

You inplug RJ45 between 2 computers and boom you have a network,
the internet is accessible via web site WWW
http:// => is the protocol
www.omycode.tn => is the domain name
=> they both form an URL: Uniform Resource Locator

browsers allows u to visit web sites, receive, send, print, upload, download

internet # World Wide Web => they're related but not the same thing

the internet is a massive network of networks or a networking infrastructure
the internet connects millions of computers globally
while 2 or more computers are connected to the internet they can communicate in between
the information to exchange via internet have to submit to set of rules
the internet is decentralized (it doesn't have a center), it's a global network connecting millions of computers, each internet computer is independent (that means if one of the line is down not all of the net is down)

WWW: World Wide Web

it's a way to accessing information through the internet -> it's a model to share information
the web is on the top of internet
the web uses HTTP protocols and web browsers

Web Architecture

we have 2 sides of internet: clients and servers

CLIENTS can't communicate directly to the internet, they use Internet Service Provider ISP (it's a company that provides access to the internet like topnet, globalnet, telecom, orange etc)

SERVERS programs/systems that run on a special computer -> they deliver data/resources/services to other computers which we call clients over network

PROTOCOLS is a set of rules, conventions and data structures that dictate how devices exchange data across network

HTTP HyperText Transfer Protocol: it defines how msg are formatted & transmitted, determines what action web servers & browsers should take: get/post/put/delete

IP Address :Internet Protocol which can be static (defined by an admin of the network)

TCP: Transmission Control Protocol: it allows the fragmentation/defragmentation of the packets transmitted via net

since the internet is a packet-switched network the IP and TCP work together to transmit the right information

DHCP: Dynamic Host Configuration Protocol: which allow one & only IP address for each client dynamically (our box internet is a local DHCP)
when a client connect to the internet the DHCP server launch a scan and generate an IP address to locate it to the new client. Once this client leaves the internet, the IP address does not belong to it anymore and another client can use it

API : Application Programming Interface: it's a software intermediary that allows 2 apps to speak to each other

it's a set of programming code

API servers can simplify and speed up software development

API : took the request from the client, interpreted, look for the right action to do and return the answer on the used application

API Advantages:

- *automation : means now we can organize, update, give order for tasks and works via applications

- *application : an API can access application components

- *more scope : can be customized to create personalized user experience

- *unlimited access to data: allows all public information

- *efficiency:

- *integration:

- *personalization: each user/company can personalize the content /service they use

- *adaptation:

==> APIs are developer-friendly, easily accessible and incredibly-flexible

Open APIs/external/public : available for developers with minimal restrictions, they may require registration & use of an API key or they can be completely open

Internal APIs: hidden from external user, only in-house developers have the access

Partner APIs : similar to open APIs but with controlled publisher

Composite APIs: access to a several endpoints in one call

Web Service APIs: SOAP, XML-RPC, JSON-RPC, REST

REST API : Representational State Transfer: which is a very popular web architecture:

client-server architecture: the interface is separated from the back-end & data storage -> flexibility & # components to evolve independently

statelessness: no client context is stored on the server

cacheability: clients can cache responses

layered system: the API will work whether it's communicating directly or via a load balancer

Rule # 1: the URI (Uniform Resource Identifier) must operate as a resource identifier.

Rule # 2: HTTP verbs must function as the identifiers of operations.

Rule # 3: HTTP responses must operate as representations of resources.

Rule # 4: links are like a relationship between resources.

Rule # 5: The existence of a parameter like an authentication token.

HTTP Methods

Defines a set of query methods that indicate the action that we want to perform on the indicated resource.

GET: The GET method requests a representation of the indicated resource. GET requests should only be used to retrieve data.

HEAD: The HEAD method requests a response identical to a GET request for which we have omitted the body of the response (we only have the header).

POST: The POST method is used to send an entity to the indicated resource. This usually results in a change of state or a cause of side effects on the server.

PUT: The PUT method replaces all current representations of the resource targeted by the content of the request.

DELETE: The DELETE method deletes the specified resource.

CONNECT: The CONNECT method builds a tunnel to the server identified by the target resource.

OPTIONS: The OPTIONS method is used to describe the communication options with the targeted resource.

TRACE: The TRACE method performs a round trip test message by following the path of the targeted resource.

PATCH: The PATCH method is used to apply partial modifications to a resource.

DATABASE is a systematic collection of data/ support storage & manipulation of data/simplify data management

These are the database types:

Relational databases.

Object-oriented databases.

Distributed databases.

Data warehouses.

NoSQL databases.

Graph databases.

OLTP databases.

SQL "Structured Query Language". > "SQL is a standard language for storing, manipulating and retrieving data in databases."

SQL commands :

CREATE DATABASE to create a database.

CREATE TABLE to create tables.

SELECT to find/extract some data from a database.

UPDATE to make adjustments and edit data.

DELETE to delete some data.

NoSQL; not only SQL-> The most common types of NoSQL databases are key-value, document, column and graph databases.

Examples of NoSQL Databases

Many NoSQL databases were designed by technology companies like Google, Amazon, Yahoo, and Facebook to provide more effective ways to store content or process data for huge websites. Some of the most popular NoSQL databases include the following:

Apache CouchDB

An open-source, JSON document-based database that uses JavaScript as its query language.

Apache Cassandra

An open-source, wide-column store database designed to manage large amounts of data across multiple servers and clustering that spans multiple data centers.

MongoDB

An open-source document-based database that uses JSON-like documents and schema, and is the database component of the MERN stack.

Redis

A powerful in-memory key-value store used for session caching, message queues, and other specific applications.

Elasticsearch

A document-based database that includes a full-text search engine.

What's the Difference Between SQL and NoSQL?

SQL are relational <> NoSQL are non-relational.

SQL use structured query language and have a predefined schema <> NoSQL have dynamic schemas for unstructured data.

SQL vertically scalable <> NoSQL horizontally scalable.

SQL table based <> NoSQL document, key-value, graph or wide-column stores.

SQL better for multi-row transactions <> NoSQL better for unstructured data like documents or JSON.

DBMS : DataBase Management System: it's a software program allow to comprehend the DB language
Database Management System

A database typically requires a comprehensive database software program known as a database management system (DBMS).

A DBMS serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized. A DBMS also facilitates oversight and control of databases, enabling a variety of administrative operations such as performance monitoring, tuning, backup and recovery.

Using DBMS allows us to:

Securely store a large volume of data for long periods of time.

Access data efficiently.

Control access to data by multiple users at the same time. One user's actions should not affect the other.

Create new databases as well as specify their schemas.

Support for a data model.

Query and modify the database's data.

Have data redundancy.

Have more flexibility than the use of files.

Have data consistency and integrity.

DBMS: MySQL/PostgreSQL/SQLite/Oracle/Microsoft SQL Server

Web Development

Web developers need a combination of graphic design skills and technical computer skills that will allow them to create particular designs on web pages. Not only do they have to be pleasing to look at but, websites also need to be functional

and secure. It is the responsibility of web developers to create websites that meet the requirements of employers or clients.

Creating a website

There are two ways to create a website:

CMS or Content Management System is an easy-to-use system that allows you to control and manage the content within your website without necessarily having technical skills.

You can easily add and delete images, links, presentation links, and other content on your website.

Hard Coding requires specific skills and knowledge of programming languages. That is the complex way to create a website and that is what web developers do.

In hard coding, you have the freedom to change what you want and build a custom website that completely satisfies the needs of the user, the employer, and the client.

What is a CMS?

A Content Management System, often abbreviated as CMS, is software that helps users create, manage, and modify content on a website without the need for technical knowledge.

In simpler terms, a content management system is a tool that helps you build a website without needing to write all the code from scratch or even know how to code at all.

Instead of building your own system for creating web pages, storing images, and other functions, the content management system handles all the basic infrastructure tasks for you so that you can focus on more forward-facing parts of your website.

Besides websites, you can also find content management systems for other functions, like document management.

WordPress, Wix, Squarespace, Joomla!, Shopify, Drupal, Blogger, Prestashop, Magento, Bitrix...Etc

CREATE WEB

Google Chrome : Google Chrome is a web browser. You may or may not already be using Google Chrome for browsing the web. A web browser allows us to view webpages (similar to the one you're reading this very text on). We'll use Google Chrome to view the results after writing and making changes to our code.

Visual Studio Code : is a popular code editor. A code editor is a very aptly named software; it helps us edit text. It is similar to the default text editor that comes preinstalled on Mac or Windows, but has additional features like code highlighting to enhance our coding and debugging experience.

Code editor is a lightweight software like a text editor. The only difference is code editor has special functionalities than a text editor. According to the Language used to code, the code editor highlights special keywords, gives suggestions to some extent, provides automatic indentation features and oftentimes it has an integrated terminal as well. Ex: Sublime Text, Visual Studio Code.

Recap

So in the previous chapters, we have learned a few basic concepts such as:

The difference between the web and the Internet.

How does the web work?

Understand what are protocols, servers and api.

Knowing what is a database.

Learning the meaning of web development.

The different ways to create a website (with CMS and with code).

What is a CMS?

- quick definition. A content management system is for creating, managing, and optimizing your customers' digital experience. More specifically, a CMS is a software application that allows users to collaborate in the creation, editing, and production of digital content: web pages, blog posts, etc.

