Contents

FinanceBot

LaTeX

Project Area

English version

Class 4J

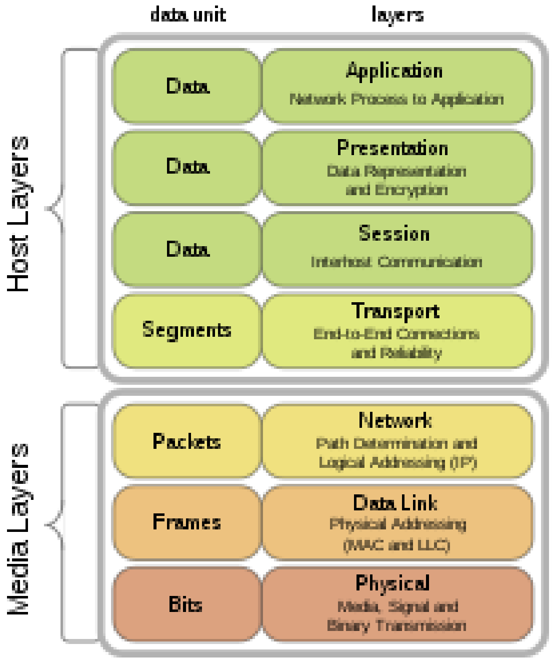
1 Introduction

1.1 Who we are

We are a class of a technical institute of computer science. Together we have built a program that monitors NASDAQ stocks through the Yahoo finance API. The processed data will then be processed by a telegram bot.

2 Network protocol

Network protocols comprise 5 protocols, the most important being IP, ARP and ICMP protocols. The transport layer allows applications running on remote terminals to communicate. It contains two protocols that allow the two applications to exchange data independently of the lower layers.Internet Protocol (IP), is a network protocol, which deals with addressing / routing, belonging to the suite of Internet protocolsTCP / IP on which the operation of the Internet is based.Modern protocols for computer networks generally use packet switching technology to send and receive messages in the form of packets, which are messages broken down into small pieces that are collected andreassembled at their destination



	data unit	layers
Host Layers	Data	Application Network Process to Application
	Data	Presentation Data Representation and Encryption
	Data	Session Interhost Communication
	Segments	Transport End-to-End Connections and Reliability
Media Layers	Packets	Network Path Determination and Logical Addressing (IP)
	Frames	Data Link Physical Addressing (MAC and LLC)
	Bits	Physical Media, Signal and Binary Transmission

Figure 1: Network protocol

3 ISO/OSI Model

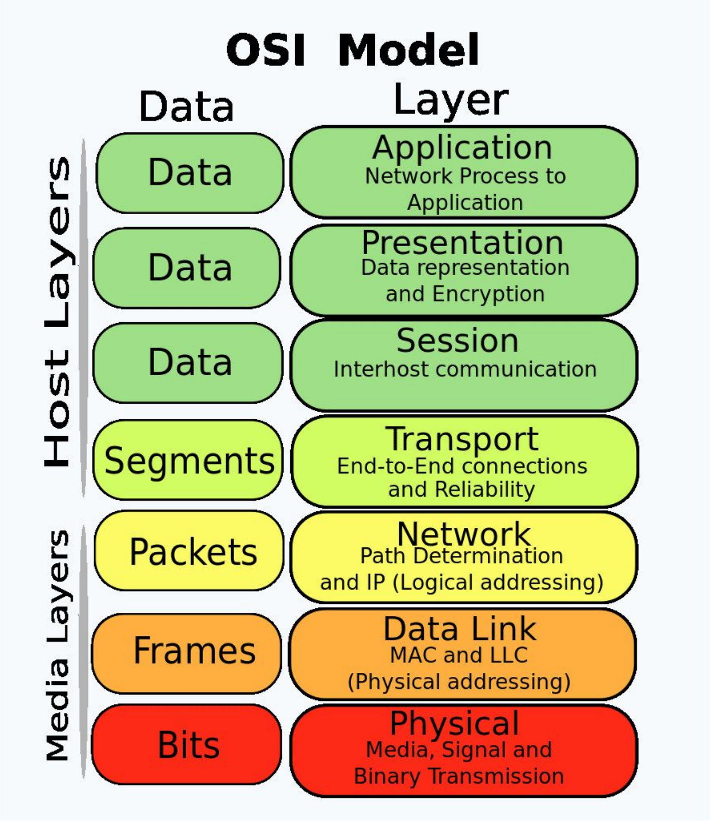

OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data representation and Encryption
Data
Session
Interhost communication
Segments
Transport
End-to-End connections and Reliability
Packets
Network
Path Determination and IP (Logical addressing)
Frames
Data Link
MAC and LLC (Physical addressing)
Bits
Physical
Media, Signal and Binary Transmission

Figure 2: Model ISO/OSI

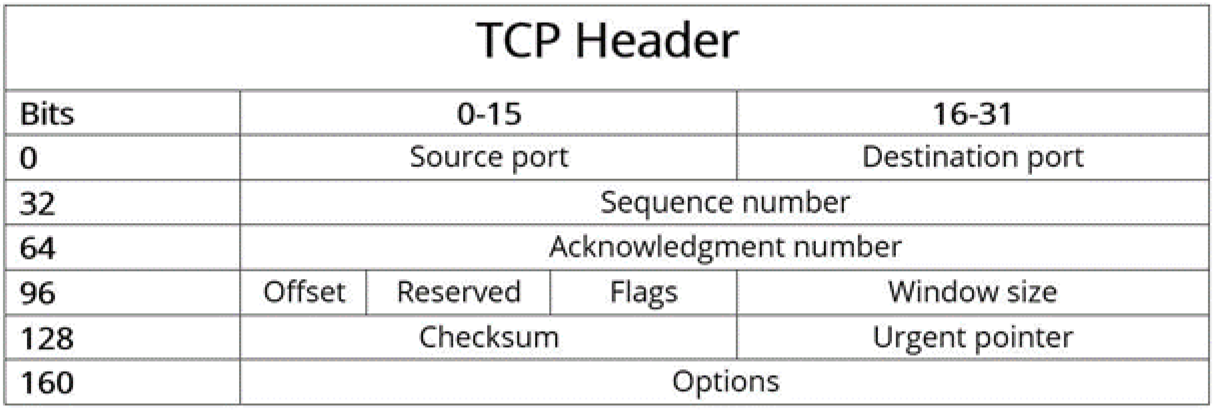
4 Level protocol

4.1 TCP protocol

The transmission control protocol (tcp) is a standardized agreement for the data transmission between different users of a computer network.The current state of development of the TCP protocol allows two endpoints within the a common computer network to make a connection through which it can take place a bidirectional transmission of data. As part of this connection, any data losses are recognized and corrected automatically; for this, the TCP comes also referred to as a reliable protocol. In the Internet protocol family, TCP constitutes, together with UDP and SCTP, the group of transport protocols which, on the basis of the OSI model, are classified in the network architecture at the transport layer. So long as,in almost all cases, the TCP protocol is based on the Internet Protocol (IP) and this one combination forms the basis for most public and local networks and for network services, often also referred to as a stack of TCP / IP protocols, meaning in a broad sense the Internet protocol family.

4.1.1 How does it work?

TCP allows the transmission of information in both directions. The systems IT professionals who communicate via TCP can therefore send and receive dataat the same time, just like during a telephone conversation. The fundamental transmission units used by the protocol are the segments (packets) which,in addition to the payload (i.e. the actual message), they may also contain information from control and are limited to a size of 1,500 bytes. The establishment and interruption of connections, classifiable as point-to-point connections, as well as transmission data itself is acquired by the TCP software in the network protocol stack of there spective operating system.



TCP Header

Bits	0-15			16-31
0	Source port			Destination port
32	Sequence number			
64	Acknowledgment number			
96	Offset	Reserved	Flags	Window size
128	Checksum			Urgent pointer
160	Options			

Figure 3: TCP

4.2 UDP Protocol

4.2.1 Applications that use UDP

Network applications that need reliable data transfer obviously don't rely on UDP, while more flexible and time-dependent applications rely on UDP instead. Furthermore, UDP is used for broadcast communications (sending to all terminals on a local network) and multicast (sending to all terminals subscribed to a service).

Application	application status protocol	transport layer protocol
email	SMTP	TCP
remote terminal access	telnet	TCP
file transfer	FTP	TCP
Web	HTTP	TCP
Streaming audio/video	RTSP/RTP	TCP (comands) + UDP (flow)
Remote file server	NFS	UDP typically
Internet telephony (VoIP)	SIP, H.323, other	UDP typicallyP
Network management	SNMP	UDP typically
Routing protocol	RIP	UDP typically
Name resolution	DNS	UDP typically

Table 1: list of the main Internet services and the protocols they adopt

5 Client-Server application

5.1 What is

A client / server application is a distributed application divided into two parts: a server application (back-end) that offers services and a client application (front-end) that manages the user interface and allows request services from the server.

The client and server components of the application are generally located on two different stations (the server part is usually located on a much more powerful computer) but they can also be located in the same station.

The client / server architecture can also be implemented at three levels, with a user interface management level, a processing level and an information management level (normally stored in a database).

It is a non-symmetrical model, also called request / replay because it is based on the succession of requests and responses.

The server can be designed to serve only one request at a time or multiple requests at the same time.

To contact the server, the client must know its address; the address may have to be known in advance, there may be a process server to which the client makes requests which are then forwarded to the desired server, or there may be a directory server which, based on the name of the desired server, returns the address.

The most common technique for creating client / server applications is the use of TCP / IP sockets.

6 DNS

DNS is the acronym for Domain Name System and, to explain it briefly, it is the system that translates IP addresses into domain names. When we browse the web, in the address bar of the browser we write the domain of a site, for example, wp.it.

What happens on a technical level is this:

- the browser takes the domain you entered;
- this domain is associated with an IP address;
- the IP address is a series of numbers that identifies a particular server, that is a computer within which the website is located;
- the DNS translates the domain name into IP address, and vice versa;

Figure 4: DNS

6.1 What is DNS for?

The dns are used to carry out a translation from the name (the domain) to the number (IP address), such as the address book of a telephone. So instead of writing the whole IP address, just type the name of the person you want to call. Another example could be a website, instead of writing the whole IP address, just type the name of the site, such as Amazon.com.

7 HTTP

HTTP (Hypertext Transfer Protocol, literally hypertext transfer protocol) is a text language that allows communication between client and server over the internet. So every time we visit a website, HTTP allows us to view the resources of that site.

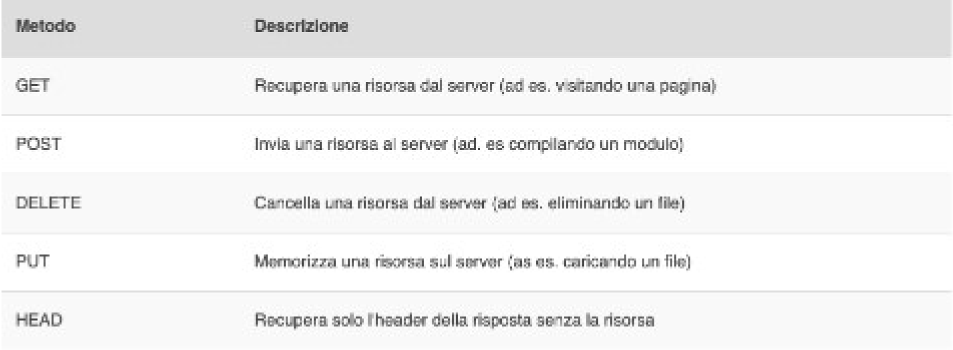
7.1 How HTTP request and response works:

After briefly introducing the HTTP protocol, now let's see how it works step by step, analyzing how an HTTP request is made and the subsequent response from the server. Request the client sends a request: Every conversation between client and server on the web begins with a request, a text message created by the client in a special format known as HTTP.

An HTTP request consists of the following parts:
[method] [URI] [version]
[headers]
[body]

The first line of an HTTP request is the most important and contains two basic parts:

1. HTTP metod (GET)
2. URL (/blog/protocol-http-what-and-how-work)



Metodo	Descrizione
GET	Recupera una risorsa dal server (ad es. visitando una pagina)
POST	Invia una risorsa al server (ad. es compilando un modulo)
DELETE	Cancella una risorsa dal server (ad es. eliminando un file)
PUT	Memorizza una risorsa sul server (as es. caricando un file)
HEAD	Recupera solo l'header della risposta senza la risorsa

Figure 5: HTTP

HTTP methods define the ways in which the client can interact with the resource. The following table shows some of the methods used during an HTTP request.

8 CSV

A CSV (Comma Separated Values) file is one of the most popular formats for storing tabular data and is mainly used to manage a large amount of data. The elements of the csv are commonly separated by a comma. Usually the columns of our table are represented in the first row, while the values are represented in the following rows.

8.1 How to read a CSV file?

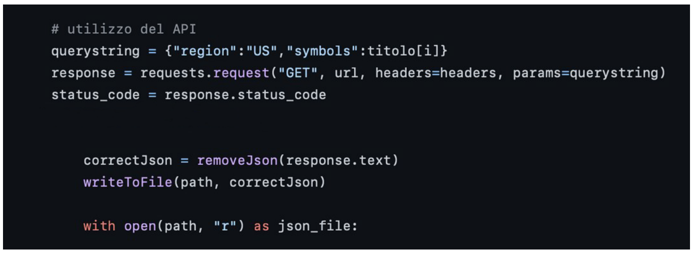
The most common practice to interact with a csv file in python is to use the library of the same name. Let's start by importing it into our script: import csv. To read the contents of our file we can use the csv.reader () method.

In the event that our file.csv uses a character other than a comma as a delimiter, we must specify it by passing a second parameter to the methodreader.csv() . This parameter is called delimiter and by default it is set with a comma. Also, by calling the method DictReader.csv() instead of the reader.csv (), we get the contents of the csv in a dictionary.

8.2 How to write a CSV file?

- To write a CSV file in Python you can use the write_row () method of the writer object, which is retrieved by calling the method writer.csv.().
- To see the result you have to go back to the script written previously to read the file, change the name of the csv with the one we have generated and run it.
- As for reading, also for writing you can define the delimiter character in the method writer.csv (file, delimiter = ";").
- Using the method DictWriter.csv() you can create a csv from a dictionary in this way.
- Keep in mind that in this case you need to define the columns you want to use.

9 API



```
# utilizzo del API
querystring = {"region":"US","symbols":titolo[i]}
response = requests.request("GET", url, headers=headers, params=querystring)
status_code = response.status_code


    correctJson = removeJson(response.text)
    writeToFile(path, correctJson)

    with open(path, "r") as json_file:
```

Figure 6: API

API is short for application programming interface, which is a set of definitions and protocols for creating and integrating application software. Intuitively, they are therefore the basis of a vastness of applications, services, websites, devices. APIs were born with the very birth of information technology, well before personal computers.

10 Bot telegram

10.1 Creation

The Bot was written in Java thanks to the ”IntelliJ” programming environment. It includes 3 object classes plus an xml file. For the creation of a Telegram Bot you need to use the Bot called ”BotFather” which will give an API to be able to synchronize the program with Telegram.

10.2 Class

- Main: allows the bot to be started and the entire program to work.
- SimpleBot: returns the output on telegram in correspondence with the requested command.
- LockFile: creates and destroys file locks.

10.3 deepening

We decided to use files.lock because they prevent the program in Python and the one in Java from interfering and operating at the same time on the same file. This functionality could have been replaced if we had decided to use Sockets to synchronize the two programs.

All the classes are stored in a special file.jar to allow the whole program to be used and with the possibility of being started also from the terminal.

11 JSON

11.1 What is JSON

JSON is a text-based serialization format for exchanging data, primarily between server and web application. JSON stands for JavaScript Object Notation, and uses the file.json extension. JSON is a competitor to XML, but has a simpler and more compact syntax than its rival.

11.2 How JSON works

1. The data is written as a key-value pair.
2. The data is separated by a comma.
3. The curly brackets contain the objects.

4. The curly brackets contain the objects.
5. Strings must be enclosed with double quotes.

12 Java programming language

Java is a high-level, class- and method-based, object-oriented programming language. Data is represented as objects and operations as methods that operate on them. This programming language was designed for developing applications on networks and to be as independent as possible from the execution Hardware platform.

12.1 advantages and disadvantages

advantages:

- cross-platform compatibility (from the Virtual Machine)
- high abstraction from the physical machine
- speed of development
- large availability of libraries
- compatibility with handheld and embedded systems
- high integration with the web

Disadvantages:

- slowness (from the Virtual Machine)
- high abstraction from the physical machine (which does not allow you to do everything you want)
- it is decompilable.

13 Python programming language

Python is a higher-level programming language than most other languages, which are also object-based. Among the common uses, Python is suitable for developing distributed applications, scripting, numerical computing, system testing, websites, software etc. In addition, there are many Python libraries available in the educational and scientific fields.

13.1 advantages and disadvantages

One of the biggest drawbacks is certainly the lack of types, hence a lack of static control. Also, the Python compiler is unable to find many errors and lack of checking of the values returned by functions. In fact, we are forced to look for all cases, and often the codes that handle errors are difficult to reach. One of the biggest advantages is the ease and cleanliness of its syntax which immediately makes the code very clear, moreover, thanks to the support of the Python Software Foundation, it makes use of a very large built-in library which makes it one of the most popular programming languages. rich and practical. Python can be defined as a pseudo-complicated and portable language, in fact an interpreter can analyze the source code and then execute it if it deems it correct since there is no separate compilation phase from which an executable file can be generated.

14 LaTeX

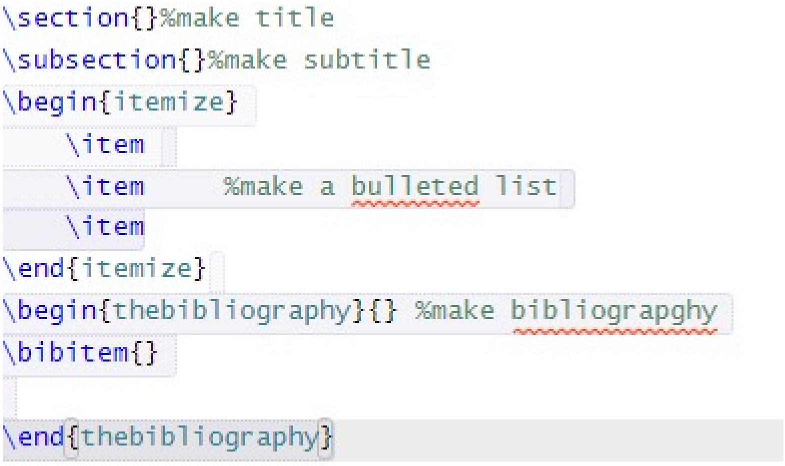
14.1 What is LaTeX?

LaTeXis a programming language developed in 1985 by Leslie Lamport. The latest version dates back to 2011. LaTeXis a markup language, in this case the most used marker is the (backslash). LaTeXis mainly used to produce pdf documents. This language to produce pdf files must be compiled.

14.2 How we used LaTeX

To make our document we used an editor and created the document from scratch. We then added an index and chapters and sub-chapters to make the pdf more orderly. We have added options such as the index and to distribute everything well we have chosen to use chapters and sub-chapters. For the further development of the code we have used some packages such as those for special characters, which includes the possibility of making the system recognize special characters, such as accented letters.

14.3 Most common commands



```
\section{}%make title
\subsection{}%make subtitle
\begin{itemize}
    \item
    \item     %make a bulleted list
    \item
\end{itemize}
\begin{thebibliography}{} %make bibliograpghy
\bibitem{}

\end{thebibliography}
```

Figure 7: common command