

html css fur dummies Document

html css fur dummies is a popular phrase among beginners who are venturing into web development, specifically focusing on HTML and CSS. If you're new to creating websites or just starting to learn how to style your web pages, this guide aims to demystify the basics of HTML and CSS, making it accessible and easy to understand. Whether you're interested in designing a simple webpage or building a more complex site, understanding the fundamentals of HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) is essential. This article will walk you through these core technologies step-by-step, providing practical insights and tips suitable for beginners and those who consider themselves "dummies" in the web development world.

Understanding HTML and CSS

What is HTML?

HTML, or HyperText Markup Language, is the backbone of any webpage. It provides the structure and content of your website by using various elements and tags. Think of HTML as the skeleton of a building ? it holds everything together and defines where each part goes.

Key points about HTML:

- It uses tags like `<h1>`, `<p>`, `<a href="#">`, ``, and others to mark up content.
- It structures text, images, links, lists, and multimedia.
- It is a markup language, not a programming language, meaning it describes content rather than performing operations.

Example of simple HTML:

```
<<<html
```

Welcome to My Website

This is a paragraph describing my website.

[Visit my site](#)

...

What is CSS?

CSS, or Cascading Style Sheets, is used to control the presentation and layout of your HTML content. It allows you to add colors, fonts, spacing, positioning, and other stylistic features to make your webpage visually appealing.

Key points about CSS:

- It separates content (HTML) from design (CSS).
- It uses selectors to target HTML elements and apply styles.
- Styles can be embedded directly in HTML, included in a separate file, or inline.

Example of simple CSS:

```
```css
```

```
body {
```

```
background-color: f0f0f0;
```

```
font-family: Arial, sans-serif;
```

```
}
```

```
h1 {
```

```
color: blue;
```

```
}
```

```
p {
```

```
font-size: 16px;
```

```
}
```

```
```
```

Getting Started with HTML and CSS

Creating Your First HTML Page

To start building your webpage, you need a text editor (like Notepad, VS Code, Sublime Text) and a web browser (Chrome, Firefox, Edge).

Basic steps:

- Open your text editor.
- Write the following code:

```
```html
My First Webpage
Hello, world!
```

This is my first webpage.

```
```
```

- Save the file as `index.html`.
- Open this file in your web browser to view your webpage.

Adding CSS to Your HTML

You can style your webpage by adding CSS.

Inline CSS example:

```
```html
Hello, world!
```

```
```
```

Internal CSS example:

```
```html
```

```
...
```

External CSS example:

- Create a file named `styles.css`:

```
```css
```

```
body {
```

```
background-color: e0e0e0;
```

```
}
```

```
h1 {
```

```
color: green;
```

```
}
```

```
...
```

- Link it in your HTML:

```
```html
```

```
...
```

## HTML and CSS Fundamentals for Dummies

### HTML Elements and Tags

HTML is composed of elements, which are defined by tags. Some common tags include:

```
- `
`to`
`: Headings
```

- `

`: Paragraphs

- ``: Links

- ``: Images

- `

` `

` `

` `

- `: Lists

- `` and ``: Containers for styling and layout

Example:

```
```html
```

My Favorite Fruits

- Apples

- Bananas

- Oranges

```
```
```

## CSS Selectors and Properties

CSS uses selectors to target elements:

- Element selectors: `h1`, `p`, `div`

- Class selectors: `.classname`

- ID selectors: `idname`

- Group selectors: `h1, h2, h3`

Common CSS properties include:

- `color`, `background-color`

- `font-family`, `font-size`

- `margin`, `padding`, `border`

- `display`, `position`, `float`

Example:

```
```css
```

```
/ Style all h2 headings /
```

```
h2 {
```

```
color: purple;
```

```
font-size: 24px;
```

```
}
```

```
/ Style elements with class 'highlight' /
```

```
.highlight {
```

```
background-color: yellow;
```

```
padding: 5px;
```

```
}
```

```
```
```

## Best Practices for Beginners

### Organize Your Code

- Use indentation for nested elements to improve readability.
- Comment your code to remember the purpose of sections:

```
```html
```

My Website

```
```
```

### Use External CSS Files

- Keep your style separate from your content for easier management.
- Link multiple pages to a single stylesheet.

## Validate Your Code

- Use tools like W3Schools Validator or the W3C Markup Validation Service to check for errors.

## Learn by Doing

- Practice by creating small projects.
- Experiment with different tags and styles.

## Common HTML and CSS Tips for Dummies

- **Start simple:** Focus on creating basic pages before adding complex features.
- **Use descriptive class and ID names:** This makes styling easier.
- **Preview often:** Regularly view your webpage in a browser to see changes.
- **Learn from examples:** Study other websites' code to understand structure and styling.
- **Ask for help:** Use forums like Stack Overflow or tutorials when stuck.

## Additional Resources for HTML & CSS Beginners

- [W3Schools HTML Tutorial](<https://www.w3schools.com/html/>)
- [W3Schools CSS Tutorial](<https://www.w3schools.com/css/>)
- [MDN Web Docs](<https://developer.mozilla.org/en-US/>)
- [freeCodeCamp](<https://www.freecodecamp.org/>)

## Conclusion

Mastering HTML and CSS is the foundation for web development, and starting with the basics can seem overwhelming at first. Remember, everyone was a beginner at some point ? the key is consistent practice and experimentation. By understanding the structure provided by HTML and the styling capabilities of CSS, you can create visually appealing and well-structured websites. Keep exploring, building, and learning, and soon you'll move from "dummies" to confident web developer.

Summary Checklist for Beginners:

- Learn HTML tags and their purposes.
- Practice creating simple HTML pages.
- Understand how to add CSS for styling.
- Experiment with different styles and layouts.
- Use online resources to expand your knowledge.
- Keep your code organized and validate it regularly.

Embark on your web development journey today ? with patience and practice, you'll soon craft websites that look great and function perfectly!

HTML & CSS for Dummies: A Comprehensive Guide for Beginners

## HTML & CSS for Dummies: A Comprehensive Guide for Beginners

**HTML & CSS for dummies** is a phrase many aspiring web developers search for when they first embark on their journey into the world of website creation. Whether you're a complete novice or someone with minimal experience, understanding the fundamentals of HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) is essential to building visually appealing, well-structured websites. This article aims to demystify these core web technologies, providing a clear, reader-friendly guide that balances technical accuracy with accessible language.

## What is HTML? The Building Blocks of the Web

### Understanding HTML

HTML is the backbone of every webpage. It provides the structure and content, organizing text, images, links, and other media into a coherent format that browsers can interpret and display. Think of HTML as the skeleton of a website ? it gives shape and form but doesn't concern itself with how things look.

### Basic Anatomy of an HTML Document

A typical HTML document consists of nested elements, each enclosed within tags. Here's a simplified outline:

```
```html
```

My First Webpage

Welcome to My Website

This is a paragraph of text.

```
```
```

- `<!DOCTYPE html>`: Declares the document type and version.
- `<html>`: The root element that encompasses all other elements.
- `<head>`: Contains metadata, scripts, stylesheets, and the page title.
- `<title>`: Sets the page's title, visible on the browser tab.
- `<body>`: Contains the content visible to users, such as headings, paragraphs, images, etc.

## Common HTML Elements for Dummies

Here are some essential tags every beginner should know:

`<h1>` to `<h6>`: Headings, with `<h1>` being the most important.

`<p>`: Paragraphs of text.

- `<a>`: Hyperlinks to other pages or resources.
- `<img>`: Embeds images.
- `<ul>`, `<ol>`: Unordered and ordered lists.
- `<div>`: A generic container for grouping elements.
- `<span>`: Inline container for styling parts of text.

## Creating Your First HTML Page

To get started:

- Use a simple text editor (like Notepad, Sublime Text, or VS Code).
- Save your file with a `.html` extension, e.g., `index.html`.
- Write your HTML code following the structure above.
- Open the file in a web browser to see your webpage come to life.

## CSS: Making Your Website Visually Appealing

## What is CSS?

CSS, or Cascading Style Sheets, controls the visual presentation of your HTML content. While HTML provides the structure, CSS handles colors, fonts, layout, spacing, and other design aspects. Think of CSS as the clothing and decoration that make your skeleton look attractive and unique.

## How CSS Works

CSS involves selecting HTML elements and applying styles to them. Styles can be defined inline, embedded within the HTML document, or stored in separate stylesheet files.

Inline CSS Example:

```
```html
```

This paragraph is blue.

```
```
```

Embedded CSS Example:

```
```html
```

```
```
```

External CSS Example:

Create a separate `styles.css` file:

```
```css
```

```
body {
```

```
background-color: f0f0f0;
```

```
}
```

```
h1 {
```

```
color: navy;
```

```
font-family: Arial, sans-serif;
```

```
}
```

```
...
```

And link it in your HTML:

```
```html
```

```
...
```

## Basic CSS Properties for Dummies

Some fundamental CSS properties include:

- `color``: Text color.
- `background-color``: Background color.
- `font-family``: Font style.
- `font-size``: Text size.
- `margin``: Space outside elements.
- `padding``: Space inside elements.
- `border``: Borders around elements.
- `display``: How elements are displayed (block, inline, flex, grid).

## Applying CSS for the First Time

Start with simple styles:

```
```css
```

```
body {
```

```
background-color: fff;
```

```
font-family: Verdana, Geneva, Tahoma, sans-serif;
```

```
margin: 20px;
```

```
}
```

```
h1 {  
  
color: 333;  
  
text-align: center;  
  
}  
  
p {  
  
line-height: 1.6;  
  
}  
  
...
```

Link this stylesheet to your HTML file, and observe how the appearance changes.

Bringing HTML and CSS Together

Separation of Structure and Style

While you can embed CSS directly in your HTML, best practices recommend keeping styles in separate `.css` files. This approach:

- Keeps code organized.
- Makes it easier to update styles globally.
- Allows multiple pages to share the same stylesheet.

Example: Building a Simple Webpage

Suppose you want a webpage with a header, a paragraph, and a footer, styled neatly:

HTML (`index.html`):

```
```html  
Simple Webpage
My Dummies Website
```

Welcome! This site is built with HTML and styled with CSS.

© 2024

```

CSS (`styles.css`):

```css

body {

font-family: Arial, sans-serif;

background-color: f9f9f9;

margin: 0;

padding: 20px;

}

header {

background-color: 4CAF50;

padding: 15px;

text-align: center;

}

h1 {

color: white;

margin: 0;

}

section {

margin-top: 20px;

```
}

footer {

margin-top: 30px;

text-align: center;

color: 777;

}

...
```

Open `index.html` in your browser to see your styled webpage.

## Common Mistakes to Avoid for Dummies

- Forgetting to close tags: Always close your HTML tags properly (`</>`).
- Incorrect nesting: Elements should be properly nested, not overlapping improperly.
- Using inline styles excessively: While quick for testing, inline styles make maintenance difficult.
- Not linking CSS correctly: Ensure your stylesheet link uses the correct path and filename.
- Ignoring responsiveness: For beginner projects, focus on desktop view before exploring mobile-friendly design.

## Next Steps: Expanding Your Web Skills

Once comfortable with HTML and CSS basics, you can explore:

- Responsive Design: Making websites look good on all devices using media queries.
- CSS Frameworks: Such as Bootstrap, to accelerate development.
- JavaScript: Adding interactivity to your sites.
- Version Control: Using Git to manage your code.
- Web Hosting: Publishing your website online.

## Conclusion: Your Journey Begins Here

Building websites might seem daunting at first, but understanding the core principles of HTML and CSS is straightforward once broken down into manageable steps. Remember, HTML provides the

structure ? the bones of your webpage ? while CSS dresses it up to be attractive and user-friendly. Starting with simple projects, practicing regularly, and gradually exploring more advanced features will set you on the path to becoming a competent web developer. Keep experimenting, stay curious, and enjoy creating your digital space.

**Question** What is HTML and why is it important for beginners? HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It's essential for beginners because it forms the foundation of all websites, allowing you to add text, images, links, and other elements. How does CSS enhance the appearance of a website? CSS (Cascading Style Sheets) is used to style and layout web pages. It allows you to control colors, fonts, spacing, and positioning, making your website visually appealing and user-friendly without changing the HTML structure. What are some basic HTML tags I should learn first? Start with tags like `

**to`**

**` for headings,`**

**` for paragraphs, `` for links, `` for images,`**

**` and`**

**- ` for lists, and `` for sections. These are fundamental for building simple web pages. How can I link CSS to my HTML file? You can link CSS to your HTML file by adding a `` tag inside the `` section, like this: ``. This connects your CSS stylesheet to your HTML document. What are some common CSS properties I should learn first? Begin with properties like `color`, `background-color`, `font-family`, `font-size`, `margin`, `padding`, `border`, and `display`. These help you control the look and layout of your webpage. Are there any free resources to learn HTML and CSS for beginners? Yes, websites like freeCodeCamp, W3Schools, Mozilla Developer Network (MDN), and Codecademy offer free tutorials and interactive lessons suitable for beginners learning HTML and CSS. How can I practice my HTML and CSS skills effectively? Practice by creating small projects like personal pages, portfolios, or simple websites. Use online editors like CodePen or JSFiddle to experiment and see live results. Also, try replicating existing website layouts to improve your skills.**

**Related keywords: HTML, CSS, web development, beginner, tutorials, coding, website design, front-end, web design, HTML basics**

# html programming for beginners answers all your q

HTML programming for beginners answers all your q ? whether you're just starting out or looking to deepen your understanding of web development, this comprehensive guide is designed to answer all your questions about HTML programming. From the basics of structure and syntax to more advanced concepts like forms and multimedia, we'll walk through everything you need to know to become confident in HTML. Understanding HTML is fundamental to creating engaging, functional websites, and this article aims to make that learning process straightforward and accessible.

## What is HTML and Why is it Important?

HTML, short for HyperText Markup Language, is the foundational language used to create web pages. It structures content on the internet, allowing browsers to interpret and display text, images, videos, links, and other multimedia elements.

## Key Reasons to Learn HTML

- **Foundation for Web Development:** HTML is the backbone of all websites.
- **Easy to Learn:** Its syntax is simple and intuitive for beginners.
- **Creates Interactive Content:** HTML works with CSS and JavaScript to build dynamic web pages.
- **High Demand:** Web development skills are highly sought after in the job market.

## Basic Structure of an HTML Document

Understanding the basic skeleton of an HTML document is crucial for beginners. Every HTML file has a specific structure that browsers recognize and render accordingly.

## Typical HTML Document Layout

### Page Title

## Explanation of Components

- : Declares the document type and version of HTML.
- : The root element that wraps all content.
- : Contains metadata, scripts, styles, and the page title.
- : Sets the title displayed in the browser tab.
- : Contains the visible content of the webpage.

## Common HTML Elements for Beginners

Learning the essential HTML tags will help you structure your content effectively.

### Text Formatting Tags

- **to**  
: Headings of different levels
- : Paragraphs
  - : Bold text
  - : *Italicized text*
  - *and*  
: Unordered and ordered lists
  - - : List items

### Link and Image Elements

- : [Creates hyperlinks](#)
- : [Embeds images](#)

### Form Elements

- : [Creates a form](#)
- : [User input fields](#)
- : [Multi-line text input](#)

- : [Clickable button](#)

## [Creating Your First Web Page](#)

[Starting with a simple HTML file is the best way to learn.](#)

### [Step-by-Step Guide](#)

- [Open a text editor \(like Notepad, VS Code, or Sublime Text\).](#)
- [Write the basic HTML structure with a title and some content.](#)
- [Save the file with a .html extension, e.g., index.html.](#)
- [Open the saved file in a web browser to see your webpage.](#)

## [Sample HTML Code for Beginners](#)

### [My First Webpage](#)

## [Hello, World!](#)

[This is my first HTML page.](#)

[Visit Example](#)

## HTML Attributes and Their Usage

*Attributes provide additional information about HTML elements and are written inside the opening tag.*

### Common Attributes

- **href**: Specifies the URL in `a` tags.
- **src**: Defines the source file in `img` and `video` tags.
- **alt**: Provides alternative text for images.
- **id**: Unique identifier for an element.
- **class**: Classifies elements for styling with CSS.
- **style**: Inline CSS styles.

### Example

[Google](#)

*This creates a link that opens in a new tab due to the target attribute.*

## Introduction to HTML Forms

*Forms are essential for collecting user input, such as login details, surveys, or search queries.*

### Basic Structure of a Form

**Name:**

## Common Input Types

- : *Single-line text input*
- : *Password input*
- : *Email address*
- : *Checkbox*
- : *Radio button*
- : *Submit button*

## Embedding Multimedia Content

*Adding images, videos, and audio enriches your webpage.*

### Embedding Images

- : *Displays an image.*

### Embedding Videos and Audio

- : *Embeds a video player.*
- : *Embeds an audio player.*

## Best Practices for Writing HTML

*To ensure your code is clean, accessible, and efficient, follow these best practices:*

### Organize Your Code

- *Use indentation to improve readability.*
- *Comment your code to explain sections.*

### Use Semantic Elements

- , , , : *Provide meaningful structure.*

### Validate Your HTML

- Use tools like the W3C Markup Validator to check for errors.

## Learning Resources and Next Steps

Once comfortable with the basics, expand your knowledge with these resources:

- [MDN Web Docs - HTML](#)
- [W3Schools HTML Tutorial](#)
- Practice by building small projects like personal pages, portfolios, or blogs.
- Learn CSS to style your HTML content and JavaScript to add interactivity.

HTML programming for beginners answers all your q ? if you're just starting out in web development, this phrase encapsulates the curiosity and eagerness many newcomers feel as they embark on their coding journey. HTML (HyperText Markup Language) is the foundational language of the web, shaping the structure and content of websites. Mastering HTML is essential for anyone interested in creating web pages, whether for personal projects, professional portfolios, or career advancement. This comprehensive guide aims to answer all your questions about HTML programming for beginners, providing clear explanations, practical insights, and helpful tips to set you on the right path.

## What is HTML and Why is it Important?

### Understanding HTML

HTML stands for HyperText Markup Language. It is a markup language used to structure content on the internet. Unlike programming languages like JavaScript or Python that add interactivity or logic, HTML describes the layout, headings, paragraphs, images, links, and other elements that make up a webpage.

### Importance of HTML in Web Development

- **Foundation of Web Pages:** Every website you visit is built using HTML, making it the backbone of the internet.
- **Universal Compatibility:** HTML is supported across all browsers and devices.
- **Ease of Learning:** HTML has a straightforward syntax, making it accessible to beginners.
- **Interoperability:** HTML works seamlessly with CSS and JavaScript to create dynamic, styled pages.

## Getting Started with HTML: Basic Concepts

## HTML Syntax Overview

*HTML documents are composed of elements, which are represented by tags enclosed in angle brackets. Elements can contain attributes that provide additional information.*

*Example:*

```
<html
```

*This is a paragraph.*

```
>
```

## Basic Structure of an HTML Document

*A minimal HTML document looks like this:*

```
<html
```

*My First Web Page*

***Hello, World!***

*Welcome to HTML programming.*

```
>
```

- `<html` declares the document type.
- `<html>` wraps the entire content.
- `<title>` contains metadata, including the title.
- `<body>` contains visible content.

## Core HTML Elements for Beginners

### Headings and Paragraphs

```
<h1>
<h2>
<h3>
<h4>
<h5>
<h6>
<p>
```

*: Define headings of different levels.*

- `

`: Represents a paragraph.

## Links and Images

- [`Link Text`](#): Creates hyperlinks.
- ``: Embeds images.

## Lists

- Ordered List: `  
`with `  
- ` items.
- Unordered List: `  
`with `  
- ` items.

## Divisions and Spans

- ``: Container for block-level grouping.
- ``: Inline grouping.

## Best Practices for Writing HTML for Beginners

### Use Semantic Elements

Semantic tags clearly describe their purpose, improving accessibility and SEO:

- ``, `

### Keep Code Organized and Indented

Proper indentation improves readability and makes debugging easier.

### Validate Your HTML

Use tools like the W3C Validator to ensure your code meets standards.

## Common Questions and Troubleshooting

### *Why is my HTML not displaying correctly?*

- *Check for syntax errors like missing closing tags.*
- *Ensure your file is saved with a `.html` extension.*
- *View the page in different browsers to identify compatibility issues.*

### *How do I link CSS and JavaScript to HTML?*

- *CSS: `<link>` inside `<head>`.*
- *JavaScript: `<script>` before `</body>`.*

### *What tools should I use to write HTML?*

- *Text editors like Visual Studio Code, Sublime Text, or Notepad++.*
- *Browser developer tools for inspecting and debugging.*

## Advantages and Disadvantages of Learning HTML

### Pros

- *Easy to learn and understand for beginners.*
- *Provides a solid foundation for web development.*
- *Widely used and supported across all platforms.*
- *Enables quick prototyping of websites.*

### Cons

- *Limited to structuring content; cannot create dynamic features.*
- *Requires knowledge of CSS and JavaScript for full functionality.*
- *Can become complex with large projects if not well-organized.*

## Progressing Beyond Basic HTML

### Learning CSS

**CSS (Cascading Style Sheets) styles your HTML content, controlling layout, colors, fonts, and responsiveness.**

## **Introduction to JavaScript**

**JavaScript adds interactivity, such as forms validation, animations, and dynamic content updates.**

## **Frameworks and Libraries**

**Once comfortable with HTML, exploring frameworks like Bootstrap (for styling) or React (for building complex UIs) can enhance your skills.**

## **Resources for HTML Beginners**

- **MDN Web Docs: Comprehensive tutorials and references.**
- **W3Schools: Interactive examples and quizzes.**
- **freeCodeCamp: Hands-on projects and certifications.**
- **Codecademy: Interactive courses for immersive learning.**
- **YouTube Channels: Traversy Media, The Net Ninja, freeCodeCamp.org.**

## **Practical Tips for HTML Beginners**

- **Practice regularly by creating small projects like a personal homepage.**
- **Experiment with different HTML elements to understand their functions.**
- **Use browser developer tools to inspect and modify code in real-time.**
- **Read and analyze existing websites' source code to learn best practices.**
- **Join online communities like Stack Overflow or Reddit's r/webdev to ask questions and share knowledge.**

## **Summary**

**HTML programming for beginners is an accessible entry point into the world of web development. By understanding the basic structure, common elements, and best practices, you can start creating simple yet effective web pages. Remember that HTML is just the beginning; complement your learning with CSS and JavaScript to build modern, responsive, and interactive websites. Patience, practice, and curiosity are your best tools as you navigate the exciting landscape of web development.**

## Final Thoughts:

*Starting with HTML might seem straightforward, but mastering it opens doors to endless creative possibilities. Whether you aim to build personal projects, contribute to open-source websites, or pursue a career in tech, a solid grasp of HTML is the first step. Keep experimenting, stay curious, and don't hesitate to seek out resources and communities for support. Happy coding!*

**Question** What is HTML and why is it important for web development? HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It is essential because it defines the layout, headings, paragraphs, links, images, and other elements that make up a webpage, forming the foundation of all websites. How do I start coding HTML as a beginner? Begin by learning basic tags like , , ,

-  
,  
, and . Use a simple text editor like Notepad or VS Code to write your code, then open the file in a web browser to see the results. Practice creating simple pages to build your skills. What are HTML tags and attributes? HTML tags are keywords enclosed in angle brackets that define elements on a webpage, like or . Attributes provide additional information about elements, such as src for images or href for links, and are added inside the opening tag. What is the difference between HTML and CSS? HTML is used to structure and organize content on a webpage, while CSS (Cascading Style Sheets) is used to style and layout that content, including colors, fonts, and positioning. How do I add images to my HTML page? Use the tag with the src attribute to specify the image URL or file path. Example: . Always include alt text for accessibility. What are semantic HTML tags and why should I use them? Semantic tags like , , , and clearly describe the purpose of the content they contain. They improve accessibility, SEO, and make your code easier to read and maintain. How can I create links in HTML? Use the tag with the href attribute to create hyperlinks. Example: [Visit Example](#). What are best practices for writing clean HTML code? Use proper indentation, meaningful tag names, comments to explain sections, and avoid inline styles. Validate your code with tools like the W3C Markup Validator to ensure it meets standards. How do I make my HTML webpage mobile-friendly? Use responsive design techniques, including setting the viewport meta tag () and using flexible layouts with CSS media queries. Where can I learn more about HTML programming for beginners? You can explore online tutorials on sites like MDN Web Docs, freeCodeCamp, W3Schools, and Codecademy, which offer comprehensive guides and interactive lessons for HTML beginners.

**Related keywords:** HTML, web development, beginner programming, HTML tutorial, HTML basics, coding for beginners, learn HTML, HTML questions, web design, HTML answers

**Read the full article online: [Html Programming For Beginners Answers All Your Q](#)**