# Validating Document

Validating documents in HTML is an important step to ensure that your HTML code follows the correct syntax and adheres to web standards. Validation helps identify errors and inconsistencies in your markup, which can improve the overall quality and compatibility of your web pages.

There are several tools available to validate HTML documents. One commonly used tool is the W3C Markup Validation Service, provided by the World Wide Web Consortium (W3C). Here is how you can use it:

1. Open your web browser and go to the W3C Markup Validation Service website: https://validator.w3.org/.
2. Choose the "Validate by Direct Input" option.
3. Open the HTML document you want to validate in a text editor or code editor.
4. Copy the entire content of the HTML document.
5. Go back to the W3C Markup Validation Service website and paste the copied HTML code into the text area provided.
6. Click on the "Check" button.

The validator will process your HTML code and display the validation results. It will highlight any errors or warnings found in your code, along with the specific line numbers where the issues occur. You can click on each error or warning to get more details and suggestions for how to fix them.

In addition to the W3C Markup Validation Service, there are other online validators available that you can use, such as the HTML5 Validator provided by the Nu Html Checker (https://validator.nu/) or the HTML Validator extension for various web browsers.

It is a good practice to validate your HTML documents during the development process to catch any mistakes early on and ensure your code meets the required standards.

# URI and URL

URI stands for Uniform Resource Identifier, and URL stands for Uniform Resource Locator. While the two terms are often used interchangeably, they have slightly different meanings:

**URI (Uniform Resource Identifier):**

A URI is a string of characters that identifies a specific resource. It can be used to identify various types of resources, not just web pages. A URI provides a unique identifier for a resource and can be used to locate or access that resource. URIs can include both URLs and URNs (Uniform Resource Names).

Example of a URI:

URI: urn:isbn:978-3-16-148410-0

Explanation: This URI identifies a specific book using its ISBN number. The "urn" scheme indicates that it is a Uniform Resource Name.

**URL (Uniform Resource Locator):**

A URL is a specific type of URI that provides the means to access a resource on the internet. It includes the protocol, domain name, and optional path or additional parameters necessary to locate the resource. URLs are commonly used to refer to web pages, images, files, and other resources accessible via the internet.

Example of a URL:

URL: https://www.example.com/index.html

Explanation: This URL specifies the protocol (https://), the domain name (www.example.com), and the path to the resource (/index.html) which is an HTML webpage.

In summary, a URI is a string of characters that identifies a resource, while a URL is a specific type of URI that provides the means to locate and access a resource on the internet. URLs are a subset of URIs and are commonly used to identify web resources.

# URI

URI (Uniform Resource Identifier) is a string of characters that identifies a resource. It provides a unique identifier for a resource and can be used to locate or access that resource. URIs are used to identify various types of resources, not just web pages. They are composed of a scheme, followed by a colon, and a scheme-specific part.

The structure of a URI is typically represented as follows:

scheme:[//authority]path[?query][#fragment]

Here is a breakdown of the components of a URI:

1. Scheme: The scheme indicates the protocol or scheme used to access the resource. It defines the rules for interpreting the rest of the URI. Common schemes include "http," "https," "ftp," "mailto," "file," etc.
2. Authority: The authority component is optional and typically includes the domain name or network location associated with the resource. It can include subdomains, port numbers, or usernames/passwords for certain schemes.
3. Path: The path component specifies the specific location or path to the resource on the server. It can be hierarchical, representing directories or folders, or it can point to a specific file or resource.
4. Query: The query component is optional and follows a question mark. It contains additional parameters or data that may be needed to retrieve or interact with the resource. The query parameters are separated by ampersands (&).
5. Fragment: The fragment component is optional and follows a hash symbol (#). It refers to a specific part or section within the resource itself, such as an anchor or identifier within an HTML page.

Example URI:

mailto:john@example.com

Explanation: This URI represents an email address using the "mailto" scheme.

https://www.example.com/path/to/resource?param1=value1&param2=value2#section2

Explanation: This URI represents a web resource with the "https" scheme, the domain "www.example.com," the path "/path/to/resource," query parameters "param1" and "param2" with corresponding values, and a fragment identifier "section2."

In summary, a URI is a string of characters that serves as a unique identifier for a resource. It can be used to locate and access various types of resources, and it consists of a scheme, authority, path, query, and fragment components. URLs are a specific type of URI that are commonly used to identify web resources.

# URL

URL (Uniform Resource Locator) is a specific type of URI (Uniform Resource Identifier) that provides the means to locate and access a resource on the internet. It is a string of characters that identifies the address or location of a resource, such as a web page, an image, a file, or an API endpoint. URLs are widely used to navigate and access resources on the web.

A URL consists of several components that specify the location and access method for the resource. The general structure of a URL is as follows:

scheme://host:port/path?query#fragment

Here is a breakdown of the components of a URL:

1. Scheme: The scheme defines the protocol or access method to be used. Common schemes include "http," "https," "ftp," "mailto," "file," etc. The scheme indicates how the resource should be accessed.
2. Host: The host component specifies the domain name or IP address of the server hosting the resource. It identifies the network location where the resource is hosted.
3. Port: The port component, which is optional, specifies the specific port number to use when establishing a connection to the server. If no port is specified, it defaults to the standard port associated with the scheme (e.g., port 80 for HTTP).
4. Path: The path component represents the specific location or path on the server where the resource is located. It may include directories, folders, or a filename, depending on the resource's structure.
5. Query: The query component, which is optional, allows passing additional parameters or data to the server. It typically appears after a question mark (?) and consists of key-value pairs separated by ampersands (&).
6. Fragment: The fragment component, also optional, refers to a specific section or anchor within the resource itself. It appears after a hash symbol (#) and is often used with HTML documents to navigate to a specific part of the page.

Example URL:

https://www.example.com/products/index.html?category=electronics#section2

Explanation: This URL represents a web page with the "https" scheme, the domain "www.example.com," the path "/products/index.html," a query parameter "category" with the value "electronics," and a fragment identifier "section2."

In summary, a URL is a specific type of URI that provides the address or location of a resource on the internet. It includes the scheme, host, optional port, path, optional query parameters, and optional fragment identifier. URLs are widely used for navigating the web and accessing various resources.

# Absolute and Relative URL

URL stands for Uniform Resource Locator, and it is a reference to a web resource, such as a webpage, an image, or a document, on the internet. URLs can be categorized into two types: relative and absolute.

**Relative URL:**

A relative URL is a URL that is relative to the current page or the current location within a website. It specifies a resource's location relative to the current context. Relative URLs are often used when linking within the same website or when creating links in a local environment. They do not contain the complete address of the resource but only provide a partial path.

Example:

Let's say you have a webpage with the URL "https://example.com/products/index.html," and you want to link to an image file located in a subdirectory called "images" on the same website. The relative URL for the image would be "images/pic.jpg". The browser would interpret this relative URL as "https://example.com/products/images/pic.jpg" since it appends the relative path to the base URL.

**Absolute URL:**

An absolute URL provides the complete address or location of a resource on the internet. It includes the scheme (such as "http" or "https"), the domain name, and the path to the resource. Absolute URLs are commonly used when linking to external websites or resources.

Example:

An absolute URL for the same image mentioned above would be "https://example.com/products/images/pic.jpg". It provides the full path to the image, including the protocol (https://), the domain name (example.com), and the path to the resource (/products/images/pic.jpg).

In summary, a relative URL is a partial URL that is relative to the current page or location, while an absolute URL is a complete URL that provides the full path to a resource on the internet, including the protocol, domain name, and path.