**An Introduction to HTML** WEEK02

Instructor: Daniel Success



**A brief history on HTML?**

HTML (Hypertext Markup Language) was created in the early 1990s. HTML was first proposed by Tim Berners-Lee, a British computer scientist, in 1989 while working at CERN, the European Organization for Nuclear Research. He wrote the first version of HTML, which was called "HTML Tags," in 1990. He then released it to the public in 1991.

Since then, there have been many different versions of HTML. The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999. Another version, XHTML, was a rewrite of HTML as an XML language. XML is a standard markup language that is used to create other markup languages. Hundreds of XML languages are in use today, including GML (Geography Markup Language), MathML, MusicML, and RSS (Really Simple Syndication). Since each of these languages was written in a common language (XML), their content can easily be shared across applications.

The first publicly available version of HTML was HTML 1.0, which was released in 1993 by the Internet Engineering Task Force (IETF) as an official standard. Since then, there have been many revisions and updates to the language, with the most recent version being HTML5.

**What is HTML?**

HTML is a type of markup language. It “marks up” data within HTML tags, which define the data and describe its purpose on the webpage. It is used to mark up the text for the browser to read and interpret as web page content. It tells the browser which parts are headings, which are paragraphs, and which are links, and the browser displays them as such.

In simple terms, the HTML describes the data to the browser, and the browser then displays the data accordingly.

That’s how the browser knows that

# This is a heading

This is a paragraph, and

*This is a link*

Even though HTML is not a programming language, it’s advisable for anyone who wants to become a programmer to pick up first. It's easy to write your first HTML program. All you need is a web browser and a text editor. There is no need to install compilers or interpreters as you might with Kotlin or Python.

Moreover, you get instant feedback. When a program works, you can see the results of your effort in a web browser immediately. The lessons an aspiring developer learns from HTML will carry forward throughout their programming career. These include:

1. The role of HTML in webpage development;
2. How code translates into software;
3. The importance of program structure;
4. How to troubleshoot code; and
5. The feeling of satisfaction when a software program works.

Developers who start with HTML create a strong foundation to learn other computer languages, such as Kotlin, Dart, C or Python.

# What HTML is not

HTML (Hypertext Markup Language) is not a programming language. It is used to structure and display content on the web. It consists of a series of tags and attributes that define the structure and layout of text, images, and other elements on a webpage. Programming languages such as JavaScript and Python are used to add interactivity and dynamic functionality to web pages.

To discuss why something is or is not a programming language, we must establish a baseline definition of what a computer programming language is. To do that, we must consider what a computer does, and more specifically, what the CPU (Central Processing Unit) does.

At the most basic level, a CPU does three things:

1. Read data out of memory;
2. Perform conditional logic on that data; and
3. Iteratively perform that logic at lightning speed.

HTML contains no programming logic. For anything to qualify as a computer programming language, it must provide a way to program these aspects of a computer. HTML does not support any of these functions. That's why HTML is not a programming language. HTML doesn't allow for logic, control, data storage and retrieval, looping, or anything else that a programming language does, it's a markup language for structuring documents. Swift, JavaScript, Dart, C, Python, Java and Kotlin all qualify as computer programming languages.

1. All these languages provide a way to declare variables and assign values to those variables. This is how computer programming languages manage data. HTML does not use variables in the traditional sense of programming languages. However, it does have attributes that can be used to assign values to elements. These attributes can be used to define the properties of an element, such as its size, position, or behavior. For example, the src attribute in an img tag can be used to specify the source URL of an image. The href attribute in an a tag can be used to specify the destination of a link. The class and id attributes can be used to identify elements and apply styles to them using CSS. It's worth noting that HTML doesn't have any built-in mechanism to change the value of an attribute dynamically. But we can use JavaScript or other programming languages to do that.
2. These languages support conditional if..then..else statements. This is how conditional logic is performed.
3. These languages also support for and while loops. This is how repetitive, iterative logic is performed millions of times per second.

HTML has no equivalent features to the above as HTML can't program a computer.

HTML is not a programming language. It's a markup language. In fact, that is the technology's name: HyperText Markup Language. That self-identified fact alone should settle the debate.

HTML (Hypertext Markup Language) is considered a markup language because it uses tags and attributes to describe the structure and layout of text and other elements on a webpage, rather than describing the specific steps the computer should take to perform a task like a programming language. In HTML, tags are used to enclose text or other elements, and attributes are used to provide additional information about those elements. For example, an h1 tag might be used to enclose a heading, and the class attribute might be used to specify a CSS class that can be used to apply styles to that heading. The tags and attributes in HTML are used to define the structure and layout of a webpage, but do not include programming constructs such as loops or conditionals. They simply describe the presentation of the content and the browser interprets them to display the content accordingly. This separation of presentation and logic is what makes HTML a markup language.

One person could reasonably argue that HTML is a programming language because it's a computer language, structured to tell a computer what to do, specifically how to display a page. "There are rules, and if you break them, it won't do what you want," he said.

Whether HTML is technically a programming language or not doesn't diminish its importance in the world of software development. It doesn't diminish the skills of the people who create amazing websites with HTML. If you love HTML development and the visual nature of website design, don't let the programming language debate diminish your love for the craft. It's just a label. And it's a silly one at that.

And even more, HTML really shines when you use it in conjunction with an actual programming language, such as when using a web framework or library like React. That’s when you can start serving up dynamically created web pages and database applications.

# A Simple HTML Document

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head> <body>

<h1>This is my First Heading</h1>

<p>This is my first paragraph.</p>

</body>

</html>

# Simple Document Explained

● The <!DOCTYPE html> declaration, defines that this document is an HTML5 document.

On the HTML document you have often seen that there is a <!DOCTYPE html> declaration before the <html> tag. HTML <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called the document type declaration (DTD). Technically <!DOCTYPE > is not a tag/element, it is just an instruction to the browser about the document type. It is a null element which does not contain the closing tag, and must not include any content within it. Actually, there are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc. The doctype declaration differs between HTML versions. The HTML 5 doctype declaration is given below.

**Syntax**

<!DOCTYPE html>

Note: It is always a good practice to add a declaration to your HTML documents to enable web browser to recognize that what type of document to expect.

* The <html> element is the root element of an HTML page
* The <head> element contains meta information about the HTML page
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The <h1> element defines a large heading
* The <p> element defines a paragraph

# Common HTML Elements

1. HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading:

1. HTML Paragraphs

HTML paragraphs are defined with the <p> tag.

1. HTML Links

HTML links are defined with the <a> tag:

<a href="https://www.deebuginstitute.com">This is a link</a>

1. HTML Images

HTML images are defined with the <img> tag.

The source file (src), alternative text (alt), width, and height are provided as attributes: <img src="bleach.jpg" alt="deebuginstitute.com" width="104" height="142">

# HTML Elements in View

The HTML **element** is everything from the start tag to the end tag: <tagname>Content goes here...</tagname>

Examples of some HTML elements:

<h1>My First Heading</h1>

<p>My first paragraph.</p>



HTML elements with no content are called empty elements.

The <br> tag defines a line break, and is an empty element without a closing tag.

Example:

<p>This is a <br> paragraph with a line break.</p>

# Nested HTML Elements

HTML elements can be nested (this simply means that elements can contain other elements within them). Pretty much all HTML documents consist of nested HTML elements.

The “Simple HTML document” we looked at earlier, contains four HTML elements:

<html>, <body>, <h1> and <p>

# HTML Attributes

* All HTML elements can have **attributes**
* Attributes provide **additional information** about elements
* Attributes are always specified in **the start tag**
* The Attribute should always be applied with its name and value pair. **Attributes usually come in name/value pairs like: name="value"**
* The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
* You can add multiple attributes in one HTML element, but need to give space between two attributes.

Take a look at this little code snippet:

<img src="deebug\_logo.jpg" alt="deebug" width="104" height="142">

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed.

The <img> tag can also contain the width and height attributes, which specifies the width and height of the image (in pixels):

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

# Headings are Important

* HTML headings are titles or subtitles that you want to display on a webpage.
* HTML headings are defined with the <h1> to <h6> tags.
* <h1> defines the most important heading and <h6> defines the least important heading.
* Search engines use the headings to index the structure and content of your web pages.
* Note: Use HTML headings for headings only. Don't use headings to make text BIG or bold.

Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute, using the CSS font-size property:

<h1 style="font-size:60px;">Heading 1</h1>

# HTML Paragraphs

A paragraph always starts on a new line, and is usually a block of text.

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

You cannot be sure how HTML will be displayed. Large or small screens, and resized windows will create different results.

With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code. The browser will automatically remove any extra spaces and lines when the page is displayed.

# Semantic Elements

A semantic element in HTML is an element that provides meaning or context to the content within it, rather than just describing the visual appearance of the content. In other words, a semantic element describes the purpose or function of the content it contains, rather than just how it should look on the page.

Examples of semantic elements in HTML include:

* <header>: Indicates the beginning of a page or section header.
* <nav>: Represents a section of the page that contains navigation links.
* <article>: Represents a self-contained piece of content, such as a blog post or

news article.

* <section>: Groups related content together.
* <aside>: Represents content that is related to the main content of the page, but

not essential to its meaning.

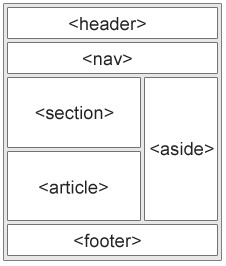
* <footer>: Indicates the end of a page or section footer.

Using semantic elements in HTML helps search engines, screen readers, and other tools to better understand the structure and meaning of your content, which can improve

accessibility and search engine optimization.

Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.

# Semantic Elements Structure



# The <section> element

The <section> element defines a section in a document. This is a somewhat thematic grouping of content, usually with a heading of some sort. Examples of where a <section> element can be used:

Chapters, Introduction, News items, Contact information, etc.

A web page could normally be split into sections for introduction, content, and contact information.

# The <article> element

The <article> element specifies independent, self-contained content. An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site. Examples of where the <article> element can be used:

Forum posts, Blog posts, User comments, Product cards, etc.

# The <header> element

The <header> element represents a container for introductory content or a set of navigational links. A <header> element typically contains:

one or more heading elements (<h1> - <h6>), logo or icon, authorship information

**Note:** You can have several <header> elements in one HTML document. However,

<header> cannot be placed within a <footer>, <address> or another <header> element.

**HTML Block & Inline**

Every HTML element has a default display value, depending on what type of element it is.

There are two display values: block and inline.

# Block-level Elements

* A block-level element always starts on a new line.
* A block-level element always takes up the full width available (stretches out to the left and right as far as it can).
* A block level element has a top and a bottom margin, whereas an inline element does not.

The <div> element is a block-level element:

<div>Hello World</div>

**Block-level Elements in HTML**

[<address>](https://www.w3schools.com/tags/tag_address.asp) [<article>](https://www.w3schools.com/tags/tag_article.asp) [<aside>](https://www.w3schools.com/tags/tag_aside.asp) [<blockquote>](https://www.w3schools.com/tags/tag_blockquote.asp) [<canvas>](https://www.w3schools.com/tags/tag_canvas.asp) [<dd>](https://www.w3schools.com/tags/tag_dd.asp) [<div>](https://www.w3schools.com/tags/tag_div.asp) [<dl>](https://www.w3schools.com/tags/tag_dl.asp) etc.

# Inline Elements

* An inline element does not start on a new line.
* An inline element only takes up as much width as necessary.

<span>Hello World</span>

# Inline Elements in HTML

[<a>](https://www.w3schools.com/tags/tag_a.asp) [<abbr>](https://www.w3schools.com/tags/tag_abbr.asp) [<acronym>](https://www.w3schools.com/tags/tag_acronym.asp) [<b>](https://www.w3schools.com/tags/tag_b.asp) [<bdo>](https://www.w3schools.com/tags/tag_bdo.asp) [<big>](https://www.w3schools.com/tags/tag_big.asp) [<br>](https://www.w3schools.com/tags/tag_br.asp) [<button>](https://www.w3schools.com/tags/tag_button.asp)

**Note:** An inline element cannot contain a block element

# BREAK TAG

In HTML, the <br> tag is used to insert a line break or a new line in a text block. It is a self-closing tag, which means it does not require a closing tag.

Here are some examples of using the <br> tag in HTML:

1. To insert a single line break:

<p>This is the first line.<br>This is the second line.</p> This will display:

This is the first line.

This is the second line.

1. To insert multiple line breaks:

<p>This is the first line.<br><br><br>This is the second line.</p>

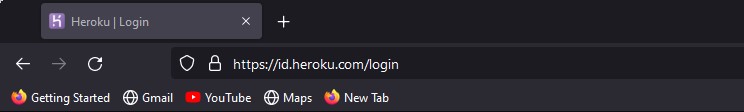
This will display:

This is the first line.

This is the second line.

# The Favicon

A favicon is a small image displayed next to the page title in the browser tab.



## Add a Favicon in HTML

You can use any image as your favicon. You can also create your own favicon on sites like:

* [https://favicon.io](https://favicon.cc/)
* [https://favicon.cc](https://favicon.cc/)
* <https://realfavicongenerator.net/>

**Note:** A favicon is a small image, so it should be a simple image with high contrast.

To add a favicon to your website, either save your favicon image to the root directory of your webserver, or create a folder in the root directory called images, and save your favicon image in this folder. A common name for a favicon image is "favicon.ico".

Next, add a <link> element to your "index.html" file, after the <title> element, like this: <head>

<title>My Page Title</title>

<link rel="icon" type="image/x-icon" href="images/favicon.ico">

</head>

The "rel" attribute specifies the relationship between the HTML document and the linked file, in this case, it's "icon" indicating that the linked file is an icon.

The "href" attribute specifies the location of the file, in this case, it's the local path of the icon

The "type" attribute specifies the MIME type. MIME (Multipurpose Internet Mail Extensions) type is a standard way to identify the type of a file on the internet. It is a string of text that identifies a specific format of data or file, and it's used by web servers to transmit the file correctly to a web browser or other client application. MIME types are important for web development because they ensure that a browser or other application can properly display or interpret the content of a file.

For example, if a web server sends an HTML file with a MIME type of "text/html", a browser will know to interpret the file as an HTML document and render it accordingly. If the server sends a JPEG image with a MIME type of "image/jpeg", a browser will know to display the image rather than trying to interpret it as text. There are many standard MIME types, such as "text/html" for HTML files, "image/jpeg" for JPEG images, "application/pdf" for PDF files, and so on. Developers can also define their own MIME types for custom file formats or data types.

## HTML File Paths

A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files, like: Web pages, Images, Stylesheet and even JavaScript files. There are two main types of file paths:

**1. Relative File Paths**

A relative file path points to a file relative to the current page.

They are often used when referring to files within the same website or directory structure. Relative paths can be categorized into the following types:

* Relative to the Current File: If the file you want to reference is in the same directory as the current HTML file, you can simply use the file name.

For example: <img src="codan-institution.png" alt="logo">

* Relative to the Current Directory: If the file you want to reference is in a subdirectory within the current directory, you can specify the path to the file relative to the current directory.

For example: <img src="images/ codan-institution.png " alt="logo">

* Relative to the Root Directory: If the file you want to reference is in a different directory relative to the root directory of the website, you can use a forward slash (/) at the beginning of the path.

For example: <img src="/images/ codan-institution.png " alt="logo">

* Relative to the Parent Directory: If the file you want to reference is in a parent directory of the current directory, you can use two dots (..) to traverse up the directory structure.

For example: <img src="../codan-institution.png " alt="logo"> refers to a file one level up from the current directory.

**2. Absolute File Paths**

An absolute file path is the full URL to a file. Absolute paths specify the complete URL or file path to the desired resource. They are used when referencing files on external websites or when you need to provide the full path to a file on your server. Absolute paths typically start with a protocol such as http:// or https:// for web resources, or with the file system path for local files. For example:

<img

src="https://www.hollywoodreporter.com/wp-content/uploads/2021/10/Man-of-Steel-Evere tt-H-2021.jpg?w=681&h=383&crop=1" alt="Superman">

**Note:**

* It is best practice to use relative file paths (if possible). When using relative file paths, your web pages will not be bound to your current base URL. All links will work on your own computer (localhost) as well as on your current public domain and your future public domains.
* It's important to use the appropriate file path depending on the location and structure of your files and directories to ensure the correct resources are loaded in your HTML documents.

## HTML <kbd> For Keyboard Input

The HTML <kbd> element is one of the HTML code elements and it is used for defining user input. The content inside is displayed in the browser's default monospace font.

<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>

Result: Save the document by pressing Ctrl + S

## HTML Symbols with Entities

Symbols which aren’t present on your keyboard can be added to your web pages using entities.

## HTML Entities

Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML. A character entity looks like this:

&entity\_name; OR &#entity\_number;

To display a less than sign (<) we must write: &lt; or &#60;

**Advantage of using an entity name:** An entity name is easy to remember.

**Disadvantage of using an entity name:** Browsers may not support all entity names, but the support for entity numbers is good.

## Creating Symbols using Entities

Many mathematical, technical, and currency symbols are not present on a normal keyboard. To add such symbols to an HTML page, you can use the entity name or the entity number (a decimal or a hexadecimal reference) for the symbol.

Display the euro sign, €, with an entity name, a decimal, and a hexadecimal value:

<p>I will display &euro;</p>

<p>I will display &#8364;</p>

<p>I will display &#x20AC;</p>

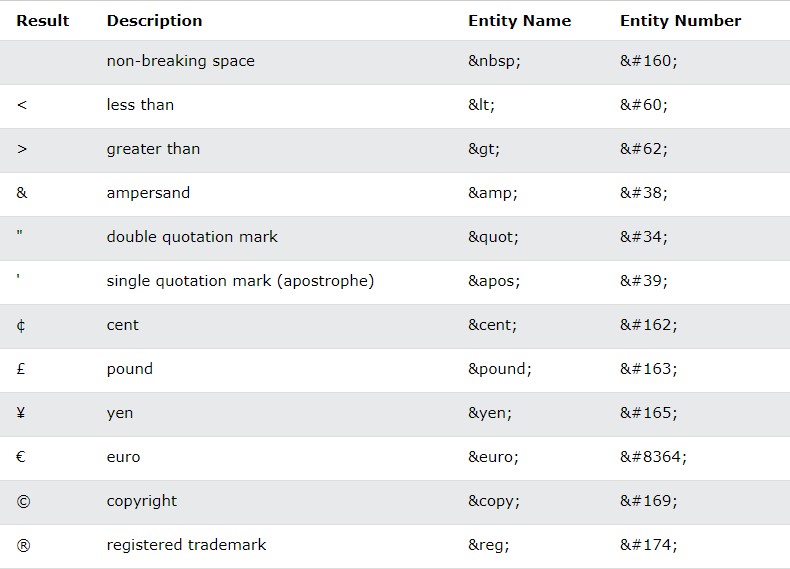
**Result:**

I will display €

I will display €

I will display €

## Useful HTML Character Entities



## Non-breaking Space

A commonly used entity in HTML is the non-breaking space: **&nbsp;**

Non-breaking space (nbsp) is used to prevent the browser from breaking a line of text at a specific point, thereby preserving the visual formatting of the text. It is especially useful in situations where you want to ensure that certain elements of a sentence or phrase remain together on the same line, even if the browser would normally break the line at that point due to limited space.

Here are some common uses of non-breaking space in HTML:

1. Between names and initials: To prevent a name or initials from being split across two lines, non-breaking spaces can be inserted between them. For example, "J. N. M. John" should be written as "J. N. M. John" to ensure that the author's name is not

separated onto two lines.

1. Between numerical values and units: When expressing numerical values and units, non-breaking spaces can be used to prevent the number and the unit from being split across two lines. For example, "10 km", “10 PM” or "3.14 lb".
2. In forms: In web forms, non-breaking spaces can be used to ensure that multiple fields appear on the same line and do not break onto separate lines. For example, "First name: " and "Last name: " can be used to ensure that the form fields appear on the same line.

Overall, non-breaking space is a useful tool for ensuring that text displays correctly on a web page, especially when dealing with situations where line breaks can cause visual confusion or disruption to the formatting.

This is handy when breaking the words might be disruptive.

Another common use of the non-breaking space is to prevent browsers from truncating spaces in HTML pages.

If you write 10 spaces in your text, the browser will remove 9 of them.

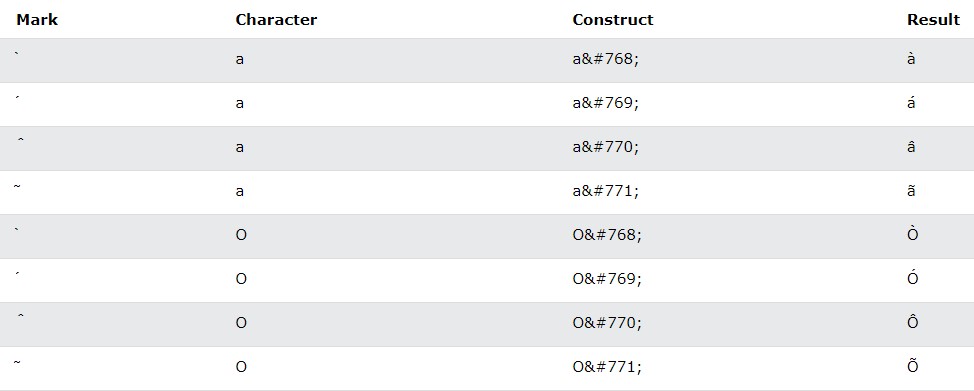
So to add real spaces to your text, you can use the **&nbsp;** character entity.

## Combining Diacritical Marks

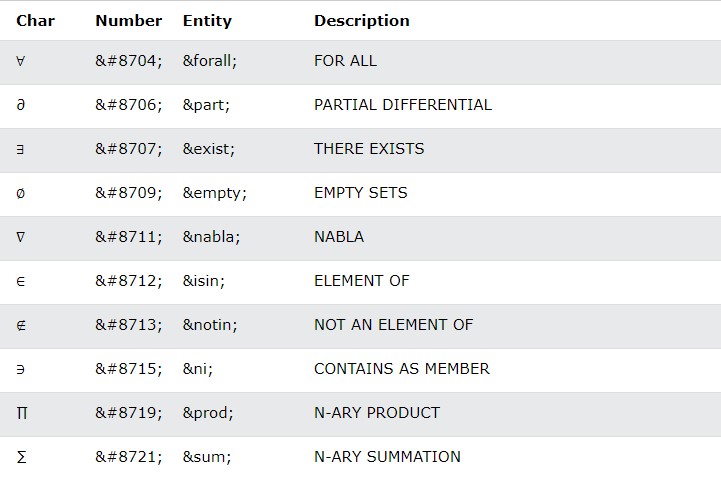
Diacritical marks are typical symbols that lay stress on the pronunciation of a particular word. Grave( ̀) and acute( ́) are the commonly used diacritical marks. We can use diacritical marks with alphanumeric characters to produce new characters unavailable in the character set.

Diacritical marks can appear both above and below a letter, inside a letter, and between two letters.

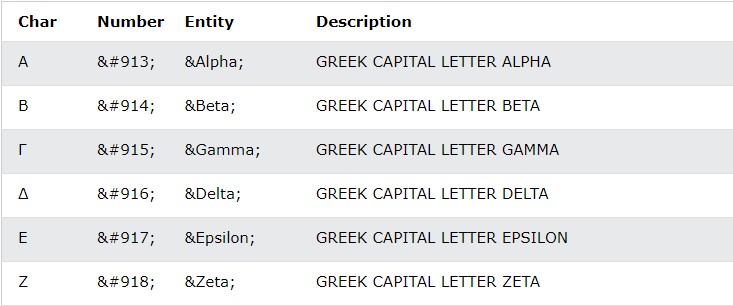
Some Examples;



## Some Mathematical Symbols Supported by HTML



## Some Greek Letters supported by HTML



## Some other entities supported by HTML



## The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page. This is specified in the <meta> tag

<meta charset="UTF-8">

If not specified, UTF-8 is the default character set in HTML.

Many UTF-8 characters cannot be typed on a keyboard, but they can always be displayed using numbers (called entity numbers):

* A is 65
* B is 66
* C is 67

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

</head>

<body>

<p>I will display A B C</p>

<p>I will display &#65; &#66; &#67;</p>

</body>

</html>

To let the browser understand that you are displaying a character, you must start the entity number with &# and end it with ; (semicolon). Both paragraphs will produce the same thing.

# Using Emojis in HTML

Emojis are characters from the UTF-8 character set: 😄😍💗

**What are Emojis?**

Emojis look like images, or icons, but they are not, they are letters (characters) from the UTF-8 (Unicode) character set.

UTF-8 covers almost all of the characters and symbols in the world.

Since Emojis are characters, they can be copied, displayed, and sized just like any other character in HTML.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

</head>

<body>

<h1>Sized Emojis</h1>

<p style="font-size:48px">

&#128512; &#128516; &#128525; &#128151;

</p>

</body>

</html>

## Some Emoji Symbols in UTF-8



## HTML Encoding (Character Sets)

A character set is a set of characters that are recognized by a computer system. These characters can include letters, numbers, punctuation marks, and other symbols. Character sets in HTML are used to define which characters can be used in a document.

The most common character set used in HTML is UTF-8, which is an encoding standard that supports all of the characters in the Unicode standard. Unicode is a character encoding standard that supports over 1 million characters, including letters, symbols, and emoji.

You have to understand that the reason Character sets are important is because they ensure that all browsers and devices can render text properly, regardless of the language or script being used.

To display an HTML page correctly, a web browser must know the character set used in the page.

This is specified in the <meta> tag:

<meta charset="UTF-8">

## From ASCII to UTF-8

ASCII was the first character encoding standard. ASCII defined 128 different characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like ! $ + - ( ) @ < > .

ISO-8859-1 was the default character set for HTML 4. This character set supported 256 different character codes. HTML 4 also supported UTF-8.

ANSI (Windows-1252) was the original Windows character set. ANSI is identical to ISO-8859-1, except that ANSI has 32 extra characters.

The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!

## HTML Uniform Resource Locators

A URL is another word for a web address.

A URL can be composed of words (e.g. deebuginstitute.com), or an Internet Protocol (IP) address (e.g. 192.68.20.50). Most people enter the name when surfing, because names are easier to remember than numbers. Web browsers request pages from web servers by using a URL. IP addresses and ports are important concepts in networking that are used to identify and communicate with devices and services on a network. They allow data to be sent and received across the network, and also help to ensure that data is sent to the correct application or service.

A Uniform Resource Locator (URL) is used to address a document (or other data) on the web.

A web address like <https://www.deebuginstitute.com/about_us> follows these syntax rules: scheme://prefix.domain:port/subdirectory

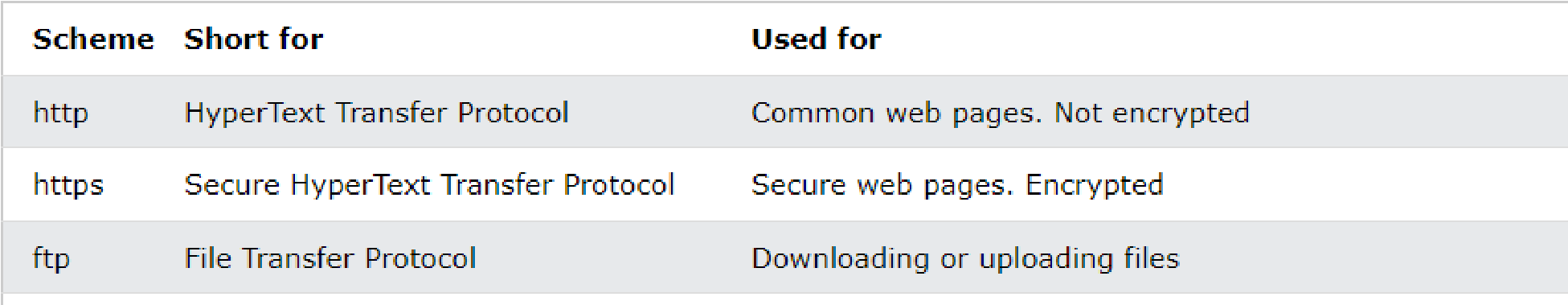
[https://www.deebuginstitute.com/about\_us](http://www.deebuginstitute.com/)

* **scheme** - defines the **type** of Internet service (most common is **http or https**)
* **prefix** - defines a domain **prefix** (default for http is **www**)
* **domain** - defines the Internet **domain name** (like deebuginstitute.com)
* **port** - defines the **port number** at the host. If no port is specified, the default port for the given scheme is assumed (default for http is **80** or **443** for HTTPS)
* **subdirectory**- This refers to a specific **directory or path** on the server that you want to access. A subdirectory in a URL scheme is a way to specify a particular location within a website’s directory structure.

Note: Not all URLs show the port number. In fact, most URLs don't show the port number at all. That's because most web servers use the default port for HTTP (port 80) and HTTPS (port 443), so it's not necessary to specify the port in the URL.

However, if a website is using a non-standard port, then it will be included in the URL. For example, if a website is running on port 8000, the URL might look like this: http://example.com:8000.

## Common URL Schemes



FTP stands for File Transfer Protocol. It's a standard network protocol used for transferring files between computers over a network. It's often used to upload and download files from a web server, such as when uploading images or other media files to a website. FTP is a very old protocol, and it has some security vulnerabilities, so it's not always recommended for transferring sensitive data. Instead, it's typically replaced by secure protocols like SFTP (Secure File Transfer Protocol) or HTTPS (Hypertext Transfer Protocol Secure).