**IT FUNDAMENTALS SUMMARİES**

**İNDEX:**

1. **Computer Fundamentals**
2. **Computational Thinking**

# **Linux**

# **GİT/GİTHUB**

# **Python**

# **SQL**

# **SDLC**

# **Agile**

# **Jira**

# **HTML**

# **CSS**

**1. COMPUTER FUNDAMENTALS**

**Computer Four Tasks:**

receiving input ------- storing ------- processing ------- giving output.

The Brain Of The Computer : CPU, The Central Processing Unit

Store Data Temporarily : RAM, Random Access Memory

Store The Data Permanently : Storage Devices

**Operating System (OS)** is the main program that runs on a computer. It communicates with the hardware. It’s the bridge between the hardware and all other software programs.

**Deadlock** is a situation when two or more processes wait for each other to finish and none of them ever finish.

**Operating Systems use three types of operations:**

**Multiprogramming**, different programs run simultaneously and share the same CPU. For example while playing a game, you can listen to music in the background.

**Multiprocessing** is using multiple CPUs for running different programs at the same time.

**Multithreading** operation allows a single program to run different tasks at the same time.

**Open Source** Software is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose.

**Free software** is the software that grants the user the freedom to share, study, and modify it.

**Shell** is an interface between the user and the kernel

All the detailed information about a *file* stored on the computer is called the **metadata** of that file. n bits yields 2^n patterns

#### 

#### File Attributes: Read-Only (R) System (S) Hidden (H)

Everything in a computer is 0's and Anything with two separate states can store 1 bit. Group 8 bits together to make 1 byte. One byte = collection of 8 bits.

**ASCII** (American Standard Code for Information Interchange) is an encoding representing

**Unicode** is a universal character encoding standard.The standard ASCII character set only supports 128 characters, while Unicode can support roughly 1,000,000 characters.

**Software** is a set of instructions, data or programs used to operate computers and execute specific tasks.

**Machine language**. As mentioned, this language has only two letters: 0 and 1. The binary code.Computers have different types of machine language. These are **Assembly Language** (low level, uses mnemonic words) and **high-level programming languages**.

An **assembler** is a program that converts assembly language into machine code.

**High-level languages** emerged in the 1950s. These languages are similar to human-level languages.

A program in a **high-level language** is called source program or source code. High-level languages should use tools called *compiler* or *interpreter* to make this translation.

A **software library** generally consists of pre-written code, classes, procedures, scripts, configuration data and more.

**Software Package** is a set of software that fulfills a particular function, for example, installation on the desktop.

**Package Managers** are the tool used to manage Software and [Libraries](https://lms.clarusway.com/mod/lesson/view.php?id=917) (together called Packages) in your OS.

**Frameworks** are like the father of software [libraries](https://lms.clarusway.com/mod/lesson/view.php?id=917). Frameworks are collections of [libraries](https://lms.clarusway.com/mod/lesson/view.php?id=917), classes, functions, and constants designed to complete a task easier. (node, django, express)

**The Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components:

**Model**, the data represented in some way,

**View**, what the user sees. .

**Controller**, the code that manipulates the data in the database.

**Frontend** (“client-side”, user experiences),and Backend“server-side”,storing and organizing data) development terms are related to web and application development.

The **internet** is the network of networks.This is not a centralized system. it’s a **fully distributed**, **end-to-end communication system**.

**Local Area Networks (LAN)** are small scale networks.(Ethernet, WiFi)

**A wide area network (WAN)** is a *geographically* distributed private telecommunications network that interconnects multiple (LANs)

TCP/IP, **Transmission Control Protocol/Internet Protocols** are separate and individual protocols in the suite of internet protocols.It describes how the data will get transmitted and routed from end to end communication.

**IP number system** has two versions.IPv4 (32 bit), IPv6(128 bit)

## Domain Name System (DNS)

**"https://clarusway.com/about/"** and includes the following elements:

1. **https://:** Specifies the protocol used to access web site address
2. **clarusway.com:** Domain name
3. **/about/:** The path to the directory.
4. **www:** Subdomain
5. **com:**Top Level Domain
6. **www.clarusway.com :** Fully Qualified Domain Name

**DNS** is an internet address mapping process with the local name. We can also call it as an internet phonebook.

**Servers** are computers that provide data to other computers.

Many types of servers exist, including web servers, mail servers, and file servers.

**Databases** are collections of data,A database is usually controlled by a database management system **(DBMS)**.

There are two broad types of databases. These are **SQL** and **NoSQL**.SQL databases are table-based, (**relational** databases. NoSQL database has a dynamic schema for unstructured data. (**nonrelational** database)

**Big data** is data that contains greater **variety** arriving in increasing **volumes** and with ever-higher **velocity**.

**The cloud** refers to [servers](https://lms.clarusway.com/mod/lesson/view.php?id=1015) that are accessed over the Internet and the software and [databases](https://lms.clarusway.com/mod/lesson/view.php?id=995) that run on those [servers](https://lms.clarusway.com/mod/lesson/view.php?id=1015)

**Virtualization** allows for the creation of a simulated, digital-only virtual computer/machine that behaves as if it were a physical computer with its hardware

### 

### Service models:

**Software-as-a-Service (SaaS)**: Instead of users installing an application on their device,

**Platform-as-a-Service (PaaS):** In this model, companies pay for the things they need to build their own applications.

**Infrastructure-as-a-Service (IaaS):** In this model, a company rents the [servers](https://lms.clarusway.com/mod/lesson/view.php?id=1015) and storage they need from a cloud provider.

**Function-as-a-Service (FaaS)**: FaaS, also known as serverless computing, breaks cloud applications down into even smaller components

### Types of Cloud Deployments

Private cloud Public cloud Hybrid cloud Multicloud

A **browser** is a software designed to find and display content on the **World Wide Web**. This content might be a web page, a pdf document, a picture, a video or any other content. Web browsers are also able to run **CSS (Cascading Style Sheet)** and Javascript code.

Web browsers have **seven high level components**. These are *User Interface, Browser Engine, Rendering Engine, Networking, Javascript Interpreter, UI Backend and Data Storage.*

**Hypertext Transfer Protocol (HTTP)** is an application layer designed within the framework of internet protocol suite. It is used for transferring text, image, sound, video or any other type of multimedia [files](https://lms.clarusway.com/mod/lesson/view.php?id=1052).

The statement written in the address bar is called **Uniform Resource Locator (URL)**. he system that is responsible to hold the values of these keys is [Domain Name System (**DNS**)](https://lms.clarusway.com/mod/lesson/view.php?id=943) [servers](https://lms.clarusway.com/mod/lesson/view.php?id=1015). If the DNS server knows the address, it answers

**Following are the header fields in HTTP:**

* **General header:** It applies for both request and response message.
* **Request header**: It contains information for the request message.
* **Response header**: It is used to contain response header information sent by the web server.
* **Entity header**: It is used to contain more information about the body of the entity.

**SSL stands for Secure Sockets Layer** and, in short, it's the standard technology for keeping an internet connection secure

**TLS (Transport Layer Security)** is just an updated, more secure, version of SSL.

**HTTPS (Hyper Text Transfer Protocol Secure)** appears in the URL when a website is secured by an SSL certificate.

A **cookie** is a tiny little file that’s stored on your computer.Cookie provides a simple way to identify session among a group of HTTP/HTML requests.

**Security** is the state of being free from danger or threat. 3 main types:

*1. Physical Security 2. Software Security 3. Network Security*

**Access Control** are *Identification, Authentication, and Authorization*.

## Cyber Attacks

**Phishing** is the fraudulent attempt to obtain sensitive information

The term **malware** is a contraction of malicious software. Put simply, malware is any piece of software that was written with the intent of damaging devices, stealing data, and generally causing a mess.

**Spyware** is a malware that aims to steal data about the organization or person.

**SQL Injection (SQLi)** is a code injection attack where an attacker manipulates the data

**Cross-Site Scripting (XSS)** attacks are a type of injection,

**Cross-site Request Forgery (CSRF)** is a very common vulnerability. It's an attack that forces a user to execute unwanted actions

**Artificial Intelligence** is a field of computer science wherein the cognitive functions of the human brain are studied and tried to be replicated on a machine/system.

**John McCarthy,** who is the father of Artificial Intelligence, described Artificial Intelligence as “The science and engineering of making intelligent machines, especially intelligent computer programs”.

**The Turing test** is a method to test a *machine’s ability* to match the *human-level intelligence*.

**Machine Learning?** It’s the science of getting computers to act by feeding them data so that they can learn a few tricks on their own, without being explicitly programmed to do so.

**Regression**  : try to predict one variable's value based on other (known) variables.

**Classification** : Try to categorize variables.

Machine learning algorithms are often categorized as **supervised** or **unsupervised**. In supervised classification, the images are manually fed and interpreted by the Machine Learning expert to create feature classes.

In **unsupervised classification**, the Machine Learning software creates feature classes based on image pixel values.

**Deep learning** is a subset of machine learning, and machine learning is a subset of Artificial Intelligence

**Natural language processing (NLP)** is a branch of *artificial intelligence* that helps computers understand, interpret and manipulate human language.

**Robotics** is a branch of AI, which is composed of **Electrical Engineering**, **Mechanical Engineering**, and **Computer Science** for *designing*, *construction*, and *application* of robots.

**2. COMPUTATIONAL THINKING:**

Computational thinking is an approach to solving problems using concepts and ideas from computer science and expressing solutions to those problems so that they can be run on a computer.

### Pillars of Computational Thinking

1. **Decomposition**: Breaking down data, processes, or problems into smaller, manageable parts

2. **Pattern Recognition**: Observing patterns, trends, regularities in data

3. **Abstraction**: Focusing on important parts only, ignoring irrelevant details

4. **Algorithm Design**: Developing the step by step instructions for solving this and similar problems

### 

### There are two main ways to *represent an algorithm*: Pseudocode and flowchart.

### 

### Pseudocode

Pseudocode is an informal high-level description of a computer program or algorithm.

***INPUT*** asks a question. ***OUTPUT*** prints a message on the screen.

Pseudocode makes creating programs easier.

**Advantages of pseudocode:**

* Pseudocode is understood by the programmers of all types.
* It enables the programmer to concentrate only on the algorithm part of the code development.
* It cannot be compiled into an executable program.

### 

### 

### How to Write Pseudocode

### 

### Statements

A statement is defined as an instruction that directs the computer to perform a specific action.

* **Mathematical operations** They allow us to manipulate the values we have stored.
* **Keywords** can be commands or parameters.

### Conditionals IF — ELSE IF — ELSE

**CASE**

Case structures are used if we want to compare a single variable against several conditions.

**The OTHERS** clause with its statement is optional. Conditions are normally numbers or characters.

### 

### Iteration

To iterate is to repeat a set of instructions in order to generate a sequence of outcomes. We iterate so that we can achieve a certain goal.

**FOR structure**

The FOR loop takes a group of elements and runs the code within the loop for each element.

**WHILE structure**

a way to repeat a block of code as long as a predefined condition remains true.

**Functions**

every-time we need them to run.

### 

### Program Wrapping

This is to improve readability and make the execution flow easier to understand.

### 

### Exception Handling

An exception is an event which occurs during program execution that disrupts the normal flow of the instructions. These are events that are non-desirable.

### Conclusion

There are no technical rules for Pseudocode. It is meant to be human-readable and still convey meaning and flow.

**Tree structure**

A style of depiction often used to indicate hierarchical relationships, such as the relationships among the subproblems and the bigger problems.

### 

### How to Write Pseudocode

1. Understand the Uses

2. Pseudocode is Subjective

3. Algorithms and Basic Constructs

- Understand Algorithms –

- Know Algorithm Flow –

- Combine the Pieces –

4. Standard Procedure

- One Statement Per Line –

- Capitalize Directions – (for example, “READ”).

- Focus on Meaning –

- Standard Programming Structures – .

- Utilize Blocks –

5. Important Tips

- Keep It Simple –

- Explain Everything –

- Practice Makes Perfect –

- Review the Pseudocode –

- Translate into Programming Language –

### Flowchart

A flowchart is a diagram that represents a set of instructions. Flowcharts normally use standard symbols to represent different instructions.

What links each instruction in a flowchart? An arrow

There are few real rules about the level of detail needed in a flowchart.

Sometimes flowcharts are broken down into many steps to provide a lot of detail about exactly what is happening.

**3. LİNUX:**

Linux is an **operating system** just like Windows, iOS, and Mac OS.

***Linus Torvalds*** invented Linux.

In ***1991*** he released version 0.02; Version 1.0 of the Linux kernel, the core of the operating system, was released in **1994**.

At about the same time, American software developer ***Richard Stallman*** and the FSF made efforts to create an open-source UNIX-like operating system called **GNU**.

* Linus announces the kernel to the world (1991).
* Release of the first "major" Linux distributions (1993).
* Linux kernel hits 1.0 (1994).
* KDE (1996) / GNOME (1999) projects are released.
* First release of Linux-based Android (1998).
* Kernel development moves to Git (2005).
* Today Linux is everywhere. Supercomputers, smartphones, desktop, web servers, tablets, laptops and home appliances like washing machines, DVD players, routers, modems, cars, refrigerators, etc use Linux OS.

### The components of Linux

## Hardware physical components

## Boot-loader manage the boot procesS

## Kernel the core part of the OS. lowest level of the OS.

## GNU Core core utilities exist on every OS

## Graphical User Interface (GUI) This is the piece that the users actually interact with. (GNOME, Cinnamon, Mate, Pantheon, Enlightenment, KDE, Xfce).

## Applications

### Popular Linux Distributions

## Linux Mint - Debian (appeared first) - Ubuntu - OpenSUSE - Manjaro - Fedora - Red Hat Enterprise Linux (RHEL)

***Embedded System***

This system is embedded as part of a complete computer system including hardware such as mechanical and electrical components.

Many systems that use Linux embedded are:

Smart TVs -- Tablet PCs -- Navigation devices -- Wireless routers

Other industrial and consumer electronic equipment

### What is open source?

**Open-source software** is software with source code that anyone can inspect, modify, and enhance.*LibreOffice* and the *GNU Image Manipulation Program* are examples of open-source software.

## 

## *Types of open source license*

**Permissive licenses** can be summarized as follows:

* Do whatever you want with the code
* Use at your own risk
* Acknowledge the author/contributor

**Copyleft licenses** also require that:

* If you distribute binaries, you must make the source code for those binaries available
* The source code must be available under the same copyleft terms under which you got the code
* You cannot place additional restrictions on the licensee's exercise of the license

## 

## *Popular Licenses*

* Apache License 2.0
* BSD 3-Clause "New" or "Revised" license
* BSD 2-Clause "Simplified" or "FreeBSD" license
* GNU General Public License (GPL)
* GNU Library or "Lesser" General Public License (LGPL)
* MIT license
* Mozilla Public License 2.0
* Common Development and Distribution License
* Eclipse Public License version 2.0

## Free Software Foundation (FSF)

The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom.

The Free Software Foundation is working to secure freedom for computer users by promoting the development and use of free (as in freedom) software and documentation

## 

## *Open Source Initiative (OSI)*

The Open Source Initiative (OSI) is a non-profit organization dedicated to the promotion of open-source software. OSI was founded in **1998 by Bruce Perens and Eric Raymond**. OSI is quite distinct from the Free Software Foundation (FSF) led by Richard Stallman. Although they have similar history and motivation, OSI considers its ends as more pragmatic and business-driven, while FSF is based on anti-establishment and moralistic viewpoints.

### 

### Desktop Applications

## Firefox- Thunderbird- LibreOffice- GIMP-

### 

### Server Applications

## Apache Web Server- NGINX- MySQL- Samba- ownCloud

### 

### Development Languages

## Shell - C - Java - JavaScript - Perl - Python - PHP

### 

### Package Management Tools

Contemporary distributions of Linux-based operating systems install software in pre-compiled packages, which are archives that contain binaries of software, configuration files, and information about dependencies.

## 

## dpkg “ -- rpm: Red Hat Package Manager

## apt-get -- yum: yellowdog updater modified

### SHELL

* Shell is an interface between an end-user and the Linux system.
* In other words, Shell is a program that receives commands from the user, relays them to the operating system to process and displays the output. Shell is one of the main parts of Linux OS. Each Linux distro comes with a GUI (Graphical User Interface), but essentially Linux has a CLI (Command-Line Interface).

### *Shell Types:*

**C Shell** : If you are using a C-type shell, t% is the default prompt.

* C shell (csh)
* TENEX/TOPS C shell (tcsh)

**Bourne Shell** : If you are using a Bourne-type shell, $ is the default prompt.

Bourne shell (sh) --- Korn shell (ksh)

Bourne Again shell (bash) ---- POSIX shell (sh)

An enhanced version of SH is called BASH (which stands for Bourne Again SHell) and serves as the main shell program on the most Linux systems.

The standard Linux shell (BASH) is both a command-line interpreter and a programming language.

**The most common interpreter is BASH or the Bourne Again Shell, but there are others available as well and some of them does not use the dollar sign.**

### Command Prompt

The command prompt at the beginning of the command line is a short text string.

Dollar sign ($) means you are a normal user and indicates you are logged in with the normal permissions.

Hash (#) means you are the system administrator (root) and indicates you are logged in with root privileges.

The "root" account on a Linux computer is the account with full privileges.

Root access is often necessary for performing commands in Linux, especially commands that affect system files.

### Basic Shell Commands

This is a list of most frequently used Linux commands.

* The **~ (tilde)** symbol stands for your home directory.
* The **pwd** (stands for print working directory) command will allow you to know in which directory you're located.
* The **cp** command will make a copy of a file.
* The **cd** command will allow you to change directories.
* The **rm** command removes or deletes a file in your directory.
* The **rmdir** command will delete an empty directory.
* The **mkdir** command will allow you to create directories
* The **mv** command will move a file to a different location or will rename a file.
* **cd -** Navigate to the last directory you were working in.
* **cd ~** or just cd Navigate to the current user's home directory.
* **cd ..** Go to the parent directory of current directory

### 

### 

### 

### Quoting

There are three types of quotes:

1. **Double** Quotes : The double quote " preserve the literal value of most characters contained within the quotes, exceptions include $ (for variables), ' (for single quoting), \ (for escaping a character) .
2. **Single** Quotes: The single quote ( 'quote' ) protects everything enclosed between single quotation marks.
3. **Backslash**: Use the backslash to change the special meaning of the characters or to escape special characters within the text such as quotation marks.

### File Permission

Each file and directory on Linux system has 3 types of owners assigned,

* ***User*** : A user is the owner of the file.
* ***Group*** : A user- group can contain multiple users.
* ***Other/All*** : Any other user who has access to a file.

### 

### Permission (chmod)

Each file and directory on your Linux system has 3 permissions defined for all the 3 owners.

* ***r - (7)- Read :*** The read permission gives you the authority to open and read a file.
* ***w - (2) - Write*** : The write permission gives you the authority to modify the contents of a file.
* ***x- (1) - Execute*** : In Linux, you cannot run a program unless the execute permission is set.
* ***-*** no permision

**754 code says;**

* Owner can read, write and execute
* Usergroup can read and execute
* Other can only read

### Ping Command

Ping or Packet Internet Groper is a network administration utility used to check the connectivity status between a source and a destination computer/device over an

IP network. It also helps you assess the time it takes to send and receive a response from the network.

ping host-name/IP

### SSH

* ssh stands for “Secure Shell”.
* It is a protocol used to securely connect to a remote server/system.
* ssh is secure in the sense that it transfers the data in an encrypted form between the host and the client

ssh user@host(IP/Domain\_name)

### *whoami*

Displays user, group and privileges information for the user who is currently logged on to the local system.

### *Man Pages*

* A man page (short for manual page) is a form of software documentation usually found on a Unix or Unix-like operating system.
* Man pages are the traditional package documentation for application usage.

The man page for a particular command is invoked by preceding the command with man.

man <command> man ls

### NAME

Program or Function name(s) followed by descriptions of functionality.

### 

### SYNOPSIS

A short overview of available options

### 

### DESCRIPTION

Detailed information about arguments and options.

### 

### Info Pages

Info pages are additional documentation with more robust capability in detail.

### 

The info page for a particular command is invoked by preceding the command with info.

info <command> info echo

**The main difference between Man and Info is the amount of content that they have; Info contains a whole lot more than Man does.**

***If no info page exists, info can pull documentation from the man page.***

***The file system hierarchy standard (FHS)*** defines the structure of the file systems on Linux.

* In the FHS, all files and directories appear under the root directory / , even if they are stored on different physical or virtual devices.
* ***/ (Root) :*** Primary hierarchy root and root directory of the entire file system hierarchy. Every single file and directory starts from the root directory.
* ***/bin :*** Essential command binaries that need to be available in single user mode.
* ***/boot :*** Boot loader files.
* ***/dev :*** Essential device files.
* ***/etc :*** Host-specific system-wide configuration files.
* ***/home :*** Users’ home directories, containing saved files. Home directories for all users to store their personal files
* ***/lib :*** Libraries essential for the binaries in /bin/ and /sbin/
* ***/mnt*** : Temporarily mounted filesystems.
* ***/opt :*** Optional application software packages.
* ***/proc :*** Contains information about system process.
* ***/sbin :*** Essential system binaries, e.g., fsck, init, route.
* ***/srv :*** Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories
* ***/tmp :*** Temporary files. Often not preserved between system reboots, and may be severely size restricted.
* ***/usr :*** Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

Hidden files can be listed via ls using the -a option (meaning show all files).

### 

### *Absolute and Relative Paths*

The path to a file or directory location can be specified as an absolute path (starts with a /), or a relative path. ,

### *Relative Path*

Below, the cat command is used to show the content of file1 specified with a relative path. [cloud\_user@ip]$ cat file1

### *Absolute Path*

Below, the cat command is used to show the content of same file1 specified with a absolute path. [cloud\_user@ip]$ cat /home/cloud\_user/file1

***Creating, moving, and deleting files and directories.***

### Working with directories

Create a new directory mkdir <NAME>

Copy a directory cp -r <SOURCE> <DESTINATION>

Move a directory mv <SOURCE> <DESTINATION>

Delete a directory rm -r <DIRECTORY>

Delete an empty directory rmdir <DIRECTORY>

### 

### Working with files

Create a new file touch filename

Removing Files rm -option filename

Option -r includes the contents of a directory and the contents of all subdirectories when you remove a directory.

Option -i prevents the accidental removal of existing files or directories.

### *Case Sensitivity*

Most of the common Linux file systems are case sensitive;

### 

### Simple Globbing

Globbing is primarily used to match patterns in filenames or text by using a wildcard character to create the pattern.

**? (Question mark)** : Match any single character

**\* (Asterisk) :** Match any number of character(s)

**[] (Brackets) :** Match character from a range

**^ (Caret) :** Used to match starting character

**$ (Dollar sign) :** Used to match ending character

**{} (Curly brace)** : Used to match more than one pattern

***| Pipe : Used for applying more than one condition***

* **[[:upper:]] or [[A-Z]] - match upper-case character**
* **[[:lower:]] or [[a-z]] - match lower-case character**
* **[[:digit:]] or [[0-9]] - match digits**
* **[[:alpha:]] or [[a-zA-Z]] - match either case character**
* **[[:alphanum:]] or [[a-zA-Z0-9]] - match alphanumeric**

## 

## What’s Vim:

Vim stands for “Vi Improved”. Vim is a highly configurable **text editor** for efficiently creating and changing any kind of text. It is included as "vi" with most UNIX systems and with Apple OS X.

Vim can operate in two modes: **command mode** and **insert mode**.

| a | append: it moves the cursor one position to the right before switching to insert mode |
| --- | --- |
| i | insert |
| o | insert a blank line under the current cursor position and move the cursor to that line What’s nano: GNU nano is a small and friendly text editor. Besides basic text editing, |

## What is Linux?

Linux is an operating system based on UNIX and was first introduced by Linus Torvalds. It is based on the Linux Kernel and can run on different hardware platforms manufactured by Intel, MIPS, HP, IBM, SPARC, and Motorola. Another popular element in Linux is its mascot, a penguin figure named Tux.

## What is the difference between UNIX and LINUX?

Unix originally began as a propriety operating system from Bell Laboratories, which later on spawned into different commercial versions. On the other hand, Linux is free, open source and intended as a non-propriety operating system for the masses

## What is BASH?

BASH is short for Bourne Again SHell. It was written by Steve Bourne as a replacement to the original Bourne Shell (represented by /bin/sh). It combines all the features from the original version of Bourne Shell, plus additional functions to make it easier and more convenient to use. It has since been adapted as the default shell for most systems running Linux.

## What is Linux Kernel?

The Linux Kernel is a low-level systems software whose main role is to manage hardware resources for the user. It is also used to provide an interface for user-level interaction.

## What is the advantage of open source?

Open source allows you to distribute your software, including source codes freely to anyone who is interested. People would then be able to add features and even debug and correct errors that are in the source code. They can even make it run better and then redistribute these enhanced source code freely again. This eventually benefits everyone in the community

## What are the basic components of Linux?

Just like any other typical operating system, Linux has all of these components: kernel, shells and GUIs, system utilities, and an application program. What makes Linux advantageous over other operating system is that every aspect comes with additional features and all codes for these are downloadable for free.

## Describe the root account.

The root account is like a systems administrator account and allows you full control of the system. Here you can create and maintain user accounts, assigning different permissions for each account. It is the default account every time you install Linux.

## What is CLI?

CLI is short for Command Line Interface. This interface allows the user to type declarative commands to instruct the computer to perform operations. CLI offers greater flexibility. However, other users who are already accustomed to using GUI find it difficult to remember commands including attributes that come with it.

## What is GUI?

GUI, or Graphical User Interface, make use of images and icons that users click and manipulate as a way of communicating with the computer. Instead of having to remember and type commands, the use of graphical elements makes it easier to interact with the system, as well as adding more attraction through images, icons, and colors.

## 

## How do you change permissions under Linux?

Assuming you are the system administrator or the owner of a file or directory, you can grant permission using the chmod command. Use + symbol to add permission or – symbol to deny permission, along with any of the following letters: u (user), g (group), o (others), a (all), r (read), w (write) and x (execute). For example, the command chmod go+rw FILE1.TXT grants read and write access to the file FILE1.TXT, which is assigned to groups and others.

## What are filenames that are preceded by a dot?

In general, filenames that are preceded by a dot are hidden files. These files can be configuration files that hold important data or setup info. Setting these files as hidden makes it less likely to be accidentally deleted.

## What are the different modes when using vi editor?

There are 3 modes under vi:- Command mode – this is the mode where you start in- Edit mode – this is the mode that allows you to do text editing- Ex mode – this is the mode wherein you interact with vi with instructions to process a file

## What is a Shell script?

As the name suggests, the shell script is the script written for the shell. This is a program file or says a flat text file where certain Linux commands are executed one after another. Although the execution speed is slow, Shell script is easy to debug and can also simplify everyday automation processes.

## Name the Linux loader.

LILO is the Linux loader.

## What is the maximum length for a filename in Linux?

**255** characters.

## How to delete information from a file in vi?

The following commands are used to delete information from vi editors.

1. **x** deletes a current character.
2. **dd** deletes the current line.

**4.GIT / GITHUB**

GIT is a ***free and open-source*** **version control system** designed to handle everything from small to very large projects with speed and efficiency.

GIT is a distributed version control system for **tracking changes in source code** during software development.

GIT has the ***functionality***, ***performance***, ***security*** and ***flexibility*** that most teams and individual developers need. Git is the most broadly adopted tool of its kind.

**Brief History of Git**

In 2002, the Linux kernel project began using a proprietary Version Control System called ***BitKeeper***.

In 2005, the relationship between the community that developed the Linux kernel and the commercial company that developed BitKeeper broke down, and the tool’s free-of-charge status was revoked. This prompted the Linux development community (and in particular ***Linus Torvalds***, the creator of Linux) to develop their own tool based on some of the lessons they learned while using BitKeeper. **That's how Git was borned**.

***Git allows you to:***

* Revert files to previous state,
* Revert entire project back to previous state,
* Compare changes over time,
* See who modified what? **And much more...**

It means if you screw things up or lose files, you can easily recover.

Use cases:

Individual development, Collaborative development, Offline usage.

**Why Git?**

* Everything is local (full history tree available offline),
* Everything is fast,
* Snapshots, not diffs,
* It is distributed not centralized,
* Great for those who hate: CVS/SVN (earlier [version control systems](https://lms.clarusway.com/mod/lesson/view.php?id=642)).

*A* ***V****ersion* ***C****ontrol* ***S****ystem* (also called Revision/Source Control System) is a software designed to record changes made to files over time so that you can recall specific versions later.

A ***VCS*** is a tool to track changes of source code and learning

* What had changed
* When it changed
* Why it changed
* Who changed it

Here are the reasons to use it. Files need to be stored somewhere. You can store them anywhere you like , but if you store them in a VCS, you never lose them. Every reported change (check-in) associated with a file are also available. All the previous versions of a file can be easily extracted/restored.

There are three types of version control systems:

File-based, Client-server type, Distributed.

**File-based** version control systems are obsolete and not much used. Distributed VCS's are more common on the market. Git is a distributed type of VCS.

**Distributed version control systems** create replicas of the repository on each computer. Every user has to work on a replica and can do so even being disconnected from the network. They are suited for large projects and independent developers who can work independently and commit the changes for merging.

### What is a Repository?

A repository is a directory or storage space where your projects can live. Sometimes it is shortened to “**repo**.” It can be local to a folder on your computer, or it can be a storage space on the cloud. You can keep code files, text files, image files, etc. inside a repository.

Git understands all the versions of the files and folders in your project and saves them in a special way. All your files and folders will be stored in **Git Database**.

In a Git repository your file can reside in three main states:

* **Modified** means that you have changed the file but have not committed it to your database (repo) yet.
* **Staged** means that you have marked a modified file in its current version to go into your next commit snapshot.
* **Committed** means that the data is safely stored in your local database.

This leads us to the three main sections of a Git project:

**The working area** is a single checkout of one version of the project. These files are pulled out of the compressed database in the Git directory and placed on disk for you to use or modify.

**The staging area** is a file, generally contained in your Git directory, that stores information about what will go into your next commit. Its technical name in Git terminology is the “index”, but the phrase “staging area” works just as well.

**The Git directory** (.git) is where Git stores the metadata and object database for your project.

**GitHub**

GitHub is a **Git repository hosting** (***Source Code Hosting***) service , but it adds many of its own features. It is a web-based platform used for version control and it provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

Like GitHub, there are other source code hosting platforms but GitHub is the most popular one.

**What is the difference between Git and GitHub?**

Git is a version control system that lets you manage and keep track of your source code history locally. GitHub is a cloud-based hosting service that lets you manage Git repositories.

## What is Git?

Git is a free and open source, distributed version-control system for tracking changes in source code during software development.

## What is the major focus of a Version Control System?

A major focus of version control system (also known as revision control or source control) is to manage the changes to the files, programs, logs, and other information related to code development, code deployment, and code operation.

## What is a Repository?

A repository is a directory or storage space where your projects can live. You can keep code files, text files, image files, etc. inside a repository.

## What are the Git states?

o Modified, o Staged, o Committed.

**How do you create an empty local repo?**

I execute the command $ git init in the file that I want to create the repo.

## How do you add your files to staging area from the working area?

With the command $ git add .

## Configure tooling

Configure user information for all local repositories

$ git config --global user.name "[name]"

Sets the name you want attached to your commit transactions

$ git config --global user.email "[email address]"

Sets the email you want attached to your commit transactions

$ git config --global color.ui auto

Enables helpful colorization of command line output

## Branches

Branches are an important part of working with Git. Any commits you make will be made on the branch you’re currently “checked out” to. Use git status to see which branch that is.

$ git branch [branch-name]

Creates a new branch

$ git checkout [branch-name]

Switches to the specified branch and updates the working directory

$ git merge [branch]

Combines the specified branch’s history into the current branch. This is usually done in pull requests, but is an important Git operation.

$ git branch -d [branch-name]

Deletes the specified branch

## Make changes

Browse and inspect the evolution of project files

$ git log

Lists version history for the current branch

$ git log --follow [file]

Lists version history for a file, beyond renames (works only for a single file)

$ git diff [first-branch]...[second-branch]

Shows content differences between two branches

$ git show [commit]

Outputs metadata and content changes of the specified commit

$ git add [file]

Snapshots the file in preparation for versioning

$ git commit -m "[descriptive message]"

Records file snapshots permanently in version history

## Redo commits

Erase mistakes and craft replacement history

$ git reset [commit]

Undoes all commits after [commit], preserving changes locally

$ git reset --hard [commit]

Discards all history and changes back to the specified commit

CAUTION! Changing history can have nasty side effects. If you need to change commits that exist on GitHub (the remote), proceed with caution. If you need help, reach out at [github.community](https://github.community/) or contact support.

## Create repositories

A new repository can either be created locally, or an existing repository can be cloned. When a repository was initialized locally, you have to push it to GitHub afterwards.

$ git init

The git init command turns an existing directory into a new Git repository inside the folder you are running this command. After using the git init command, link the local repository to an empty GitHub repository using the following command:

$ git remote add origin [url]

Specifies the remote repository for your local repository. The url points to a repository on GitHub.

$ git clone [url]

Clone (download) a repository that already exists on GitHub, including all of the files, branches, and commits

## The .gitignore file

Sometimes it may be a good idea to exclude files from being tracked with Git. This is typically done in a special file named .gitignore. You can find helpful templates for .gitignore files at [github.com/github/gitignore](https://github.com/github/gitignore).

## Synchronize changes

Synchronize your local repository with the remote repository on GitHub.com

$ git fetch

Downloads all history from the remote tracking branches

$ git merge

Combines remote tracking branches into current local branch

$ git push

Uploads all local branch commits to GitHub

$ git pull

Updates your current local working branch with all new commits from the corresponding remote branch on GitHub. combination of git fetch and git merge

## Glossary

* **git**: an open source, distributed version-control system
* **GitHub**: a platform for hosting and collaborating on Git repositories
* **commit**: a Git object, a snapshot of your entire repository compressed into a SHA
* **branch**: a lightweight movable pointer to a commit
* **clone**: a local version of a repository, including all commits and branches
* **remote**: a common repository on GitHub that all team members use to exchange their changes
* **fork**: a copy of a repository on GitHub owned by a different user
* **pull request**: a place to compare and discuss the differences introduced on a branch with reviews, comments, integrated tests, and more
* **HEAD**: representing your current working directory, the HEAD pointer can be moved to different branches, tags, or commits when using git checkout

**5. PYTHON BASİC, PYTHON PLUS:**

**Python** is a ***programming language*** (such as Java, C ++, R, Ruby, and so on). Like other programming languages, it allows you to control the machine in front of you, the computer.

The benefits of Pythons are that it is **simple** and **easy**, **portable**, **extensible**.

The IT industry is booming with Data science applications using,

* Artificial intelligence,
* Deep learning and
* Machine learning algorithms.

### The Programming Language of the Agile Era

Python Brochure Vol.I 2nd Edition, named: Case Studies & Success Stories, which was prepared by the Python Software Foundation (PSF) to introduce Python.

Python is a very lean programming language. Python programs are a great deal shorter than code written in other modern programming languages.

A comprehensive standard library and thousands of additional libraries in the Python Package Index provide developers with high-quality solutions that they can easily integrate into their applications to meet virtually any requirement.

**The Master Key for System Integration :**

Python offers unique benefits for system integration. On the one hand, there are huge numbers of Python libraries, with which virtually any third-party system can be integrated.

On the other hand, the libraries of many other programming languages can also be used in Python.

The Python programming language is easy to learn.

If you have any problem, answer on several sites like [stackoverflow.com](https://stackoverflow.com/)

### 

### Historical Development of Python

This programming language was developed in the early 90s by a Dutch programmer called **Guido van Rossum.**

The version of Python 3.X was released in 2008. It makes Python more *readable* and *consistent* than previous versions.

**What is PEP 8?**

A: PEP stands for Python Enhancement Proposal. PEP 8 is a coding convention, a set of recommendation, about how to write your Python code more readable. In other words, PEP 8 is a document that gives coding conventions for the Python code comprising the standard library in the main Python distribution.

We will show you some important PEP 8 traditional rules that you can follow.

· Limit all lines to a maximum of 79 characters. For flowing long blocks of text with fewer structural restrictions (docstrings or comments), the line length should be limited to 72 characters.

You can check if your code complies with traditional PEP 8 rules using this [module](http://pep8online.com/).

There are several Integrated Development Environments (IDE) that we can write and run Python codes on them. /JUPİTER,ANACONDA, GOOGLE COLAB VS.

Surrounding the expression with triple quotes: """...""" or '''...''' ensures that the code returns no error, especially in long texts.

**Comments**

Comments are used to explain code when the basic code itself isn't clear. Python ignores comments.

There are **three types of commenting methods**. These are :

* **Single-line comments** begin with the hash character # and are terminated by the end of the line.
* **Inline comments** also begin with hash character # and start from the end of a code line.
* **Multi-line comments** basically consist of multiple comment lines.

**Docstrings** are - unlike regular comments - stored as an attribute of the function or the module they document, meaning that you can access them programmatically.

Docstring runs as an explanatory text of codes and it should be written between triple quotes. Like: """docstring""".

Normally, when we want to call docstring of a function or module to read, we will use \_\_doc\_\_ (the keyword doc enclosed by double underscores) syntax.

### 

## Basic Data Types

Simple data types commonly used in Python:

· String,

· Signed Integer,

· Floating Point,

· Complex,

· Boolean.

Boolean types are called bool and their values are the two constant objects True and False.

* We can print the types of data using type() function.

***Arithmetic Operators***

1. parentheses : ()

2. power : \*\*

3. unary minus : -3

4. multiplication and division : \*, /

5. addition and subtraction : +, -

6. -= decrements the variable in place,

7. += increment the variable in place,

8. \*= multiply the variable in place,

9. /= divide the variable in place,

10. //= ﬂoor divide the variable in place,

11. %= returns the modulus of the variable in place,

12. \*\*= raise to power in place.

### 

### *Escape Sequences*

· \n : means new line,

· \t : means tab mark,

· \b : means backspace.

### Boolean Logic Expressions

Python has three built-in boolean operators: and, or and not.

The and operator evaluates all expressions and returns the last expression if all expressions are evaluated True. Otherwise, it returns the ﬁrst value that evaluated False.

The or operator evaluates the expressions left to right and returns the ﬁrst value that evaluated True or the last value (if none is True).

***Order of Priority***

1. not

2. and

3. or

For example : x = True and not True, the value of x returns False.

The values of non-boolean types (integers, strings, etc.) are considered **truthy** or **falsy** when used with logical operations, depending on whether they are seen as True or False.

The following values are considered False, in that they evaluate to False when applied to a boolean operator:

* None.
* Zero of any numeric type: 0, 0.0, 0j
* Empty sequences and collections: '', [], {}.
* Other than above values, any remaining value is evaluated as True.

The formula syntax of string indexing is : string[start:stop:step].

### String Formatting with Arithmetic Syntax

· Arithmetic syntax (+, \*, and =),

· % operator formatting,

· string.format() method,

· f-string formatting.

What are the string.startswith() and string.endswith() method used for?

A: To search patterns in a string there are two useful methods called startswith() and endswith() that search for the particular pattern in the immediate beginning or end of a string and return True if the expression is found.

* str.replace(old, new[, count]) replaces all occurrences of old with the new
* str.swapcase() converts upper case to lower case and vice versa.
* str.capitalize() changes the first character of the string to the upper case and the rest to the lower case.
* str.upper() converts all characters of the string to the upper case.
* str.lower() converts all characters of the string to the lower case.
* str.title() converts the first character of each word to upper case.

str.strip() : removes all spaces (or specified characters) from both sides.

str.rstrip() : removes spaces (or specified characters) from the right side.

str.lstrip() : removes spaces (or specified characters) from the left side.

Q: In Python what is slicing?

A: A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

### 

### 

### Collection Types

We can add an element into a list using .append() or .insert() methods.

We can remove the elements in lists using list.remove() method or sort the elements using list.sort() method.

Likewise, the length of the list elements can be calculated with the len() function

What does list[::-1] do?

A: list[::-1] is used to reverse the order of a sequence of the elements in the list.

The most important difference from the list is that the tuple is immutable.

* Tuples are faster than list.
* Remember to always use a comma when defining a singleton tuple.

Because of its immutability, the data stored in a tuple can not be altered by mistake.

A dictionary in Python is a collection of key-value pairs called items of a dictionary.

* Note that keys and values can be of different types.

You can access all items using the .items() method, all keys using the .keys() method, and all values using the .values() method:

You can also remove an item using the del function:

Dictionaries strongly resemble JSON syntax. The native json module in the Python standard library can be used to convert between JSON and dictionaries.

Sets

A set is a collection of elements with no repeats and without insertion order but sorted order.

Explanation: The iterable argument given for the set must be used in a correct way.

· .add() : Adds a new item to the set.

· .remove() : Allows us to delete an item.

· .intersection() : Returns the intersection of two sets.

· .union() : Returns the unification of two sets.

· .difference() : Gets the difference of two sets

## 

## 

## 

## 

## Conditional Statements

The if statements check the condition. The condition is always a Boolean expression, that is, its value equals either True or False

* Note that the condition ends with a colon and a new line starts with an indentation.

| Operator | How it works ? |
| --- | --- |
| == | Returns True if two values are equal or False if different |
| != | Returns True if two values are not equal or False if equal |
| > | Returns True if the value on the left is greater than the value on the right otherwise returns False |
| < | Returns True if the value on the left is less than the value on the right otherwise returns False |
| >= | Returns True if the value on the left is greater than or equal to the value on the right otherwise returns False |
| <= | Returns True if the value on the left is less than or equal to the value on the right otherwise returns False |

* Do not use assignment operator = in comparison statements. In comparisons, you have to use == operator for equality.

In Python you can deﬁne a series of conditionals using :

· if for the ﬁrst one,

· elif for the rest, up until the ﬁnal (optional),

· else for anything not caught by the other conditionals.

## 

## *Loops*

**Q: What are the two major loop statements?**

A: for and while loops.

To create a for loop, you need a variable and an iterable object.

**Q: How does for loop and while loop differ in Python and when do you choose to use them?**

A: For loop is generally used to iterate through the elements of various collection types such as list, tuple, set and dictionary.

**Q: What are Python iterators?**

A: Iterators in Python are array-like objects which allow moving on the next element. We use them in traversing a loop, for example, in a for loop.

The range() function creates an iterable sequence of numbers. And it can be simply converted into an iterable object: list, set, and tuple.

Starred expression 👉🏻\* can separate other iterable objects. For example, you can separate a string: print(\*('separate'))

zip() function make an iterator that aggregates elements from each of the iterables.

**Q: What is function in Python?**

A: A function is a block of code which is executed only when it is called. To define a Python function, the def keyword is used.

Reading a function is very easy in Python. For example; multiply(2, 5) or multiply(no1, no2). In this example, multiply is the name of the function, 2 and 5 are its arguments that we passed into the parameters which are the variables (no1 and no2). Calling a function means using it.

### Built-in Functions

### 

There are a range of functions and types built into the Python interpreter, so they are always usable.ok at the official Python documentation for the built-in functions mentioned below.

* Some of them return bool type according to the conditional algorithm in it. For example; all(*iterable*), any(*iterable*), and callable(*object*).
* Some of them help you convert data types into each other. For example; bool(), float(), int(), and str().
* Some others allow you to create and process the collection types. Such as : dict(), list(), tuple(), set(), len(), frozenset(), zip(), filter(*function*, *iterable*), and enumerate(*iterable*).
* Some others tackle numbers. Such as : max(), min(), sum(), and round().
* The others are built for special purposes. They do some complicated implementations. For example : map(*function*, *iterable*, *...*), eval(*expression*[, *globals*[, *locals*]]), sorted(*iterable*), open(), dir([*object*]), hash(), and help([*object*]).

### Q: Explain Python functions.

### A: A function is a section of the program or a block of code that is written once and can be executed whenever required in the program. A function is a block of self-contained statements which has a valid name, parameters list, and body. Functions make programming more functional and modular to perform modular tasks. Python provides several built-in functions to complete tasks and also allows a user to create new functions as well. There are two types of functions: Built-In Functions: copy(), len(), count() are the some built-in functions. User-defined Functions: Functions which are defined by a user known as user-defined functions

**Q: How do we write a function in Python?**

A: We can create a Python function in the following manner.

Step-1: to begin the function, start writing with the keyword def and then mention the function name.

Step-2: We can now pass the arguments and enclose them using the parentheses. A colon, in the end, marks the end of the function header.

Step-3: After pressing an enter, we can add the desired Python statements for execution.

**Q: What is the return keyword used for in Python?**

A: The purpose of a function is to receive the inputs and return some output. The return is a Python statement which we can use in a function for sending a value back to its caller.

When calling a function with positional arguments, they must be passed in order from **left to right**.

#### 

**Q: What does this mean: \*args, \*\*kwargs? And why would we use it?**

A: We use \*args when we aren’t sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. \*\*kwargs is used when we don’t know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The identifiers args and kwargs are a convention, you could also use \*bob and \*\*billy but that would not be wise.

## Scope of the Variables (Optional)

#### What is Scope?

A scope is a concept describes where or in which space the variables are defined in the program stream. This concept has a significant place in programming. In other words, a scope is a *textual region* of a Python program where a *namespace* is directly accessible. “Directly accessible” here means that an unqualified reference to a name attempts to find the name in the namespace.

**Q: What is the namespace in Python?**

A: The namespace is a fundamental idea to structure and organize the code that is more useful in large projects. A namespace is defined as a simple system to control the names in a program. It ensures that names are unique and won't lead to any conflict. Also, Python implements namespaces in the form of dictionaries and maintains name-to-object mapping where names act as keys and the objects as values.

**What are local variables and global variables in Python?**

A: Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program. Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space. When you try to access the local variable outside the function, it will give an error.

LEGB Ranking Rule

When you call an object (*method or variable*), the *interpreter* looks for its name in the following order:

1. Locals. The space which is searched first, contains the local names defined in a function body.

2. Enclosing. The scopes of any enclosing functions, which are searched starting with the nearest enclosing scope (from inner to outer), contains non-local, but also non-global names.

3. Globals. It contains the current module’s global names. The variables defined at the top-level of its module.

4. Built-in. The outermost scope (searched last) is the namespace containing built-in names.

#### 

You can not change the value assigned to a globally defined variable within a function. To do this we use the keyword global.

#### 

You can use the keyword nonlocal to extend the scope of the *local variable* to an upper scope.

### 

### Defining a Lambda Function

Another way to define functions in Python is lambda functions. Lambda functions are also called anonymous functions since they have no name.

Lambda functions can have any number of arguments but only one statement.

lambda x: 'odd' if x % 2 != 0 else 'even'

Lambda's most important advantages and uses are:

1. You can use it with its own syntax using *parentheses*,

2. You can also *assign* it to a *variable*,

3. You can use it in several *built-in* functions,

4. It can be useful inside user-defined functions (def).

**Q: What Are The Principal Differences Between The Lambda And Def?**

A:

- Def can hold multiple expressions while lambda is a uni-expression function.

- Def generates a function and designates a name to call it later. Lambda forms a function object and returns it.

- Def can have a return statement. Lambda can’t have return statements.

- Lambda supports to get used inside a list and dictionary.

**Q: What is map function in Python?**

A: map function executes the function given as the first argument on all the elements of the iterable given as the second argument. If the function given takes in more than 1 arguments, then many iterables are given.

iterable = [1, 2, 3, 4, 5]

map(lambda x:x\*\*2, iterable)

filter() filters the given sequence (iterable objects) with the help of a function (lambda) that tests each element in the sequence to be true or not.

first\_ten = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

even = filter(lambda x:x%2==0, first\_ten)

## Loading Modules

* If you open and use this file (with a .py extension) *directly*, that is script, and
* If you *load* (import)this file (with a .py extension) and call any function from it, that's a module this time.

**· Q: How to import modules in Python?**

A: Modules can be imported using the import keyword. You can import modules in three ways. Example:

· import array # importing using the original module name

· import array as arr # importing using an alias name

· from array import \* # imports everything present in the array module

### Built-in Modules

Python comes with a huge library of standard modules many of which are built into the interpreter. These modules make our code more effective by providing useful functions and data structures.

* The built-in function dir() is used to find out which names a module defines. It returns a sorted list of strings.

**Q: What are python modules? Name some commonly used built-in modules in Python.**

A: Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code. Some of the commonly used built-in modules are: os, sys, math, random, datetime.

**Q: How can you generate random numbers in Python?**

A: Random module is the standard module that is used to generate a random number. The method is defined as:

import random

print(random.random())

**Q: What is module and package in Python?**

A: In Python, module is the way to structure the program. Each Python program file is a module, which imports other modules like objects and attributes. The folder of Python program is a package of modules. A package can have modules or subfolders.

Pip allows you to install/uninstall, and manage additional packages that are not part of the [Python standard modules](https://docs.python.org/3/py-modindex.html).

### 

## Errors

The name of module - *Traceback* - appears when your code causes an error and it reports detailed information on that specific error, demonstrating the particular files in which the error occurred. In these error messages, the most important thing that a programmer should be interested in is the last lines in the most cases.

**Common Errors**

The most common syntax errors made by programmers can be listed as follows.

Quotes

Wrong Parentheses

Wrong Spelling & Typo

Indents

| **Difference** | |
| --- | --- |
| **Syntax Error** | **Exception Error** |
| These types of errors are detected during compiling the program into byte-code. | These types of errors are detected during the program execution (interpretation) process. |

**Q: Give some examples of standard errors that occour in Python.**

A:

* TypeError- It occurs when the expected type does not match with the given type of a variable.
* ValueError- It occurs when an expected value is not given, suppose you are expecting 6 elements in a list and you gave 2.
* NameError- It occurs when you are trying to access an undefined variable or a function.
* IOError- It occurs when you are trying to access a file that does not exist.
* IndexError- It occurs when you are trying to access an invalid index of a sequence.
* KeyError- It occurs when you use an invalid key to access a value in the dictionary.

## Exception Handling

We can ensure that our program continues without hesitation by using the try-except statement. The basic structure of try-except statement looks like :

try:

code block to be normally executed

except:

code block to be exceptionally executed

* First, Python executes all statements between try and except.
* If there is no exception, then the try block is successfully executed and finished.
* If an exception raises, the rest of the try block is ignored. After that, Python interpreter checks if the type of exception matches the exception specified after the except keyword, it executes the except block and resumes executing the program after the try-except block.
* If an exception doesn’t match the exception named in the except syntax, it is known an unhandled exception and execution of your program stops with a traceback error message.
* The else-block is only executed if there raised no exceptions.
* We can use a finally syntax at the end of the block. A finally syntax is always executed before leaving the try-except block, whether an exception was raised or not.

| **'mode' Parameter Attributes** | |
| --- | --- |
| Char | What it's used for? |
| 'r' | Open for reading (default). If the file doesn't exist, FileNotFoundError will raise. |
| 'a' | Open for writing. It will append to the end of the file if it already exists. If there is no file, it will create it. |
| 'w' | Open for writing. It will be overwritten if the file already exists. If there is no file, it will create it. |
| 'x' | Open for exclusive creation, it will fail if the file already exists. |
| 'b' | Open in binary mode. |
| 't' | Open as a text file (default). |
| '+' | Open for updating (reading and writing). |

· By default, open() function executes opening the file for *reading* ('r') as a *text* ('t'). In other words, it defaults to 'r' or 'rt' which means open for reading in text mode.

You can use the following methods for reading file :

· .read(size)

· .readline(size)

· .readlines()

· using loops

Reading Methods of the CSV Files

There are several ways to read CSV files in Python.

1.You can use the ordinary way of reading using open() function and .read() method just you learned in the previous lesson.

2.You can use [csv Module](https://docs.python.org/3/library/csv.html#module-csv).

3.You can use the Pandas library for which created data analysis purposes.

**6. SQL(Structured Query Language):**

Database is a collection of data stored in a computer system.

Structured data here means table

**In database world;**

A column is also called a field or attribute,

A row is also called a record or a tuple,

A table is also called a relation.

two main database storage types: Relational Database - **SQL**

Non-Relational Database - **NoSQL**

A software system used to maintain relational databases is called a Relational Database Management System **(RDBMS)**.  Here are some examples of RDBMS:

Amazon Aurora, Amazon RDS, Microsoft SQL Server, MySQL, SQLite

SQL is accepted as the standard (RDBMS) language.

SQL is a declarative language, not a procedural language

With SQL, you can access or manipulate data stored in the database.

**Types of access:**

Retrieval of data, Insertion of new data,

Updating the data, Deletion of data

**A query**is a statement asking for the retrieval of information from the database.

**SELECT, FROM** words, are keywords and they are special commands for SQL.

Writing SQL commands upper-case is the most common and preferred style.

SELECT column1, column2 FROM table1;

SELECT \* FROM table1; (all columns)

**Keyword:**  elements which are predefined. example SELECT and FROM

**Clause:** It's a part of a SQL statement. example, SELECT first name,

**Statement:** The complete query is a statement. Consist of two or more clauses.

For no **duplicated rows** in a column: **SELECT DISTINCT**

The WHERE, FİLTER clauses is used to filter records.

SELECT \* FROM student\_table WHERE grade > 70 LIMIT 2;

The ORDER BY keyword sorts the result-set in descending or ascending order.

SELECT \* FROM employees WHERE salary > 80 ORDER BY first\_name DESC;

FROM employees ORDER BY salary DESC;

SELECT \* FROM employees ORDER BY gender ASC, first\_name DESC;

SELECT \* FROM employees ORDER BY hire\_date ASC LIMIT 1;

SELECT first\_name, salary FROM employees ORDER BY salary DESC;

SELECT \* FROM employees WHERE job\_title = 'Data Scientist' AND gender = 'Male';

SELECT \*

FROM employees

WHERE salary BETWEEN 75000 AND 95000;

SELECT \*

FROM employees

WHERE job\_title IN ('Data Scientist', 'Business Analyst');

SQL provides two special characters for constructing patterns. These are also called wildcards.

* Percent (%): The % character matches any sequence of zero or more characters.
* Underscore ( \_ ): The \_ character matches any single character.

However, if you want to make LIKE operator case-sensitive, we need to use PRAGMA statement as follows:

SELECT \*

FROM student\_info

WHERE county LIKE 'Wo%';

PRAGMA case\_sensitive\_like = true;

SELECT \*

FROM student\_info

WHERE field LIKE '%developer';

SELECT first\_name

FROM employees

WHERE first\_name LIKE 'El\_is';

As a side note, statements are also called "commands" in some sources.

Actually these statements are grouped into four main categories:

* DDL - Data Definition Language
* DML - Data Manipulation Language
* DCL - Data Control Language
* TCL - Transaction Control Language

Data Definition Language (DDL) specifies the database schema. The statements used in DDL are CREATE, ALTER, DROP. DDL statements are typically used to set up and configure a new database before we insert data.

[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Data Manipulation Language (DML) enables users to access or manipulate data. INSERT, UPDATE, DELETE, SELECT are the statements used in DML.

[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Data Control Language (DCL) is used to grant or revoke access control. Its statements are REVOKE and GRANT.

[✱](https://www.toptal.com/designers/htmlarrows/symbols/heavy-asterisk/) Transaction Control Language (TCL) controls the transactions of DML and DDL commands. Its statements are BEGIN and COMMIT.

**SQLite, there are five primitive data types. They are called storage classes. Here are the details of them:**

* NULL: The value is a NULL value. NULL values mean missing information or unknown.
* INTEGER: The value is a whole number (positive or negative), stored in 1, 2, 3, 4, 6 or 8 bytes depending on the magnitude of the value.
* REAL: The value is a floating-point value, stored as an 8-byte float. They are real numbers with decimal values.
* TEXT: The value is a text string, stored using the database encoding (UTF-8, UTF-16BE or UTF-16LE).
* BLOB: BLOB stands for a binary large object that can store any kind of data. The value is a blob of data, stored exactly as it was input.

**SQLite determines the data type of a value based on its data type in accordance with the following rules:**

* If a value is enclosed by single or double quotes, it is assigned the TEXT storage class.
* If a value has no enclosing quotes and decimal point or exponent, it is assigned INTEGER storage class.
* If a value has no enclosing quotes, but decimal point or exponent, it is assigned REAL storage class.
* If a value is NULL without quotes, it is assigned NULL storage class.
* If a value has X'ABCD' or x'abcd', it is assigned BLOB storage class.

**What are Constraints in SQL?**

A: NOT NULL - Restricts NULL value from being inserted into a column. DEFAULT - Automatically assigns a default value if no value has been specified for the field. UNIQUE - Ensures unique values to be inserted into the field. PRIMARY KEY - Uniquely identifies each record in a table. FOREIGN KEY - Ensures referential integrity for a record in another table.

The primary key is used to uniquely identify each record(ROW)

The primary key column cannot contain NULL values.

A FOREIGN KEY comprises of single or collection of fields in a table that essentially refer to the PRIMARY KEY in another table. Foreign key constraint ensures referential integrity in the relation between two tables. The table with the foreign key constraint is labelled as the child table, and the table containing the candidate key is labelled as the referenced or parent table.

Unlike other database systems, SQLite supports a limited functionality of the ALTER TABLE statement.

## Some of The Most Important SQL Commands

* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index

## Joins

JOINS are used to combine rows from two or more tables, based on a related column between those tables. The following are the types of joins:

* [INNER JOIN:](https://www.edureka.co/blog/sql-commands#INNER%20JOIN:) This join returns those records which have matching values in both the tables.
* [FULL JOIN:](https://www.edureka.co/blog/sql-commands#FULL%20JOIN:) This join returns all those records which either have a match in the left or the right table.
* [LEFT JOIN:](https://www.edureka.co/blog/sql-commands#LEFT%20JOIN:) This join returns records from the left table, and also those records which satisfy the condition from the right table.
* [RIGHT JOIN:](https://www.edureka.co/blog/sql-commands#RIGHT%20JOIN:) This join returns records from the right table, and also those records which satisfy the condition from the left table.

**7. SDLC:**

The  **Software Development Life Cycle** (SDLC) is a process that is used to develop software.

SDLC is a step by step procedure need to be followed by the organization to design and develop a high-quality product.

The main purpose of SDLC is to produce software with the highest quality and lowest cost in the shortest time.

**The SDLC process**

* **Requirement Phase (most  critical)**

. To achieve this we need to continuous communication with customers to gather all requirements.This is done through an **SRS (Software Requirement Specification)** document.

“””SRS is a detailed description of a software system to be developed with requirements. The SRS is developed based on the agreement between customers and contractors. It may include the use cases of how a user is going to interact with a software system.”””

* **Design Phase**

high priority. **logical designing**  is converted into **physical designing**.

* **Build/Development Phase**

**coding phase. L**ongest phase.Work/task is divided into small units or modules,

* **Testing Phase**

This phase is where you focus on investigation and discovery.

The testing phase begins when the software is complete.

* **Deployment/Deliver Phase**
* **Maintenance**

It is an important part of a tester's role to be familiar with the common SDLC models

**There are several characteristics of good testing:**

* For every development activity, there is a corresponding test activity
* Each test level has test objectives specific to that level
* Test analysis and design for a given test level begin during the corresponding development activity
* Testers participate in discussions to define and refine requirements and design, and they are involved in reviewing work products as soon as drafts are available part of testing is focused on **verification** testing, and another part is focused on **validation** testing.

**Verification** focuses on the question **Is the deliverable built according to the specification?**.

**Validation** focuses on the question **Is the deliverable fit for purpose, and does it provide a solution to the problem?**.

**SDLC Models**

1. Waterfall Model

2. Iterative Model

3. Spiral Model

4. V-Model

5. RAD Model

6. Agile Model

7. DevOps Model

**The Waterfall Model**

**The sequential phases in the Waterfall model are;**

* Requirement Gathering and analysis
* System Design
* Implementation
* Integration and Testing
* Deployment of system
* Maintenance

**Many situations where the most effective use of the Waterfall model are;**

* The requirements are documented very well.
* The definition of a product is stable.
* Technology is comprehensible.
* Requirements are not ambiguous.
* Ample resources are available with the required expertise to support the product.
* The project is short.

**Some of the major advantages of the Waterfall Model**

* It allows control and departmentalization.
* A schedule may be set with deadlines for each stage of development, and a product may proceed one by one through phases of the development process model.
* Every development phase proceeds in strict order.
* Easy and simple to use and understand.
* Easy to manage because of model rigidity. There are specific deliverables and a review process for each phase.
* Phases are processed one at a time and completed.
* Works well for smaller projects where requirements are very well understood.
* Phases clearly defined.
* The coding and testing steps are very short, as the requirements and design are clearly defined during the analysis and design phases.
* The number of errors during the test phase is very small.
* Tasks are easy to arrange.
* There is good documentation of the process and results.

### 

**Some of the major disadvantages of the Waterfall Model**

* It does not allow a great deal of reflection or revision.
* Once an application is in the phase of testing, it is very hard to go back and alter something that was not well-documented.
* No working software is produced until late during the life cycle.
* There is a high amount of risk and uncertainty.
* The time-loss due to the upper phase errors is quite high.
* It is not a good model for complex and object-oriented projects.
* It is a poor model for long and ongoing projects.
* It is not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty are high with this process model.
* The progress in phases can be difficult to measure.
* The product has to wait until the end of all the phases.
* It can not adapt to changing requirements.
* Scope adjustment over the life cycle can end a project.

## 

## DevOps Model

The term DevOps is the short form of ***Development and Operations***.

It focuses on collaboration between developers and other roles.

DevOps is a practice that allows a single team to manage the entire application development life cycle, that is, development, testing, deployment, operations.

DevOps is an evolution of the ***Agile Model*** of software development.

DevOps addressed the gap between Developers and Operations.

### 

### DevOps Phases

**Plan -** Business owners and development team discuss project goals and create a plan.

**Code -** Programmers then design and code the application

**Build -** Build tools like Maven and Gradle, take code from different repositories and combine them to build the complete application.

**Test -** Application is tested using automation

**Integrate -** When testing is complete, new features are integrated automatically to the already existing codebase.

**Deploy -** Application is packaged after release and deployed from the development server to the production server.

**Operate -** Once the software is deployed, the operations team performs activities such as configuring servers and provisioning them with the required resources.

**Monitor -** Monitoring allows IT organizations to identify specific issues of specific releases and understand the impact on end-users.

**Some of the DevOps principles are;**

* Create production-like systems for development and testing environment
* Deployments need to be iterative and frequent. Ensure a reliable and repeatable process
* Continuously monitor and validate operational quality characteristics
* Amplify feedback loop

**Some of the DevOps advantages are;**

* Time taken to create and deliver software is reduced
* The complexity of maintaining an application is reduced
* Improved collaboration between developers and operations team
* Continuous integration and delivery ensure faster time to market

**Some of the DevOps disadvantages are;**

* Technology investments in the automation tools required for DevOps are costly and will take a great deal of time to identify and implement.
* It needs a specialist who can cover each stage of the software delivery pipeline rather than investing in a smaller number of full-stack developers.
* It is easy to add new features, but more features may not always better, even if they can be implemented efficiently.

**8. AGİLE:**

What is agile: A new method for project management,

Uses iterative planning and getting feedback ensures

With customer's requirements

Helps companies build the right product

**The Agile Manifesto:**

In 2001, 17 independent Software Leaders met in America for brainstorming.

The Manifesto was only 68 words, four values:

1. Individuals and interactions over processes and tools

2. Working software over comprehensive documentation

3. Customer collaboration over contract negotiation

4. Responding to change over following a plan

**Stemming from these values there are 12 principles in agile:**

satisfy the customer --- welcome changing requirements

face to face conversation ---- work together

motivated individuals ---- shorter timescale

measure of progress ---- sustainable development

maximizing the amount of work --- self-organizing teams  
 continuous attention,good design--- regular intervals

**two most common agile methodologies: Scrum and Kanban**

Scrum Framework and Sprint Concept:

Scrum is the most common methodology to implement the Agile. It is an iterative development model. In scrum, larger projects are divided into smaller parts that can be managed with **sprints** . Sprints are the periods from **1 to 4 weeks**.

**Scrum Roles:**

Product Owner, Scrum Master, and Development Team(3-9 people).

**Epic”**  : business needs

**User Story**  : customer needs, type of user, what and why they want it.

**Task Concepts** : solution,pieces of work, realize a user story.

Tasks are displayed on a **scrum board.**

The board is usually divided into 3 categories: *to do*, *in progress*, and *done*

**The Scrum framework defines three major artifacts:**

**product backlog prioritizing the requirements**

list everything that needs to be done to complete the project.

**sprint backlog**  issues, completed during the sprint constitute

**product increment**  sum of all the product backlog items.

The product increment aims to invest in small amounts in the new features of the main product.

**Ceremonies in Scrum**

**1, Sprint Planning Meeting:** At the start of each sprint,

- The scrum team decides the scope and the goals of the sprint.

**2. Grooming Meeting:** no specific date,

- Review the backlog

**3. Daily Stand Up Meeting or Daily Scrum**: 15 min.

- What they completed yesterday,

**-** What they will focus on today,

- What are the things impeding them?

**4. Sprint Review Meeting:** After each sprint

- What they **accomplished** during the sprint and receives feedback

**5. Sprint Retrospective Meeting:** After each sprint

- What went well in the previous sprint?

- What didn’t work well?

- What are the improvement areas to increase team performance?

**Scrum projects also include some other artifacts:**

**Burn-down chart** demonstrates the amount of work remained to complete a project. So, the burn-down chart is used to trace the progress of a project.

**The timeboxing** aims to define and limit the amount of time dedicated to a certain event.

**Story Points** are decided upon and used by individual scrum teams. A Story Point is a relative unit of measure to provide relative predictions of effort for completing tasks or user stories.

**Velocity** is an indication of the average amount of items from the product backlog turned into a product increment In scrum, velocity is the key metric.

story points 80, first sprint is 20. team needs 4 more sprints to complete the whole project

**KANBAN**

The word "kanban" is in Japanese and can be translated as "the card you can see" or "signboard". it is a visual framework used to implement agile and shows **what to produce, when to produce , how much to produce**.

Kanban focuses on **visualizing the workflow process.**

Kanban's story dates back to the **1940s** (Toyota)

Optimizes the flow between the organization and the consumer

The kanban method controls the entire chain from production to the end consumer

**David J. Anderson** introduced the idea of using the kanban concept for software development, and in 2010 he wrote a book named "Kanban: Successfully Evolutionary Change for your Technology Business ”

A Kanban is like a flash card carrying all the information about the **current status** of your work and the **work required** to be done on the product at each stage of the software development process.

Any given time, the organization can see **progress**, **capacity**, **productivity**, and **efficiency**.

**‘Scrum' and ‘Kanban':**

Scrum is a better choice when you need a more **prominent process.**

if you want improvement in running the process without much changes in the whole scenario, you should use Kanban.

**A kanban board** is a tool designed to visualize the work and increase efficiency using cards and columns.

**There are three main components of a kanban board:**

**visual signals**: what they working on and what is the current situation

**columns:** stage of a project, workflow

**work-in-progress limits** (WIP). maximum number of cards

**Four (4) principles of kanban:**

1. Start with what you are doing now:

2. Agree to pursue incremental, evolutionary change:

3. Initially, respect current roles, responsibilities, and job-titles:

4. Encourage acts of leadership at all levels:

## Six (6) Practices of Kanban:

**1. Visualizing the Workflow**

Visualization is the first step to start with kanban. You need to visualize the current workflow on either a physical whiteboard or a digital Kanban board.

**2. Limit Work in Progress (WIP)**

The maximum number of cards that can be in a column at any given time is called WIP limits. 1 to 1.5 times the number of people taking part in each stage or each column

**3. Managing Flow**

Relates to **improving** the flow of the process. Lead time

### 4. Making Process Policies Explicit:

Allows the team to follow the process easily and make proposals for the necessary improvements

### 5. Implementing Feedback Loops:

Get feedback from people not involved in your system.

### 6. Improving Collaboratively, Evolving Experimentally (Using the Scientific Methods)

**İnterwiew Questions:**

## What is the use of burn-down charts?

A burn-down chart demonstrates the amount of work remained to complete a project. So, the burn-down chart is used to trace the progress of a project.

## What is the role of the Scrum Master?

Scrum Master coaches the team, protects the team from organizational distraction, clears any obstacles encountered and helps team members focus on what they do. Scrum master ensures that scrum is understood well by the team members and it is working properly. Scrum master constantly improves the team's environment. While product owner has a directing role, scrum master has an enabling role in a scrum team.

## What is the scrum of scrums?

Suppose there are 6 teams working on a project and each team has 6 members. Each team leads its own special scrum meeting. However, in order to coordinate and communicate with different teams, it is required to organize a separate scrum meeting. The meeting organized to hold coordination between scrum teams is called the scrum of scrums. There is one team leader from every group, known as ambassador, who is responsible to represent his team in the scrum of scrums.

## Explain the term ‘increment' in Scrum.

The Product Increment is the sum of all the product backlog items finished during the sprint. In other words, by the end of each sprint, the development team creates a new software that gets built into the main product and this new software is called product increment. The product increment aims to invest in small amounts in the new features of the main product. This helps to shorten the time before receiving feedback. As the name implies, product increment continues to increase within the subsequent sprints. That means each product increment includes all the previous sprint increment values as it is cumulative.

## Describe the places where ‘Scrum' and ‘Kanban' are used?

Scrum is a better choice when you need a more prominent process. However, if you want improvement in running the process without much changes in the whole scenario, you should use Kanban.

## What is a user story?

It describes a software feature from the customer’s perspective and includes the type of user, what they want, and why they want it. Therefore, it answers the ‘who’, ‘what’ and ‘why’ in a simple language. The product owner has the responsibility of user stories. Leaving out the technical aspect, it should describe the behavior from a user’s perspective. Examples of User Storie: As a registered user, I want to add items to the cart so that I can purchase multiple items at once.

## Why aren't user stories simply estimated in man-hours?

Estimation of user stories on the basis of man-hours is possible but not preferred. Because in that case, you won't be able to concentrate on the quality product to be delivered to the customer. In addition to that, you will concentrate on the cost and budget of the management while using man-hours. Rather than man-hours, story points are used, as it provides a complete idea about both the complexity of work and required efforts.

## Explain what is Kanban.

A Kanban is like a flash card carrying all the information about the current status of your work and the work required to be done on the product at each stage of the software development process.

## How does Kanban improve visibility?

Kanban uses digital or physical boards to demonstrate the team’s workflow. The tasks demonstrated by cards move from left to right representing the progress. So at any given time, the organization can see progress, capacity, productivity, and efficiency.

## Ideally, how WIP limit is calculated with respect to team size?

You can start with a WIP limit of 1 to 1.5 times the number of people taking part in each stage or each column. For example, if team size is 4 in a particular stage, max 6 items can be in progress at any given time.

## What is lead time in Kanban?

Lead time is the period between creating a task in your workflow and its final departure from the kanban board. Therefore, kanban helps you monitor workflow, identify blockages, and make adjustments to improve the flow for reducing the lead time.

## Are there any drawbacks of the Agile model? If yes, please explain.

Yes, there are some drawbacks of the Agile method, some of them are as follows:

- It is not easy to make an estimation of the effort required to complete a task. It becomes more complex in the case of large projects as it becomes difficult to make a prediction about the total effort required.

- In case the desired requirements of the client are not understood properly, the final project will not meet the customer requirements. Thus, this will lead to customer dissatisfaction.

- Only the leader who has considerable experience in the Agile model is capable to take important decisions. The team members with less or no experience are not involved in the decision-making process, thus they don’t have a chance to advance their knowledge.

**9. JİRA**

Jira is a tool developed to help teams for **project management, bug tracking, and issue tracking**. (by Australian Company [Atlassian](https://www.atlassian.com/))

The name "Jira" is derived from the truncation of *Gojira*, the Japanese word for *Godzilla*.

When Jira was released in 2002, it was seen as an issue tracking software used by software developers. Later, this application was accepted as a project management tool

Some of the worldwide companies that use Jira at certain times for bug-tracking and project management are Samsung, Coca Cola, Domino’s, Nestle, Hitachi, Pfizer, Visa, Costco, Delta Airlines, Office Depot and so on.

**The four products of the Jira family**

1. **Jira Software**

This is a software development tool used by agile teams.

It is very suitable for software development teams who want to use agile methodologies such as Scrum and Kanban. Jira Software is designed to plan, monitor, and release great software for each member of the software team.

### Jira Align

Allows the company leadership to monitor current developments and link them to business outcomes at an enterprise scale.

1. **Jira Core**

Developed for general-purpose project management. optimized for business teams, with all the field customizations and workflow capabilities. Jira Core keeps the teams organized by managing projects, monitoring details, and measuring performance.

1. **Jira Service Desk**

It is developed for general-purpose project management. This is similar to the classic Jira, optimized for business teams, with all the field customizations and workflow capabilities. Jira Core keeps the teams organized by managing projects, monitoring details, and measuring performance.

**Jira Software** has several special features that can be useful not just for software development teams, but for any business teams.

### Hosting Options

Teams can use Jira Software with two hosting options: **cloud** and **self-managed**.

There are three hosting options available: **Atlassian Cloud**, **Server**, and **Data Center** (Amazon Web Services (AWS) and Microsoft Azure).

With **Jira Software Cloud,** the *Jira Software site* is hosted and installed in the cloud for you.

**Jira Software Server** allows you to host Jira Software on your own hardware and customize it to suit your needs when installing.

### Data Center

With this option, Jira Software can be hosted on your hardware or public cloud services such as Amazon Web Services (AWS) and Microsoft Azure. Jira Software Data Center is often the best deployment option for enterprise teams that need uninterrupted access and high performance.

Since Jira Software Cloud is a web application, you do not need to download and install it on your computer.

Atlassian offers Jira Software Cloud free of charge for up to 10 users with a 2GB storage capability.

There are two types of boards in Jira Software: classic and next-gen.

In Jira, sidebars are color-coded.

**Blue** indicates that you're in a general Jira menu.

**Gray** tells that you're in a project.

### What is Board?

A board allows you viewing issues from one or more projects and gives you simplicity in managing, viewing, and reporting ongoing work. There are three types of boards in the Jira Software:

## Scrum Board

allows the issues of one or more projects to be displayed in columns that are not started, in progress and done.

| **Type of Board** | **For Which Teams?** |
| --- | --- |
| Scrum board | Teams that plan their work in sprints. |
| Kanban board | Teams that focus on managing and constraining their work-in-progress. |
| Next-gen board | Teams who are new to agile. |

### What is Board?

A board allows you viewing issues from one or more projects and gives you simplicity in managing, viewing, and reporting ongoing work. There are three types of boards in the Jira Software:

* **JQL** stands for Jira Query Language and is the most powerful and flexible way to search for your issues in Jira.

A **backlog** is a list of issues that can be created for your project. You can create issues and sprints in the backlog. Then, you can add issues to a sprint so that your team can work on it. These issues are not detailed features in your backlog.

In Jira Software, the versions represent the points of a project over time. With versioning, you can plan the order in which new features and fixes for the product will be released to your customers.

**What is an issue**

In Jira, teams use issues to track individual pieces of work that must be completed. During the sprint planning, users can create issues that they plan to work on and add them to the backlog.

## What is referred as issues in Jira?

Depending on how a team uses Jira, an issue can represent followings:

* Project task
* Feature
* Help-desk ticket
* Leave request form
* Enhancement
* Software bug

Depending on how a team uses Jira, an issue can represent followings: Project task, Feature, Help-desk ticket, Leave request form, Enhancement, Software bug

### 

### Issue Types

* Epic: An epic is a set of jobs that can be divided into manageable user stories.
* Story: A story is a short requirement written from an end-user perspective.
* Task: A task represents the job that needs to be done.
* Subtask: A subtask can be considered as a smaller piece of a story.
* Bug: A bug impairs expected functionality of the product.
* The **hierarchy** for units of work in Jira Software is as follows: ***Project > Epics/Components > Stories > Tasks > Subtasks***
* Each issue type can have different fields, screens and workflows.

Below are a few examples of typical issues.

Epic : Full website redesign

Story : As a registered user, I want to send message to sellers, So that I can chat with sellers.

Task : Create a new logo

Sub-task : Select colors for new logo

Bug : URL directs to nonexistent web page

### 

### İssue Priorities

| **Priority** | **Description** |
| --- | --- |
| **Highest** | This problem will block progress. |
| **High** | Serious problem that could block progress. |
| **Medium** | Has the potential to affect progress. |
| **Low** | Minor problem or easily worked around. |
| **Lowest** | Trivial problem with little or no impact on progress. |

You can define the life cycle of an issue by setting up its **workflow**.

TO DO --- İN PROGRESS --- DONE

* **Status** represents the stage of an issue at a particular time. An issue can be only in one status at a given time.
* **Transition** is the link between two statuses that allows an issue to move from one status to another. A transition is necessary when an issue moves from one status to another.

An ***epic*** is a set of jobs that can be divided into manageable and shippable user stories based on the needs of customers. Epic is the best way to group similar user stories. It may take several sprints to complete an epic.

Epics do not show up as cards in sprints or in the backlog.

* If you find that a subtask is large enough at an issue level, you can convert it into an issue.
* Likewise, if you see that an issue is really just a subtask of another issue, you can convert it into a subtask.

Which format that you need to use for logging time : **4h 30m**

**Cloning an issue** (copying an issue) allows you to quickly create a copy of an issue within the same project.

* A clone issue and original issue are separate entities.
* Operations in the original issue do not affect the information related to the clone issue. Likewise, operations in the clone issue do not affect the information related to the original issue.
* The only connection that relates the clone to the original issue can be a link if one exists.

***The fields that aren't copied:***

| Time tracking | Comments |
| --- | --- |
| Issue history | Links to Confluence pages |

**A sprint** is a container for all of the work a team decides to do for a specific time period. Once you have plenty of issues in your backlog, you can start a sprint.

* You can plan multiple sprints at one time.
* You won't be able to start a new sprint until the active sprint is completed unless your Jira administrator has configured parallel sprints.

### 

### Reporting a Sprint's Progress

Reports will also be useful during ***sprint retrospective meetings*** because they provide detailed information on how sprint is progressing.

**The reports generated by any Scrum project in JIRA are:** **Burndown Chart, Burnup Chart, Sprint Report, Velocity Chart, Cumulative Flow Diagram, Version Report, Epic Report, Control Chart, Epic Burndown, and Release Burndown.**

**A Burndown chart** is a graphical representation of the work left to be done ***(STORY POINTS)*** versus the time remained in the active sprint ***(TIME)***.

**The Sprint report** displays a condensed burndown chart and lists which issues are ***complete*** and which are still ***incomplete***.

**The Velocity Chart** shows the amount of value delivered in each sprint, enabling you to predict the amount of work the team can **get done in future sprints.**

## What is a subtask in Jira Software?

A subtask can be created to divide an issue into smaller pieces and allow them to be assigned to different people. If you find that a subtask is large enough at an issue level, you can convert it into an issue. Likewise, if you see that an issue is really just a subtask of another issue, you can convert it into a subtask.

**A project key is a unique code for your project.** Jira Software will automatically generate a short project key in accordance with your project name. However, if you want to specify this auto-generated key yourself, you can change it

**10.HTML:**

HTML stands for **H**yper**T**ext **M**arkup **L**anguage

HTML was originally developed by Tim Berners-Lee while at CERN

| **Version** | **Year** |
| --- | --- |
| HTML | 1991 |
| XHTML | 2000 |
| HTML5 | 2014 |

HTML is a markup language used to create documents on the web. It describes the structure and layout of a web page.

HTML is composed of elements. A tag and the content between it is called an HTML element. Elements are usually made up of two tags: An opening tag and a closing tag.

* The <!DOCTYPE html> declaration defines this document to be HTML5.
* The <html> element is the root element of an HTML page.
* The <head> element contains meta information about the document.
* The <title> element specifies a title for the document.
* The <body> element contains the visible page content.

<!DOCTYPE html>

<html>

<head>

<h1> to <h6>

<head/>

<body>

<p>This is a <sub>subscripted</sub> text</p>

</body>

</html>

If a HTML element does not have any content, it is called empty element.

HTML tags are element names surrounded by angle brackets.

### HTML Attributes provide additional information about the contents of an element.

A <div> section in a document that is styled with CSS:“Cascading Style Sheets”

HTML has six "levels" of headings, which are graded according to importance.

The HTML <p> element stands for paragraph.

<hr> tag creates a horizontal line

<br> tag stands for a line break.

Comments: <!-- This is a comment→

<b>: bold text. <strong>: strong text,the semantic important texts.

<i>:italic <em>: emphasized text

<small>:smaller text

<sup>:superscriptth<sub>: subscriptedth

<mark>:marked <ins> inserted <del> deleted

<ol> create ordered lists, Each list item is defined by the <li> tag.

<ol>

<li>elenemt1</li>

<li>element2</li>

<li>element3</li>

</ol>

<ul> create unordered lists

In order to create a table in HTML, we use <table> tag.

* The <tr> tag is used to define each table row.
* The <th> tag defines a table header.
* A table cell is defined with the <td> tag.

<table>

<tr>

<th>Day</th>

<th>Lesson</th>

</tr>

<tr>

<td>Monday</td>

<td>Java</td>

</tr>

<tr>

<td>Tuesday</td>

<td>HTML</td>

</tr>

</table>

* We can span two or more columns by using the colspan attribute < td colspan="2" >Java< /td >
* We can span two or more rows by using the rowspan attribute  
  < td rowspan="2" >Java< /td >a
* To change table or cell position,<td align="center">java</td >

### What is HTML?

### HTML stands for Hyper Text Markup Language. It is a language of World Wide Web. It is a standard text formatting language which is used to create and display pages on the Web. It makes the text more interactive and dynamic. It can turn text into images, tables, links.

### 

### Do all HTML tags have an end tag?

### No. There are some HTML tags that don't need a closing tag. For example: <image> tag, <br> tag.

### Are the HTML tags and elements the same thing?

No. HTML elements are defined by a starting tag, may contain some content and a closing tag.For example, <h1>Heading 1</h1> is a HTML element but just <h1> is a starting tag and </h1> is a closing tag.

### What are tags and attributes in HTML?

Tags are the primary component of the HTML that defines how the content will be structured/ formatted, whereas Attributes are used along with the HTML tags to define the characteristics of the element. For example, <p align=” center”>Interview questions</p>, in this the ‘align’ is the attribute using which we will align the paragraph to show in the center of the view.

### What are void elements in HTML?

HTML elements which do not have closing tags or do not need to be closed are Void elements. For Example <br />, <img />, <hr />, etc.

### What are the various formatting tags in HTML?

HTML has various formatting tags:

* <b> - makes text bold
* <i> - makes text italic
* <em> - makes text italic but with added semantics importance
* <big> - increases the font size of the text by one unit
* <small> - decreases the font size of the text by one unit
* <sub> - makes the text a subscript
* <sup> - makes the text a superscript
* <del> - displays as strike out text
* <strong> - marks the text as important
* <mark> - highlights the text
* <ins> - displays as added text

### How can we club two or more rows or columns into a single row or column in an HTML table?

HTML provides two table attributes “rowspan” and “colspan” to make a cell span to multiple rows and columns respectively.

### Difference between link tag <link> and anchor tag <a>?

The anchor tag <a> is used to create a hyperlink to another webpage or to a certain part of the webpage and these links are clickable, whereas, link tag <link> defines a link between a document and an external resource and these are not clickable.

### What are some of the advantages of HTML5 over its previous versions?

Some advantages of HTML5 are:-

* It has Multimedia Support.
* It has the capabilities to store offline data using SQL databases and application cache.
* Javascript can be run in the background.
* HTML5 also allows users to draw various shapes like rectangles, circles, triangles, etc.
* Included new Semantic tags and form control tags.

### What are the New tags in Media Elements in HTML5?

* <audio> - Used for sounds, audio streams, or music, embed audio content without any additional plug-in.
* <video> - Used for video streams, embed video content etc.
* <source> - Used for multiple media resources in media elements, such as audio, video, etc.
* <embed> - Used for an external application or embedded content.
* <track> - Used for subtitles in the media elements such as video or audio.

**11. CSS**

**CSS** stands for ***Cascading Style Sheets.***

CSS helps elements how to be displayed on the web.

CSS files the external style sheets which consist of the styles for **HTML** file.

**Hakon Wium Lie** released the first draft of “Cascading HTML Style Sheets” in **October 1994.** With the release of CSS 3, capabilities were broken into modules.Between June 2011 and June 2012, the following four modules were released as formal recommendations: color, selectors level 3, namespaces and media queries.

CSS is used to style your web pages.

Basic CSS rule consists of a selector and a declaration: The selector points to the HTML element you want to style

The declaration block contains one or more declarations separated by semicolons. In this example { color: blue; } is the declaration block.

The styles can be located in a different file, so with an external style sheet, the design of the website can be changed from this file.

To include an external stylesheet, <link> element must be used as a reference to the external style sheet in the <head> section of the HTML page.

All styles which are applied to the same element will apply in the cascade order of rules.

All elements in the HTML file are considered as boxes. The "Box Model" term is used to describe a box that surrounds an element.

**It consists of: margins, borders, padding, and content.**

**CSS element Selector:**

An element selector uses element name to select the HTML element.

In what order do the properties work in the box?

Select one:

top-right-bottom-left

p selector is used to select all <p> elements in the HTML file.

The id selector uses id attribute to select the element in HTML file.The id of an element must be unique in a page. A hash (#) character is used to select an element with a specific id.

The class selector uses class attribute to select all elements of that class in HTML file.

A period (.) character is used to select elements with a specific class.

**The Universal Selector**

The universal selector \* selects all HTML

elements on the page.

**In what order do the properties work in the box?**

top-right-bottom-left

**How do you insert a comment in a CSS file?**

/\* this is a comment \*/

A Media query is a CSS3 feature that makes a webpage adapt its layout to different screen sizes and media types.

@media media type and (condition: breakpoint) { // CSS rules }

Always put your media queries at the end of your CSS file.

* all — for all media types
* print — for printers
* screen — for computer screens, tablets and, smart-phones
* speech — for screen readers that “read” the page out loud

@media screen and (max-width: 480px) { .text { font-size: 16px; } }

A breakpoint is a key to determine when to change the layout and adapt the new rules inside the media queries

**Common breakpoints for widths of devices:**

* 320px — 480px: Mobile devices
* 481px — 768px: iPads, Tablets
* 769px — 1024px: Small screens, laptops
* 1025px — 1200px: Desktops, large screens
* 1201px and more —  Extra large screens, TV

The logical keywords not or `only`` can be used **optionally** to include or exclude specific media types or screen sizes.

We can use and and , to separate or combine conditions.