

**B** Academy  
**RO**

**Manual**

**Lesson 1**



# Components of the Internet

## Computers and Servers

**Computers:** We all know what these are.

**Servers:** Special powerful computers that store information and send it to us when we request to open websites or receive emails.

## Routers and Switches

Imagine the internet as a huge road network. Routers are like guides that decide which road to send information on to reach the right destination—either to you or to another computer.

You might remember these if you had a home phone and internet, with a router deciding where calls and web information go.

## Network Nodes

These are places where networks meet and exchange information, like intersections or stations where cars change routes or switch to other buses.

Think of them as railroad switches.

# Components of the Internet

## Internet Service Providers

Companies that provide us with internet access, connecting our computers to the internet via cables or wireless signals.

## Data Centers

Large warehouses of information where websites and other online resources are stored. When we want to open a website, the server sends it to our computer so we can view it.

**The internet** is a network where information travels between computers via cables, with routers and servers helping the information reach its destination.

# Process of Loading a Web Page

## 01. Entering the Address

When you enter a web page address in the browser's address bar and press Enter, your browser starts exchanging data with the server where the website is stored. This is done using the HTTP protocol or its secure version, HTTPS.

## 02. DNS Request (Domain Name System)

Before the browser can contact the web server, it needs to know where the server is located. DNS converts the human-readable domain address into a numerical IP address that servers use to communicate.

Every computer has an IP (Internet Protocol) address. A server is essentially a computer, so it also has an IP. Your computer can be a server for a website if you fix the IP with the provider.

## 03. Establishing a TCP Connection (Transmission Control Protocol)

The browser establishes a TCP connection with the server where the website is hosted. TCP is a data transfer protocol that ensures reliable and stable data transmission between computers.

# Process of Loading a Web Page

## 04. Sending an HTTP Request to the Server

Once the browser knows the server's IP address, it sends an HTTP request for the desired web page. The server processes this request.

## 05. Processing the Request on the Server

The server receives the HTTP request and processes it. If the requested page exists and the server can find it, it sends it back to the browser.

## 06. Sending the HTTP Response

The server sends an HTTP response to the browser and starts sending back the requested data—HTML, CSS, JavaScript, and other resources needed to display the web page. It also sends information about the request's status (e.g., successfully completed or an error was found).

# Process of Loading a Web Page

## 07. Receiving Data

The browser receives the HTTP response from the server and starts loading the web page content, including text, images, styles, scripts, and other resources.

## 08. Rendering the Page

The browser processes the received information and starts "rendering" the page, displaying it in the browser window as it should appear to the user.

## 09. Completing the Load

Once all resources are loaded and processed, the browser displays the web page content on your screen. This can include text, images, videos, audio, and other elements that make up a complete website.

After the page is fully loaded, we can interact with it. For example, we can click on a link, fill out a form, or submit a comment. When we perform such actions, the browser sends a new request to the server, and the information exchange process repeats from step 4.

# How Data is Exchanged

## 01. Breaking into "Packets"

When sending an HTTP request, all data is broken into "packets" the size of a byte, each containing information such as:

- Home address
- Destination address
- Number of parts it was broken into
- How to reassemble it

Think of your data as a big family on a trip: everyone takes different cars but agrees to meet at the same place at the same time. To avoid getting lost, each person has the meeting place and their home address written on their hand with a permanent marker. They look up the destination address in a phone book (DNS) and hit the open road (TCP).

# How Data is Exchanged

## 02. How Data Moves Between Servers

Talking about the "information superhighway" in terms of roads and cars is useful as it parallels network communication: traffic routes, exchange operations, maps, intersections, and maintenance closely mirror how data moves on the internet.

Just like driving rules, there are protocols that govern data transmission. For example, road width affects car size, parking spaces must be a certain number in specific places, correct types of street signs are needed, and proper highway access and road signs tell drivers which side of the road to drive on.

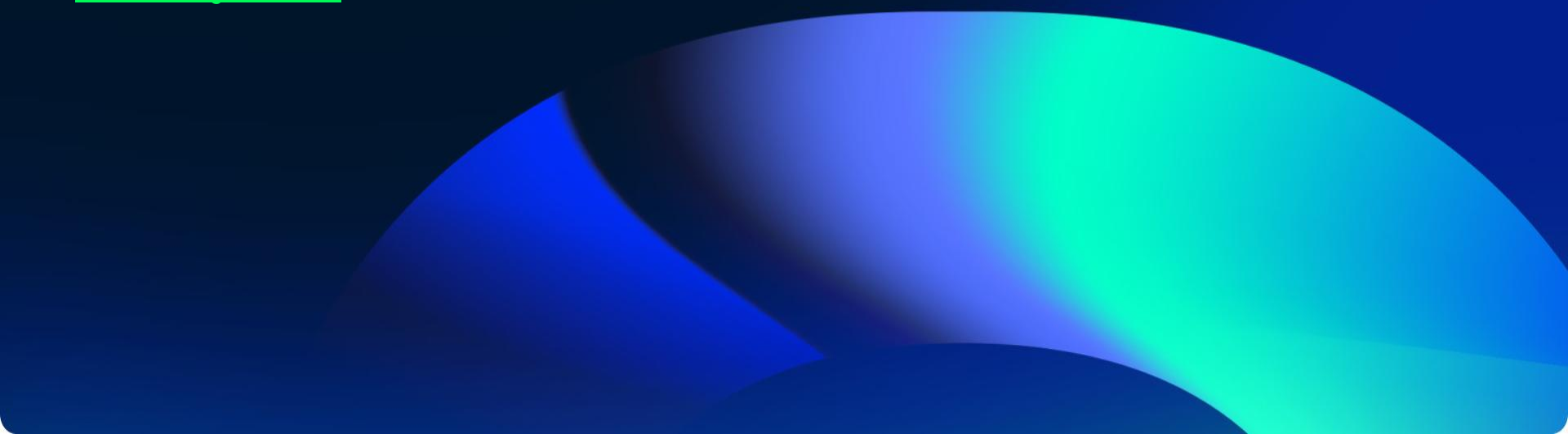
Similarly, technical rules dictate how data is transmitted. You don't need to worry about these protocols establishing and reinforcing technical standards (e.g., how data is broken into packets or how fast packets are reassembled). However, it's helpful to know there's a whole network of protocols maintaining these levels of data transmission.



# Browsers

For developers, the browser engine type is crucial as it determines how the page is rendered. Different browsers have different engines. For example, Safari always uses its own engine, leading to specific bugs. Browsers on different engines may or may not support certain HTML elements, render styles differently, and interpret JavaScript behavior in unique ways. Thus, websites can look different across browsers. For example, Chrome, Firefox, and Safari differ in font rendering and default styles.

[Browser Usage Statistics](#)

Abstract decorative graphic consisting of large, overlapping curved shapes in shades of blue and cyan, located in the bottom right portion of the slide.

# Browser Engines

The most popular browser engine in the world is Blink, used by Chrome and browsers based on it: Opera, Samsung Internet, etc. Google is a key developer of the open-source Blink engine, with active participation from many companies.

**WebKit:** Another popular engine, similar to Blink. In 2013, Blink was copied from WebKit. Google parted ways with Apple, the main developer of WebKit, to work differently. Mobile and desktop Safari browsers now use the new WebKit, while the old version runs on Android browsers up to KitKat.

**Gecko:** Powers Firefox, once very popular, now maintaining a small share but playing a crucial role in web and technology development. Gecko uses its own prefixes (moz — Mozilla) but supports some WebKit properties for compatibility. Full-fledged Firefox on Gecko runs on desktop platforms and Android.

**EdgeHTML:** Used by Edge on all modern Microsoft platforms, including mobile and Xbox. EdgeHTML was copied from Trident or MSHTML, which powered Internet Explorer. This is similar to Blink and WebKit, though the engines have different capabilities.

# Critical rendering path

---

## 01. DOM

Every HTML tag is an object in the document's object model. Nested tags are "children" of the parent element. Text inside a tag is also an object. The DOM tree is like a family tree.

---

## 02. CSSOM

The CSS code is parsed into CSSOM, where each style rule becomes an object.

---

## 03. Render tree

DOM and CSSOM combine to create the Render Tree, defining which elements on the page are visible and how they will look.

---

## 04. Layout

The browser calculates the position and size of each element on the page, similar to arranging furniture in a room.

---

## 05. Paint

The browser "paints" each element on the screen according to their position and style, like drawing a picture on a canvas.

---