



# LINUX

ROS (Robot Operating System), Eskişehir



Supported by ROSIN - ROS-Industrial Quality-Assured Robot Software Components.  
More information: [rosin-project.eu](https://rosin-project.eu)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732287.

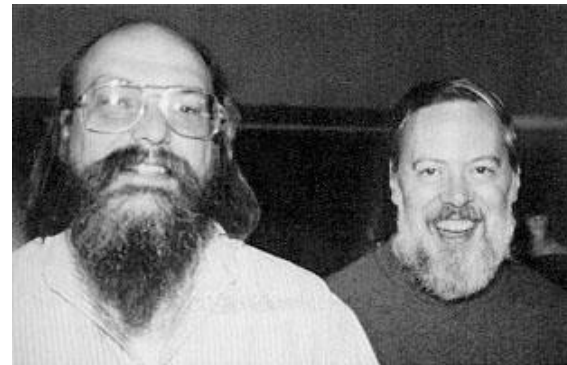
# Context

- History
- UNIX
- GNU
- Linux
- Distributions

# Bell Labs

- Radio astronomy
- Transistor
- Lazer
- Photovoltaic cell
- CCD
- Information Theory
- C, C++
- 9 Nobel Prize

□ and **UNIX**



*Ken Thompson ve Dennis Ritchie*

# UNIX (1970's)

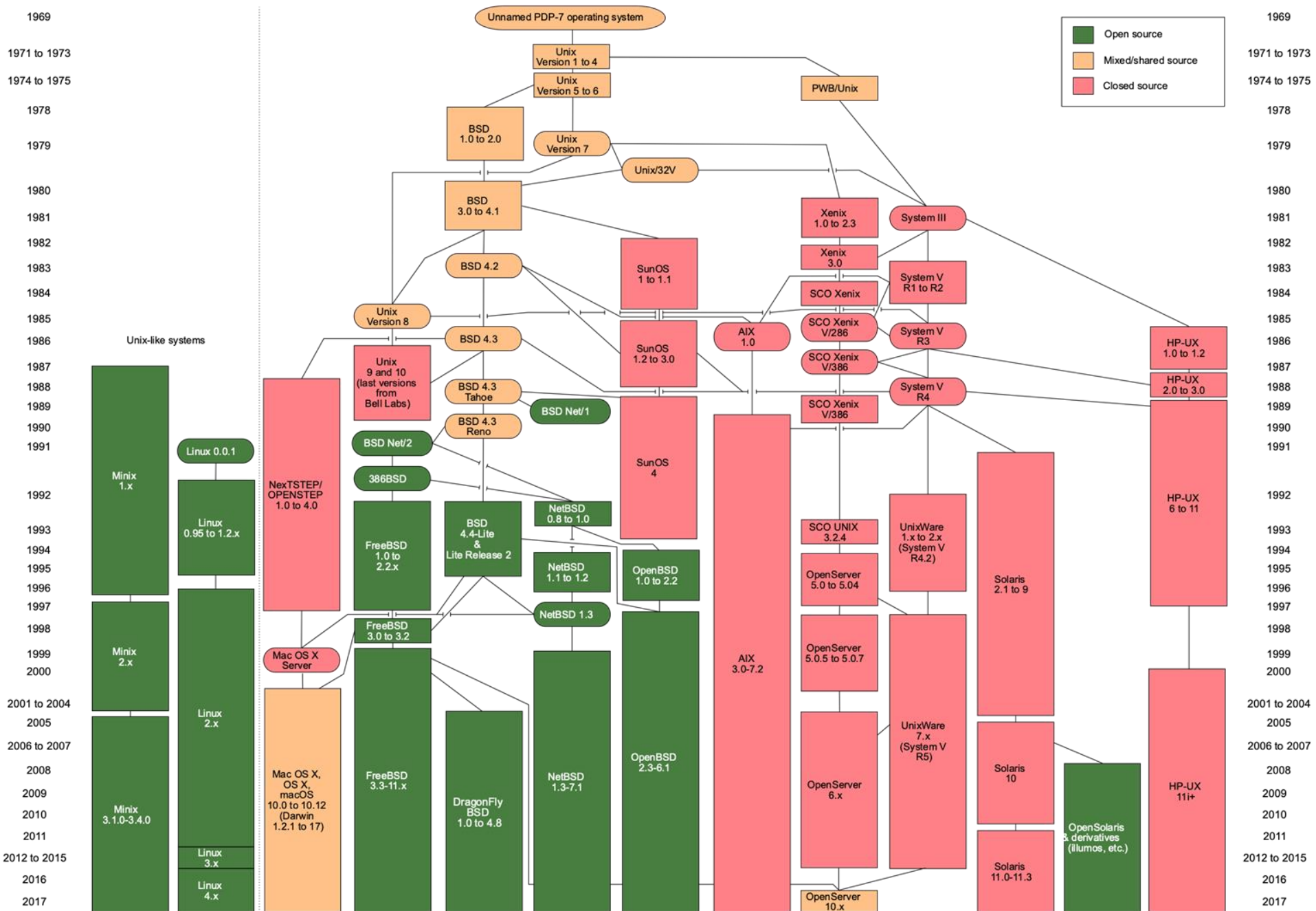
## □ Philosophy

- Worse is better.
- Simple, easy to use, minimalist, reusable software.
- 4 rules: Simple, accurate, consistent, complementary.
- Each program does a job but does it best.
- Instead of adding new features to old programs, write new ones from scratch.
- Let the output of one program be the input of another program.

**Everything is a file.**

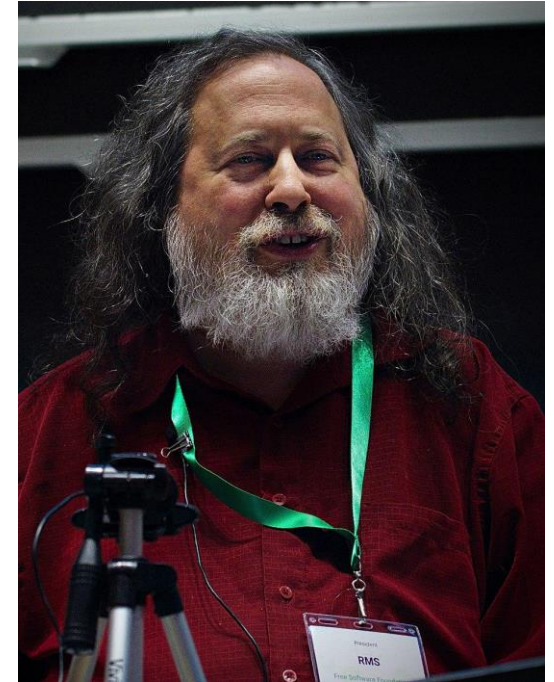
# UNIX

- ❑ BSD (Berkeley)
- ❑ Xenix (Microsoft)
- ❑ AIX (IBM)
- ❑ Solaris (Sun)
- ❑ MacOS (Apple)

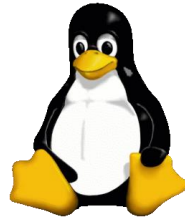


# GNU (*GNU's Not Unix*) - 1983

- ❑ Kernel : HURD
- ❑ GNU General Public License (GPL)
- ❑ Free Software Foundation (FSF)
- ❑ Richard Matthew Stallman (rms)



# Linux (1991)



- Linus Torvalds
- Version 1.0 (1994)





# Distributions

- Linux kernel
  - GNU tools and libraries
  - Window manager
  - Desktop environment
  - Package manager
- 
- Old: Slackware ve Debian (1993)
  - More than 500 active distributions.

# Package Managers

Debian	apt (dpkg, .deb)
Red Hat, CentOS, Fedora	yum (.rpm)
SUSE	YaST
Arch	pacman
Pardus	PiSi

# Ubuntu

- ❑ Canonical Ltd.
- ❑ Mark Shuttleworth
- ❑ ~450 employees
- ❑ 2 versions per year(.04 ve .10)
- ❑ 16.04, 18.04, 20.04 (LTS)



# GUI vs. CLI

- Graphic User Interface
- Command-line Interface

# Shell

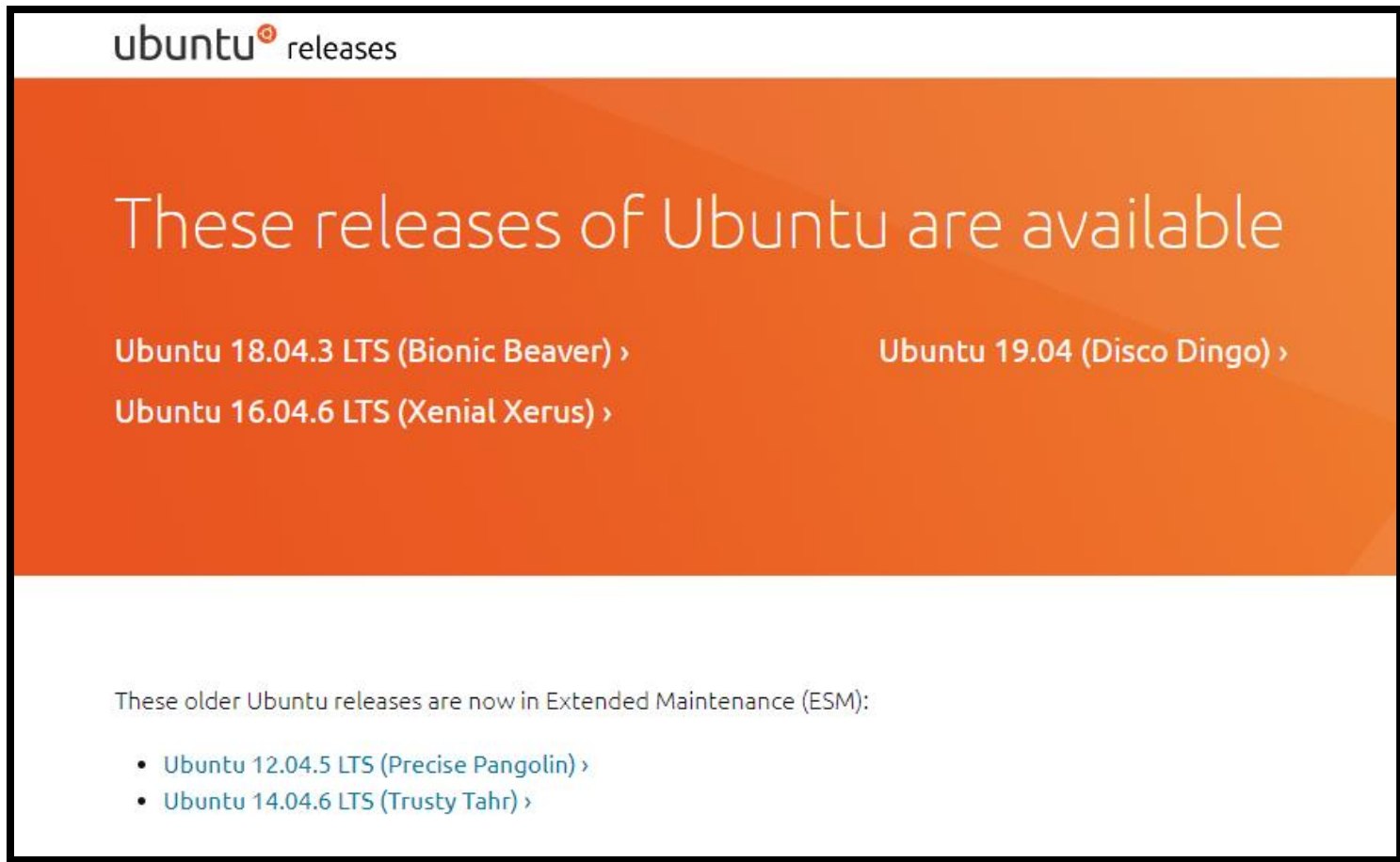
- Can be CLI or GUI
- It is the interface that provides access to the services of the operating system.
- csh, tcsh, ksh, zsh, ash, CMD.EXE
- **GNU Bash**

```
Prompt> command param1 param2 ...
```

# Setting up Ubuntu

- All available ubuntu versions are available at <http://releases.ubuntu.com/>.
- The downloaded Ubuntu .iso file should be written to Usb memory or CD.
- Unetbootin or Rufus programs can be used for printing.

# Setting up Ubuntu

A screenshot of the Ubuntu releases page. The page has a white header with the 'ubuntu' logo and the word 'releases'. Below the header is a large orange section with the text 'These releases of Ubuntu are available'. Inside this orange section, there are three links: 'Ubuntu 18.04.3 LTS (Bionic Beaver) >', 'Ubuntu 16.04.6 LTS (Xenial Xerus) >', and 'Ubuntu 19.04 (Disco Dingo) >'. Below the orange section is a white section with the text 'These older Ubuntu releases are now in Extended Maintenance (ESM):'. Below this text is a bulleted list with two items: 'Ubuntu 12.04.5 LTS (Precise Pangolin) >' and 'Ubuntu 14.04.6 LTS (Trusty Tahr) >'.

ubuntu releases

These releases of Ubuntu are available

Ubuntu 18.04.3 LTS (Bionic Beaver) ›

Ubuntu 16.04.6 LTS (Xenial Xerus) ›

Ubuntu 19.04 (Disco Dingo) ›

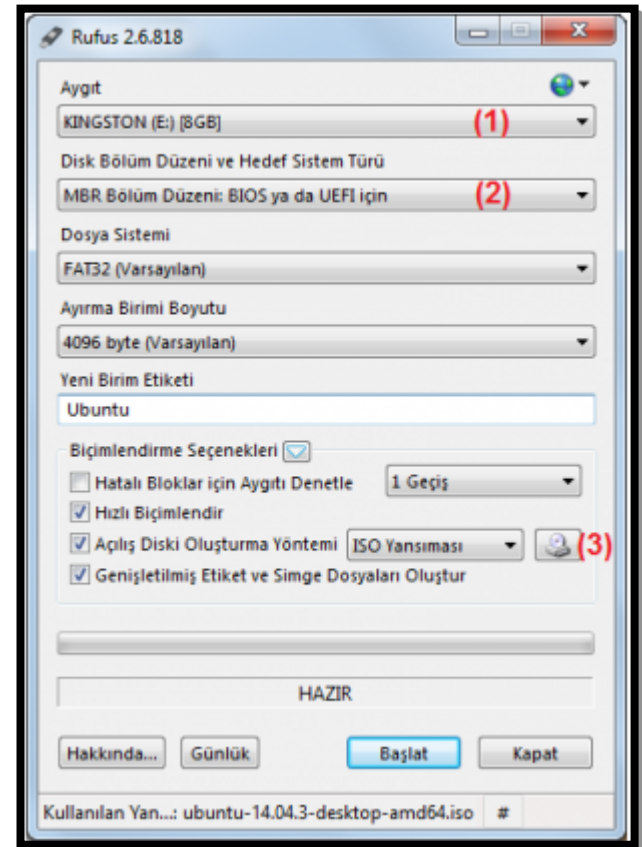
These older Ubuntu releases are now in Extended Maintenance (ESM):

- Ubuntu 12.04.5 LTS (Precise Pangolin) ›
- Ubuntu 14.04.6 LTS (Trusty Tahr) ›



# Setting up Ubuntu

- ❑ After downloading the Rufus program :
- Usb is selected from the device section.
- Boot type is selected.
- The directory of the .iso file is displayed.
- Start button is pressed.
- Close key is pressed when memory is ready.



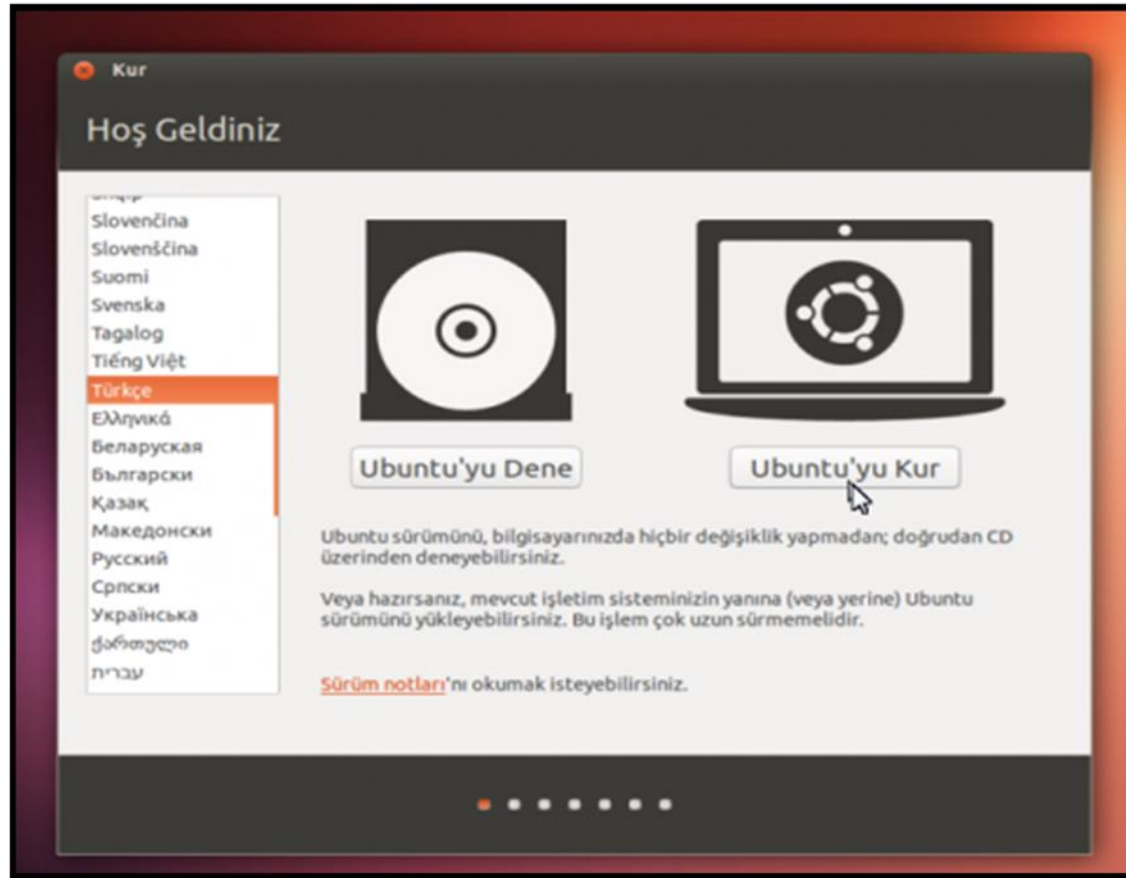
# Setting up Ubuntu

□ Insert the USB memory prepared while the computer is booted and press F2 or F12 to choose the USB memory to be used for Ubuntu installation from the boot settings.

# Setting up Ubuntu

- There is an option that you can try without installing Ubuntu on the screen.
- With this, the Ubuntu version can be tried and exited without any registration.
- Or, Ubuntu installation can be started during the trial..
-

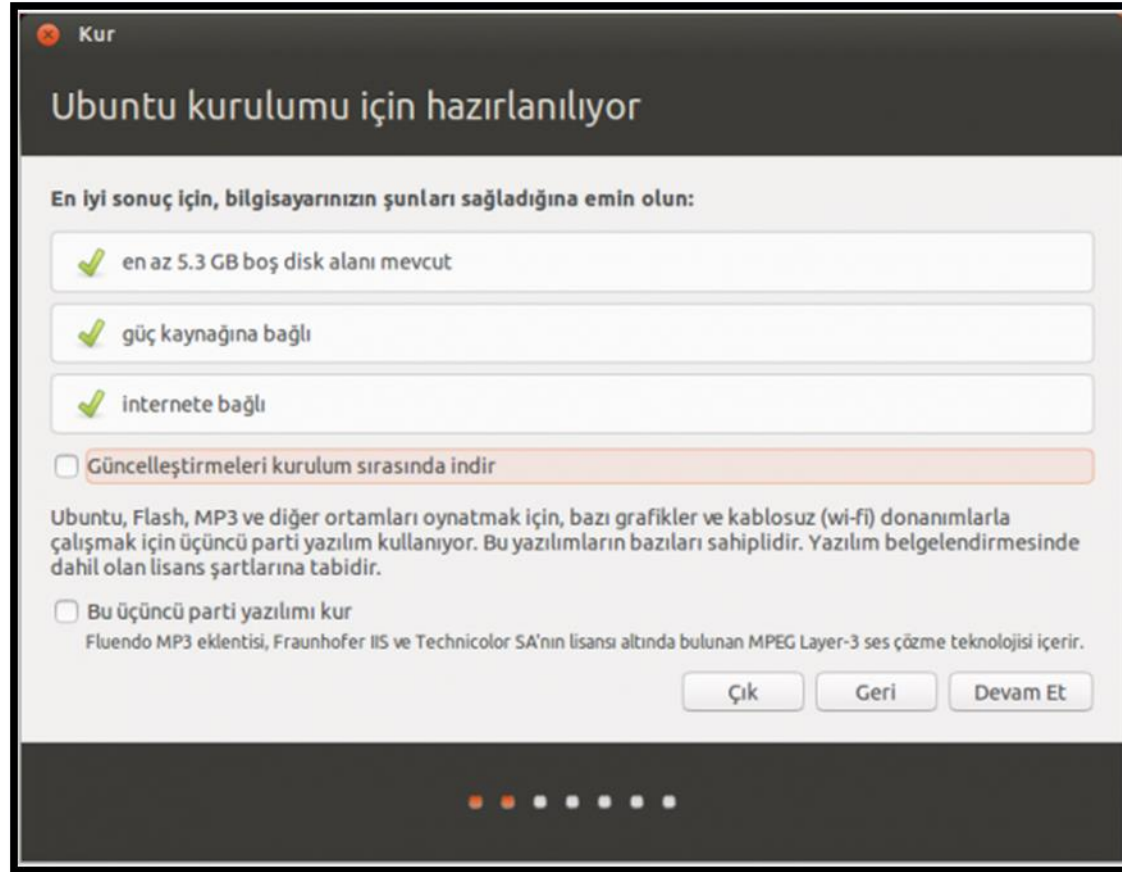
# Setting up Ubuntu



# Setting up Ubuntu

- Ubuntu installation starts with the Install Ubuntu option.
- The following options can be selected for installing updates that have been installed during installation and for installing side technologies. But they are not mandatory.

# Setting up Ubuntu

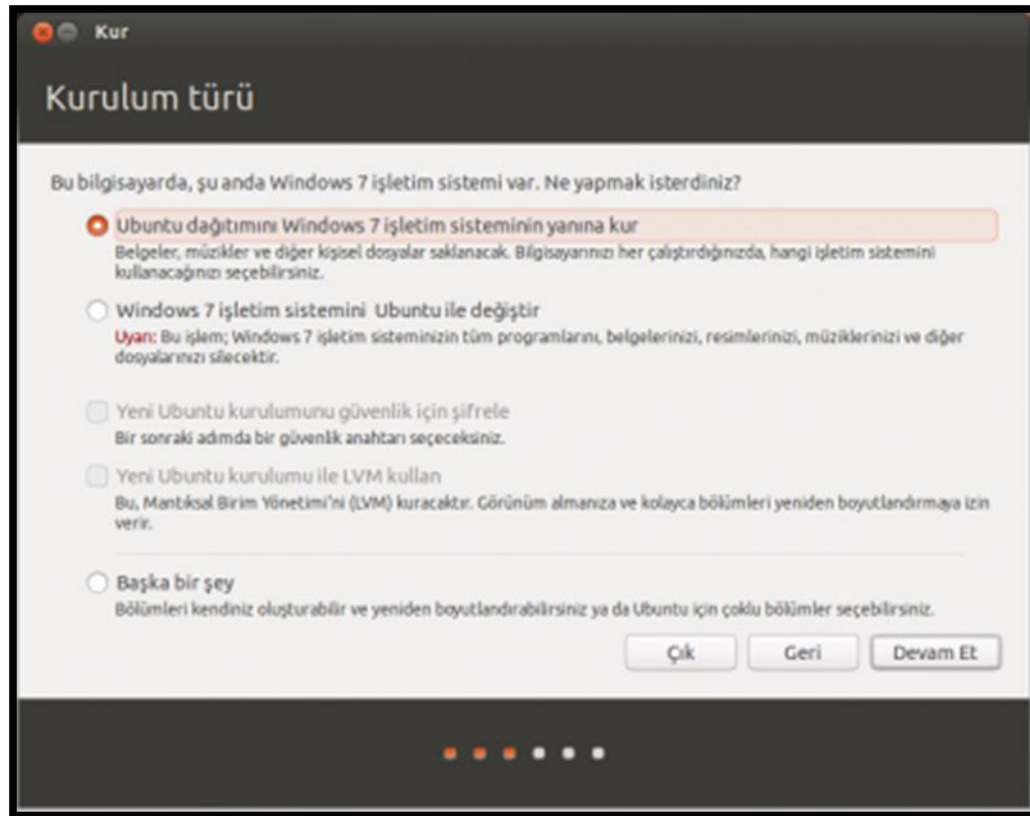


# Setting up Ubuntu

## □ In the Setup Type window

- Replace the Windows operating system with Ubuntu: This option deletes Windows and the documents on your computer and installs Ubuntu instead.
- Encrypt the new Ubuntu installation for security: With this option, we can control the access and intervention for non-ourselves by adding a password to the installation.
- Use LVM with new Ubuntu installation: With this option, we install on the disk that is configured to be expandable. This does not require a special disc. If we make this selection, our disk is configured in accordance with the feature of increasing the size of our disk without damaging the data in the future.
- Another thing: This option allows you to configure the disc to your own needs. Using this option is somewhat complicated and requires experience.
- If no operating system is installed on your computer, the first option will be Erase Disk and install Ubuntu..

# Setting up Ubuntu





# Setting up Ubuntu

- Choose your location

- Then "Where are you?" The screen where we will answer the question comes. Here you can select Istanbul to Turkey.

# Setting up Ubuntu



# Setting up Ubuntu

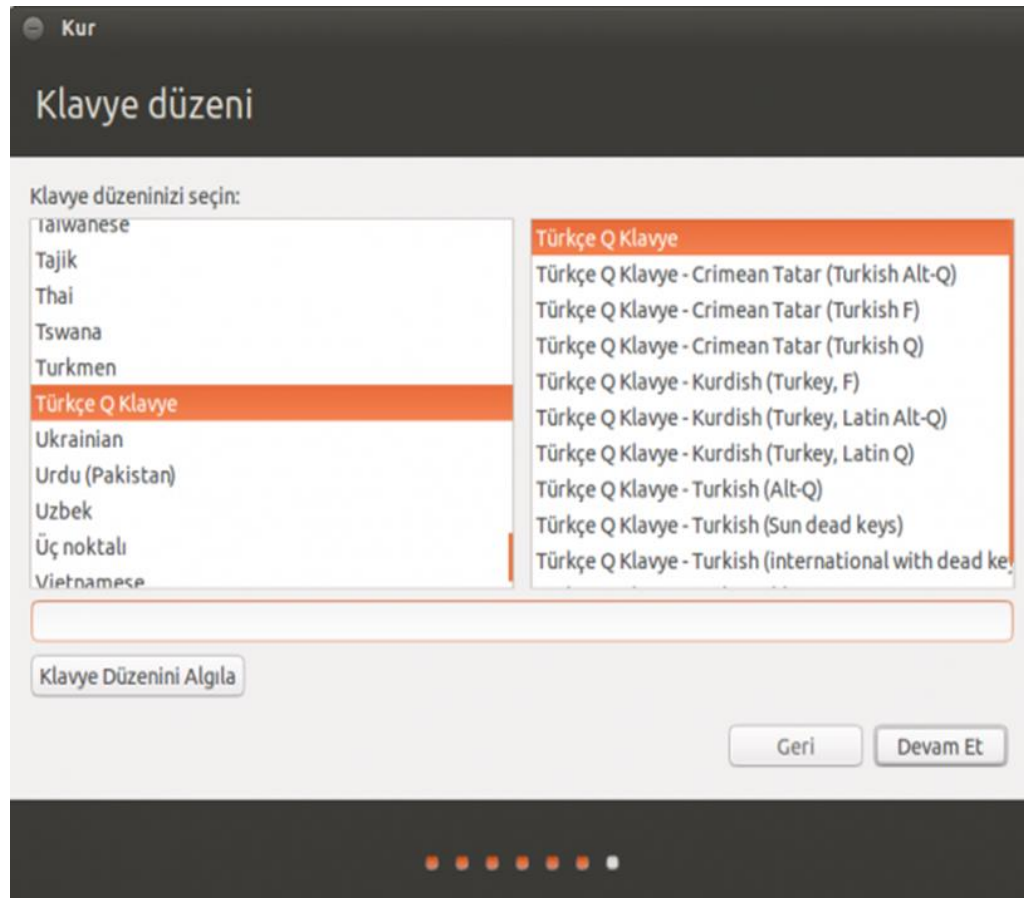
## ❑ Select keyboard layout

■ Then, we see the screen for determining the "Keyboard layout".

■ Since our keyboard layout will be determined automatically under normal conditions here, we usually do not need to take any action. The 'Detect Keyboard Layout' button located here can also be used to detect the keyboard.

■ After selecting a keyboard language, you can check the correctness of the language in question, that is, compatibility with your keyboard by entering some Turkish characters from the keyboard in the box here.

# Setting up Ubuntu

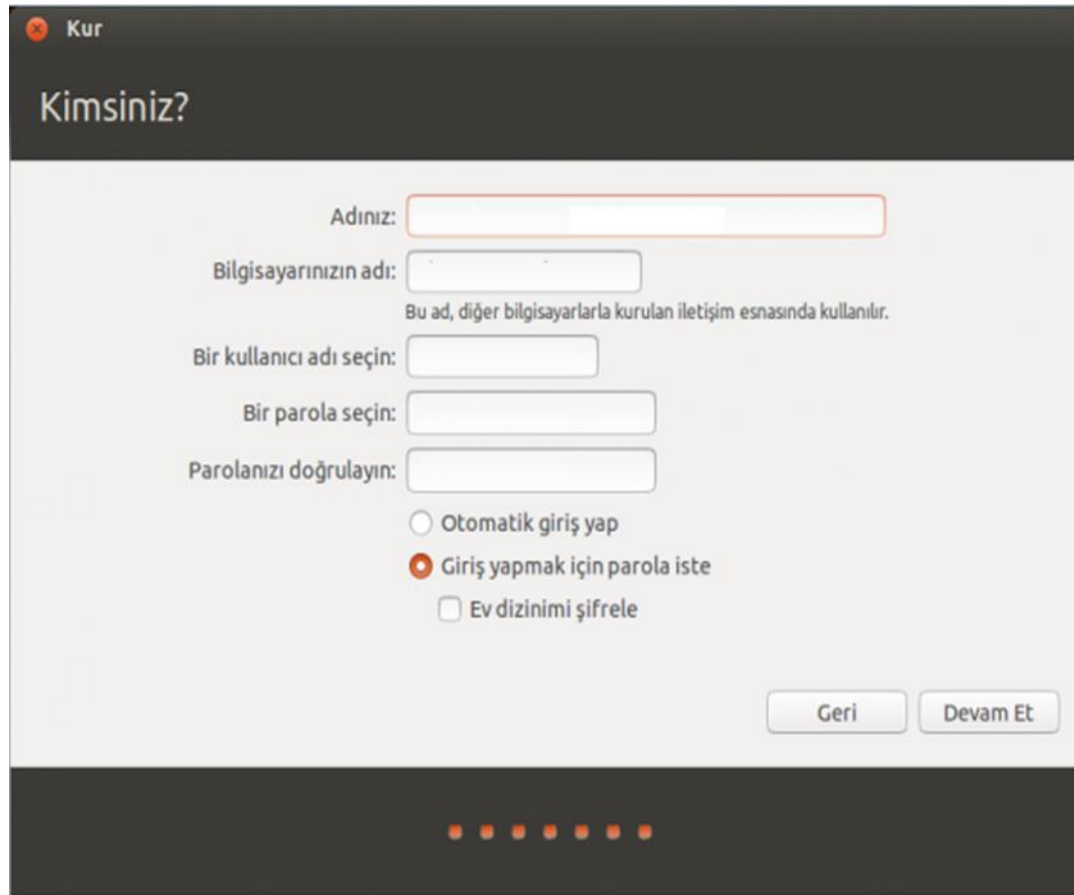


# Setting up Ubuntu

## □ Setting username and password

- At this stage, we enter the name of our computer and the name of our user account. If the password we set is at least 6 characters long and includes characters such as letters, numbers, ? \* - etc., Ubuntu setup will like our password and say 'Good Password'.
- Auto login: You can select this option if you want to log in directly without entering your user password every time you open Ubuntu. Even if you do not select this option, you can also enable automatic user login through Settings after installing Ubuntu.
- Require password to login: It is the opposite of the option above. This option is checked by default.
- Encrypt my home directory: By checking this option, the user home directory under the folder named / Home can be encrypted for access outside of us by encrypting our home directory.
- Then, when we use the 'Continue' button, the "Install" screen, which comes up with the "Installing System" message, starts to complete the installation

# Setting up Ubuntu



Kur

Kimsiniz?

Adınız:

Bilgisayarınızın adı:   
Bu ad, diğer bilgisayarlarla kurulan iletişim esnasında kullanılır.

Bir kullanıcı adı seçin:

Bir parola seçin:

Parolanızı doğrulayın:

☐ Otomatik giriş yap

☒ Giriş yapmak için parola iste

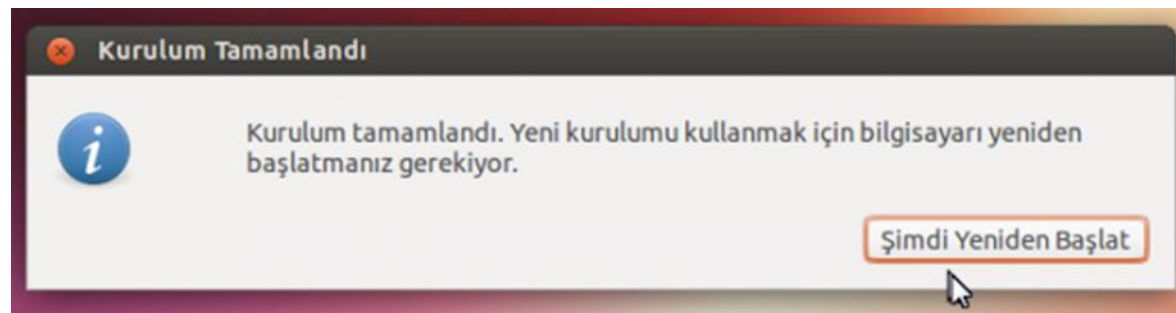
☐ Ev dizinimi şifrele

Geri Devam Et

Progress bar with 8 dots, the first 7 are filled.

# Setting up Ubuntu

- Wait for installation to finish
- When the installation is complete, we receive the 'Installation Complete' message. After saying 'Restart Now' we can now start using Ubuntu.



# Prompt

```
gokhan@esogu:~$ █
```



# Bash - Beginning

```
gokhan@esogu:~$ whoami
gokhan
gokhan@esogu:~$ hostname
esogu
gokhan@esogu:~$ lscpu

gokhan@esogu:~$ uname
Linux
gokhan@esogu:~$ uname -r
5.0.0-29-generic
```

# Bash - Beginning

```
gokhan@esogu:~$ w
```

# Bash - folders

```
$ mkdir siirler  
$ cd siirler  
$ pwd  
/home/gokhan/siirler
```

# Downloading files

```
$ wget http://188.132.181.140/otuzbes.txt  
$ wget http://188.132.181.140/handuvar.txt  
$ wget http://188.132.181.140/sessiz.txt
```

# Listing files

```
$ ls  
handuvar.txt  otuzbes.txt  sessiz.txt
```

```
$ ls -l  
handuvar.txt  
otuzbes.txt  
sessiz.txt
```

```
$ ls -l  
-rw-r--r-- 1 gokhan gokhan 6662 Sep 23 01:46 handuvar.txt  
-rw-r--r-- 1 gokhan gokhan 1163 Sep 21 02:09 otuzbes.txt  
-rw-r--r-- 1 gokhan gokhan  528 Sep 23 01:49 sessiz.txt
```

# Wiewing Files

```
$ more otuzbes.txt  
$ less handuvar.txt  
$ cat sessiz.txt
```

# head and tale

```
$ head otuzbes.txt  
$ tale handuvar.txt  
$ head -n 5 sessiz.txt
```

# cat (*Concatenation*)

```
$ cat otuzbes.txt handuvar.txt sessiz.txt  
$ cat *
```



# Standard Flows

input  
***stdin***  
0

program  
or  
*process*

output  
***stdout***  
1

error  
***stderr***  
2

cat -

```
$ cat -
```

```
$ cat otuzbes.txt -
```

# Redirecting >

```
$ cat otuzbes.txt handuvar.txt > ikisiir.txt  
$ more ikisiir.txt
```

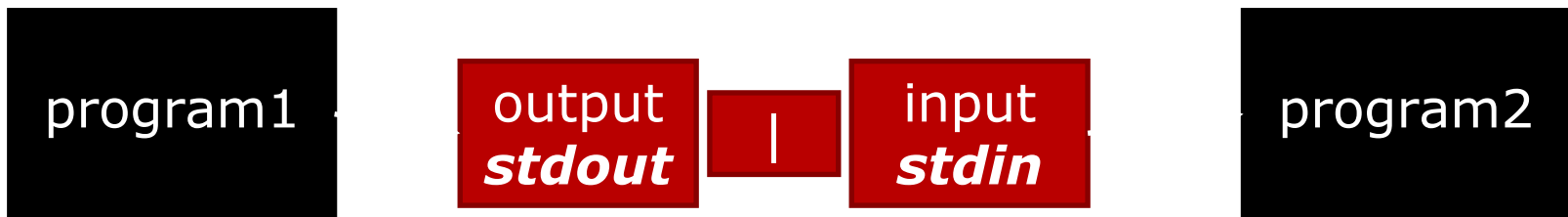
```
$ cat otuzbes.txt handuvar.txt sessiz.txt > tumsiirler.txt  
$ more tumsiirler.txt
```

```
$ ls > dosya_listesi.txt  
$ more dosya_listesi.txt
```

# Redirecting <

```
$ cat < handuvar.txt
```

# Redirecting | (pipe)



```
$ echo "Benim Siirim" | cat sessiz.txt -
```

# Regular expressions (*regex*)

```
$ grep "yollar" otuzbes.txt  
Dostlarla da yollar ayrildi bir bir;
```

^ refers to the beginning of the line.

```
$ grep "^B" sessiz.txt  
Bicare gonuller! Ne giden son gemidir bu!  
Bilmez ki giden sevgililer donmeyecekler.  
Bir cok gidenin her biri memnun ki yerinden,  
Bir cok seneler gecti; donen yok seferinden.
```

# Regular expressions(*regex*)

\$ refers to the end of the line.

```
$ grep '!$' handuvar.txt
Yalniz arabacinin dudaginda bir islik!
Artik bahtin aciktir, uzun etme, arkadas!
Basucumda gordugum su satirlarla yandim!
Ey Marasli Seyhoglu, evliyalari adagi!
Bahtina lanet olsun asmadinsa bu dagi!
Donmeyen yolculara aglayan yasli yollar!
```

# Regular expressions(*regex*)

. refers to any character.

```
$ cat handuvar.txt otuzbes.txt sessiz.txt | grep "d..l" -  
Basini kaldırarak boslugu dinliyordu.  
Son yokus noktasından duzluge cevrilince  
Yol, hep yol, daima yol... Bitmiyor duzluk yine.  
Bu dort misra degil, sanki dort damla kandi.
```



# tr

```
$ head otuzbes.txt | tr 'a' 'A'
```

```
$ head otuzbes.txt | tr 'ai' 'AI'
```

```
$ head otuzbes.txt | tr 'abu' 'AI '
```

# tr -d

```
$ head otuzbes.txt | tr -d 'a'
```

```
$ head otuzbes.txt | tr -cd 'a'
```

# Getting Information About Commands

```
$ man tr
```

# cut -f

```
$ wget http://188.132.181.140/tablo.txt  
$ cat tablo.txt
```

```
$ cat tablo.txt | cut -f 4  
Eskisehir  
Bursa  
Antalya  
Istanbul  
Ordu  
Bursa  
Gaziantep  
Van  
Trabzon  
Izmir
```

# cut -c

```
$ cat tablo.txt | cut -c 1-5
```

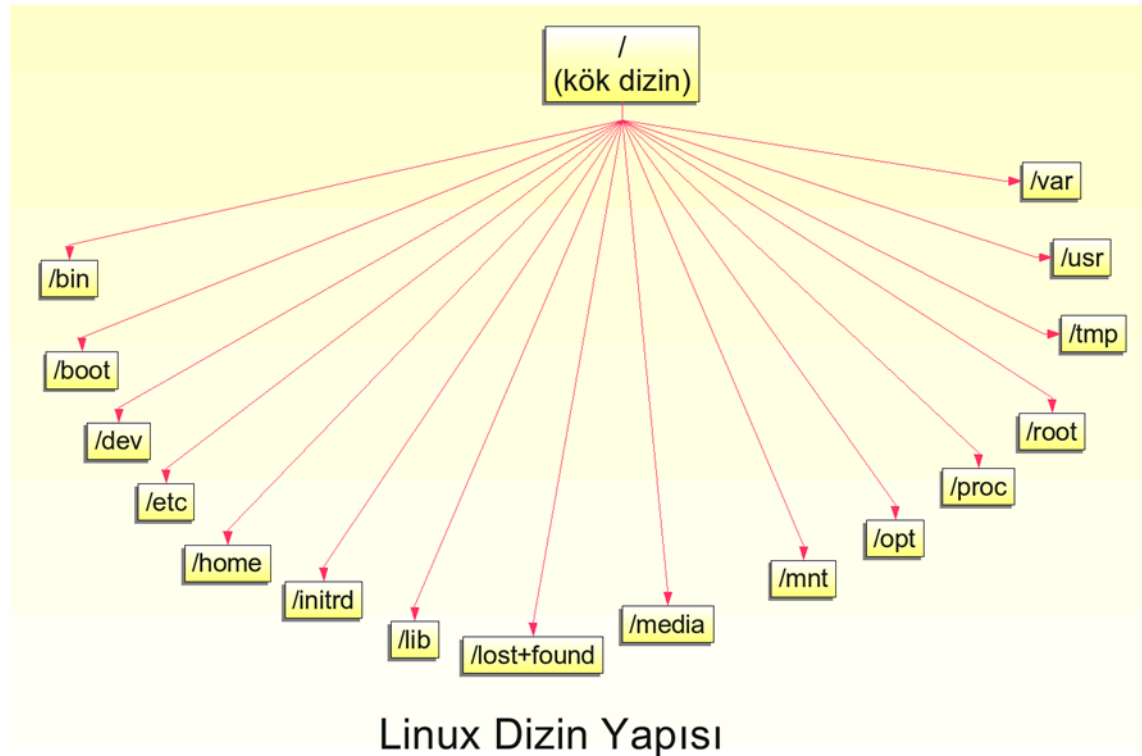
```
$ cat tablo.txt | cut -c 20-
```

# Linux File Structure

/ root is directory.

```
$ ls -1 /  
bin  
boot  
dev  
etc  
home  
lib  
media  
mnt  
opt  
proc  
root  
run  
sbin  
sys  
tmp  
usr  
var
```

# Linux file structure



# Linux File Structure

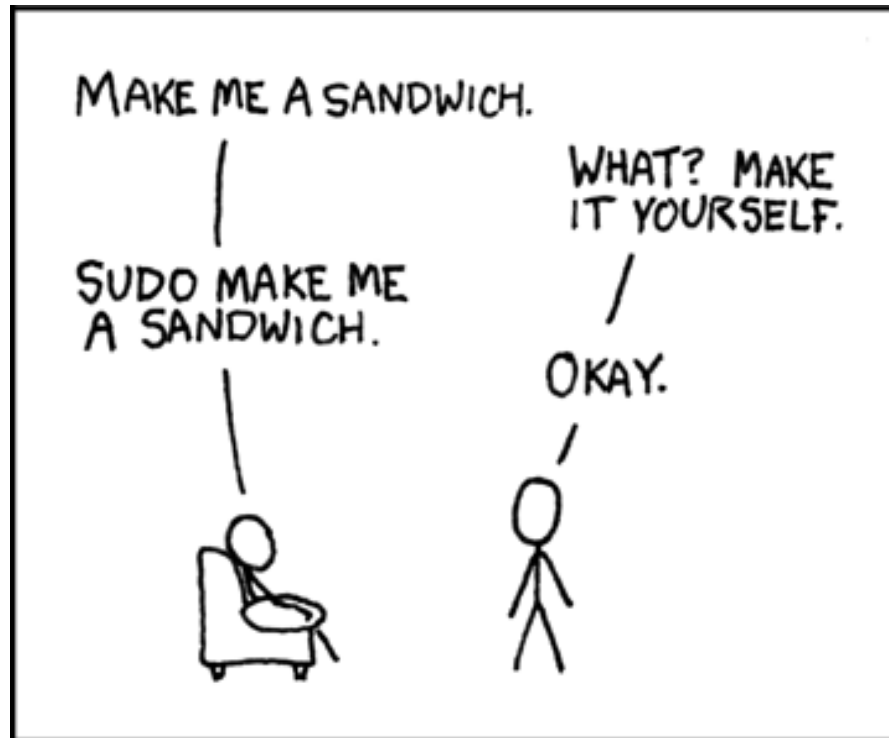
## □ root altındaki temel klasörler

- /bin: Includes must-have scripts
- /boot: Contains files necessary for startup
- /dev: Contains hardware files
- /etc: Contains system settings
- /lib: Contains library files and kernel modules
- /media: It is the folder where removable devices (CD-Rom, Flash memory etc ...) are added to the system.
- /mnt: Used to temporarily add a file system.
- /opt: For installing extra programs
- /sbin: Keeps executable files related to the system administrator.
- /srv: Related to the services provided by the system
- /tmp: For holding temporary files
- /usr: A secondary hierarchy
- /var: Stores variable data



# sudo

□ super user do



# Users

```
$ cat /etc/group
```

## □ Adding new user

```
$ man adduser
```

# File Permissions

- .Each file has an owner and a group.
- .Rights are defined for a file for the following users:
  - .User (u)
  - .Group (g)
  - .Others (o)
- .Identifiable rights are as follows:
  - .Read– 4
  - .Write– 2
  - .Execute– 1
  - .4:2:1 (r:w:x)

# File Permissions

```
$ ls -l
-rw-r--r-- 1 gokhan gokhan 6662 Sep 23 01:46 handuvar.txt
-rw-r--r-- 1 gokhan gokhan 1163 Sep 21 02:09 otuzbes.txt
-rw-r--r-- 1 gokhan gokhan  528 Sep 23 01:49 sessiz.txt

$ chmod go-w handuvar.txt
$ chmod u+x otuzbes.txt
```

# File Permissions

```
$ chown root.root handuvar.txt
$ sudo chown root.root handuvar.txt
$ ls -l

$ wget http://188.132.181.140/meteo.sh
$ ./meteo.sh
$ bash meteo.sh
$ nano meteo.sh
```









# Shebang or Hashbang - #!

```
#!/usr/bin/python  
#!/usr/bin/python3  
#!/bin/bash
```

```
$ env
```

```
#!/usr/bin/env python
```

# Basic Network Commands

```
$ ip addr  
$ ping 127.0.0.1  
$ ifconfig -a  
$ ssh gokhan@localhost
```

# Processes

- `ps`: Refers to Processes. Lists processes running on Linux system.
- `ps -aux | grep <action_name>`: Returns the specific process or processes running on Linux.
- `kill -9 <action_ID>`: Allows killing the process running on Linux and having the ID with the command above.
- The first running process (number 1) is `init`
- `top` (table of processes)
-

# Package Installations

- ❑ `wget '<download_url>'`: Downloads the file specified on the internet to the folder on the computer.
- ❑ `sudo apt-get install <package_name>`: Searches for the repository and installs it on the computer.
- ❑ `sudo apt-get remove <package_name>`: Searches for and deletes the package from the computer.
- ❑ `sudo apt-get update`: retrieves the information of the repository stored in the `sources.list` file to the computer.
- ❑ `sudo apt-get upgrade`: Updates the packages on the computer.
- ❑ `apt-cache search <package_name>`: Searches and fetches the related package in repository.
- ❑ `sudo chmod <permission_type> <folder_or_file_name>`: Gives file permissions to the relevant folder or document. Using `777` as the permission type means Read-Write-Execute.

# Services

- ❑ `sudo service <service_name> start`: Starts the service running within Linux.
- ❑ `sudo service <service_name> stop`: Terminates the service running within Linux.
- ❑ `sudo service <service_name> restart`: Restarts the service running under Linux.
- ❑ Startable services
  - FTP
  - Web server (NGINX, Apache)
  - Databases (MySQL, PostgreSQL)
  - SSHD (remote connection)
  - ...
  -

# Text Editors

□ CLI based:

■ pico, nano

■ vi, vim

■ Emacs

□ GUI based:

■ Gedit

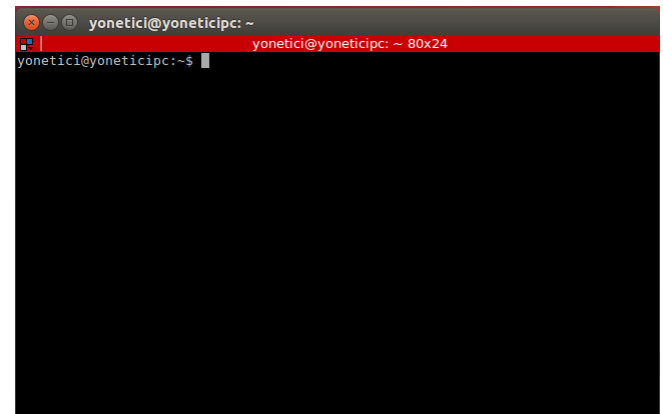
■ Atom

■ VS Code

■ Sublime Text

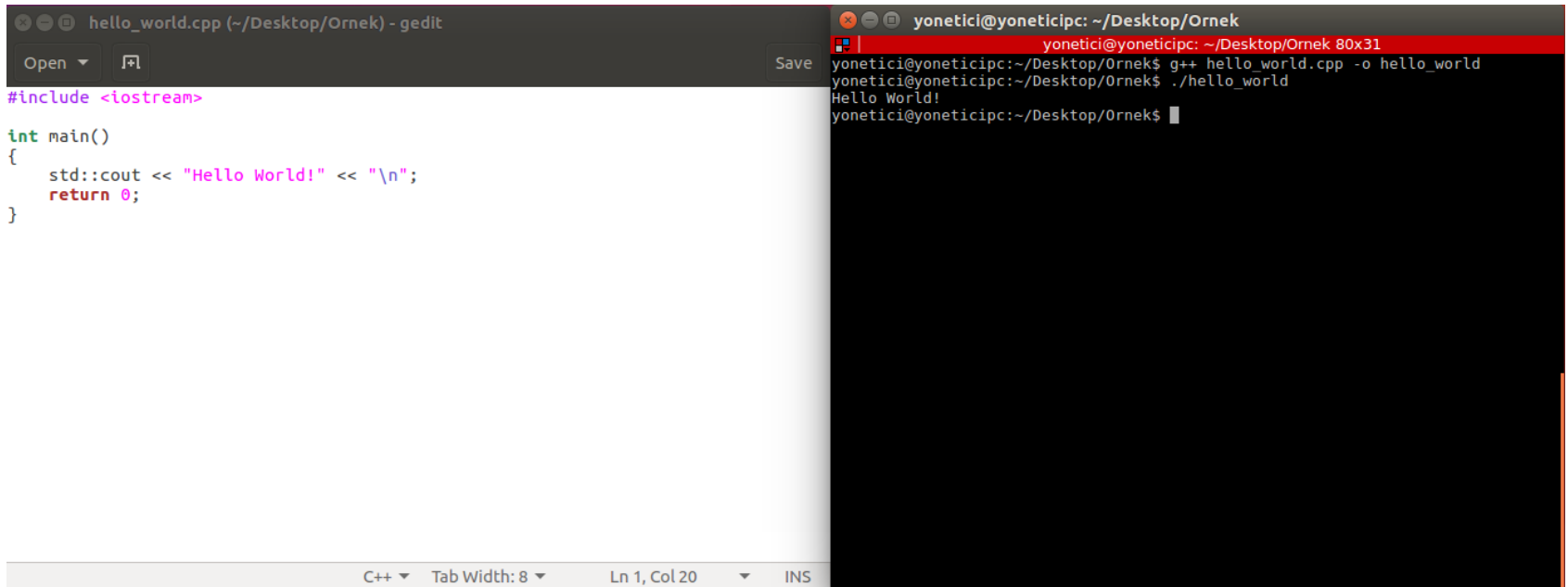
# Linux Terminal

- ❑ Terminal: Ctrl + Alt + T
- ❑ Copy: Ctrl + Shift + C
- ❑ Paste: Ctrl + Shift + V
- ❑ terminate: Ctrl + C
- ❑ admin@adminpc:
  - admin: user name
  - yoneticipc: computer name



# Running C ++ Code

- Go to the directory with the file
- To compile code
- `g++ hello_world.cpp -o hello_world`
- To run
- `./hello_world`



The image shows a code editor window on the left and a terminal window on the right. The code editor displays the source code for `hello_world.cpp`, which includes `<iostream>` and has a `main` function that prints "Hello World!". The terminal window shows the execution of the compilation command `g++ hello_world.cpp -o hello_world` and the subsequent execution of the program `./hello_world`, which outputs "Hello World!".

```
hello_world.cpp (~/Desktop/Ornek) - gedit
Open Save
#include <iostream>

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

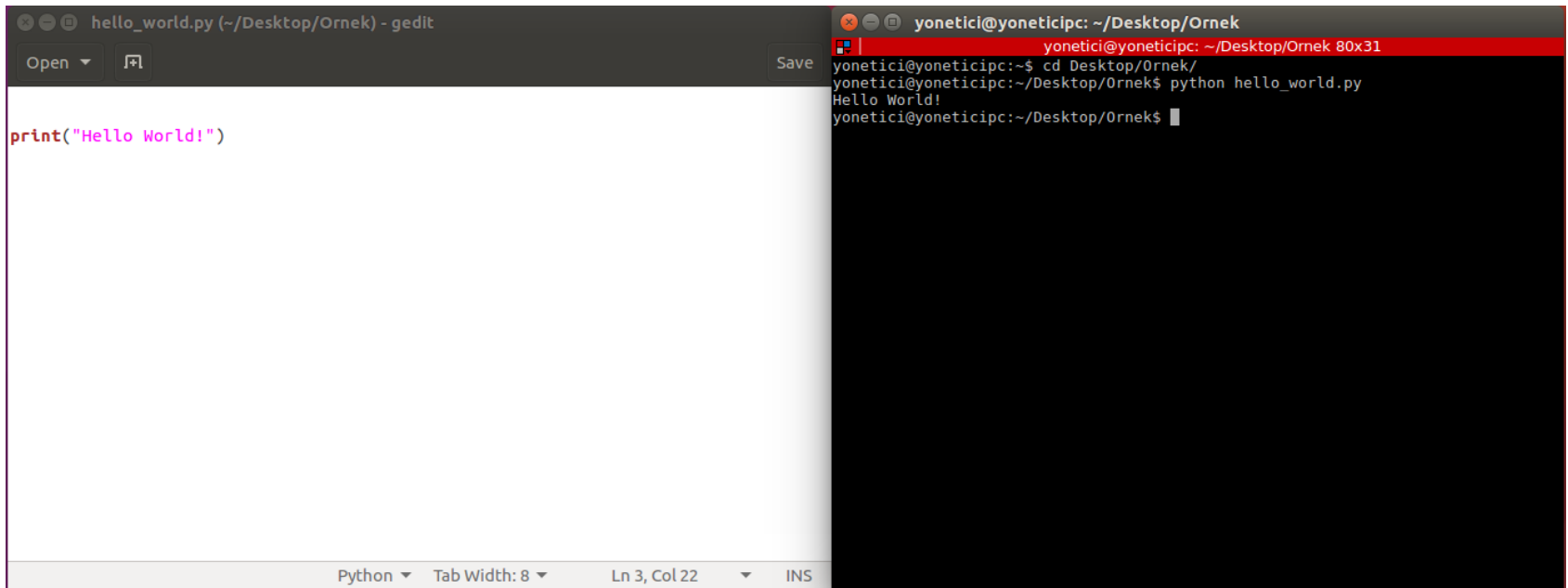
yonetici@yoneticipc: ~/Desktop/Ornek
yonetici@yoneticipc: ~/Desktop/Ornek 80x31
yonetici@yoneticipc:~/Desktop/Ornek$ g++ hello_world.cpp -o hello_world
yonetici@yoneticipc:~/Desktop/Ornek$ ./hello_world
Hello World!
yonetici@yoneticipc:~/Desktop/Ornek$
```

C++ Tab Width: 8 Ln 1, Col 20 INS



# Running Python Code

- Go to the directory with the file
- To run code
- `python hello_world.py`



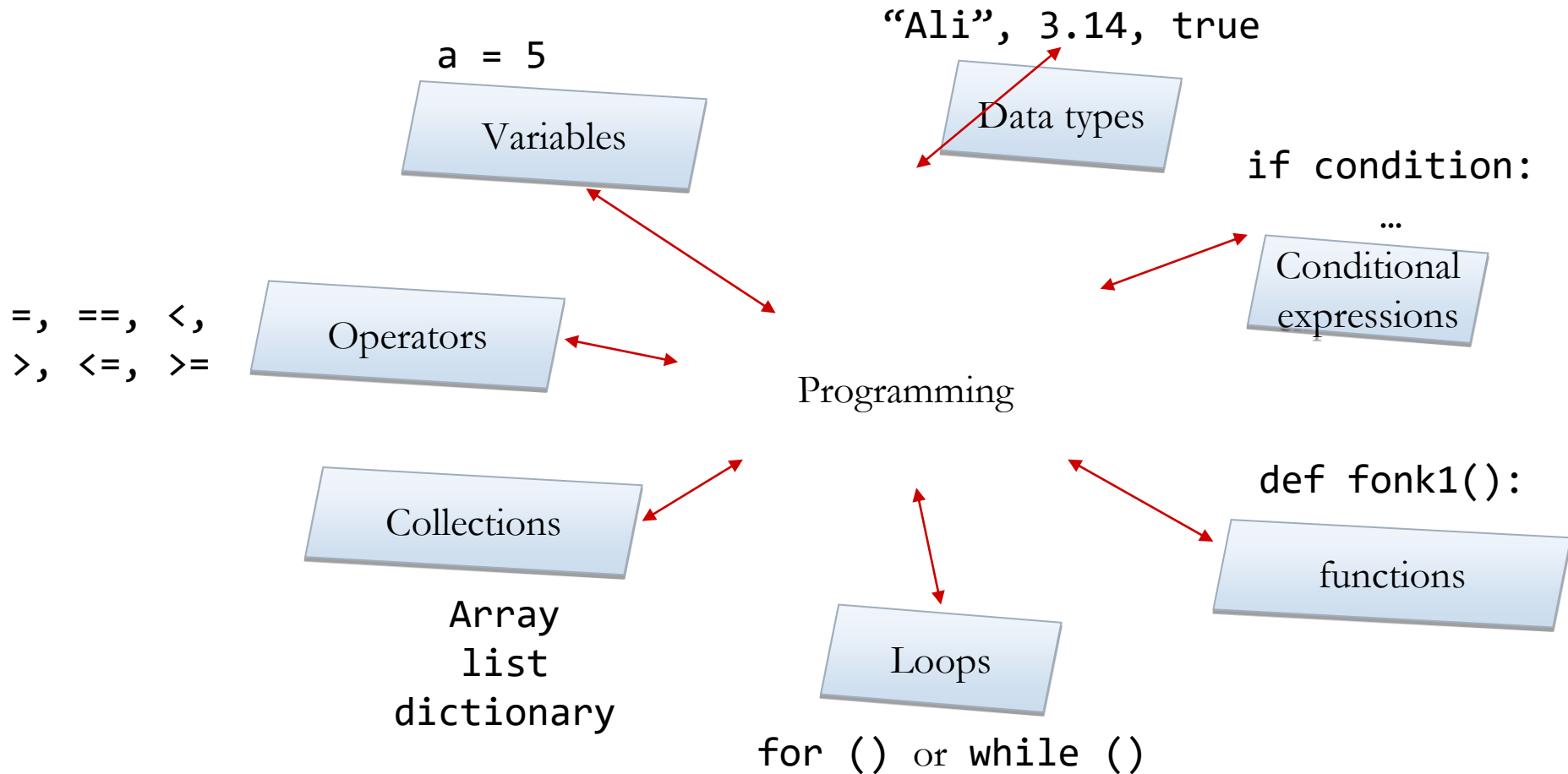
The image shows a side-by-side comparison of a code editor and a terminal window. The left window is a code editor titled 'hello\_world.py (~/Desktop/Ornek) - gedit'. It contains a single line of Python code: `print("Hello World!")`. The right window is a terminal titled 'yonetici@yoneticipc: ~/Desktop/Ornek'. It shows the execution of the script: `cd Desktop/Ornek/` followed by `python hello_world.py`, which outputs 'Hello World!'.

```
hello_world.py (~/Desktop/Ornek) - gedit
Open  Save
print("Hello World!")

Python  Tab Width: 8  Ln 3, Col 22  INS
```

```
yonetici@yoneticipc: ~/Desktop/Ornek
yonetici@yoneticipc:~$ cd Desktop/Ornek/
yonetici@yoneticipc:~/Desktop/Ornek$ python hello_world.py
Hello World!
yonetici@yoneticipc:~/Desktop/Ornek$
```

# Introduction to Python Programming



# Variables

- `a = 5`
- `b = 'Deneme'`
- `c, d = 10, 'Deneme2'`

# Data types

- integer

- 0, 1, 2, 3, ... -1, -2, -3, ...

- floating

- 1.234, 0.94, 1099.75, -100.32, ...

- String

- "Ali", "Türkiye", "Fenerbahçe", ...

- Boolean

- True, False

# Conditional expressions

□ We use three statements to indicate condition conditions:

■ if

■ elif

■ else

# Conditional expressions

```
if condition_1:  
    result_1
```

```
elif condition_2:  
    result_2
```

```
elif condition_3:  
    result_3
```

```
else:  
    result_4
```

# Functions

```
def basitFonksiyon():  
    print('Deneme')
```

```
basitFonksiyon()
```

# Functions

```
def toplama(a, b):  
    sonuc = a + b  
    return sonuc
```

```
sonuc = toplama(5, 10)
```



# Loops

- while condition:

...

- for i in range(3, 20):  
    print(i)

- for harf in metin:  
    print(harf)

# Loops

□ pass

```
parola = input("parola giriniz: ")
```

```
if not parola:  
    pass
```

■ If the user passes the password blank, it means doing nothing.

# Loops

## □ Break

```
while True:
```

```
    sayi = input("Bir sayı girin: ")
```

```
    if sayi == "iptal":
```

```
        break
```

■ If the user passes the password blank, it means doing nothing.

# Loops

## □ Continue

```
for val in "string":  
    if val == "i":  
        continue
```

■ Continue command is used to jump without leaving the loop when it satisfies the condition.

# Collections

```
□ liste = [50, 60, 'Merhaba', True, 99]
■ print(liste[2])                # 'Merhaba'
■ print(liste[1:3])              # 60, 'Merhaba'
■ print(len(liste))              # 5
■ liste.append('add_new_component')
■ liste.sort()
■
```

# Collections

```
□sozluk = {  
    'isim': 'Ayse',  
    'sehir': 'Eskisehir',  
    'yas': 22  
}
```

```
■print(sozluk['isim'])    # Ayse
```

# Operators

□ = append

□ == equal

□ > bigger

□ < smaller

□ >= greater than or equal

□ <= smaller than or equal

□ != not equal

# Finding Prime Numbers

```
sayi = int(input("Sayi Giriniz:"))  
sayac = 0
```

```
for i in range(2, (sayi + 1)):  
    kontrol = True  
    for j in range(2, i):  
        if(i % j == 0):  
            kontrol = False  
            break  
    if kontrol:  
        print(i)  
        sayac += 1
```

```
print("Toplam ", sayac, " adet asal sayı vardır.")
```



# Finding Prime Numbers

```
Sayi Giriniz:25
```

```
2
```

```
3
```

```
5
```

```
7
```

```
11
```

```
13
```

```
17
```

```
19
```

```
23
```

```
Toplam 9 adet asal sayı vardır.
```

# Matplotlib

- To download the Matplotlib library

- `pip install matplotlib`

- 

- To download Pip

- `sudo apt install python-pip`

# Plot Application

```
import matplotlib.pyplot as plt      # importing matplotlib library

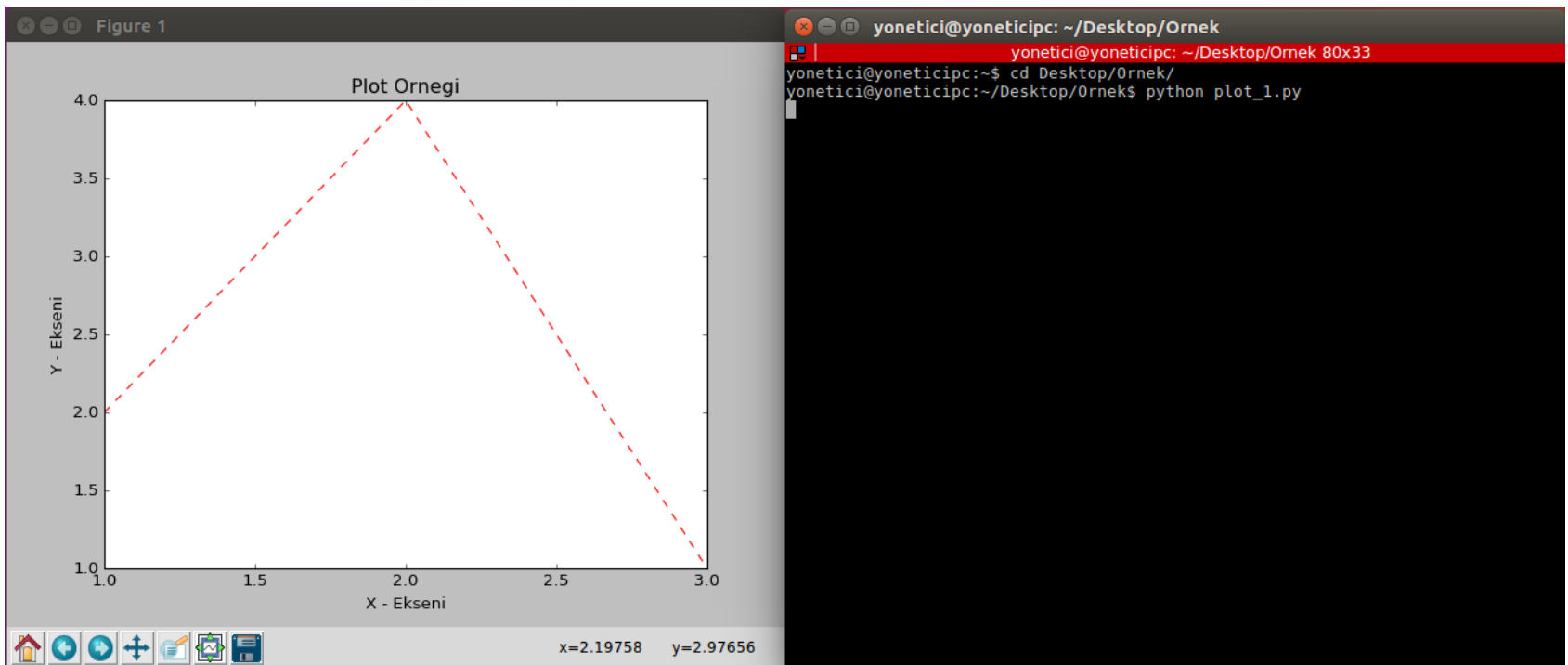
x = [1, 2, 3]                       # x axis values
y = [2, 4, 1]                       # y axis values

plt.plot(x, y, "--r", label = "Line 1") # to draw plot with dots

plt.xlabel('X - Ekseni')            # x axis name
plt.ylabel('Y - Ekseni')            # y axis name

plt.title('Plot Ornegi')             # adding header to plot
plt.show()                          # showing plot function
```

# Plot Application



# Dual Plot Application

```
import matplotlib.pyplot as plt

x1 = [1,2,3]
y1 = [2,4,1]
plt.plot(x1, y1, "--r", label = "line 1")

x2 = [1,2,3]
y2 = [4,1,3]
plt.plot(x2, y2, "-.g", label = "line 2")

plt.xlabel('X - Ekseni')
plt.ylabel('Y - Ekseni')
plt.title('Ikili Plot Ornegi')

plt.legend()
plt.show()
```

# Dual Plot Application

