mdn web docs

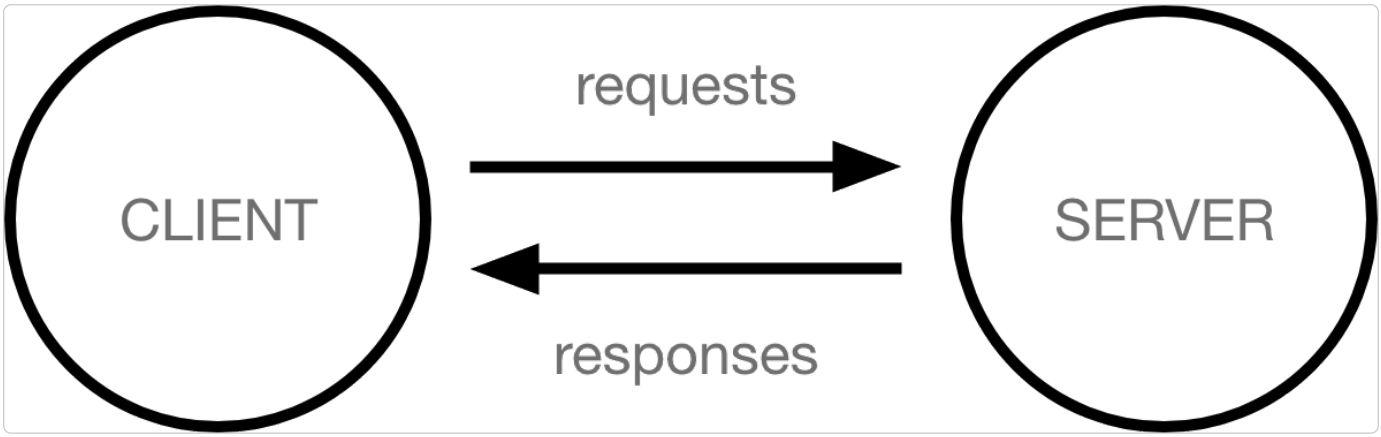
How the web works

How the web works provides a simplified view of what happens when you view a webpage in a web browser on your computer or phone.

This theory is not essential to writing web code in the short term, but before long you'll really start to benefit from understanding what's happening in the background.

Clients and servers

Computers connected to the internet are called clients and servers. A simplified diagram of how they interact might look like this:



- Clients are the typical web user's internet-connected devices (for example, your computer connected to your Wi-Fi, or your phone connected to your mobile network) and web-accessing software available on those devices (usually a web browser like Firefox or Chrome).
- Servers are computers that store webpages, sites, or apps. When a client device wants to access a webpage, a copy of the webpage is downloaded from the server onto the client machine to be displayed in the user's web browser.

The other parts of the toolbox

The client and server we've described above don't tell the whole story. There are many other parts involved, and we'll describe them below.

For now, let's imagine that the web is a road. On one end of the road is the client, which is like your house. On the other end of the road is the server, which is a shop you want to buy something from.

In addition to the client and the server, we also need to say hello to:

- Your internet connection: Allows you to send and receive data on the web. It's basically like the street between your house and the shop.
- TCP/IP: Transmission Control Protocol and Internet Protocol are communication protocols that define how data should travel across the internet. This is like the transport mechanisms that let you place an order, go to the shop, and buy your goods.

In our example, this is like a car or a bike (or however else you might get around).

- DNS: Domain Name System is like an address book for websites. When you type a web address in your browser, the browser looks at the DNS to find the website's IP address before it can retrieve the website. The browser needs to find out which server the website lives on, so it can send HTTP messages to the right place (see below). This is like looking up the address of the shop so you can access it.
- HTTP: Hypertext Transfer Protocol is an application protocol that defines a language for clients and servers to speak to each other. This is like the language you use to order your goods.
- Component files: A website is made up of many different files, which are like the different parts of the goods you buy from the shop. These files come in two main types:
  - Code files: Websites are built primarily from HTML, CSS, and JavaScript, though you'll meet other technologies a bit later.
  - Assets: This is a collective name for all the other stuff that makes up a website, such as images, music, video, Word documents, and PDFs.

So what happens, exactly?

When you type a web address into your browser (for our analogy that's like walking to the shop):

1. The browser goes to the DNS server, and finds the real address of the server that the website lives on (you find the address of the shop).
2. The browser sends an HTTP request message to the server, asking it to send a copy of the website to the client (you go to the shop and order your goods). This message, and all other data sent between the client and the server, is sent across your internet connection using TCP/IP.
3. If the server approves the client's request, the server sends the client a "200 OK" message, which means "Of course you can look at that website! Here it is", and then starts sending the website's files to the browser as a series of small chunks called data packets (the shop gives you your goods, and you bring them back to your house).

4. The browser assembles the small chunks into a complete web page and displays it to you (the goods arrive at your door — new shiny stuff, awesome!).

Order in which component files are parsed

When browsers send requests to servers for HTML files, those HTML files often contain `<link>` elements referencing external CSS stylesheets and `<script>` elements referencing external JavaScript scripts. It's important to know the order in which those files are parsed by the browser as the browser loads the page:

- The browser parses the HTML file first, and that leads to the browser recognizing any `<link>` -element references to external CSS stylesheets and any `<script>` -element references to scripts.
- As the browser parses the HTML, it sends requests back to the server for any CSS files it has found from `<link>` elements, and any JavaScript files it has found from `<script>` elements, and from those, then parses the CSS and JavaScript.
- The browser generates an in-memory DOM tree from the parsed HTML, generates an in-memory CSSOM structure from the parsed CSS, and compiles and executes the parsed JavaScript.
- As the browser builds the DOM tree and applies the styles from the CSSOM tree and executes the JavaScript, a visual representation of the page is painted to the screen, and the user sees the page content and can begin to interact with it.

DNS explained

Real web addresses aren't the nice, memorable strings you type into your address bar to find your favorite websites. They are special numbers that look like this: `63.245.215.20` .

This is called an IP address, and it represents a unique location on the web. However, it's not very easy to remember, is it? That's why Domain Name Servers were invented. These are special servers that match up a web address you type into your browser (like "mozilla.org") to the website's real (IP) address.

Websites can be reached directly via their IP addresses. You can use a DNS lookup tool to find the IP address of a website.

Packets explained

Earlier we used the term "packets" to describe the format in which the data is sent from server to client. What do we mean here? Basically, when data is sent across the web, it is sent in thousands of small chunks. There are multiple reasons why data is sent in small packets. They are sometimes dropped or corrupted, and it's easier to replace small chunks when this happens. Additionally, the packets can be routed along different paths, making the exchange faster and allowing many different users to download the same website at the same time. If each website was sent as a single big chunk, only one user could download it at a time, which obviously would make the web very inefficient and not much fun to use.

See also

- How the Internet works
- HTTP — an Application-Level Protocol
- HTTP: Let's GET It On!
- HTTP: Response Codes

Credit

Street photo: Street composing , by kevin digga .

This page was last modified on Feb 16, 2023 by MDN contributors.