Αυτοματοποιημένη Διαχείριση Συστημάτων

Διάλεξη 3η

Θωμάς Καμαλάκης

Χαροκόπειο Πανεπιστήμιο Αθηνών

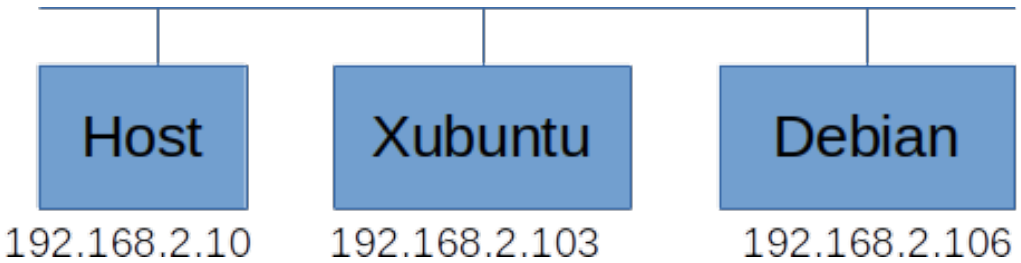
Οκτώβριος 2022

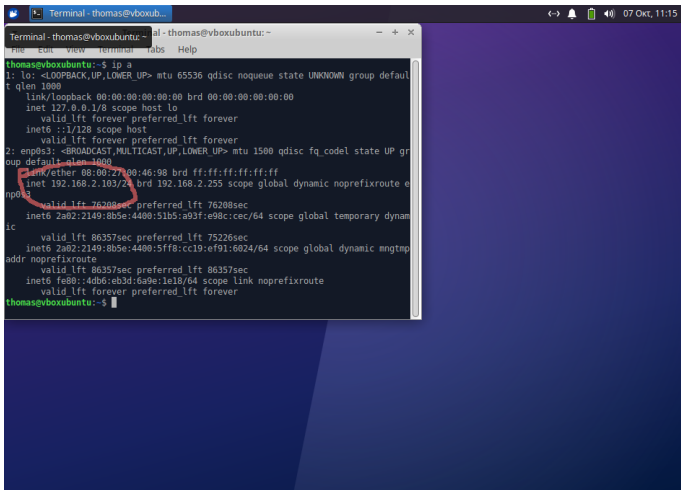
Περιεχόμενα

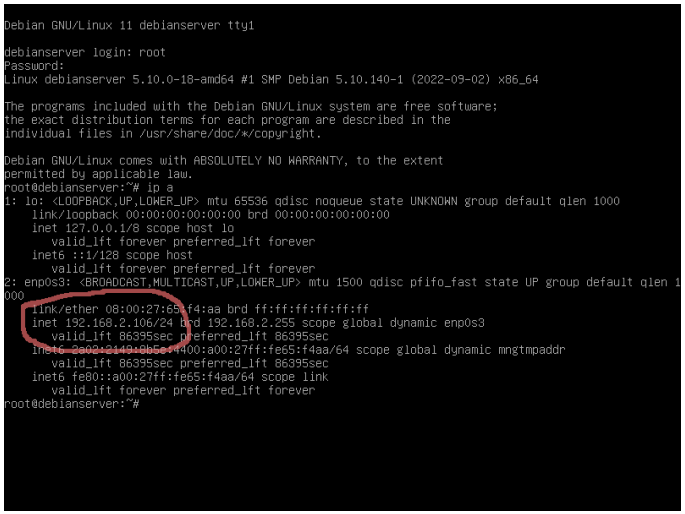
Networking

- Μέχρι τώρα έχουμε φτιάξει δύο ιδεατές μηχανές
- Μία που είναι ο client (xubuntu)
- Η άλλη είναι ο server (debian)
- Πως αυτά βλέπουν το ένα το άλλο;

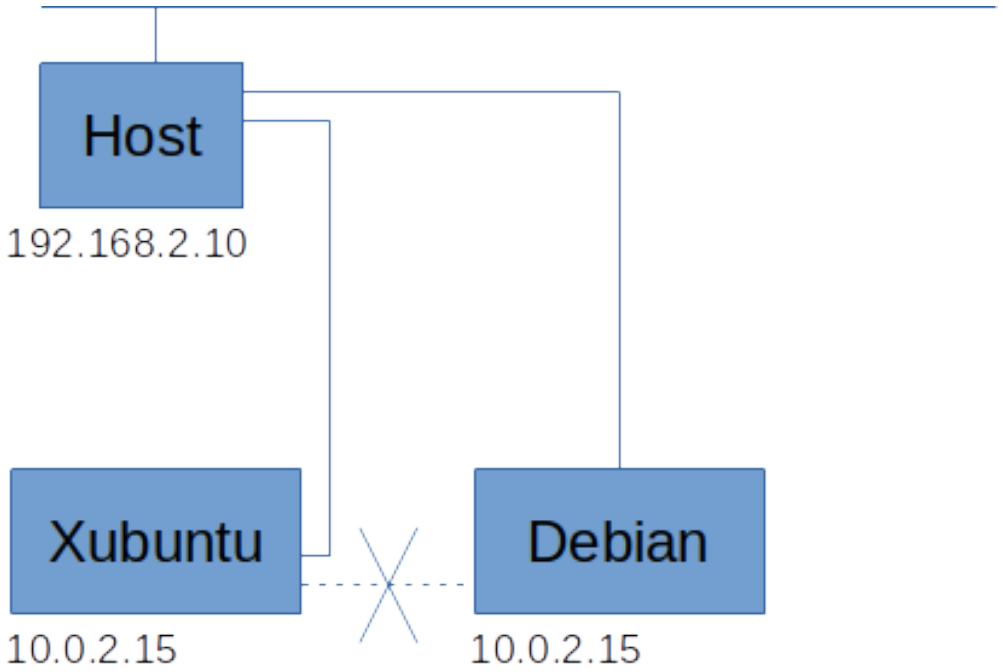
Bridged Networking



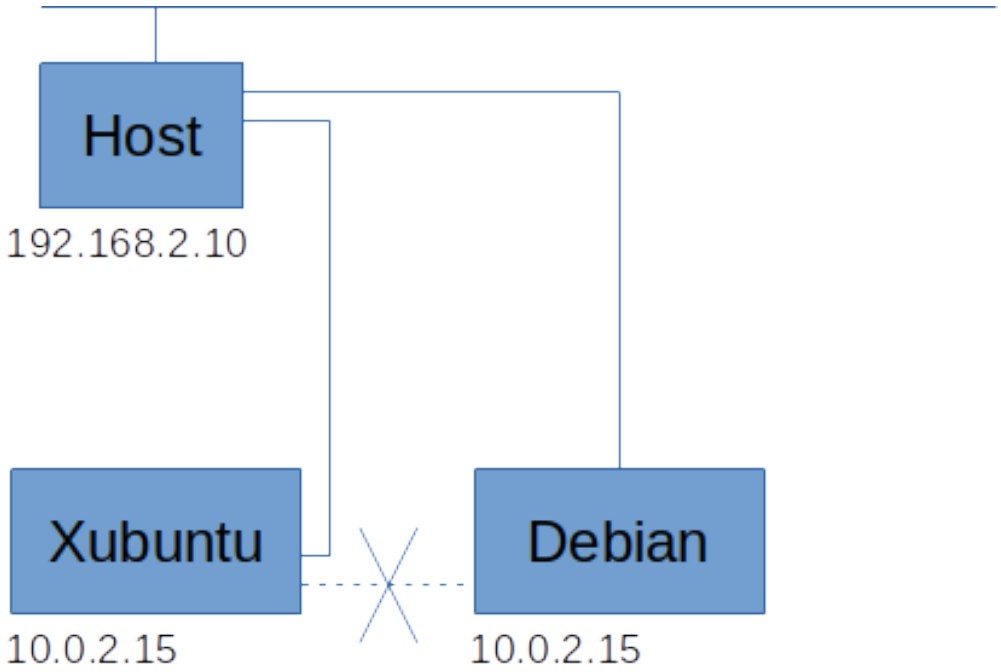




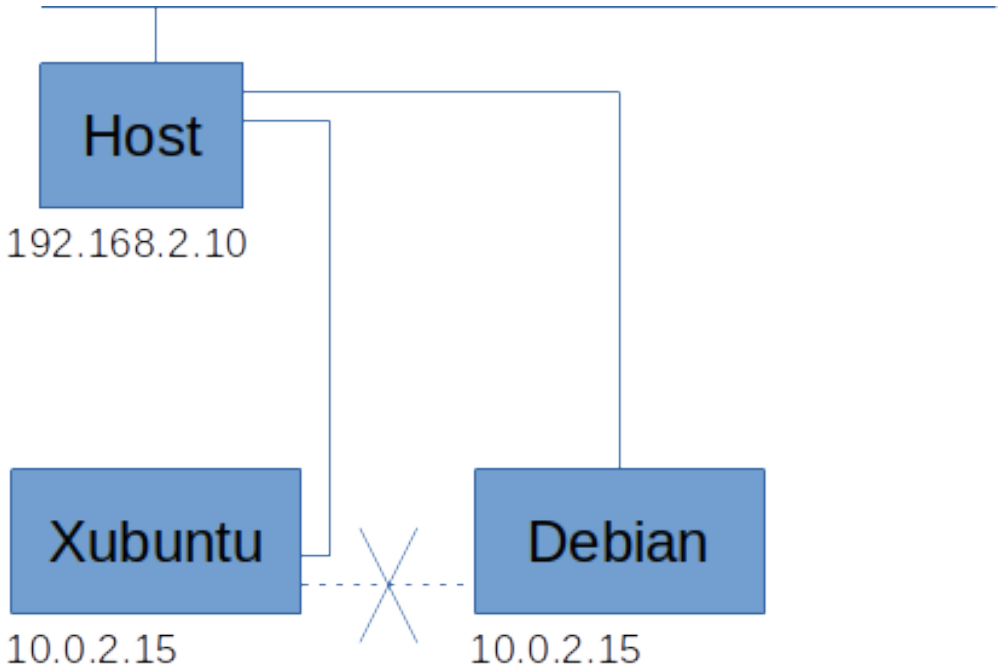
NAT Networking



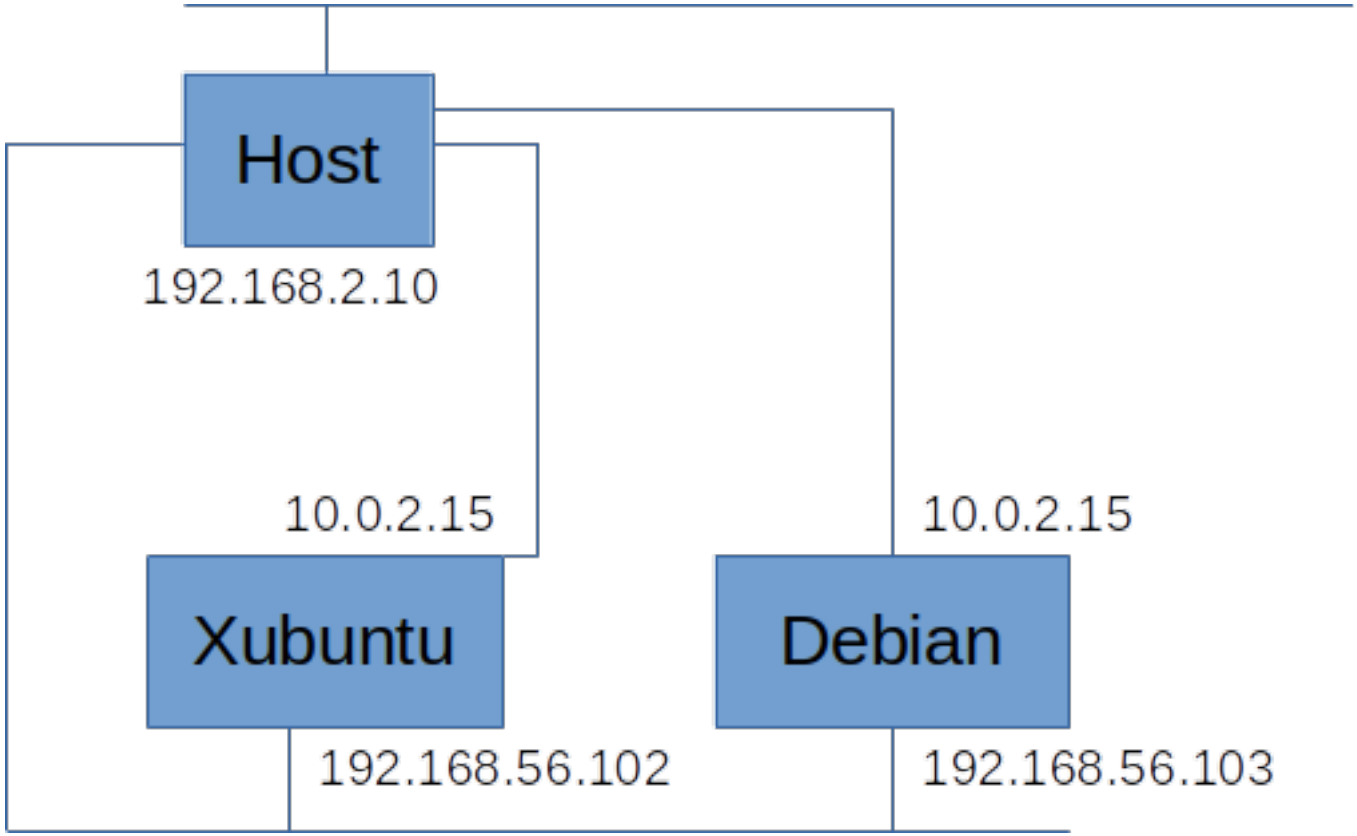
NAT Networking



NAT Networking



NAT Networking + Host-only Network

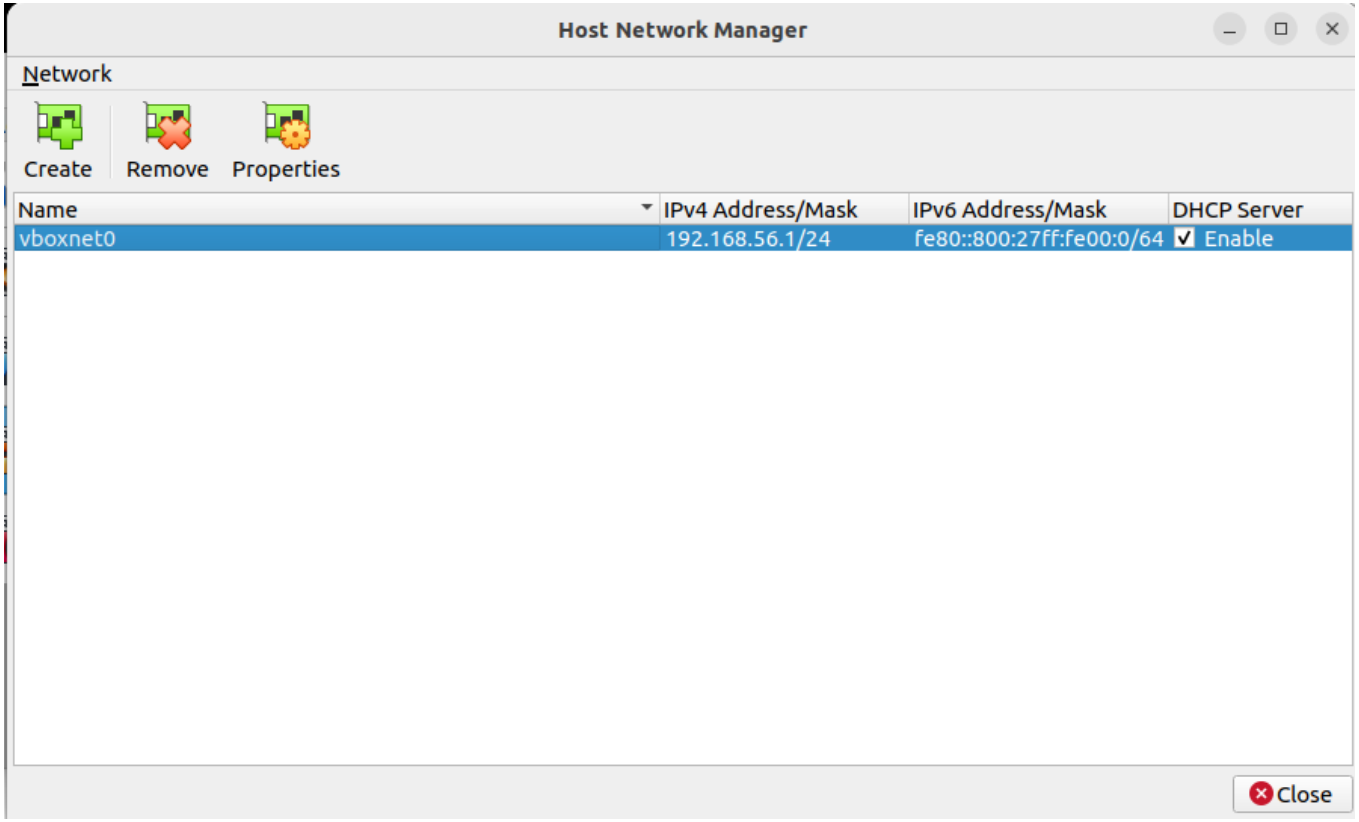


NAT Networking + Host-only Network

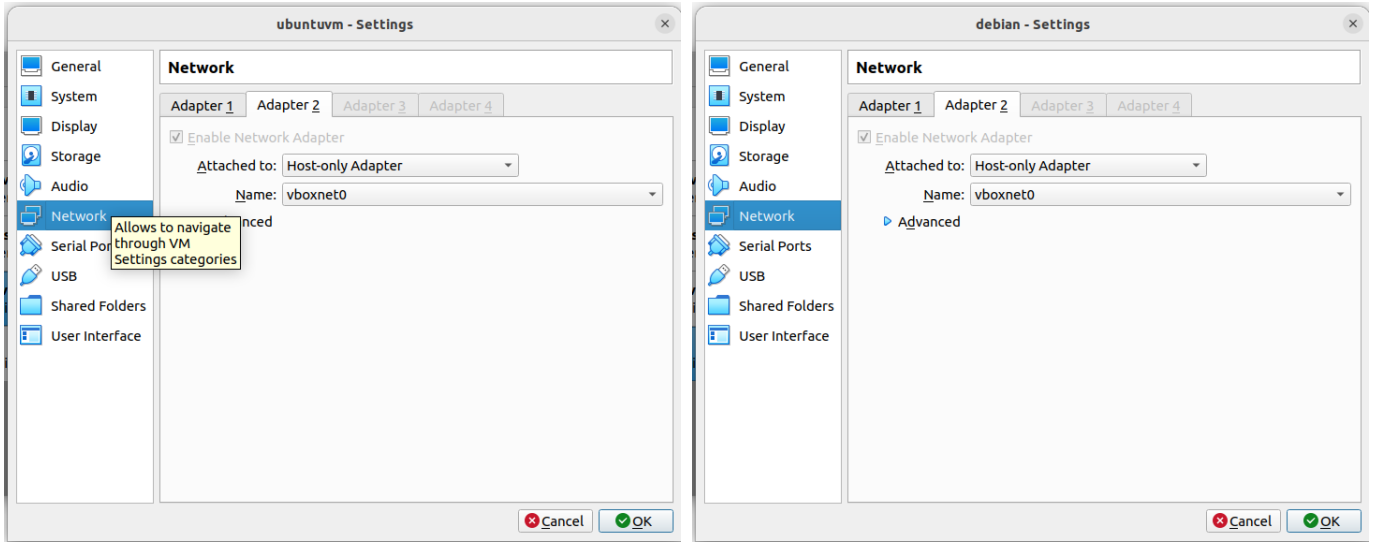
Με αυτή την αρχιτεκτονική τα δύο μηχανήματα:

- έχουν internet
- επικοινωνούν μεταξύ τους
- τα βλέπει και ο host

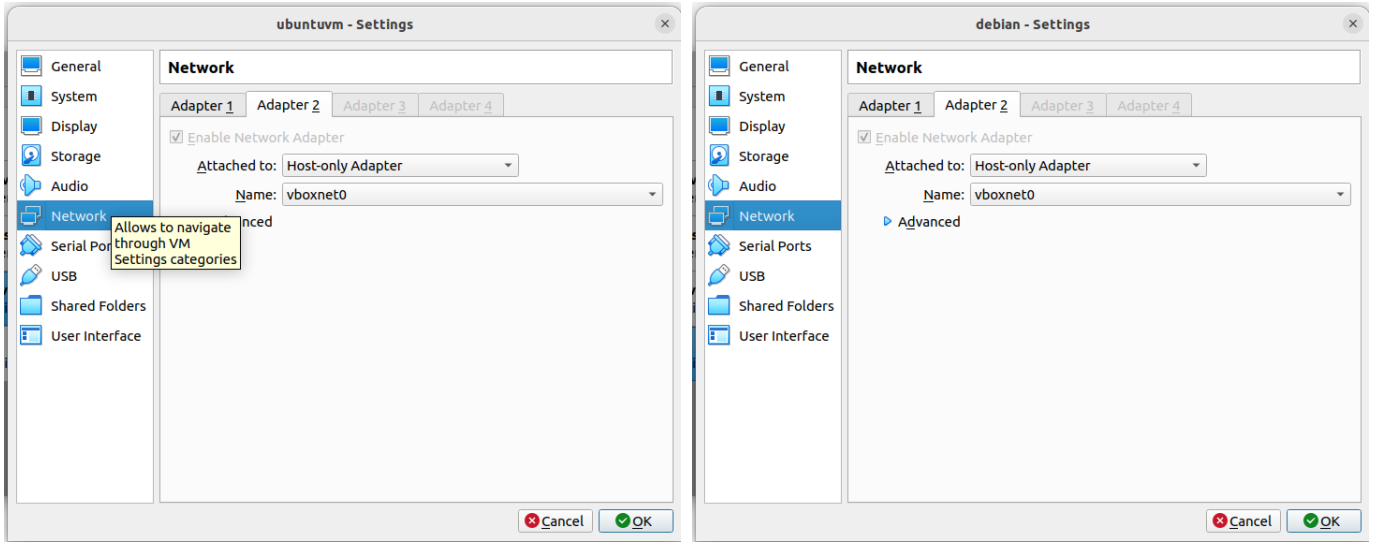
NAT Networking + Host-only Network



NAT Networking + Host-only Network



NAT Networking + Host-only Network



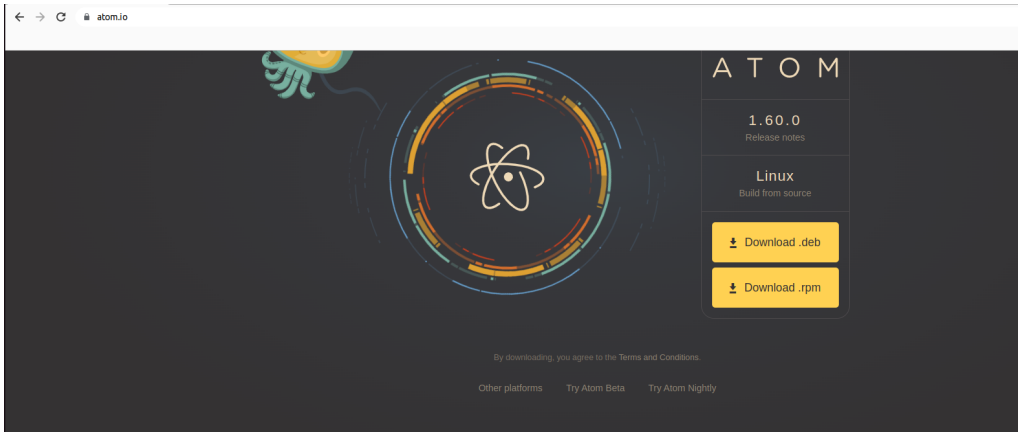
NAT Networking + Host-only Network

```
  GNU nano 5.4                          /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp

# virtualbox host-only network
allow-hotplug enp0s8
iface enp0s8 inet dhcp

                                [ Read 16 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Oct  7 18:31:31 EEST 2022 on tty1
root@debianserver:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:65:f4:aa brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 86394sec preferred_lft 86394sec
    inet6 fe80::a00:27ff:fe65:f4aa/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:78:d5:2e brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 594sec preferred_lft 594sec
    inet6 fe80::a00:27ff:fe78:d52e/64 scope link
       valid_lft forever preferred_lft forever
root@debianserver:~# ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=1.34 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.738 ms
^C
--- 192.168.56.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.738/1.040/1.343/0.302 ms
root@debianserver:~#
```

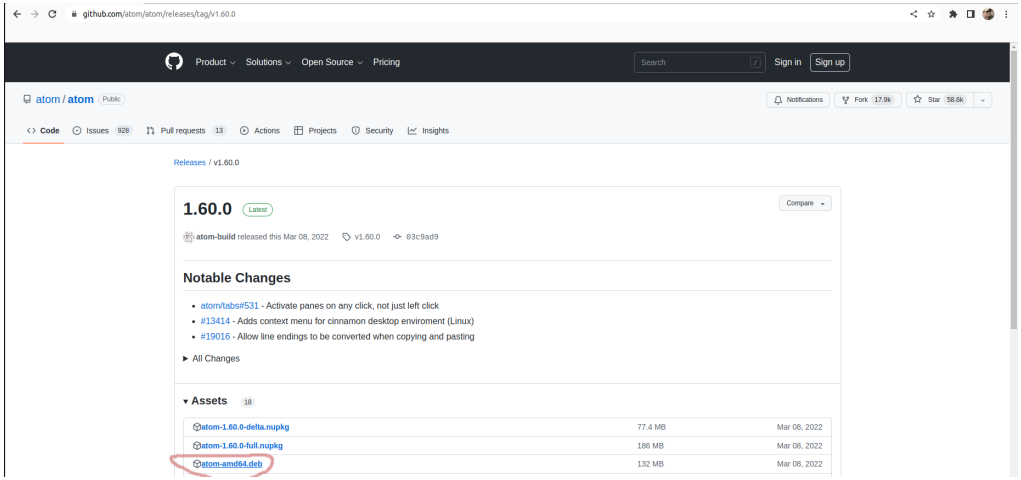
Εγκατάσταση του Atom

- Ένας καλός Editor πάντα χρειάζεται
- Μπορούμε να βάλουμε τον Atom, `https://atom.io`



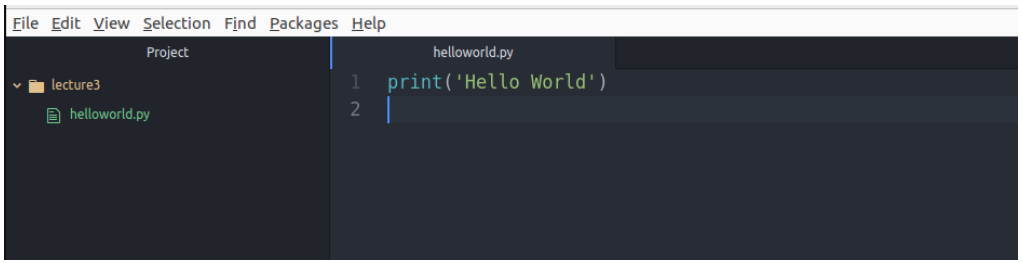
Εγκατάσταση του Atom

- Κατεβάζουμε το .deb αρχείο στο Xubuntu και γράφουμε σε ένα terminal lstinline|sudo dpkg -i atom-amd64.deb|.



Hello world!

- Φτιάχνουμε ένα αρχείο helloworld.py στον Atom που περιέχει μόνο μία εντολή
- print('Hello world!')
- το τρέχουμε με python3 helloworld.py στο terminal




```
thomas@kirklaptop:~/Documents/mscpython/code/lecture3$ python3 helloworld.py
Hello World
thomas@kirklaptop:~/Documents/mscpython/code/lecture3$
```

Hello world - C

```c
#include <stdio.h>

int main() {
    printf("Hello World");
    return 0;
}
```

Hello world - C++

```cpp
#include <iostream>

int main() {
    std::cout << "Hello World
    return 0;
}
```

Hello world - C#

```
namespace HelloWorld
{
    class Hello {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World");
        }
    }
}
```

Hello world - COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. Hello-world.
PROCEDURE DIVISION.
    DISPLAY "Hello World".
```

Hello world - Java

```
'
```

```java
/*package whatever //do not write package name here */

import java.io.*;

class GFG {
    public static void main (String[] args) {
       System.out.println("Hello World");
    }
}
```

Hello world - Javascript

```
console.log("Hello World");
```

Ιστορία

- Ξεκινάει το 1989 από τον Guido van Rossum
- Θεωρείται Benevolent dictator for life
- Πήρε το όνομα της από τους Monty Python
- Η έκδοση 2.0 ξεκίνησε το 2000
- Το 2008 βγήκε η έκδοση 3.0
- Προσοχή! Ποτέ δεν γράφουμε Python 2!

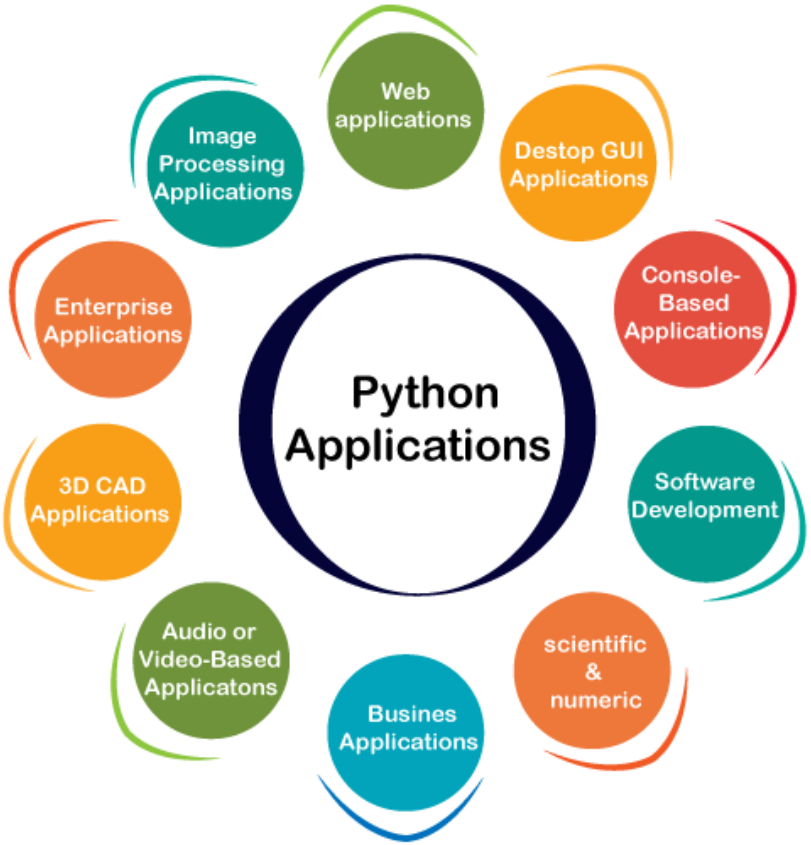
Zen of Python

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren’t special enough to break the rules.
- Although practicality beats purity.

Zen of Python

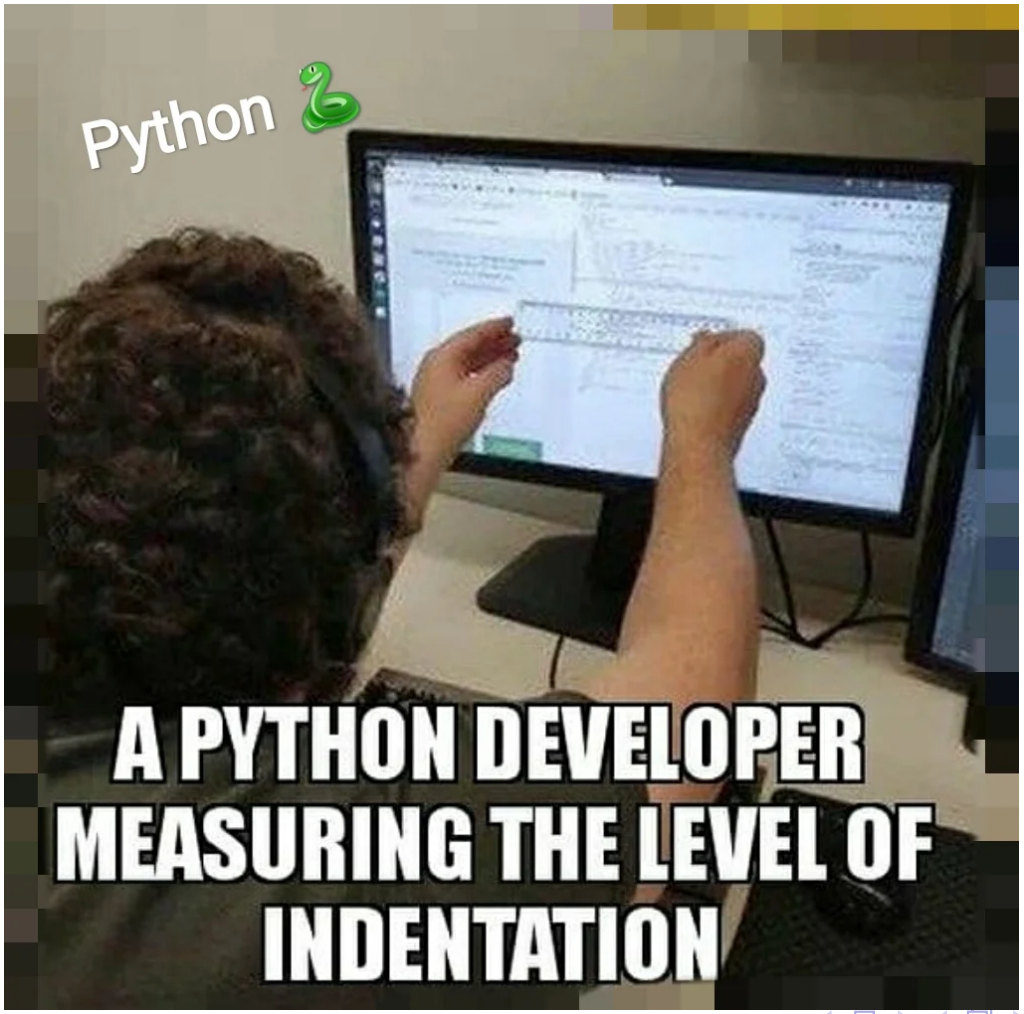
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one– and preferably only one –obvious way to do it.
- Although that way may not be obvious at first unless you’re Dutch.
- Now is better than never.
- Although never is often better than *right* now.
- If the implementation is hard to explain, it’s a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea – let’s do more of those!

Εφαρμογές

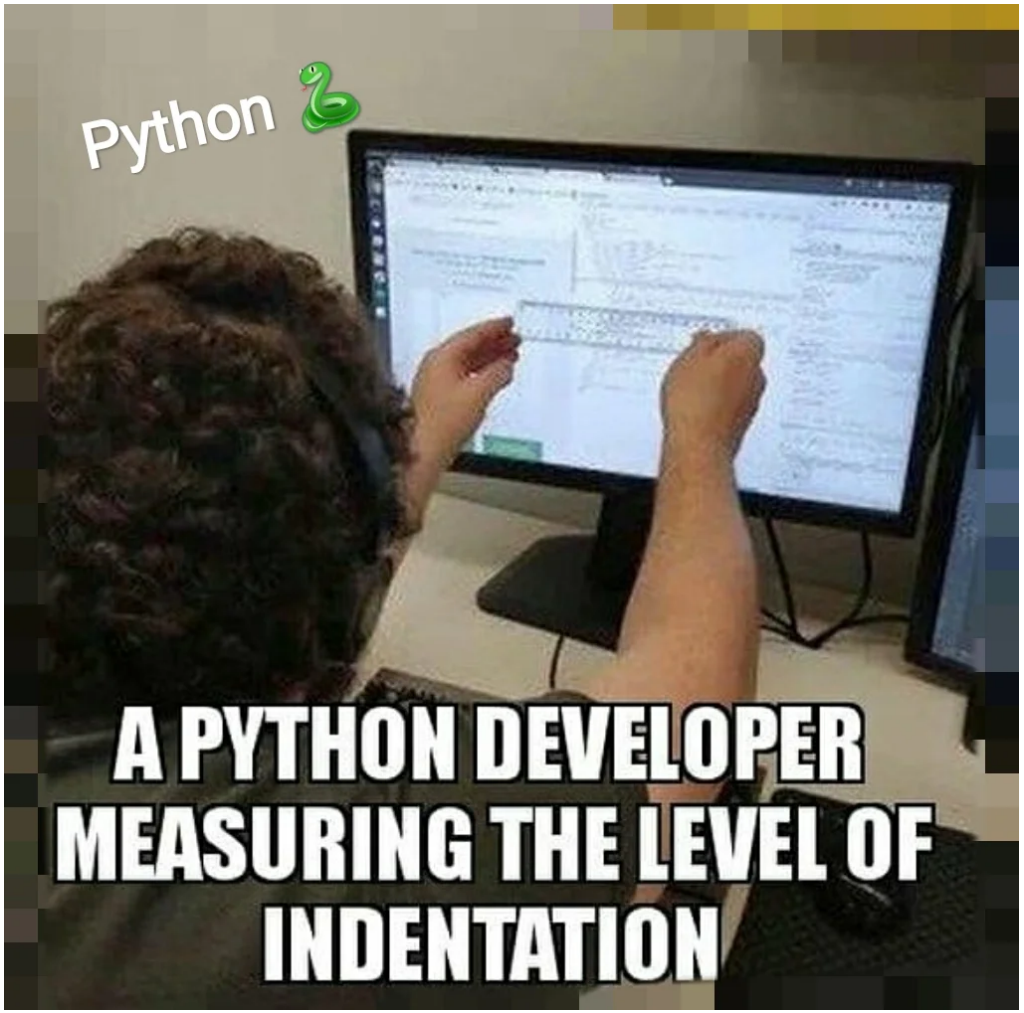

Python Applications
Web applications
Destop GUI Applications
Console-Based Applications
Software Development
scientific & numeric
Busines Applications
Audio or Video-Based Applicatons
3D CAD Applications
Enterprise Applications
Image Processing Applications

Στοίχιση

"Readability counts"



Στοίχιση


Python
A PYTHON DEVELOPER
MEASURING THE LEVEL OF
INDENTATION

Στοίχιση

```
1  x = 0
2  if x == 0:
3      print('x is zero')
4
```

```
thomas@kirklaptop:~/Documents/mscpython/code/lecture3$ python3 indentation.py
x is zero
```

Στοίχιση

```
1  x = 0
2  if x == 0:
3  print('x is zero')
4
```

```
thomas@kirklaptop:~/Documents/mscpython/code/lecture3$ python3 indentation.py
  File "/home/thomas/Documents/mscpython/code/lecture3/indentation.py", line 3
    print('x is zero')
    ^
IndentationError: expected an indented block after 'if' statement on line 2
```

Σχόλια

"Readability counts"

```
1 # Set variable to zero
2 x = 0
3
4 # Check whether variable is zeros
5 if x == 0:
6     print('x is zero')
```

Σχόλια

"Readability counts"

```
1
2 x = 0 # Set variable to zero
3
4
5 if x == 0: # Check whether variable is zeros
6     print('x is zero')
```

Σχόλια

"Readability counts"

```
"""
An example to demonstrate line and block comments

Created by Thomas Kamalakis
"""

x = 0 # Set variable to zero

if x == 0: # Check whether variable is zeros
    print('x is zero')
```

Μεταβλητές

- Μία *μεταβλητή* είναι κάτι που μπορεί να αλλάξει μέσα στο πρόγραμμα.
- Στην Python υπάρχουν διάφορα είδη μεταβλητών
- *ακέραια* μεταβλητή (int)
- μεταβλητή *πραγματικού αριθμού* (float)
- *αλφαριθμητική* μεταβλητή (string)
- μπορούμε να χρησιμοποιήσουμε και πιο πολύπλοκους τύπους μεταβλητών

Μεταβλητές

```
1  x = 1           # Integer
2  s = 'Thomas'    # String
3  S = "Thomas"    # String
4  y = 1.2         # Float
5
6  # Check out variable types
7  print( 'x is: ', type(x) )
8  print( 's is: ', type(s) )
9  print( 'S is: ', type(S) )
10 print( 'y is: ', type(y) )
```

```
thomas@kirklaptop:~/Documents/mscpython/code/lecture3$ python3 variables.py
x is:  <class 'int'>
s is:  <class 'str'>
S is:  <class 'str'>
y is:  <class 'float'>
```

Μεταβλητές

```
1 """
2 Simultaneous assignments
3 """
4 x, y, z = 0, "Thomas", 1.2
5
6 print(x)
7 print(y)
8 print(z)
```

```
thomas@kirklaptop:~/Documents/mscpython/code/lecture3$ python3 variables2.py
0
Thomas
1.2
```

Μεταβλητές - Casting

- Μπορούμε να αλλάξουμε τον τύπο μίας μεταβλητής, π.χ. από string σε ακέραιο.

```
1  """
2  Casting of variables
3  """
4  x = 2
5  s = str(x)
6  print( type(s) )
7  print(s)
8
9  S = "4"
10 y = int(S)
11 print( type(y) )
12 print(y)
```

```
thomas@kirkoffice:~/Documents/mscpython/code/lecture3$ python3 casting.py
<class 'str'>
2
<class 'int'>
4
```

Strings - Ανάθεση

```
1  """
2  Strings
3  """
4  s = "Thomas"
5  print(s)
6
7  S = """
8  Thomas
9  Kamalakis
10 """
11 print(S)
```

```
Thomas

Thomas
Kamalakis
```

Τμήματα από strings

- Μπορούμε να πάρουμε τμήματα από ένα string
- s[i], ο i+1 χαρακτήρας.
- s[i:j], το string μεταξύ του (i+1) και του j χαρακτήρα.
- s[i:], το string μεταξύ του (i+1) και του τελευταίου χαρακτήρα.
- s[:j], το string μεταξύ του πρώτου και του j χαρακτήρα.

```
1  """
2  Slicing Strings
3  """
4  s = "Thomas Kamalakis"
5  print(s)
6
7  print(s[1])
8  print(s[1:5])
9  print(s[:5])
10 print(s[5:])
```

```
thomas@kirkoffice:~/Documents/mscpython/code/lecture3$ python3 slicingstrings.py
Thomas Kamalakis
h
homa
Thoma
s Kamalakis
```

String loops

- Μπορούμε να αναθέσουμε σε μία μεταβλητή τους διαδοχικούς χαρακτήρες ενός string.

```
1 """
2 String Loops
3 """
4
5 S = "Thomas Kamalakis"
6 for x in S:
7   print(x)
```

```
thomas@kirkoffice:~/Documents/mscpython/code/lecture3$ python3 stringloop.py
T
h
o
m
a
s

K
a
m
a
l
a
k
i
```

String in String

- μπορούμε να ελέγξουμε αν ένα string περιέχεται σε ένα άλλο string

```
1  """
2  String in String
3  """
4
5  S = "Thomas Kamalakis"
6  s = "Thomas"
7  if s in S:
8    print(s, 'is contained in', S)
9  else:
10   print(s, 'is not contained in', S)
11
12 s2 = "Tom"
13 if s2 in S:
14   print(s2, 'is contained in', S)
15 else:
16   print(s2, 'is not contained in', S)
```

```
thomas@kirkoffice:~/Documents/mscpython/code/lecture3$ python3 stringin.py
Thomas is contained in Thomas Kamalakis
Tom is not contained in Thomas Kamalakis
```

Μήκος string

- Η συνάρτηση length επιστρέφει τον αριθμό χαρακτήρων του string

```
1 """
2 Length of String
3 """
4
5 S = "Thomas Kamalakis"
6 l = len(S)
7 print(S, 'has ', l, 'characters')
```

```
thomas@kirkoffice:~/Documents/mscpython/code/lecture3$ python3 stringlen.py
Thomas Kamalakis has  16 characters
```

Μετασχηματισμοί String

```
"""
Transform a String
"""

S = "Thomas, Kamalakis"
print( S.upper() )
print( S.lower() )
print( S.strip() )
print( S.replace("Thomas", "Tom") )
print( S.split(",") )

S2 = "Thomas " + "Kamalakis"
print(S2)
```

```
y
THOMAS, KAMALAKIS
thomas, kamalakis
Thomas, Kamalakis
Tom, Kamalakis
['Thomas', ' Kamalakis']
Thomas Kamalakis
```