

Introduction to Cybersecurity

Cybersecurity is the practice of protecting systems, networks, and data from digital attacks.

It includes implementing measures to prevent unauthorized access, data breaches, and other cyber threats.

Cybersecurity is crucial in today's digital world to safeguard sensitive information and maintain trust with users.

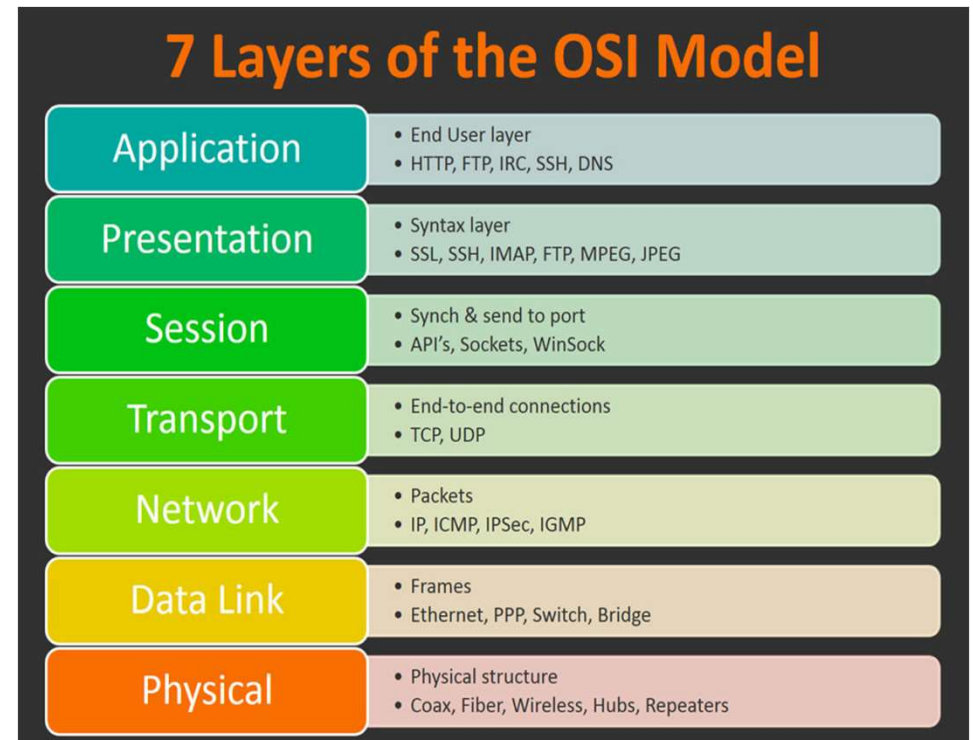


Networking TCP and OSI Model

TCP (Transmission Control Protocol) is a communication protocol that ensures reliable and ordered delivery of data packets over a network.

The OSI (Open Systems Interconnection) model is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven layers.

Understanding TCP and the OSI model is essential for designing and troubleshooting network architectures.

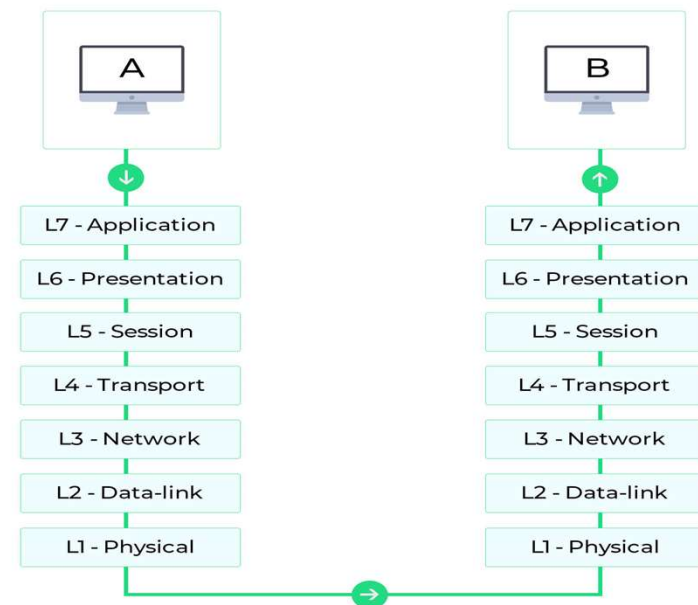


Ports and Protocols

Ports are virtual endpoints for communication in a network, allowing different applications to send and receive data.

Protocols define the rules and formats for communication between devices and systems in a network.

Common protocols include TCP, UDP, HTTP, and FTP, each serving specific purposes in data transmission.



Introduction to Python

Python is a high-level programming language known for its simplicity and readability, making it ideal for beginners and experienced developers alike.

It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Python's vast standard library and community support make it a versatile tool for various applications, including web development, data analysis, and automation.



Cybersecurity Best Practices

Use strong, unique passwords for each online account to prevent unauthorized access.

Keep software and operating systems up to date to patch vulnerabilities and protect against cyber threats.

Be cautious of phishing emails and suspicious links to avoid falling victim to social engineering attacks.



TCP/IP Protocol Suite

The TCP/IP protocol suite is the foundation of the internet, defining how data is transmitted between devices.

It consists of four layers: Application, Transport, Internet, and Network Interface, each handling specific aspects of data communication.

Understanding the TCP/IP protocol suite is essential for networking professionals to troubleshoot and optimize network performance.

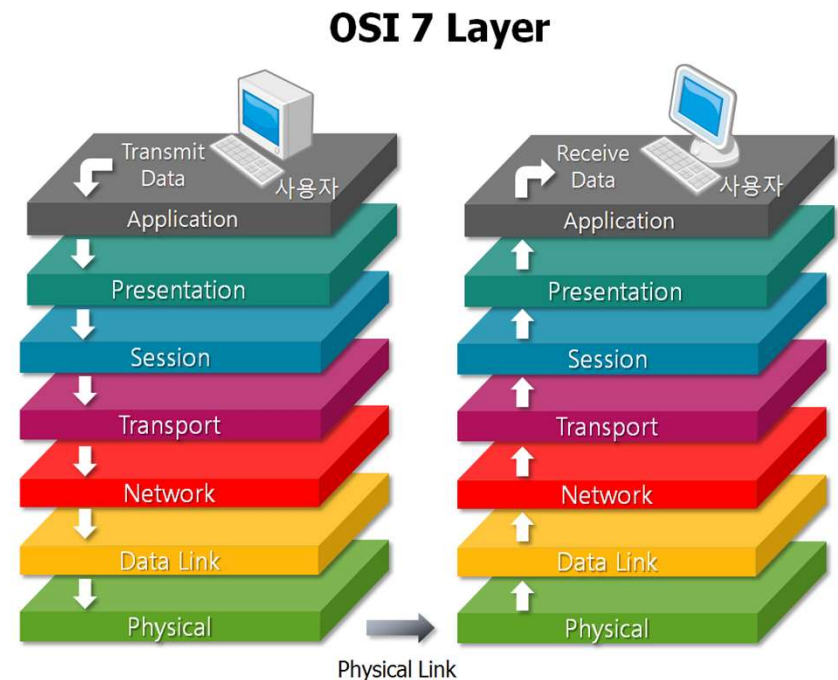
TCP/IP Model Edukedar	Layer	Protocols
	Application	HTTP,FTP,POP3, SMTP,SNMP
	Transport	TCP,UDP
	Networking	IP,ICMP
	Datalink	Ethernet, ARP

OSI Model Layers

The OSI model consists of seven layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.

Each layer serves a unique function in data transmission, from converting data into signals at the physical layer to presenting information to users at the application layer.

The OSI model provides a framework for understanding how different network components interact and communicate.



Common Network Ports

Port 80: HTTP (Hypertext Transfer Protocol) for web browsing.

Port 443: HTTPS (Hypertext Transfer Protocol Secure) for secure web communication.

Port 22: SSH (Secure Shell) for secure remote access to network devices.

COMMON PORTS

TCP/UDP Port Numbers

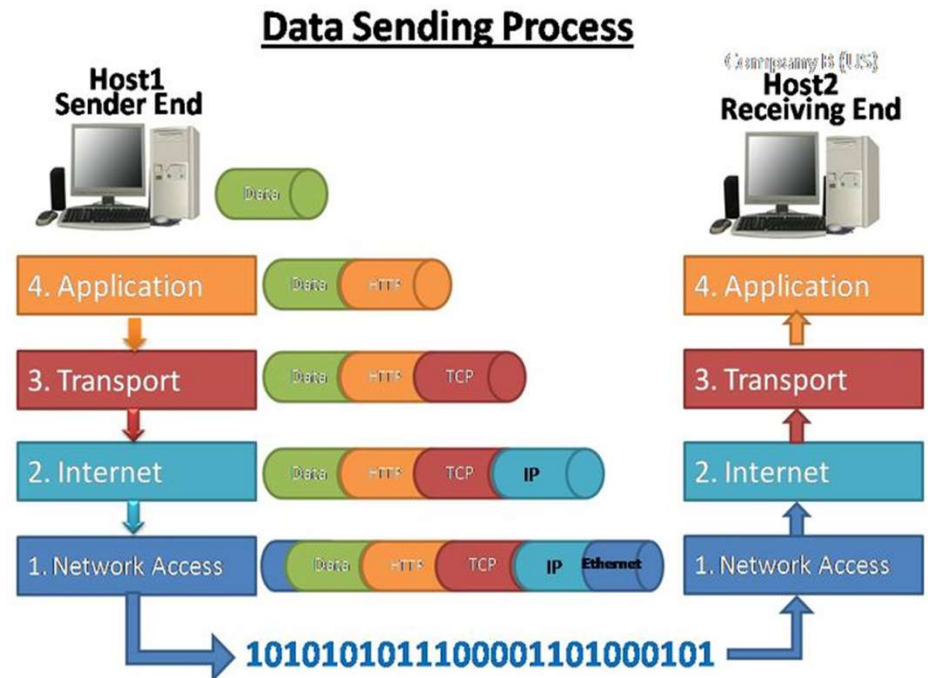
7 Echo	554 RTSP	2745 Bagle.H	6891-6901 Windows Live
19 Chargen	546-547 DHCPv6	2967 Symantec AV	6970 Quicktime
20-21 FTP	560 rmonitor	3050 Interbase DB	7212 GhostSurf
22 SSH/SCP	563 NNTP over SSL	3074 XBOX Live	7648-7649 CU-SeeMe
23 Telnet	587 SMTP	3124 HTTP Proxy	8000 Internet Radio
25 SMTP	591 FileMaker	3127 MyDoom	8080 HTTP Proxy
42 WINS Replication	593 Microsoft DCOM	3128 HTTP Proxy	8086-8087 Kaspersky AV
43 WHOIS	631 Internet Printing	3222 GLBP	8118 Privoxy
49 TACACS	636 LDAP over SSL	3260 iSCSI Target	8200 VMware Server
53 DNS	639 MSDP (PIM)	3306 MySQL	8500 Adobe ColdFusion
67-68 DHCP/BOOTP	646 LDP (MPLS)	3389 Terminal Server	8767 TeamSpeak
69 TFTP	691 MS Exchange	3689 iTunes	8866 Bagle.B
70 Gopher	860 iSCSI	3690 Subversion	9100 HP JetDirect
79 Finger	873 rsync	3724 World of Warcraft	9101-9103 Bacula
80 HTTP	902 VMware Server	3784-3785 Ventrilo	9119 MXit
88 Kerberos	989-990 FTP over SSL	4333 mSQL	9800 WebDAV
102 MS Exchange	993 IMAP4 over SSL	4444 Blaster	9898 Dabber
110 POP3	995 POP3 over SSL	4664 Google Desktop	9988 Rbot/Spybot

Common Network Protocols

TCP (Transmission Control Protocol) ensures reliable data delivery through connection-oriented communication.

UDP (User Datagram Protocol) provides fast, connectionless data transmission for real-time applications.

HTTP (Hypertext Transfer Protocol) enables web browsing and communication between web servers and clients.



Python Basics

Python uses indentation to define code blocks, promoting readability and enforcing code structure.

Variables in Python are dynamically typed, meaning you don't need to specify a data type when declaring a variable.

Python supports both functional and object-oriented programming paradigms, allowing for flexible and efficient code development.

Indentation in Python (With Examples)

Rules, Benefits, Disadvantages &
Errors

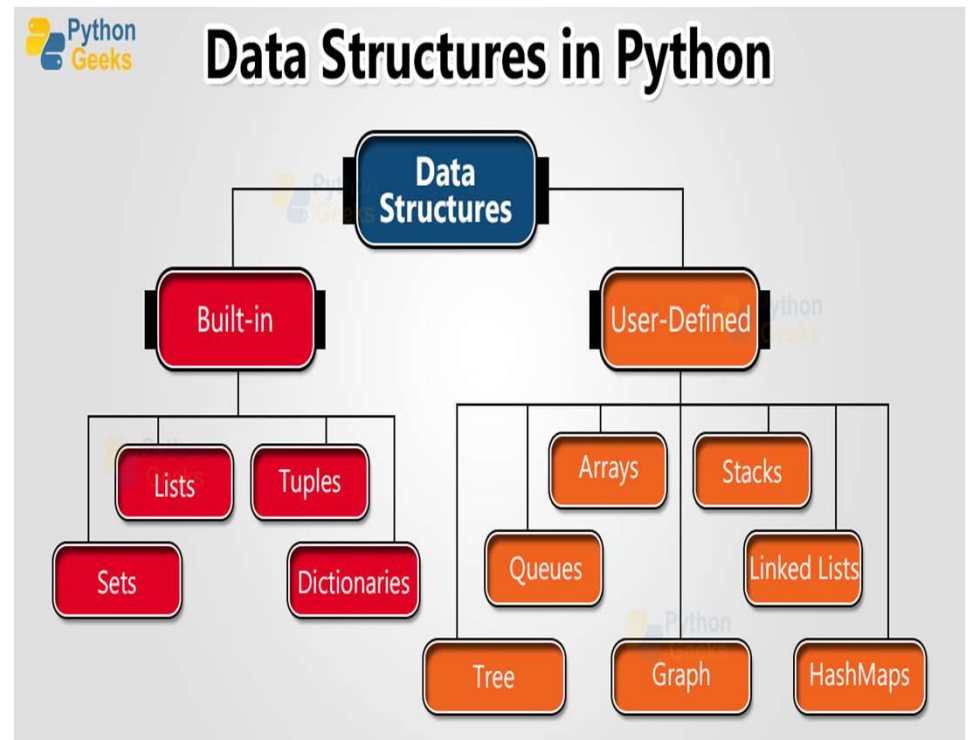


Python Data Structures

Python offers built-in data structures like lists, tuples, dictionaries, and sets for storing and organizing data.

Lists are ordered collections of items, while dictionaries store key-value pairs for efficient data retrieval.

Understanding data structures in Python is essential for manipulating and processing data in various applications.

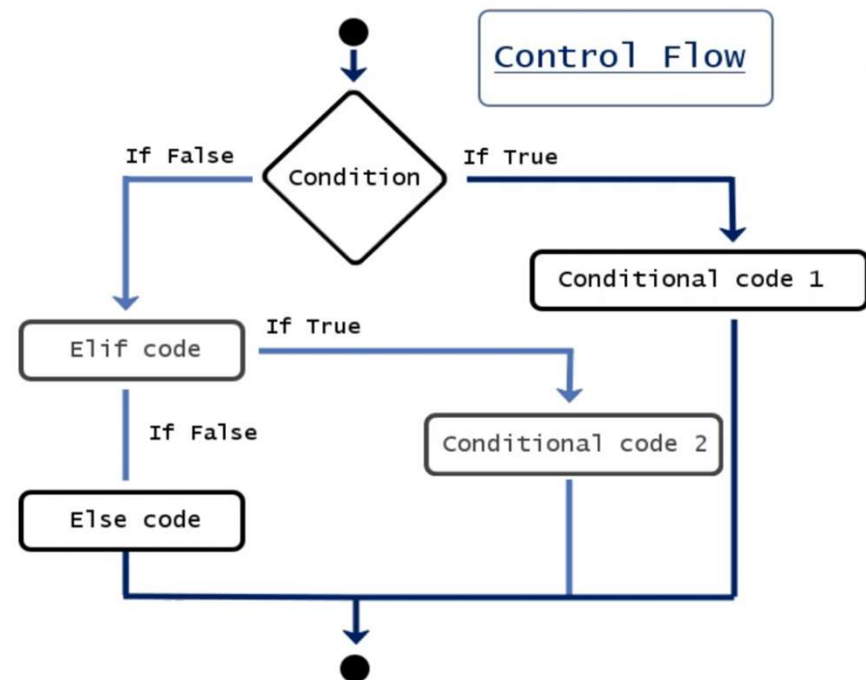


Python Control Flow

Python uses conditional statements like if, elif, and else to control the flow of a program based on specific conditions.

Loops in Python, such as for and while loops, allow for iterative execution of code blocks.

Control flow structures in Python enable developers to implement decision-making and repetitive tasks in their programs efficiently.



Python Functions

Functions in Python are reusable blocks of code that perform specific tasks when called.

Parameters and return values can be defined for functions to accept input and produce output.

Python functions promote code reusability, modularity, and maintainability in software development projects.

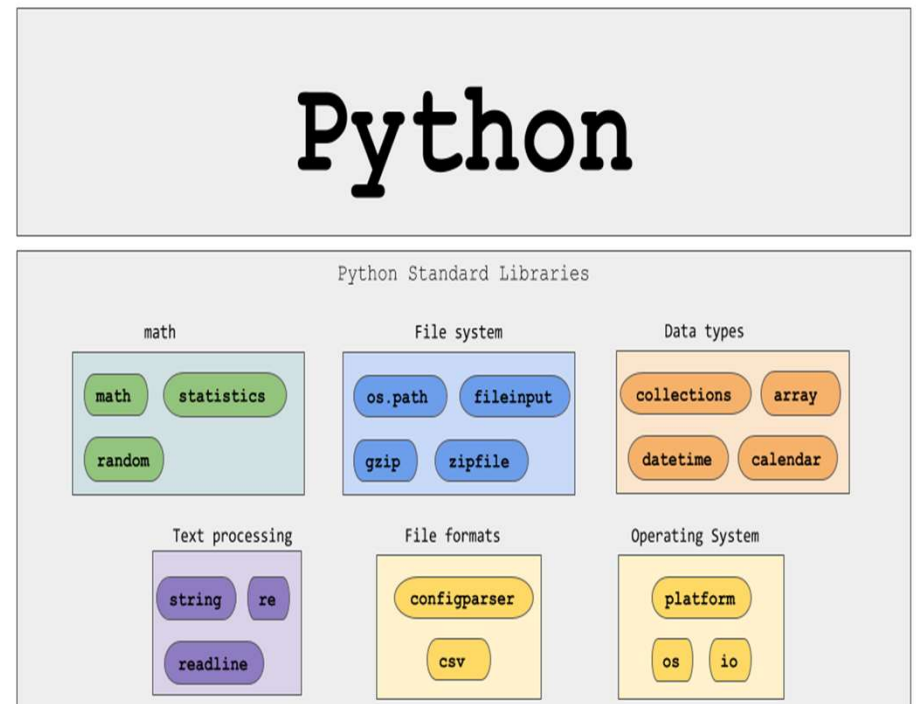


Python Libraries and Modules

Python's extensive standard library provides pre-built modules for various tasks, such as math operations, file handling, and networking.

External libraries like NumPy, pandas, and matplotlib extend Python's capabilities for scientific computing, data analysis, and visualization.

Leveraging libraries and modules in Python allows developers to streamline development processes and enhance the functionality of their applications.



Python Applications

Python is used in various fields, including web development, data science, machine learning, automation, and cybersecurity.

Popular frameworks like Django and Flask empower developers to build dynamic web applications using Python.

Python's versatility, ease of use, and vast ecosystem of tools make it a top choice for professionals across different industries.

