
Web programming-1

**Software engineering, Digital marketing,
network and security & Web and graphics
designs.**

By VERNYUY BASILE

TIMELINE/Week 2

ADRESSING OF WEB DOCUMENTS

1. Introduction to URLs (Uniform Resource Locators)

2. Types of URLs: Relative and absolute addressing of web documents

3. Working with directories and file structures

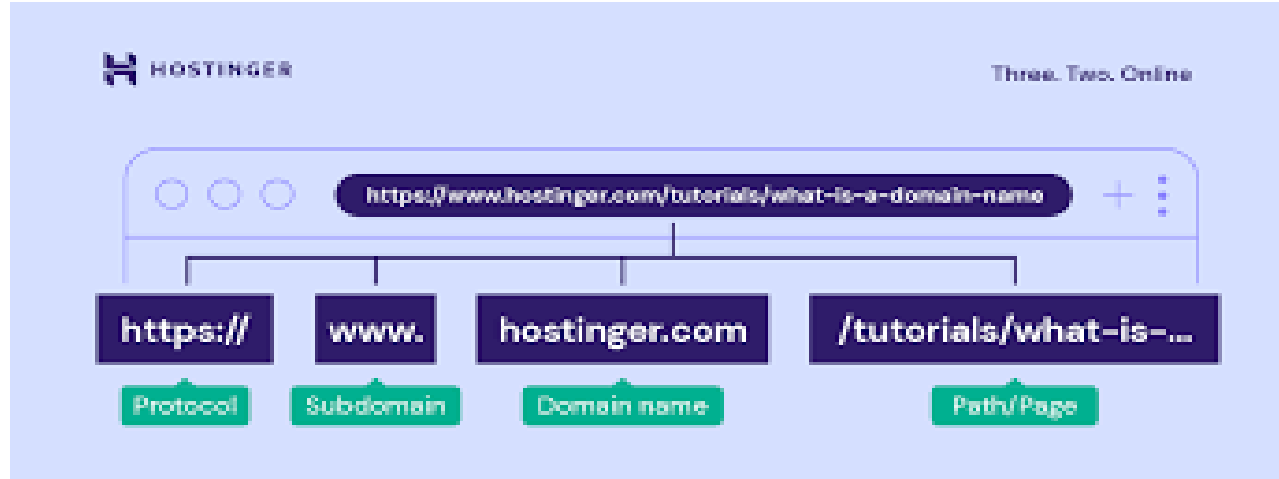
1. Introduction to URLs (Uniform Resource Locators)

What is a URL (Uniform Resource Locator)?

A URL stands for Uniform Resource Locator. It is a unique identifier used to locate a resource on the Internet. URLs are also referred to as web addresses. URLs are the standard addressing system of the www. A complete URL provides the web client with all the information it needs to contact a server and make a request for information . URLs divided into three essential parts:

Parts of URL

- **A. Protocol (http ://):** The protocol is the method used to access the resource. For example, http:// indicates that the server to be connected is a www server.
- **B. Domain name (www.beginners.com):** The domain name is the address of the website where the resource is located. In this example, beginners.com is the domain name.
- **C. The path / Resource name (index.html):** The path is the specific location of the resource on the website. The third piece of information is the path to the actual document requested. In this example, the URL indicates that the document is in the system directory and is named software.htm.



Examples of
URLs in pictures

2.Types of URLs: Relative and absolute url or addressing of web documents

Types of URLs

1. Absolute URLs: Absolute URLs contain all of the information needed to locate a resource, including the protocol, domain name, and path. For example, the following URL is an absolute URL: Absolute URLs provide all the available information to find the location of a page. For example <https://www.google.com/>

2. Relative URLs: A relative URL is a URL that only includes the path. The path is everything that comes after the domain, including the directory. A relative URL locates a resource using an absolute URL as a starting point. Here's an example of a relative URL: </about/team/>

Absolute URL vs Relative URL


`http://www.domain.com/`

in


`hello`

HTML

Hypertext – Absolute & Relative Links

Absolute Links

` UW-Milwaukee `
also called an external link

Relative Links

` Link to admissions page `

Relative links do not require the protocol and domain name

These links are “related” to the current page (The page you want to link to is within the same domain as the document you are linking from)

*Examples of
absolute and
relative URLs*

3. Working with Directories and File Structures.

1.What is a web directory?

A web directory is a website or webpage that includes a list of links to other sites that fit into a relevant topic. These directories were especially prevalent in the early days of the internet, before individuals could simply find what they were looking for via a search engine.

Web directories were one of the first ways that people found websites on the internet. Before search engines became popular, web directories were the primary way to find new websites.

A web directory would be applicable for any type of industry or business type; these links could all apply to retail, or they could include services like dentists and doctors. In other words, there's no clear and definitive format or topics. Rather, web directories can be broad, making them important for any types of businesses. Examples of some web directories are: DMOZ, Yahoo! Directory, Startpage, Business.gov, USA.gov etc.

Here are some tips for using web directories:

- Use the search bar to find the category or subcategory that you are interested in.
- Browse the categories and subcategories to find websites that look interesting.
- Read the descriptions of the websites before clicking on the links. This will help you to determine if the website is relevant to your interests.

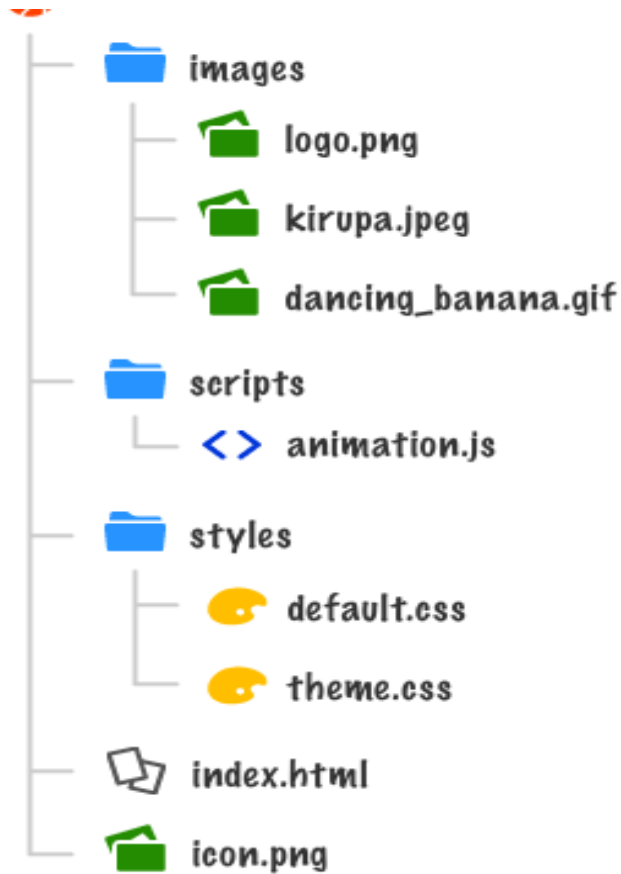
2.What is file structure?

The file structure of a website is the way that files on a particular website are organized. This includes the HTML files, CSS files, JavaScript files, image files, and other files that are needed to display the website.

A well-organized file structure can make it easier to find and manage the files on your website. It can also help to improve the performance of your website.

Here are some tips for creating a good file structure for a website:

- Use logical names for your folders and files. For example, instead of naming a folder "New Folder," name it "Images."
- Use a consistent naming convention for your folders and files. This will make it easier for you to find the files you need, and it will also make your file structure look more organized.
- Use subdirectories to organize your files. For example, you could create a subdirectory for each page on your website.



Typical example of file structures in websites

End
