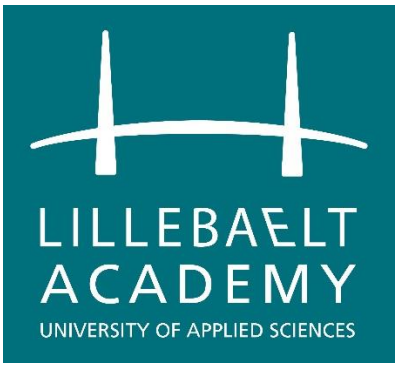
Bo Mikkelsen Mikkelsen.bo@gmail.com Erhvervsakademiet Lillebælt
Odense

IT Technology

Assignment 8 IP-Addresses



LILLEBAELT ACADEMY
UNIVERSITY OF APPLIED SCIENCE

Author
Bo Mikkelsen
Mikkelsen.bo@gmail.com

Thursday 19-01-2018

Bo Mikkelsen Mikkelsen.bo@gmail.com Erhvervsakademiet Lillebælt
Odense

Table of Contents

Bo Mikkelsen Mikkelsen.bo@gmail.com Erhvervsakademiet Lillebælt
Odense

1 Introduction

This assignment will consist of five different programs that are from tasks from the book “starting out with python second edition, programmed in python of course.

Bo Mikkelsen Mikkelsen.bo@gmail.com Erhvervsakademiet Lillebælt
Odense

2 Week 3 Programming 8 tasks

2.1 Exercise 1: valid: Ipv4-Address

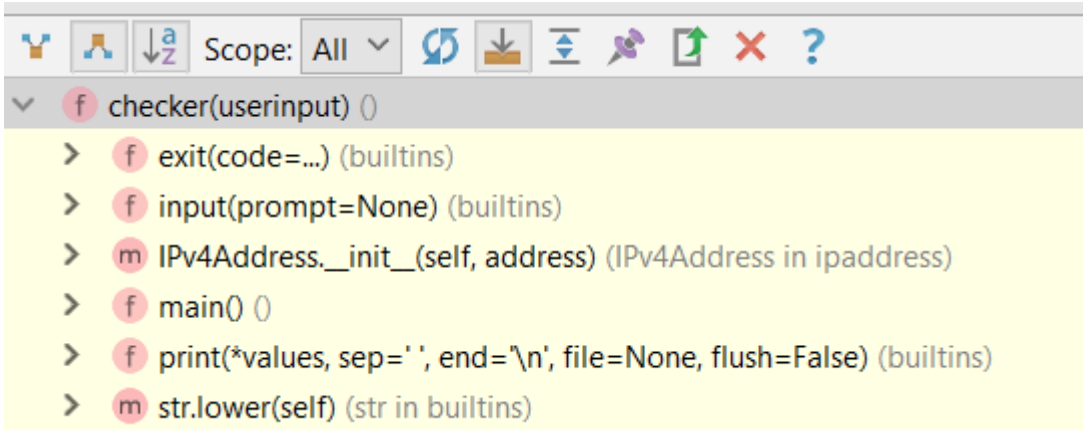
My program will let the user enter a ip-address(in ipv4) without a subnet mask. The program will then check the IP-address and tell if the address is valid or not. It will display if the address is public or private.

Here is the source code for the program is made:

```
#Bo Mikkelsen
#date 19-01-2018
#Exercise 1
#This program will tell if a given IP-address is valid and can be used.

#importing the IP address
import ipaddress

def main():

  userinput = input('Enter a IP-Addres in the format (192.168.4.1):')
  print('-'*25)
  checker(userinput)

def checker(userinput):

  try:
    valid=ipaddress.IPv4Address(userinput)
    print(valid,'This is a Valid address')

    if valid.is_private is True:
      print('-'*25)
      print('The address is a private address')

    elif valid.is_global is True:
      print('-'*25)
      print('The address is a public address')

  except ValueError:
    print('ValueError pls try again, Hint: do not use subnet mask:')
    main()

  else:
    print('-'*25)
    Again = str.lower(input('Do you want to try again?, y = yes, n = no:'))
    if Again == 'y':
      main()

    else:
      exit()

main()
```

I import the IP-address module to make life a little easier, while keeping the code short

Here is a Pycharm hierarchy diagram of my functions



It tells the functions used in the checker defined function.

Here is a proof of my exception handling in action and public and private address display.
I use both lower and upper characters in the try again statement I did, to show that they both work.

```
C:\Users\mikke\AppData\Local\Programs\Python\Python36-32\python.exe C:/Users/
Enter a IP-Addres in the format (192.168.4.1):192.168.4.2/24
-------------------------
ValueError pls try again, Hint: do not use subnet mask:
Enter a IP-Addres in the format (192.168.4.1):192.168.4.2
-------------------------
192.168.4.2 This is a Valid address
-------------------------
The address is a private address
-------------------------
Do you want to try again?, y = yes, n = no:y
Enter a IP-Addres in the format (192.168.4.1):212.189.235.123
-------------------------
212.189.235.123 This is a Valid address
-------------------------
The address is a public address
-------------------------
Do you want to try again?, y = yes, n = no:N

Process finished with exit code 0
```

The exception handling is mostly used for the user writing a subnet mask, which is not intended in this program.

2.2 Exercise 2: Ip-Address subnet

My program will ask the user for a private IPv4-Address with a subnet mask. And the program will display if the IP-address is valid for private and what subnet the address belongs to.

Bo Mikkelsen Mikkelsen.bo@gmail.com Erhvervsakademiet Lillebælt
Odense

Here is the source code I made for the program:

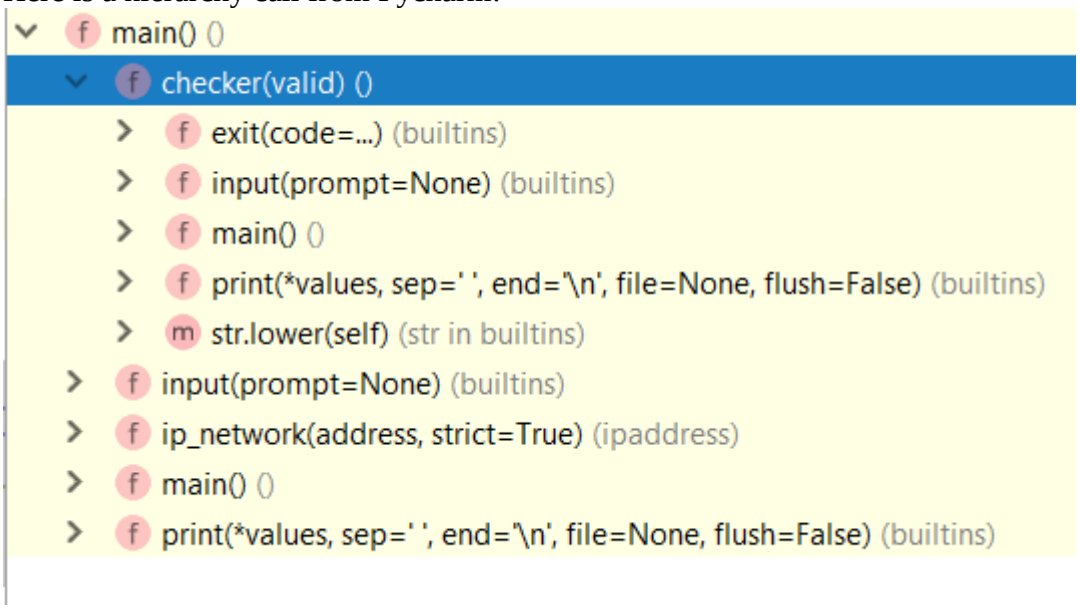
```
#Bo Mikkelsen
#date 19-01-2018
#Exercise 1
#This program will test if an entered ip address is a valid address in a subnet

#importing ip-address moduele
import ipaddress

def main():
  userinput = input("Enter a private network address in',’\n’
 CIDR format(ex.192.168.1.60/24): ")

  try:
    valid = ipaddress.ip_network(userinput,strict=False)
    if valid.is_private is True:
      print('-' * 25)
      pass

    elif valid.is_global is True:
      print('-' * 25)
      print('The address entered is public, and therefore can not',’\n’
 have a subnet mask:')
      print('Try again with a private')
      main()

    elif valid.is_multicast is True:
      print('-' * 25)
      print('The address entered is a multicast address, and therefore''\n'
            ' can not have a subnet mask:')
      print('Try again with a private address')
      main()

  except ValueError:
    print('-' * 25)
    print('This is not a valid address')
    print('Try again pls')
    main()

  else:
    checker(valid)

def checker(valid):
  print('-'*25)
  print('The subnet for that address is:',valid)
  print('This is a valid address in this subnet')
  Again = str.lower(input('Do you want to try again?, y = yes, n = no:'))

  if Again == 'y':
    main()

  else:
    print('-' * 25)
    print('The program is closing:')
    exit()
```

Bo Mikkelsen Mikkelsen.bo@gmail.com Erhvervsakademiet Lillebælt
Odense

```
main()
```

I use the ip-address module for making it a lot easier. This also gives me the option to check if the address entered is private, public or a multicast, to make the program better.

Here is a hierarchy call from Pycharm:



It shows the functions in the checker defined function, and which functions is used through the program.

Here is a working example of my program:

```
C:\Users\mikke\AppData\Local\Programs\Python\Python36-32\python.exe C:/Users/mikke/.PyChar
Enter a private network address in CIDR format(ex.192.168.1.60/24): 123.523.12.576/24
-------------------------
This is not a valid address
Try again pls
Enter a private network address in CIDR format(ex.192.168.1.60/24): 182.61.66.92/30
-------------------------
The address entered is public, and therefore can not have a subnet mask:
Try again with a private
Enter a private network address in CIDR format(ex.192.168.1.60/24): 172.16.4.2/30
-------------------------
-------------------------
The subnet for that address is: 172.16.4.0/30
This is a valid address in this subnet
Do you want to try again?, y = yes, n = no:n
-------------------------
The program is closing:

Process finished with exit code 0
```

Here I check my exception handling, with writing an address that is not valid.
Then I try with a public address.
The program gives me an error message and lets me try again, this time I use a private address as I should, and the program shows me the subnet of that ip-address. I say no to try again and the program closes.