data that can be stored in Python variables fall into the following categories.

1. Text Type: Strings (str)
2. Numeric Types: Numbers (int, float, complex)
3. Sequence Types: Lists, Tuples (list, tuple)
4. Mapping Type: Dictionaries (dict)
5. Boolean Type: bool (true/false)



# Functions.

- **Function is a group of statements that performs a specific task.**
- **Helps to break down large programs into small modular chunks.**
- **Helps keep code organized and manageable**
- **More importantly helps us follow DRY (Don't Repeat Yourself) principle.**
- **We write function once and call it multiple times**



## Syntax

```
def function_name(parameters):  
    """docstring"""  
    statement(s)
```

# Functions (Cont.)

```
import ipaddress

def check_ip(ip):
    try:
        print(ipaddress.ip_address(ip))
        print('IP Valid')
    except:
        print('-' * 50)
        print('IP is not valid')

while True:
    ip = input('Enter IP Address: ')
    if ip != 'q':
        check_ip(ip)
    else:
        print('Script finished')
        break
```

IE  
Net





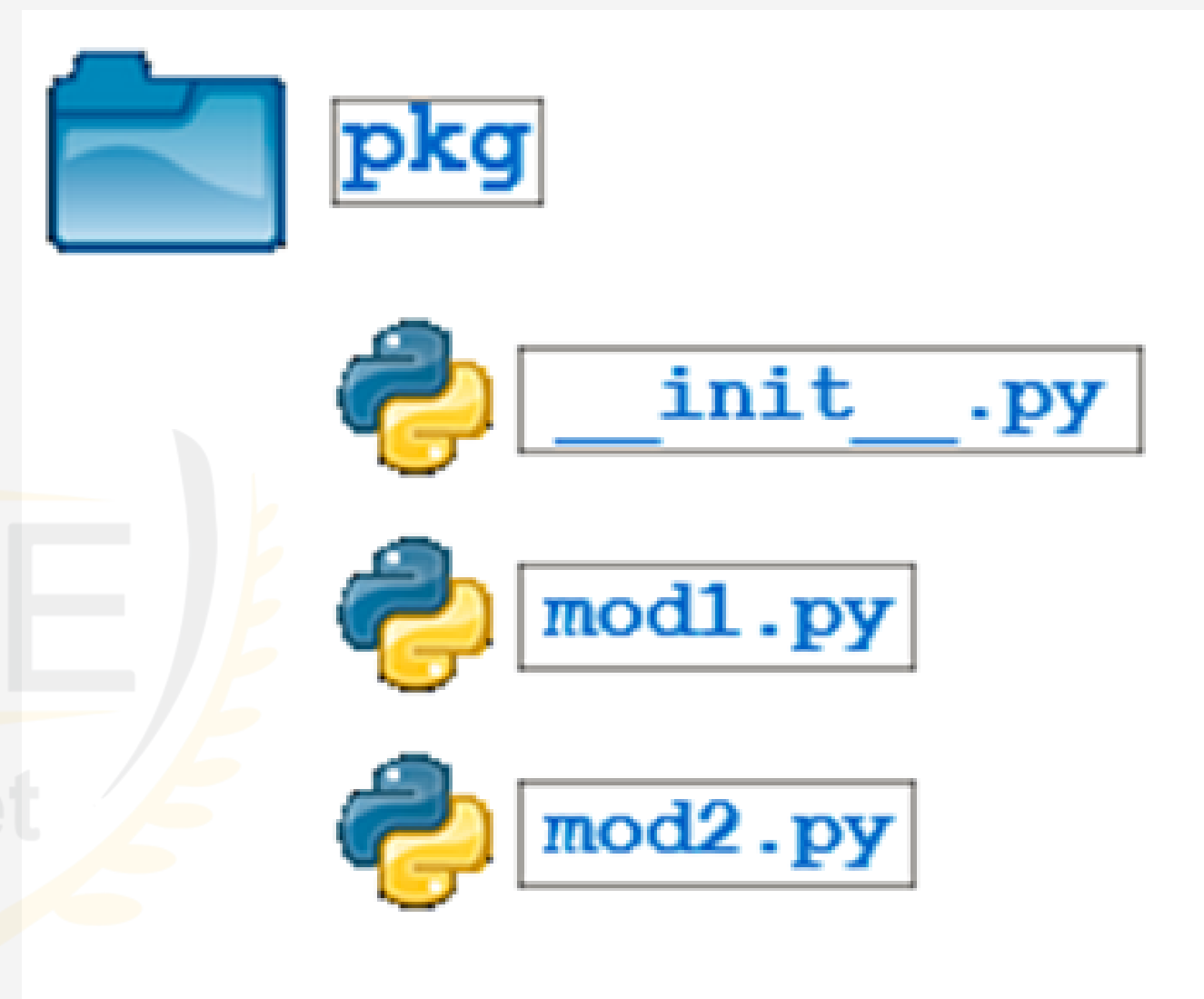
# Modules and Packages

- **Modules:**

1. Collection of functions and global variables stored as a file with .py extension.
2. It is an executable file.

- **Packages:**

1. Simple directory containing collection of modules and sub packages (sub directories).
2. It has a `__init__.py` file which is used by Python interpreter to identify this directory as Python package



# Netmiko:

- **Paramiko is a Python implementation of SSHv2 protocol.**
- **Netmiko is a multi-vendor library to simplify Paramiko SSH connections to network devices.**
- **It can-**
  1. Establish ssh connection to various devices
  2. Simplify execution of show commands used to retrieve config data.
  3. Simplify execution of configuration commands and possibly.
  4. Support working with a broad range of devices from different vendors.



# Netmiko (cont.)

```
from netmiko import ConnectHandler

#First create the device object using a dictionary
CSR = {
    'device_type': 'cisco_ios',
    'ip': '192.168.1.220',
    'username': 'roger',
    'password': 'cisco'
}

# Next establish the SSH connection
net_connect = ConnectHandler(**CSR)

# Then send the command and print the output
output = net_connect.send_command('show ip int brief')
print (output)

# Finally close the connection
net_connect.disconnect()
```

Netmiko sample code to ssh to Cisco IOS device.



DEMO.