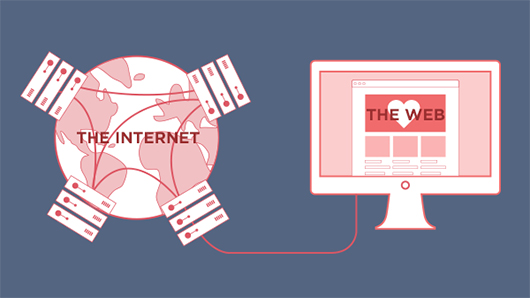
**Confusion between the terms: Internet and Web**

Many people use the terms Internet and World Wide Web (aka. the Web) interchangeably, but in fact, the two terms are not synonymous.

The Internet and the Web are two separate but related things. They are both very different.

So, the first thing to do is clear up the uncertainty between the two words.



# The Internet

The Internet is a massive network of networks or a networking infrastructure. It connects millions of computers globally, forming a network. In which any computer can communicate with any other computer as long as they are both connected to the Internet.  
Information that travels over the Internet does so via a set of rules known as protocols.

Some facts about The Internet:

* The Internet is a global network connecting millions of computers.
* The Internet is decentralized.
* Each Internet computer is independent.
* There are a variety of ways to access the Internet.
* There are more than 3.5 billion Internet users in the world. That’s almost half of the world’s population!



The Web (World Wide Web)

The World Wide Web, or simply the web, is a way of accessing information through the medium of the Internet. It is an information-sharing model that is built on top of the Internet. The web uses the HTTP protocol, one of the languages spoken over the Internet, to transmit data.

Web services that use HTTP allow applications to communicate, exchange business logic and use the web to share information. The web also utilizes browsers such as Google Chrome, Internet Explorer or Firefox, to access Web documents called web pages that are linked to each other via hyperlinks.

Web documents also contain graphics, sounds, text and video.

Some facts about The Web:

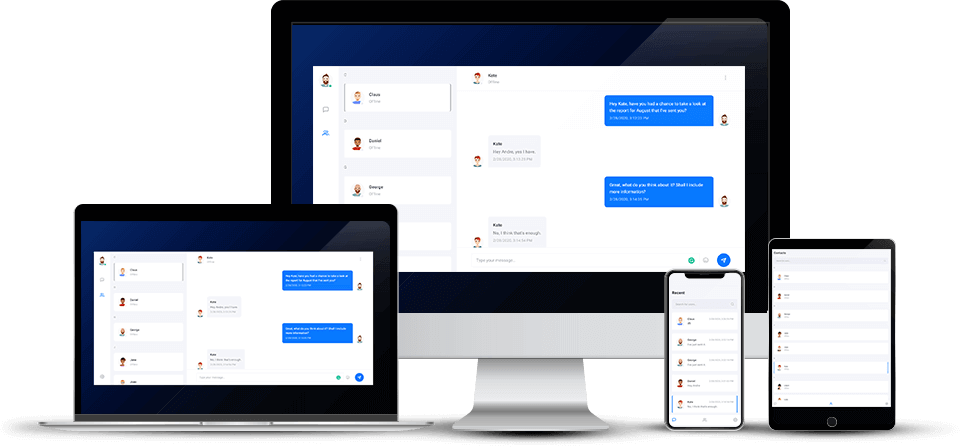
* The Web is a system of Internet servers that support specially formatted documents.
* Documents are formatted in a markup language that supports links to other documents.
* You can jump from one document to another simply by clicking on hot spots (hyperlinks).
* Applications, called web browsers, make it easy to access the World Wide Web.
* There are more than 1,275,000,000 websites.
* The Web is a a part of The Internet.

# Web architecture

The Web can be summarized as a series of interactions between two types of systems, **clients** and **servers**.  


Clients

* Clients are the user’s devices that request and render web content. Clients can be: browsers, mobile applications, TVs…Etc.  
  We live in an increasingly connected world and the number of clients that can access web content is growing rapidly.
* Clients are not connected directly to the Internet.
* Clients are connected to the Internet through their ISP (Internet Service Provider).



Servers

* Servers are programs or systems that run on a special computer ( with special hardware like network equipment and dedicated network access).
* Their job is to deliver data, resources, or services to other computers, known as clients, over a network.
* There are different types of servers such as web servers, mail servers, and virtual servers. These specialize in specific types of data or specific processing capabilities. They’re all involved in providing the content that you request or the services that you need to perform.
* Servers are connected directly to the Internet and the web pages (content) are stored on that server’s hard drive.



# Servers Identifier

As we have seen previously, each server contains the files of a specific website. But how do we differentiate between one website and another?  
Every server that contains the website files has a unique identifier (e.g.: The identifier of www.google.com is “172.217.0.0”, www.facebook.com is “31.13.86.36”)

These numbers are called IP Address.

Each machine (server or client) connected on the Internet has a unique IP Address that represents it.  
If you are using Windows as your OS, open the command line (invite de commande) and type ipconfig then press **Enter**.  
A lot of information popped up, right?  
Those numbers and addresses allow your computer to connect to the Internet. In the line where **IP Address IPv4** is written you will find yours.

We'll discover more about IP Address in the next skill.

# DNS Server

The DNS (Domain Name System) server is a service where its main function is to translate a domain name (for example facebook.com) into an IP address. It is the backbone of the world wide web.

The DNS server acts as a directory that a computer consults when accessing another computer over a network.

In other words, the DNS server is a service that makes it possible to associate a website (or a connected computer or a server) with an IP address, just as a smartphone’s contact list makes it possible to associate a telephone number with a person’s name.



Protocol

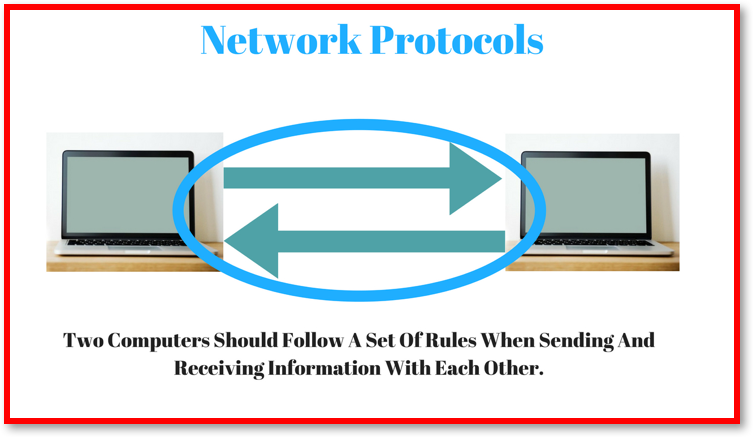
Network protocols are a set of rules, conventions, and data structures that dictate how devices exchange data across networks.

Network protocols control all aspects of network communication from sending and receiving messages, to formatting files for different types of messages.

Similar to the way that speaking the same language simplifies communication between two people, network protocols make it possible for devices to interact with each other because of predetermined rules built into devices’ software and hardware.

A network protocol will specify, for example:

* The format of data packets.
* The addressing system.
* Error-checking procedures used.



HTTP Protocol

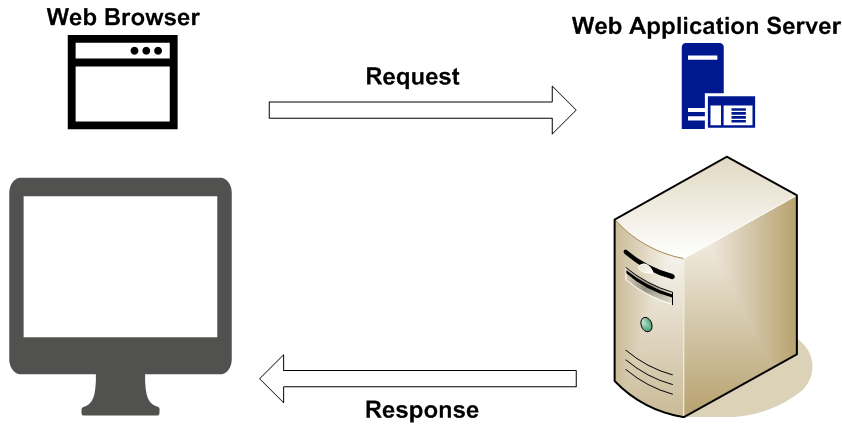
HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the Web and this protocol defines how messages are formatted and transmitted. It also determines what actions Web servers and browsers should take in response to various commands.

For example, when you enter a URL in your browser, this sends an HTTP command to the Web server asking it to fetch and transmit the requested Web page.

The HTTP request contains different types of methods to pick out the desired action from the server.

The most common methods are:

* **GET**: Retrieves data from the server.
* **POST**: Submits data to the server.
* **PUT**: Updates data that’s already on the server.
* **DELETE**: Deletes data from the server.



# IP Address

Stands for "Internet Protocol." IP provides a standard set of rules for sending and receiving data over the Internet. It allows devices running on different platforms to communicate with each other as long as they are connected to the Internet.

To be recognized by other devices, Internet-connected hosts must have an **IP address**. This may be either an IPv4 or IPv6 address. Either way, it uniquely defines a device on the Internet.

The Internet Protocol also provides basic instructions for transferring packets between devices. However, it does not establish the connection or define the ordering of the packets transmitted. These aspects are handled by the Transmission Control Protocol (TCP), which works in agreement with the Internet Protocol to transfer data between systems on the Internet.  
For this reason, connections between Internet-connected systems are often called "TCP/IP" connections.



TCP/IP

Stands for "Transmission Control Protocol". TCP is a fundamental protocol within the Internet protocol suite. it's a collection of standards and guidelines that allows systems and digital computers to communicate over the Internet.

The TCP/IP is categorized as a "transport layer" protocol since it creates and maintains connections between hosts.

* The Internet is a packet-switched network, in which information is broken down into small packets, sent individually over many different routes at the same time, and then reassembled at the receiving end.
* TCP is the component that collects and reassembles the packets of data, while IP is responsible for making sure the packets are sent to the right destination.

The two protocols are commonly grouped together and referred to as TCP/IP.

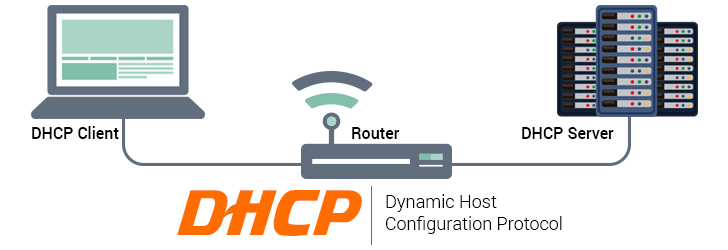
# DHCP

An acronym for "Dynamic Host Configuration Protocol.". DHCP is a protocol that automatically assigns a unique IP address to each device that connects to a network.

With DHCP, there is no need to manually assign IP addresses to new devices. Therefore, no user configuration is necessary to connect to a DHCP-based network.

Because of its ease of use and widespread support, DHCP is the default protocol used by most routers and networking equipment.

DHCP works in the background when you connect to a network. Your device is considered a client and the router is the server, the time it takes to connect via DHCP depends on the type of router and the size of the network, but it usually takes around three to ten seconds. DHCP works the same way for both wired and wireless connections. That means desktop computers, tablets, and smartphones can all connect to a DHCP-based network at the same time.



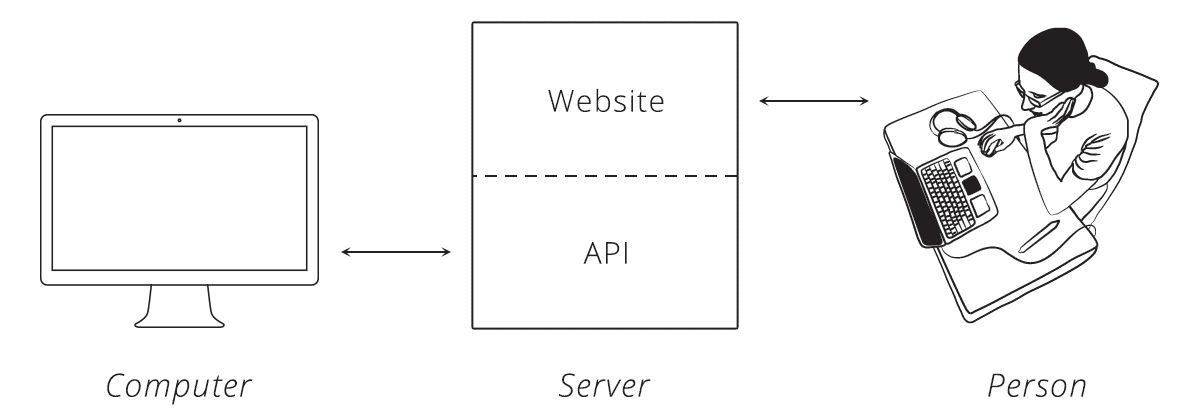
# Introduction

API is the acronym for Application Programming Interface. It is a software intermediary that allows two applications to speak to each other.

An **API** is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.

APIs serve numerous purposes. Generally, they can simplify and speed up software development.

Whenever you use an application like Facebook, send an instant message, or check the weather on your phone, you're using an API.

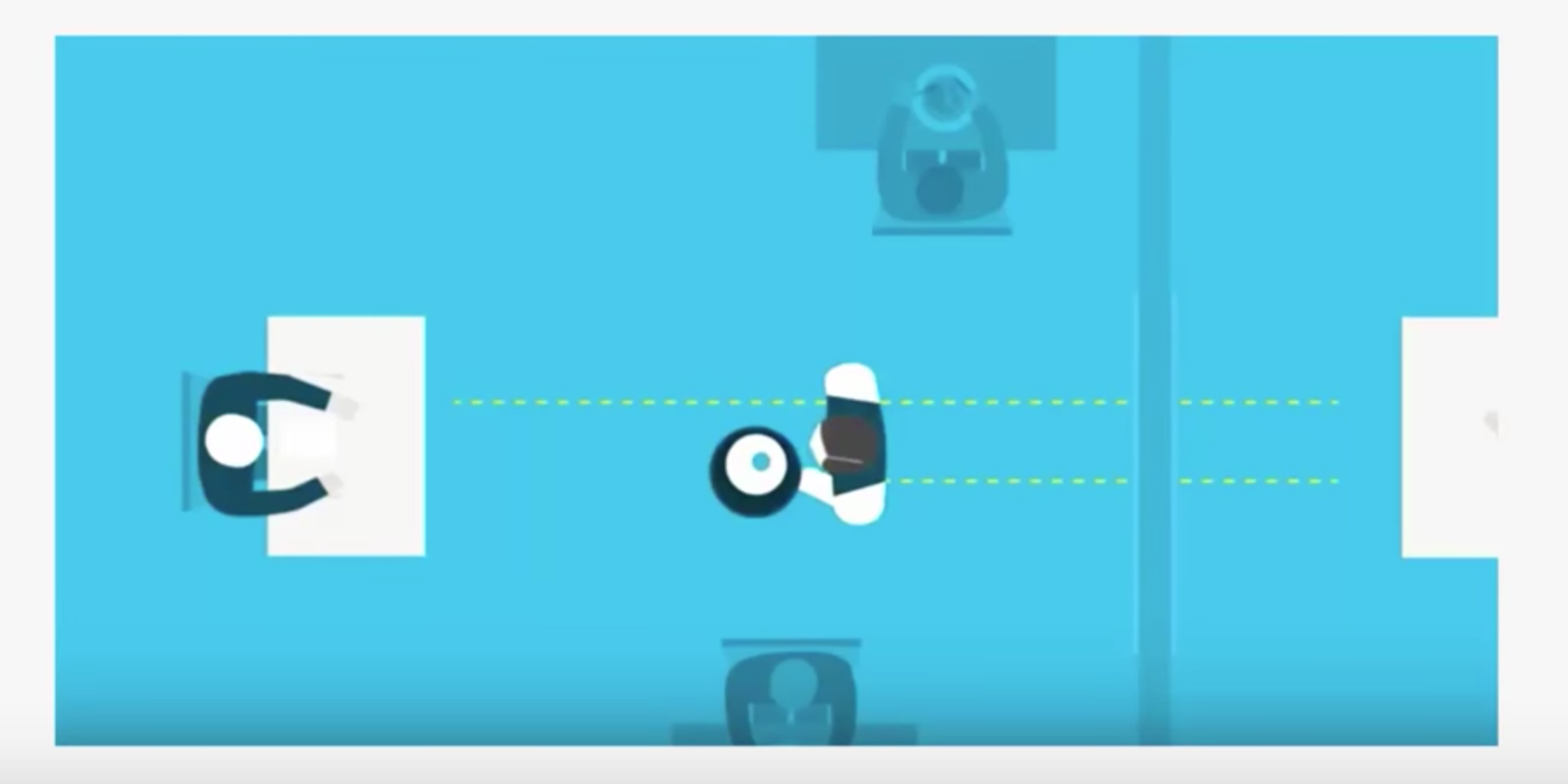


# What is an API?

When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. Afterward, the server retrieves this data, interprets it, performs the necessary actions, and returns it to your phone. The application then interprets this data and presents the desired information to you in an understandable way. That's what an API is. All of those actions are done through an API.

To better explain this, let's take a familiar example.

Imagine you are sitting at a table in a restaurant with a menu and you’re about to order. The kitchen is the part of the "system" that will prepare your order. The missing link here is the one that will communicate your order to the kitchen and deliver your food to your table. This is where the server or API comes in. The waiter is the messenger (an API) who takes your request or order and tells the kitchen (the system) what to do. Then the server brings you the answer; in this case, it's food.



What Are The Advantages of APIs?

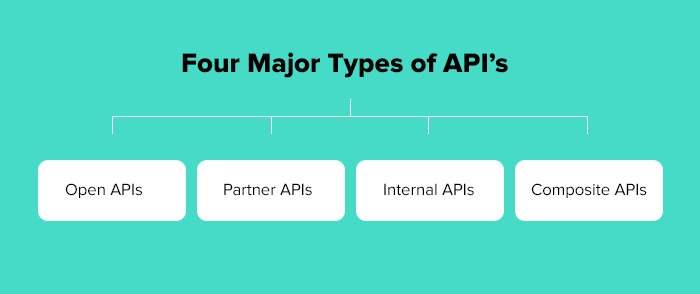
"*There is no information exchange without an API. There is no digital transformation without an API. The Open API is the real revolution!*" Says Bertrand Lafforgue.  
Microsoft Commercial Data Platform Director.

* **Automation:** with APIs, computers rather than people can manage the work. Using APIs, agencies can update workflows to make them faster and more productive.
* **Application:** APIs can access application components. The delivery of services and information is more flexible.
* **More scope:** Through the use of an API, an application layer can be created which also can be used to distribute information and services to new users. These APIs can be customized to create personalized user experiences.
* **Unlimited access to Data:** an API allows all public information that is generated by the government for general use to be accessible to all citizens without any exclusivity.
* **Efficiency:** when access is provided to an API, the generated content can be published automatically and is available for each channel. It makes it easier to share and distribute it.
* **Integration:** APIs make it easier to embed content from any site or application. This ensures smoother information flow and an integrated user experience.
* **Personalization:** Thanks to APIs, any user or company can personalize the content and services they use the most.
* **Adaptation:** Needs evolve and APIs allow you to anticipate changes. When working with this technology, data transfer is easily supported and the information is examined more closely.

In short, APIs are developer-friendly, easily accessible, and incredibly-flexible.

# API Types

Web APIs are APIs that can be accessed using the HTTP protocol.  
This API class is the most common.  
The API defines endpoints, valid requests, and response formats.  
Web APIs include the APIs used to communicate with the browser (see list). They may be services such as web notifications and web storage.



Web Service APIs

Besides the main web APIs, there are also web service APIs :

* SOAP
* XML-RPC
* JSON-RPC
* REST: For example, Twitter's [REST APIs](https://dev.twitter.com/rest/public) provide programmatic access to read and write data using which we can integrate twitter's capabilities into our own application.

*In the following course, we'll talk more about REST API.*

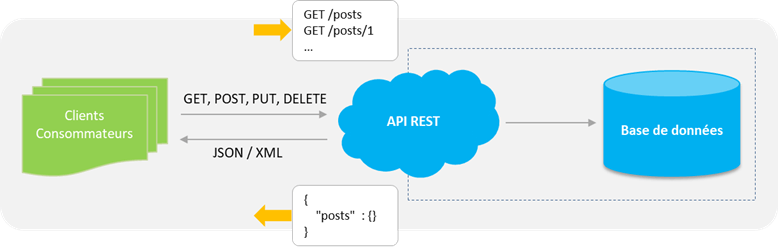
REST API

REST (representational state transfer) is a very popular web API architecture. REST API is an API that must have certain architectural constraints or principles including:

* **Client-server architecture:** the interface is separated from the backend and data storage. This allows for flexibility and for different components to evolve independently.
* **Statelessness:** no client context is stored on the server between requests.
* **Cacheability:** clients can cache responses, so a REST API response must explicitly state whether it can be cached or not.
* **Layered system:** the API will work whether it is communicating directly with a server or through an intermediary such as a load balancer.

The REST architecture uses the original instructions of the HTTP protocol.

**Rule # 1:** the URI 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. Although they have names, these methods are often called HTTP verbs. Each of them implements a different semantic but some common functionalities can be shared by different methods.

**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.

# Definition of a database

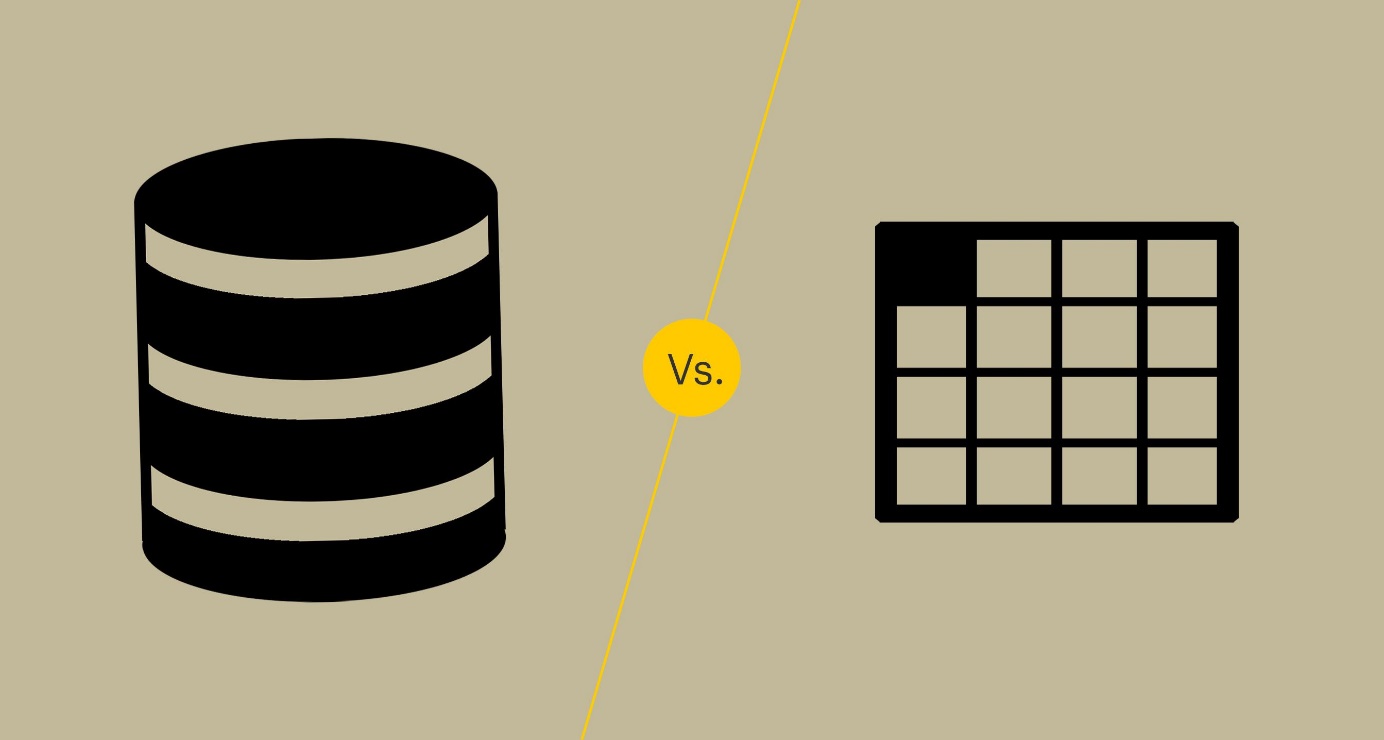
Database are a systematic collection of data.  
Databases support storage and manipulation of data.  
Databases simplify data management.

Let's take Facebook as an example. The social media platform needs to store, manipulate and present data to its members. Not only that, it needs to provide data to the whole network of friends and that includes their activities, their messages, their likes and a lot more. If we store all this data in files, it will be next to impossible to utilize it.

Fortunately, this is where Databases come into play.



Why Do We Need a Database ?



Today, companies have to manage large volumes of data, while ensuring data integrity, security, and accuracy.

Spreadsheets are great tools for making calculations and inputting data. But if there’s a requirement to store substantial amounts of data such as customers’ information, employee records, or business inventory, a more efficient data management tool could be the better option.  
In today’s digital age, substituting spreadsheets with databases is a sure-fire way for business growth.

Types of Databases:

There are many different types of databases. The best database for a specific organization depends on the organization’s purposes, plans, and projects.

These are the database types:

* Relational databases.
* Object-oriented databases.
* Distributed databases.
* Data warehouses.
* NoSQL databases.
* Graph databases.
* OLTP databases.

Database History :

The history of databases dates back to the 1960s, with the appearance of network databases and hierarchical databases. In the 1980s, it was the era where object-oriented databases have appeared. Today, the number-one databases are SQL, NoSQL and cloud databases.

In this skill, we’ll talk more about SQL and NoSQL databases.

If you want to read more:

<https://medium.com/@rpolding/databases-evolution-and-change-29b8abe9df3e>

# hat is SQL?

SQL stands for "Structured Query Language".

It's a language that allows communication with databases to properly manage its data.

### **Is SQL a Programming Language?**

* Yes, SQL is a language. It offers looping, logic directives, variables, and so on. However, it’s not a language in the same sense as Java or C++.
* SQL may not be a language compared to Java or C#, but it is a language nonetheless.
* According to W3Schools: “SQL is a standard language for storing, manipulating and retrieving data in databases.”



# How to Use SQL?

While an application might be programmed in a language like Python, PHP, or Ruby, databases are not designed to recognize these languages. Historically, databases recognize only SQL (although, this has changed significantly in recent years).  
Similar to programming languages, SQL has its own markup. This makes it a priority for a programmer to learn SQL markup before using it effectively.  
Besides markup, another feature unique to database programming is the concept of tables. A database may be represented as a number of tables. Each table has its own number of columns and rows and represents a set of data.

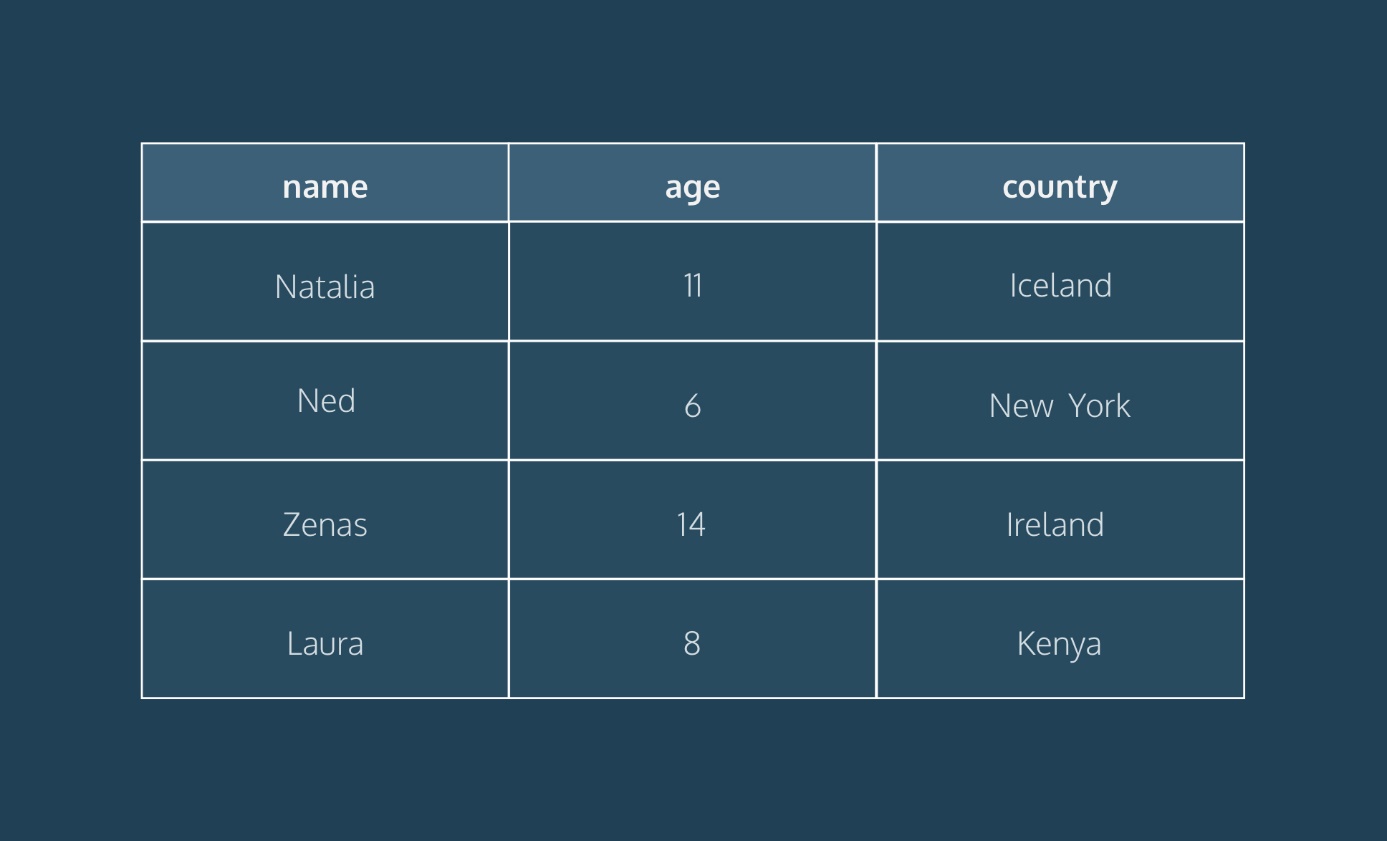
Example of SQL database management

Imagine a library. We could create a database that stores data about students in the GoMyCode community. In this case, we would need only one table, this table would allow us to store all the information we need.  
There are a few frequently used SQL commands you should be familiar with. When working with databases, a programmer might write commands such as:

* **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.

These are just the most common commands. The more complicated the database is, the more commands you as the programmer will need to use.

* Example of a database table :

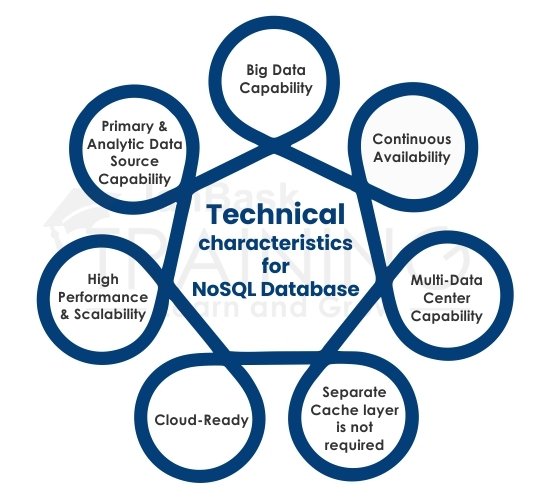


# What is NoSQL?

A NoSQL, which stands for “not only SQL,” originally referring to non SQL or non relational is a database that provides a mechanism for storage and retrieval of data.

NoSQL is an approach to database design that provides flexible schemas for the storage and retrieval of data beyond the traditional table structures found in relational databases.  
While NoSQL databases have existed for many years, it has recently become more popular in the era of cloud, big data, and high-volume web and mobile applications. They are chosen today for their attributes around scale, performance and ease of use.

The most common types of NoSQL databases are key-value, document, column and graph databases.



Examples of NoSQL Databases

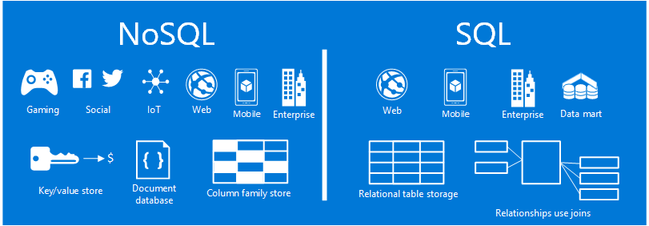
Many NoSQL databases were designed by technology companies like Google, Amazon, Yahoo, and Facebook to provide more effective ways to store content or process data for huge websites. Some of the most popular NoSQL databases include the following:

* **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?

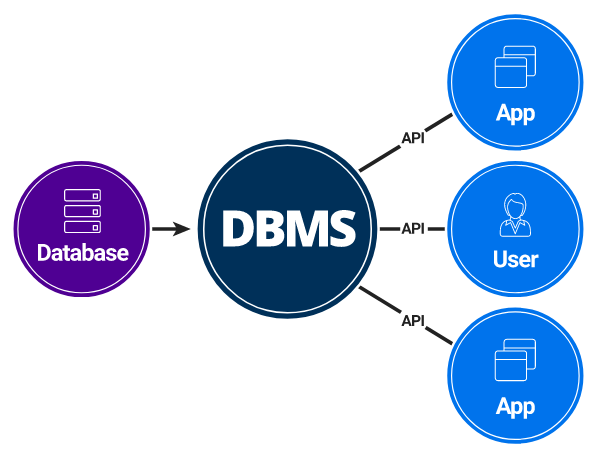
The fundamental difference between SQL and NoSQL is not that complicated. Each has a different philosophy for how data should be stored and retrieved.

1. SQL databases are relational, NoSQL are non-relational.
2. SQL databases use structured query language and have a predefined schema. NoSQL databases have dynamic schemas for unstructured data.
3. SQL databases are vertically scalable, NoSQL databases are horizontally scalable.
4. SQL databases are table based, while NoSQL databases are document, key-value, graph or wide-column stores.
5. SQL databases are better for multi-row transactions, NoSQL are better for unstructured data like documents or JSON.



# 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.

## **The Most Known DBMS:**

* **MySQL**  
  A free and open source DBMS. It is probably the best known DBMS. We will be using it in this section.
* **PostgreSQL**  
  A free and open source like MySQL but, it has more functionality and it is a little less known.
* **SQLite**  
  A free and open source. It’s lightweight but extremely limited in functionality.
* **Oracle**  
  A nonfree database, but it's used by large companies. It is undoubtedly one of the most complete DBMS.
* **Microsoft SQL Server**  
  Microsoft's DBMS. Microsoft markets at least a dozen different editions of Microsoft SQL Server. They’re aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications that have many concurrent users.