What happens when you type a URL into your browser and press enter:

1. You type a URL in your browser and press Enter
2. Browser looks up IP address for the domain
3. Browser initiates TCP connection with the server
4. Browser sends the HTTP request to the server
5. Server processes request and sends back a response
6. Browser renders the content

Step: 1

Websites are collections of files, often HTML, CSS, Javascript, and images, that tell your browser how to display the site, images, and data. They need to be accessible to anyone from anywhere at any time, so hosting them on your computer at home isn't scalable or reliable. A powerful external computer connected to the Internet, called a server, stores these files.

When you point your browser at a URL like https://www.google.com, your browser has to figure out which server on the Internet is hosting the site. It does this by looking up the domain, www.google.com, to find the address.

Each device on the Internet — servers, cell phones, your smart refrigerator — all have a unique address called an IP address. An IPv4 address contains four numbered parts:

Example: 203.0.213.112

But numbers like this are hard to remember! That's where domain names come in. www.google.com is much easier to remember than 203.0.213.112, right?

Imagine having to remember all the phone numbers of your contacts without having the Contacts app on your phone. Your Contacts app lets you look up phone numbers by name.

We do the same on the Internet. The domain name system, or DNS, is like the Contacts app on our phone. DNS helps our browser (and us) find servers on the Internet. We can do a DNS lookup to find the IP address of the server based on the domain name, www.google.com.

Step 2:

After you've typed the URL into your browser and pressed enter, the browser needs to figure out which server on the Internet to connect to. To do that, it needs to look up the IP address of the server hosting the website using the domain you typed in. It does this using a DNS lookup.

Step 3:

Browser initiates TCP connection with the server (transmission control protocol)

Step 4:

Now that the browser has a connection to the server, it follows the rules of communication for the HTTP(s) protocol. It starts with the browser sending an HTTP request to the server to request the contents of the page.

Step 5:

Server processes request and sends back a response.

Step 6:

Once the browser has received the response from the server, it inspects the response headers for information on how to render the resource.

Difference between **display: none | visibility: hidden | opacity: 0**

"display: none", "visibility: hidden", and "opacity: 0" are CSS properties that can be used to hide elements on a web page, but they work in different ways:

1. **display: none**: This property completely removes the element from the document flow. The element is not rendered at all, and its space is collapsed. It's as if the element doesn't exist in the layout. Other elements will fill in the space previously occupied by the hidden element.

2. **visibility: hidden**: This property hides the element while still preserving its space in the layout. The element is not rendered, but the space it occupies is maintained. So, the element is essentially invisible, but it still affects the positioning of other elements around it.

3. **opacity: 0**: This property makes the element completely transparent. The element is still rendered and takes up space in the layout, but it's visually transparent, so its content is not visible. Other elements will be positioned as if the transparent element is still visible.

In summary, "display: none" removes the element from the layout entirely, "visibility: hidden" keeps the space but makes the element invisible, and "opacity: 0" makes the element transparent while still being rendered and occupying space.

Block and Inline Elements in HTML

In HTML, elements can be classified as either block-level or inline-level elements based on how they are displayed and how they interact with other elements. Here's an explanation of block and inline elements along with some examples:

1. Block-level elements:
    - Block-level elements start on a new line and take up the full available width of their parent container by default.
    - They create a "block" of content and stack vertically on top of each other.

    - Examples of block-level elements: `<div>`, `<p>`, `<h1>` to `<h6>`, `<ul>`, `<li>`, `<section>`, `<article>`, `<header>`, `<footer>`, etc.
2. Inline-level elements:
    - Inline-level elements do not start on a new line and only occupy the necessary width of their content.
    - They flow alongside neighbouring elements horizontally.
    - Examples of inline-level elements: `<span>`, `<a>`, `<strong>`, `<em>`, `<img>`, `<input>`, `<label>`, `<button>`, etc.

It's important to note that the default display behaviour of elements can be changed using CSS properties like `display`. For example, you can change a block-level element to behave like an inline-level element or vice versa by modifying its `display` property in CSS.

Image Tag in HTML

The `<img>` tag in HTML is used to insert an image into a web page. It is a self-closing tag, meaning it does not require a closing tag.

Important `<img>` tag attributes :

- **src**: This attribute specifies the URL or file path of the image to be displayed.
- **alt**: This attribute provides alternative text for the image, which is displayed if the image cannot be loaded or for accessibility purposes.
- **width and height**: These attributes define the width and height dimensions of the image in pixels. They are optional, but it's generally recommended to include them to reserve space for the image and prevent layout shifting.

Example:

```
<img src="path/to/image.jpg" alt="Description of image" width="300" height="200">
```

In the example above, an image with a source of "path/to/image.jpg" is inserted. The alt attribute provides a description of the image that will be displayed if the image fails to load or for accessibility purposes. The width and height attributes define the dimensions of the image, specifying a width of 300 pixels and a height of 200 pixels.

Note that you need to replace "path/to/image.jpg" with the actual path or URL of the image file you want to display on your web page.