Html Xhtml Et Css Pour Les Nuls Manual

html xhtml et css pour les nuls

Se lancer dans le monde du développement web peut sembler intimidant au début, surtout lorsqu'il s'agit de comprendre des concepts aussi fondamentaux que HTML, XHTML et CSS. Si vous êtes débutant ou que vous souhaitez simplement avoir une compréhension claire et simple de ces technologies, cet article est fait pour vous. Nous allons explorer étape par étape ce qu?est HTML, XHTML, et comment CSS vient compléter la création de pages web attrayantes et fonctionnelles.

Qu'est-ce que HTML ? Le socle du web

Le HTML, ou HyperText Markup Language, est le langage de base utilisé pour créer des pages web. Il sert à structurer le contenu en utilisant des balises qui indiquent au navigateur comment afficher le texte, les images, les liens, et autres éléments.

Les fondamentaux du HTML

- **Balises :** Ce sont des éléments de code qui entourent le contenu. Par exemple,
pour un titre principal, ou
pour un paragraphe.
- Attributs : Ils donnent des informations supplémentaires sur une balise. Par exemple, Lien où href indique l?URL de destination.
- Structure : Une page HTML doit commencer par une déclaration et contenir des éléments comme , (pour les métadonnées), et (pour le contenu visible).

Ce que fait HTML

HTML sert à structurer le contenu d?une page. Sans lui, il serait impossible de définir où se trouve un titre, un paragraphe, une image ou un lien. C?est la base de toute page web.

Comprendre XHTML : Une version plus stricte du HTML

L?XHTML, ou eXtensible HyperText Markup Language, est une reformulation du HTML en utilisant la syntaxe XML. Il a été créé pour rendre les pages web plus conformes aux standards et plus faciles à interpréter par différents outils.

Les différences clés entre HTML et XHTML

- Syntaxe stricte : En XHTML, toutes les balises doivent être fermées, même celles qui en HTML sont optionnelles, comme .
- Utilisation de majuscules : Les noms de balises et attributs doivent être en minuscules.
- Attributs obligatoires : Certains attributs doivent être présents, par exemple, l?attribut alt dans une balise .

Pourquoi utiliser XHTML ?

- Il favorise une meilleure compatibilité avec d?autres technologies XML.
- Il encourage une écriture de code cohérente et propre.
- Il facilite la validation des pages selon des standards stricts.

HTML vs XHTML : Le choix

De nos jours, la majorité des développeurs privilégient HTML5, qui combine simplicité et puissance. Cependant, connaître XHTML peut être utile pour comprendre la structure stricte du code et pour des projets nécessitant une conformité rigoureuse aux standards.

CSS : La touche esthétique pour vos pages web

HTML et XHTML permettent de structurer votre contenu, mais pour le rendre attrayant, il faut ajouter des styles avec CSS, ou Cascading Style Sheets.

Qu?est-ce que CSS ?

CSS est un langage qui permet de définir l?apparence visuelle des éléments HTML. Il contrôle la mise en page, les couleurs, les polices, les marges, et bien plus encore.

Les bases de CSS

- Sélecteurs : Ils ciblent les éléments HTML à styler. Par exemple, p { color: blue; } applique une couleur bleue à tous les paragraphes.
- Propriétés : Définissent le style, comme font-family, background-color, margin, etc.
- Valeurs : Elles donnent les paramètres aux propriétés, par exemple font-family: Arial, sans-serif;.

Comment utiliser CSS

- Inline : Directement dans la balise HTML avec l?attribut style.
- Interne : Dans une section dans l?en-tête () de la page.
- Externe : Via un fichier CSS séparé lié avec la balise .

Exemple simple de CSS

Les bonnes pratiques pour débuter

Pour bien commencer avec HTML, XHTML et CSS, voici quelques conseils pratiques.

Organisez votre code

- Utilisez une indentation cohérente pour rendre votre code lisible.
- Commentez votre code pour expliquer les sections importantes.

Validez votre code

Utilisez des outils en ligne comme le validateur W3C pour vérifier que votre code respecte les standards. Cela évite les erreurs d?affichage et augmente la compatibilité.

Expérimentez progressivement

- Commencez par créer une page simple avec un titre, un paragraphe, et une image.
- Ajoutez du style en modifiant votre CSS étape par étape.
- Explorez de nouvelles balises et propriétés pour enrichir votre page.

Ressources pour aller plus loin

- Documentation MDN sur HTML
- Documentation MDN sur CSS
- Guide officiel W3C sur CSS
- Créez votre premier site web en suivant des tutoriels en ligne pour pratiquer concrètement.

Conclusion

Maîtriser html xhtml et css pour les nuls est une étape essentielle pour quiconque souhaite se lancer dans le développement web. En comprenant la différence entre HTML et XHTML, en maîtrisant la syntaxe de base, et en apprenant à styliser vos pages avec CSS, vous pourrez créer des sites web attrayants, fonctionnels et conformes aux standards. N?oubliez pas que la pratique régulière et la validation de votre code sont clés pour progresser rapidement. Alors, n?attendez plus, commencez à coder dès aujourd?hui et donnez vie à vos projets en ligne !

HTML, XHTML et CSS pour les Nuls : Maîtriser les bases du web en toute simplicité

Introduction : Pourquoi apprendre HTML, XHTML et CSS ?

Dans l?univers du web, la création de sites Internet repose sur trois piliers fondamentaux : HTML, XHTML et CSS. Que vous soyez débutant, passionné ou professionnel souhaitant renforcer ses compétences, comprendre ces technologies est essentiel pour construire des pages web attrayantes, fonctionnelles et conformes aux standards. Cet article vous guidera pas à pas dans l?apprentissage de ces langages, en abordant leurs spécificités, leur fonctionnement et leurs bonnes pratiques.

Qu?est-ce que HTML ?

Définition et rôle de HTML

HTML (HyperText Markup Language) est le langage de balisage utilisé pour structurer le contenu d?une page web. Il définit la hiérarchie, la présentation et le comportement des éléments présents sur une page : textes, images, liens, formulaires, etc.

Historique et évolution

- Créé par Tim Berners-Lee en 1991, HTML a connu plusieurs versions, dont HTML 2.0, 3.2, 4.01, et plus récemment HTML5.
- La dernière version, HTML5, introduit de nombreuses fonctionnalités modernes pour répondre aux exigences du web actuel (multimédia, graphiques, stockage local, etc.).

Fonctionnement de HTML

- Utilise des balises (tags) pour délimiter et définir chaque élément.
- Chaque balise peut avoir des attributs pour préciser ses propriétés.
- La structure de base d?une page HTML est composée d?un `` (entête) et d?un `` (corps).

Exemple simple de code HTML

```html
Ma première page
Bienvenue sur mon site

Ceci est un paragraphe de texte.

```

Qu?est-ce que XHTML ?

Définition et différences avec HTML

XHTML (Extensible HyperText Markup Language) est une reformulation de HTML en utilisant la syntaxe stricte de XML. Il impose des règles plus strictes pour assurer une meilleure compatibilité et une conformité aux standards du web.

Principales différences :

- Syntaxe rigoureuse : toutes les balises doivent être fermées, même celles qui sont auto-fermantes (ex : ``).
- Respect strict de la casse : les balises et attributs doivent être en minuscules.
- Document doit être bien formé pour être valide XHTML.

Pourquoi utiliser XHTML ?

- Compatibilité accrue avec les navigateurs XML.
- Facilite la validation et la maintenance du code.
- Permet une meilleure structuration pour des technologies futures.

Exemple de code XHTML

```xml

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
Ma première page XHTML
Bienvenue sur mon site

Ceci est un paragraphe de texte.

```

Transition entre HTML et XHTML

- HTML5 est conçu pour être compatible avec XHTML, mais il maintient une syntaxe souple.
- La majorité des sites modernes utilisent HTML5, mais connaître XHTML est utile pour des projets nécessitant une stricte conformité XML.

Comprendre CSS : La feuille de style du web

Qu?est-ce que CSS ?

CSS (Cascading Style Sheets) est le langage qui contrôle la présentation visuelle d?une page web. Il permet de définir les couleurs, polices, marges, alignements, dispositions et effets visuels.

Rôle de CSS dans la conception web

- Séparer le contenu (HTML) de la présentation.
- Faciliter la maintenance et la mise à jour du design.
- Créer des interfaces esthétiques cohérentes et modernes.

Syntaxe de base de CSS

- Sélecteur : cible un ou plusieurs éléments HTML.
- Déclaration : définit la propriété et la valeur.

Exemple :

```css

h1 {

color: blue;

font-size: 24px;

}

```

Intégration de CSS dans une page HTML

- En ligne : dans l?attribut `style` d?un élément.
- Interne : dans une balise `

Principles of html xhtml and dhtml

principles of html xhtml and dhtml form the foundation of modern web development, guiding how developers structure, style, and add interactivity to websites. Understanding these principles is crucial for creating accessible, maintainable, and standards-compliant web pages. HTML, XHTML, and DHTML are related technologies that, while sharing common goals, differ in syntax, capabilities, and implementation strategies. This article explores the core principles that underpin each of these web technologies, highlighting their similarities, differences, and best practices for effective web development.

Understanding HTML: The Backbone of the Web

HTML (Hypertext Markup Language) is the standard markup language used to create web pages. It provides the basic structure of sites, allowing developers to define elements such as headings, paragraphs, links, images, and more. The primary principles of HTML focus on simplicity, accessibility, and compatibility.

Core Principles of HTML

- Semantic Markup: Use meaningful tags that clearly describe the content they enclose (e.g., , ,). This improves accessibility and SEO.
- Structure and Hierarchy: Organize content logically with proper nesting of elements to reflect the document?s structure.
- Separation of Content and Presentation: Keep content separate from styling, typically through CSS, to enhance maintainability.

- Accessibility: Ensure content is accessible to all users, including those using assistive technologies, by using appropriate tags and attributes.
- Compatibility: Write code that works across different browsers and devices.

HTML?s simplicity and focus on structure make it the foundation for web development. However, as web standards evolved, XHTML was introduced to enforce stricter syntax rules, and DHTML emerged to enhance interactivity.

XHTML: Stricter, XML-Based HTML

XHTML (Extensible Hypertext Markup Language) is an XML-based reformulation of HTML. It emphasizes stricter syntax rules, making documents well-formed XML, which enables better interoperability and easier data manipulation.

Principles of XHTML

- Well-Formedness: XHTML documents must adhere to XML syntax rules, such as properly nested tags, lowercase element names, and closing all tags.
- Strict Syntax: No presentational elements or attributes are allowed; styles and formatting should be handled via CSS.
- Validation: XHTML documents are validated against DTDs (Document Type Definitions) to ensure correctness.
- Compatibility with XML Tools: XHTML documents can be processed with standard XML parsers, facilitating data exchange and transformation.
- Backward Compatibility: XHTML is designed to be compatible with HTML, but requires adherence to stricter syntax rules.

XHTML?s emphasis on correctness and formality makes it suitable for applications requiring rigorous data standards, such as XML data exchange, RSS feeds, and mobile devices.

Introducing DHTML: Dynamic HTML for Interactivity

DHTML (Dynamic HTML) is not a separate language but a collection of techniques used to create interactive and animated websites by combining HTML, CSS, and JavaScript. DHTML enables web pages to change dynamically without needing to reload, enhancing user experience.

Principles of DHTML

- Document Object Model (DOM) Manipulation: Use JavaScript to access and modify the structure, style, and content of web pages dynamically.
- Separation of Concerns: Maintain a clear separation between HTML (structure), CSS (style), and JavaScript (behavior) for better maintainability.
- Event-Driven Programming: Respond to user actions such as clicks, mouse movements, or keyboard input to create interactive features.
- Progressive Enhancement: Ensure that websites work with basic functionality even if scripting is disabled, enhancing accessibility.
- Use of Standards-Compliant Code: Follow web standards to maximize compatibility across browsers and devices.

DHTML allows developers to create engaging interfaces, such as dropdown menus, image sliders, and real-time form validation, significantly improving user engagement.

Comparison of HTML, XHTML, and DHTML Principles

While HTML, XHTML, and DHTML serve different purposes, their principles often intersect, especially regarding standards compliance, accessibility, and usability.

Key Differences and Similarities

- Syntactical Rigidity: XHTML enforces stricter syntax rules compared to HTML, which is more forgiving. DHTML relies on HTML or XHTML structures but adds dynamic behaviors.
- Separation of Concerns: Both HTML/XHTML advocate separating content from style (via CSS) and behavior (via JavaScript), a principle central to DHTML.
- Interactivity: DHTML specifically focuses on adding dynamic features through scripting, which complements HTML and XHTML's static structures.
- Compatibility and Validation: XHTML documents need validation against DTDs, whereas HTML documents follow the DOCTYPE declaration. DHTML relies on standards-compliant code to ensure cross-browser functionality.
- Use of Markup Languages: HTML and XHTML are markup languages defining document structure, while DHTML is a collection of techniques that enhance those structures with interactivity.

Best Practices for Applying Principles in Web Development

Implementing the principles of HTML, XHTML, and DHTML effectively requires adherence to best practices that ensure accessibility, maintainability, and compatibility.

Designing with HTML and XHTML

- Use semantic tags to improve accessibility and SEO.
- Validate your code regularly using validators like W3C Validator to ensure standards compliance.
- Keep your code clean and well-organized with proper indentation and commenting.
- Use external CSS for styling to separate presentation from content.
- Ensure your documents are well-formed and valid XHTML if using XML syntax.

Enhancing with DHTML

- Leverage the DOM API for robust manipulation of page elements.
- Implement event listeners for user interactions to create dynamic behaviors.
- Optimize scripts for performance and responsiveness.
- Test across multiple browsers and devices to ensure consistency.
- Maintain graceful degradation so that functionality is preserved even if scripting fails.

The Future of Web Principles: Embracing Standards and Accessibility

As web technologies evolve, the core principles of these languages emphasize creating accessible, semantic, and responsive content. Modern standards encourage developers to write clean, valid code, and utilize progressive enhancement techniques. The adoption of HTML5, which consolidates many features of HTML and XHTML, further exemplifies these principles by promoting semantic markup, multimedia support, and better support for dynamic content.

Key Takeaways

- Adhere to semantic and well-formed markup to ensure accessibility and compatibility.
- Separate content, style, and behavior for maintainability.
- Utilize JavaScript and DOM manipulation to create engaging, interactive experiences (DHTML principles).
- Validate and test your code regularly against web standards.
- Prioritize accessibility and progressive enhancement to serve all users effectively.

By following these principles, developers can craft websites that are not only visually appealing but also robust, accessible, and future-proof.

Conclusion

The principles of HTML, XHTML, and DHTML form a cohesive framework for building modern

websites. HTML provides the structural foundation, XHTML enforces stricter syntax and data consistency, and DHTML introduces interactive capabilities through dynamic manipulation. Emphasizing semantic markup, separation of concerns, validation, and accessibility ensures that websites are maintainable, compatible across platforms, and user-friendly. As web standards continue to evolve, embracing these core principles remains essential for creating effective, efficient, and accessible digital experiences.

Principles of HTML, XHTML, and DHTML: An Expert Overview

In the rapidly evolving landscape of web development, understanding the foundational technologies that shape the internet is essential for developers, designers, and digital strategists alike. Among these, HTML, XHTML, and DHTML stand out as pivotal components that have significantly influenced how content is created, structured, and rendered across browsers and devices. This article delves into the core principles of each technology, examining their syntax, philosophy, compatibility, and the role they play in modern web development.

Understanding HTML: The Backbone of the Web

What is HTML?

HTML, or HyperText Markup Language, is the standard markup language used to create web pages. It provides the structural skeleton of a webpage, organizing content into elements such as headings, paragraphs, links, images, and more. Developed initially in the late 1980s by Tim Berners-Lee, HTML has undergone numerous revisions, evolving from simple text tags to a sophisticated language supporting multimedia, scripting, and semantic structures.

Principles of HTML

- **Simplicity and Readability**

HTML was designed to be accessible and straightforward, enabling authors with minimal technical background to create web pages. Its syntax is based on tags enclosed within angle brackets, such as `

` for paragraphs or `` for links. This simplicity fosters an environment where content creators can focus on the message without being bogged down by complex syntax.

- **Content Structure and Semantics**

At its core, HTML emphasizes the importance of structuring content meaningfully. Elements like ``, ``, `

- Browser Compatibility

HTML's principles are rooted in ensuring that content displays consistently across browsers. This is achieved through standardized tags and attributes, although variations and quirks have historically posed challenges, leading to the development of standards and best practices.

- Progressive Enhancement

HTML encourages a philosophy where basic content and functionality are accessible to all users, regardless of their browser capabilities. Additional features like CSS and JavaScript enhance the experience but are not essential for core content accessibility.

Key Features of HTML

- **Tags and Elements: Building blocks of content structure.**
- **Attributes: Provide additional information about elements, such as `href` in `` tags.**
- **Forms and Input: Enable user interaction and data collection.**
- **Multimedia Support: Incorporation of images, audio, and video.**
- **Links and Navigation: Hypertext links connecting pages and resources.**

XHTML: Extending and Enforcing XML Principles

What is XHTML?

XHTML, or Extensible HyperText Markup Language, is a reformulation of HTML as an XML application. Developed in the early 2000s by the World Wide Web Consortium (W3C), XHTML aims to combine the flexibility of HTML with the rigor and extensibility of XML. It enforces stricter syntax rules, promoting well-formed documents that are easier to parse and validate.

Principles of XHTML

- Stricter Syntax and Well-Formed Documents

Unlike HTML, which is forgiving of minor syntax errors, XHTML requires documents to be well-formed. This means:

- **All tags must be properly closed (e.g., `` instead of ``).**
- **Attribute names and values must be in lowercase.**
- **All attribute values must be quoted.**
- **Nesting must be correct, and elements must be properly closed.**

This strictness ensures documents are valid XML, facilitating better interoperability, validation, and tool support.

- **XML Compatibility**

Since XHTML is based on XML, it inherits XML?s features:

- **Namespace support: Allows combining multiple XML vocabularies.**
- **Validation: Documents can be validated against DTDs or XML schemas.**
- **Extensibility: Developers can create custom tags and attributes.**

- **Enhanced Parsing and Data Interchange**

XHTML documents are more predictable to parsers, making them suitable for data interchange and integration with other XML-based systems. This predictability improves consistency across different platforms and tools.

- **Separation of Content and Presentation**

XHTML encourages the separation of structure (HTML/XHTML) from presentation (CSS), aligning with best practices for maintainable, scalable web development.

Transition from HTML to XHTML

- **XHTML was designed to be a stricter, cleaner version of HTML.**
- **Many developers adopted XHTML to leverage XML tools and validation.**
- **However, XHTML's strict syntax meant that browsers needed to be XHTML-aware or**

served XHTML as application/xhtml+xml rather than text/html.

DHTML: Dynamic HTML for Interactive Web Experiences

What is DHTML?

DHTML, or Dynamic HTML, is not a single technology but a collection of HTML, CSS, and JavaScript techniques used to create interactive and animated content within web pages. Emerging in the late 1990s, DHTML allowed developers to modify webpage content and presentation dynamically without reloading the page, paving the way for richer, more engaging user interfaces.

Principles of DHTML

- **Separation and Integration of Technologies**

DHTML leverages:

- **HTML for structure.**
- **CSS for styling and layout.**
- **JavaScript for scripting and interaction.**

This separation of concerns enables dynamic updates while maintaining a clear code architecture.

- **Client-Side Manipulation**

DHTML relies heavily on client-side scripting to modify document object models (DOM) in real-time. This allows:

- **Moving, showing, hiding, or resizing elements.**
- **Changing styles dynamically.**
- **Responding instantly to user actions like clicks, hovers, or keyboard input.**

- **Event-Driven Programming**

DHTML is fundamentally event-driven. Developers attach event listeners to elements to trigger scripts, enabling rich interactions such as sliders, drop-down menus, and animations.

- Cross-Browser Compatibility

In its early days, DHTML faced challenges due to inconsistent browser implementations. Developers had to write code that gracefully degraded or used browser-specific code to ensure broad compatibility.

Principles in Practice

- Manipulating the DOM: Using JavaScript to add, remove, or modify elements and attributes.
- CSS Dynamic Styling: Changing classes, inline styles, or computed styles to animate or alter presentation.
- Event Handling: Responding to user actions for interactive features.

Limitations and Evolution

While DHTML was revolutionary, it was also complex and often inconsistent across browsers. Over time, many of its features have been standardized through DOM specifications and modern JavaScript frameworks. Today, DHTML concepts underpin many web applications, but are implemented via contemporary practices like AJAX, React, Angular, and Vue.js for more robust, cross-browser compatible solutions.

Comparison and Interrelation of HTML, XHTML, and DHTML

Core Differentiators

| Aspect | HTML | XHTML | DHTML |

|---------|-------|--------|--------|

| Definition | Standard markup language for webpage structure | XML-based, well-formed version of HTML | Collection of technologies for dynamic and interactive content |

| Syntax | Flexible, forgiving | Strict, XML-compliant | Not about syntax, but dynamic manipulation via scripts |

| Standards | Developed by W3C, evolving | Stricter, XML-based standards | Combines HTML, CSS, JavaScript for client-side interactivity |

| Validation | Optional but recommended | Mandatory for well-formedness | Not applicable; relates to interactivity features |

| Use Case | Content structure | Structured, validated content | Dynamic, interactive user interfaces |

Interrelation and Evolution

- HTML forms the baseline; it's simple and widely supported.
- XHTML emerged to impose stricter syntax, making documents more predictable and compatible with XML tools, but faced challenges with browser support.
- DHTML builds upon HTML (or XHTML), adding dynamic capabilities through scripting and styling, enabling modern web interactivity.

Conclusion: Principles as the Foundation for Modern Web Development

Understanding the principles of HTML, XHTML, and DHTML is crucial for anyone involved in web development. Each technology reflects a different philosophy and set of best practices:

- HTML emphasizes simplicity, accessibility, and broad compatibility, making it the fundamental language of the web.
- XHTML enforces stricter, well-formed syntax aligned with XML standards, promoting validation and data interchange.
- DHTML leverages these markup languages combined with CSS and JavaScript to create rich, interactive experiences.

As the web continues to evolve, these principles underpin contemporary frameworks and standards. Modern developers should appreciate the historical context and technical foundations these technologies provide, enabling them to craft robust, accessible, and engaging digital experiences. Whether working with raw HTML, validating with XHTML, or scripting dynamic interfaces using DHTML concepts, understanding these core principles remains essential in navigating the complexities of the modern web landscape.

QuestionAnswer What are the main differences between HTML and XHTML? HTML is a flexible markup language used for creating web pages with loose syntax rules, while XHTML

is an XML-based version of HTML that enforces stricter syntax, such as proper nesting and case sensitivity, ensuring better consistency and compatibility across platforms. What are the core principles of DHTML? DHTML (Dynamic HTML) combines HTML, CSS, and JavaScript to create interactive and animated web pages. Its core principles include dynamic content manipulation, event-driven programming, and seamless integration of style and behavior to enhance user experience. Why is it important to use proper nesting and syntax in XHTML? Proper nesting and syntax in XHTML are crucial because XHTML is XML-compliant, meaning that incorrect syntax can cause parsing errors, rendering issues, and decreased compatibility across different browsers and devices. How do HTML, XHTML, and DHTML relate to each other? HTML is the foundational markup language, XHTML is a stricter XML-based version of HTML, and DHTML leverages HTML or XHTML combined with CSS and JavaScript to create dynamic, interactive web pages. DHTML works within the framework of HTML or XHTML to enhance functionality. What are the advantages of using XHTML over HTML? XHTML's advantages include stricter syntax rules that promote cleaner code, improved interoperability across different systems, better compatibility with XML tools, and enhanced accessibility and maintainability of web pages. What principles should be followed when designing with DHTML? Design principles for DHTML include ensuring cross-browser compatibility, separating content from presentation using CSS, writing clean and efficient JavaScript, and creating responsive, accessible interfaces that enhance user interaction without compromising performance. Can you convert an HTML page to XHTML? What should be considered? Yes, converting HTML to XHTML involves ensuring proper syntax, such as closing all tags, using lowercase for tags and attributes, adding proper quotes around attribute values, and validating the code with an XHTML validator to comply with XML standards. What role does CSS play in the principles of HTML, XHTML, and DHTML? CSS is essential for defining the presentation and layout of web pages in HTML, XHTML, and DHTML. It separates style from content, enabling consistent and maintainable designs, especially important in DHTML for creating dynamic visual effects. Why is understanding the principles of HTML, XHTML, and DHTML important for modern web development? Understanding these principles is vital because they form the foundation for creating accessible, maintainable, and interactive websites. Knowledge of syntax, standards, and dynamic techniques ensures compatibility, better user experience, and adherence to best practices in web development.

Related keywords: HTML, XHTML, DHTML, web development, markup languages, web design, W3C standards, DOM, CSS, scripting languages

Read the full article online: Principles of html xhtml and dhtml